

THE MaxBook



Max/MSP リファレンスマニュアル

本書はMax/MSPに付属するオブジェクト・リファレンスの日本語訳です。翻訳にあたっては細心の注意を払いましたが、お気付きの点などございましたら、何なりとお知らせいただければ幸いです。

また、正誤表など本書に関する情報をWEBサイトで公開する予定ですので、合わせてご活用いただけるようご案内いたします。

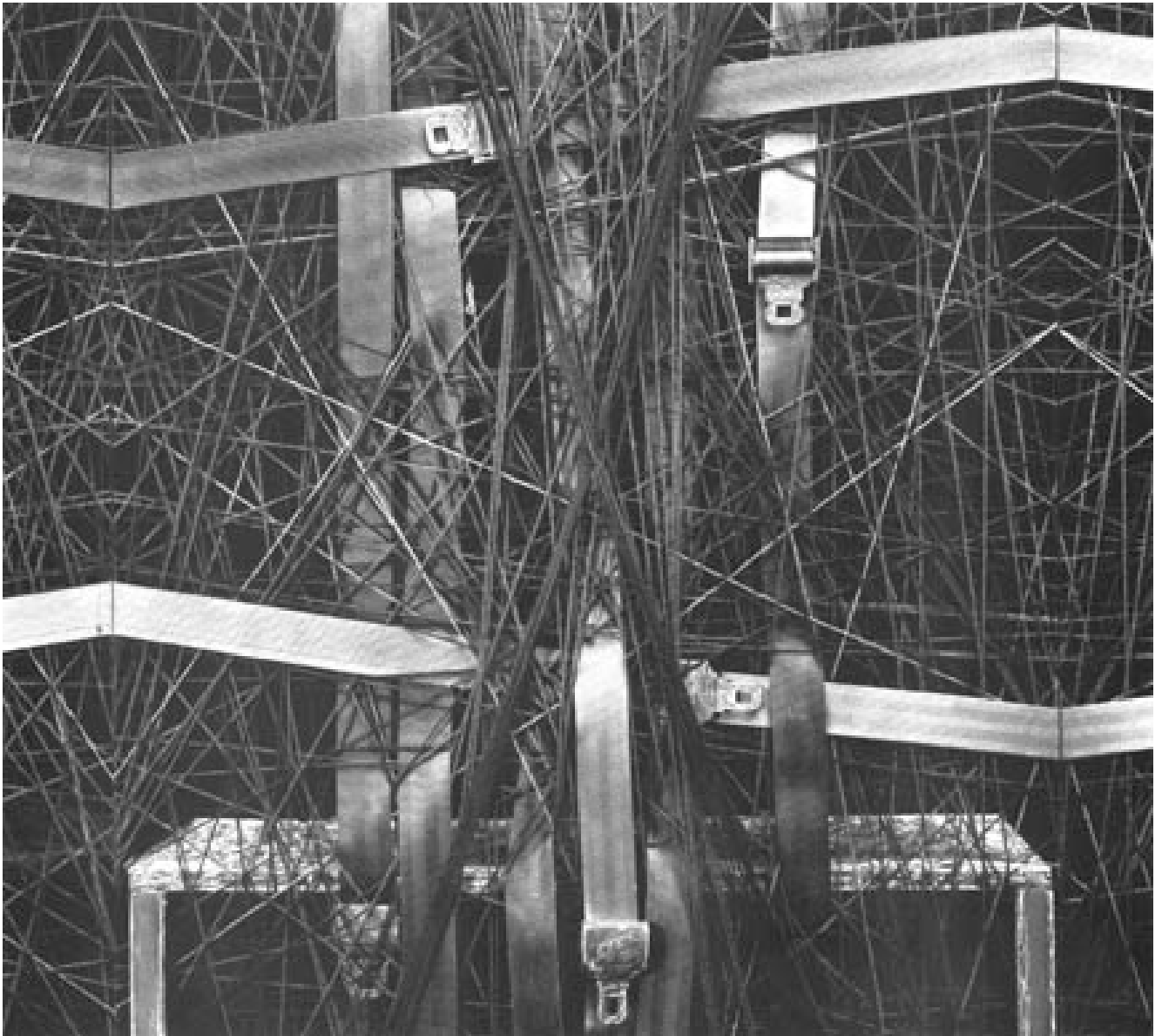
連絡先：

info@cameo.co.jp

WEBサイト：

<http://www.cameo.co.jp/products/cycling74/maxmsp.html>

MAX



リブアレンス

オブジェクトは-オブジェクトとよく似ているが、インレットが逆である。

入力

- 整数** 第1インレット：第2インレットに受け取った数値からこの数値を減算し、その結果をアウトレットから出力する。
第2インレット：この数値は内部に格納され、第1インレットに受け取った数値によって減算される。
- 実数** !-オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。
- bang** 第1インレット：現在格納されている値を元に減算を行う。アーギュメントが設定されていない場合は初期値の0が適用される。

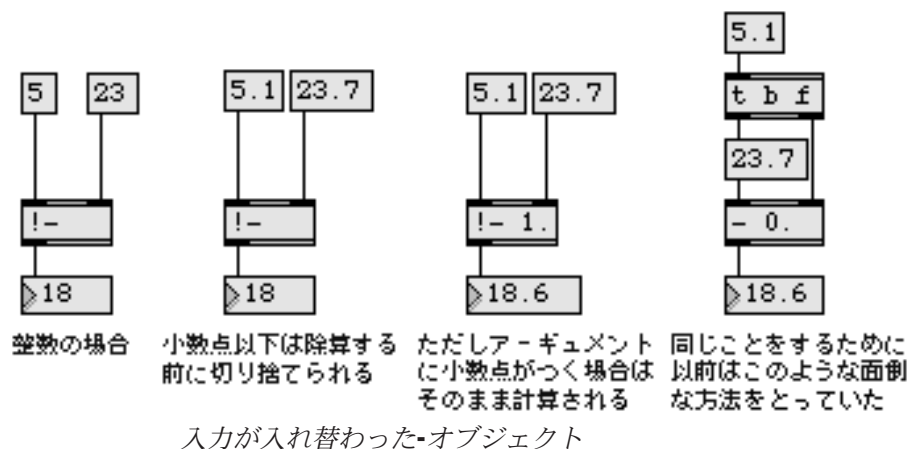
アーギュメント

整数または実数 オプション。第1インレットに受け取った値によって減算される初期値を設定する。実数のアーギュメントを与えた場合は、実数として数値の減算を行う。

出力

- 整数** インレットに受け取った2つの数値の差。
- 実数** アーギュメントに小数点がある時は、2つの数値の差が実数として出力される。

利用例



参照

expr

数式の値を求める

**!/
!=**

数値の除算を行う (**!**オブジェクトとはインレットが逆)

!

2つの数値の比較し、等しくなければ1を出力する

+

2つの数値を加算し、結果を出力する

-

2つの数値を減算し、結果を出力する

2つの数値を乗算し、結果を出力する

/

2つの数値を除算し、結果を出力する

%

数値の除算を行い、余りを出力する

Tutorial 8

Doing Math in Max (英文PDF参照)

!/

数値の除算を行う
(!/オブジェクトとはインレットが逆)

!/オブジェクトは/オブジェクトとよく似ているが、インレットが逆である。

入力

整数 第1インレット：第2インレットに受け取った数値をこの数値によって除算し、その結果をアウトレットから出力する。
第2インレット：この数値は内部に格納され、第1インレットに受け取った数値によって除算される。

実数 !/オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。

bang 第1インレット：現在格納されている値を元に除算を行う。

アーギュメント

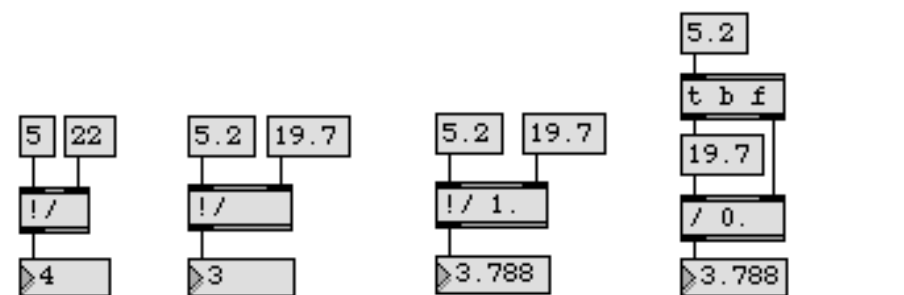
整数または実数 オプション。除算のための初期値を設定する。アーギュメントがない場合、除数は1に設定される。実数のアーギュメントを与えた場合は、実数として数値の除算を行う。(0による除算は許されていない。整数の0による除算は1による除算と同じ結果になる。実数の0による除算は、常に -2^{31} の出力になる。)

出力

整数 インレットに受け取った2つの数値を除算し、結果をアウトレットから出力する。

実数 アーギュメントに小数点がある時は、2つの数値の商が実数として出力される。

利用例



入力が入れ替わった!/オブジェクト

参照

expr	数式の値を求める
!-	数値の減算を行う (-オブジェクトとはインレットが逆)
!=	2つの数値の比較し、等しくなければ1を出力する
+	2つの数値を加算し、結果を出力する
-	2つの数値を減算し、結果を出力する
*	2つの数値を乗算し、結果を出力する
/	2つの数値を除算し、結果を出力する
%	数値の除算を行い、余りを出力する

Tutorial 8 Doing Math in Max (英文PDF参照)

入力

- 整数** 第1インレット：入力された数値は第2インレットの数値と比較され、2つの数値が等しくなければ1が、等しければ0が出力される。
- 第2インレット：入力された数値は内部に格納され、第1インレットに入力される値と比較される。
- 実数** !=オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。
- bang** 第1インレット：現在、格納されている数値で比較を行う。アーギュメントがない場合、初期値は0に設定される。
- リスト** 第1インレット：1つ目の数値と2つ目の数値を比較し、2つの数値が等しくなければ1が、等しければ0が出力される。

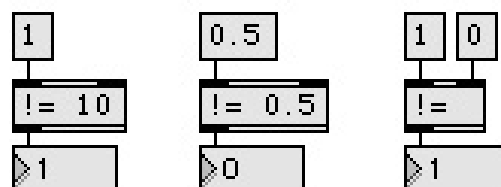
アーギュメント

整数または実数 オプション。初期値を設定すると、第1インレットに入力される数値と比較されるのに用いられる。実数のアーギュメントであれば、実数として数値の比較を行う。

出力

入力される数値が等しくなければ1が、等しければ0が出力される。

利用例



二つの数値が等しくないかどうかを調べる

参照

- select** 特定の入力を選択し、それ以外は通過させる
- split** 数値を範囲によって振り分ける
- <** 2つの数値を比較し、より小さいことを判断する
- <=** 2つの数値を比較し、より小さいか等しいことを判断する
- ==** 2つの数値を比較し、等しければ1を出力する
- >** 2つの数値を比較し、より大きいことを判断する
- >=** 2つの数値を比較し、より大きいか等しいことを判断する

入力

- 整数** 第1インレット：この数値に第2インレットの値を加算し、結果が出力される。
- 第2インレット：この数値は内部に格納され、第1インレットに受け取った数値に加算される。
- 実数** +オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。
- bang** 第1インレット：現在、格納されている値を元に加算を行う。アーギュメントがない場合、初期値は0に設定される。
- リスト** 第1インレット：1つ目の数値が2つ目の数値に加算され、アウトレットから結果が出力される。
- set** 第1インレット：**set**に続けて数値を受け取ると、結果の出力をせずに、その値は第2インレットの値に加算される。続けて**bang**を受け取ると結果が出力される。
- set**メッセージは他の **+**, **-**, *****, **/**, **%**, **<**, **<=**, **==**, **!-**, **!/**, **!=**, **>=**, **>**, **&&**, **||**, **&**, **|**, **<<**, **>>** といった、すべての算術演算子、論理演算子、ビット処理演算子のオブジェクトに対しても同様に機能する。数値は第1インレットに入る値として計算されるが、結果は出力されない。

アーギュメント

- 整数または実数** オプション。加算のための初期値を設定する。この値は第1インレットで受け取った数値に加算される。実数のアーギュメントであれば、実数として数値の加算を行う。

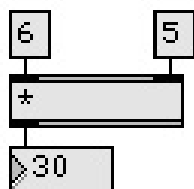
出力

- 整数** インレットに受け取った2つの数値の合計が出力される。
- 実数** アーギュメントに小数点がある時は、2つの数値の和が実数として出力される。

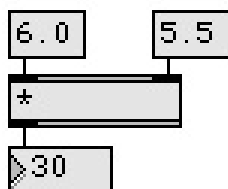
+

2つの数値を加算し、
結果を出力する

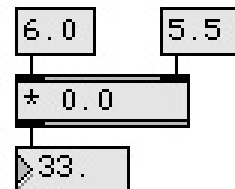
利用例



通常の整数同士の加算



実数は加算の前に小数点以下を
切り捨てられる...



...実数のアーギュメントを
持たない限り

参照

expr

- !- 数式の値を求める
- !- 数値の減算を行う (-オブジェクトとはインレットが逆)
- !/ 数値の除算を行う (/オブジェクトとはインレットが逆)
- 2つの数値を減算し、結果を出力する
- * 2つの数値を乗算し、結果を出力する
- / 2つの数値を除算し、結果を出力する
- % 数値の除算を行い、余りを出力する

Tutorial 8

Doing math in Max (英文PDF参照)

入力

- 整数** 第1インレット：第2インレットに受け取った数値をこの数値から減算し、その結果がアウトレットから出力される。
- 第2インレット：受け取った数値は内部に格納され、第1インレットに受け取った数値から減算される。
- 実数** オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。
- bang** 第1インレット：現在格納されている値を元に減算を行う。アーギュメントがない場合、初期値は0に設定される。
- リスト** 第1インレット：2つ目の数値を1つ目の数値から減算し、その結果が出力される。

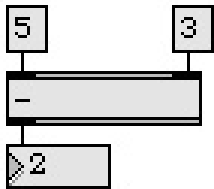
アーギュメント

- 整数または実数** オプション。初期値を設定し、第1インレットに受け取った値から減算する。実数のアーギュメントであれば、実数として数値の減算を行う。

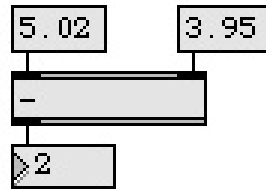
出力

- 整数** インレットに受け取った2つの数値を減算し、その結果がアウトレットから出力される。
- 実数** アーギュメントに小数点がある時は、2つの数値の差が実数として出力される。

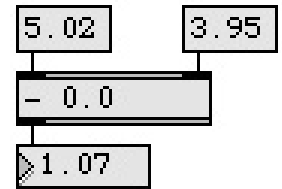
利用例



整数の引き算をする



実数は減算の前に小数点以下を
切り捨てられる...



...実数のアーギュメント
がない限り

参照

expr

!-	数式の値を求める
!-	数値の減算を行う (-オブジェクトとはインレットが逆)
!/	数値の除算を行う (/オブジェクトとはインレットが逆)
+	2つの数値を加算し、結果を出力する
-	2つの数値を減算し、結果を出力する
*	2つの数値を乗算し、結果を出力する
/	2つの数値を除算し、結果を出力する
%	数値の除算を行い、余りを出力する

Tutorial 8

Doing math in Max (英文PDF参照)

入力

- 整数** 第1インレット：この数値は第2インレットの値と乗算され、その結果が出力される。
- 第2インレット：この数値は内部に格納され、第1インレットに受け取った数値と乗算される。
- 実数** *オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。
- bang** 第1インレット：現在、格納されている値を元に乗算を行う。アーギュメントがない場合、初期値は0に設定される。
- リスト** 第1インレット：1つ目の数値と2つ目の数値で乗算をし、アウトレットから結果を出力する。

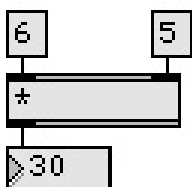
アーギュメント

- 整数または実数** オプション。乗算のための初期値を設定する。この値は第1インレットに入力される数値に乗算される。実数のアーギュメントであれば、実数として数値の乗算を行う。

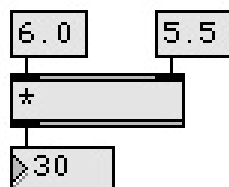
出力

- 整数** インレットに受け取った2つの数値で演算した結果が出力される。
- 実数** アーギュメントに小数点がある時は、2つの数値の積が実数として出力される。

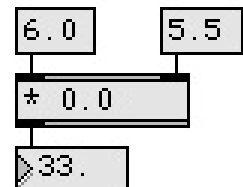
利用例



整数同士の乗算



実数は乗算処理の前に小数点以下を切り捨てられる...



...実数のアーギュメントを持たない限り

参照

expr	数式の値を求める
!-	数値の減算を行う (-オブジェクトとはインレットが逆)
!/	数値の除算を行う (/オブジェクトとはインレットが逆)
!=	2つの数値の比較し、等しくなければ1を出力する
+	2つの数値を加算し、結果を出力する
-	2つの数値を減算し、結果を出力する
/	2つの数値を除算し、結果を出力する
%	数値の除算を行い、余りを出力する
<<	数値のビットを左へシフトする

Tutorial 8 Doing math in Max (英文PDF参照)

入力

- 整数** 第1インレット：入力された値を第2インレットの値で除算し、結果を出力する。
- 第2インレット：入力された値は第1インレットに入力される値を除算するため、格納される。
- 実数** /オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。
- bang** 第1インレット：現在、格納されている値を元に除算を行う。
- リスト** 第1インレット：1つ目の数値が2つ目の数値で除算され、アウトレットから結果が出力される。

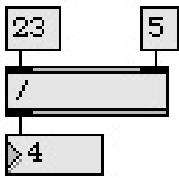
アーギュメント

- 整数または実数** オプション。除数の初期値を設定する。アーギュメントがない場合の除数の初期値は1である。アーギュメントが実数であれば、実数として数値の除算を行う。(0による除算は禁止されている。整数の除算では、0での除算は、1で除算したのと同じ結果が出力される。実数の除算では、0で除算した場合、常に -2^{31} が出力される。)

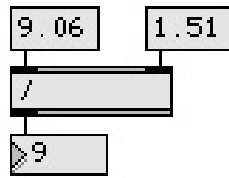
出力

- 整数** 入力された2つの数値で除算を行った結果が出力される。
- 実数** アーギュメントに小数点がある時は、2つの数値の除算の結果が実数として出力される。

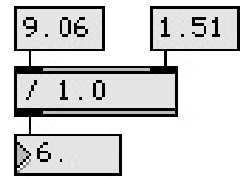
利用例



残りは捨てられる



実数は除算の前に小数点以下を
切り捨てられる...



...実数のアーギュメントを
持たない限り

参照

expr

- !- 数式の値を求める
- !- 数値の減算を行う (-オブジェクトとはインレットが逆)
- !/ 数値の除算を行う (/オブジェクトとはインレットが逆)
- + 2つの数値を加算し、結果を出力する
- 2つの数値を減算し、結果を出力する
- * 2つの数値を乗算し、結果を出力する
- % 数値の除算を行い、余りを出力する

Tutorial 8

Doing math in Max (英文PDF参照)

入力

整数 第1インレット：入力された値を第2インレットの値で除算し、その余りが出力される。

第2インレット：入力された値は内部に格納され、余りを計算するための除数（第1インレットに入力される値を除算する数値）になる。

実数 整数に変換される。

bang 第1インレット：現在、格納されている2つの数値で演算を行う。

リスト 第1インレット：1つ目の数値が2つ目の数値で除算され、アウトレットから余りが出力される。

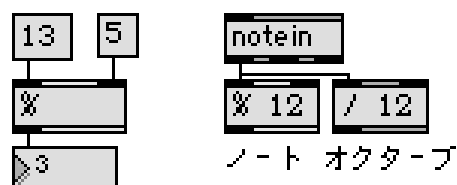
アーギュメント

整数 オプション。除数の初期値を設定する。アーギュメントがない場合、除数は1に設定される。

出力

整数 インレットに受け取った2つの数値を除算し、その余りがアウトレットから出力される。%はモジュロ演算子と呼ばれる。

利用例



除算の余りを求める

参照

expr

- !- 数式の値を求める
- ! 数値の減算を行う（-オブジェクトとはインレットが逆）
- / 数値の除算を行う（/オブジェクトとはインレットが逆）
- +
-
- *
- /

入力

- 整数** 第1インレット：この数値が第2インレットの数値よりも小さければ、1が出力される。それ以外の場合は0が出力される。
- 第2インレット：この数値は内部に格納され、第1インレットに受け取った数値と比較される。
- 実数** <オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。
- bang** 第1インレット：現在、格納されている値を元に比較を行う。アーギュメントがない場合、初期値は0に設定される。
- リスト** 第1インレット：1つ目の数値が2つ目の数値より小さい場合は1が、それ以外の場合は0が出力される。

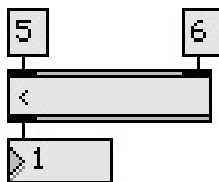
アーギュメント

- 整数または実数** オプション。比較のための初期値を設定し、第1インレットに受け取った数値と比較される。実数のアーギュメントであれば、実数として数値の比較を行う。

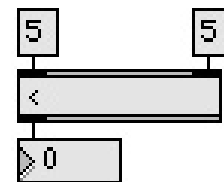
出力

- 整数** 第1インレットの値が第2インレットの値より小さければ1が、第1インレットの値が第2インレットの値よりも大きいか等しければ0が出力される。

利用例



左の数値が右側の数値より小さい



左の数値が右側の数値より小さくない

参照

- !=** 2つの数値の比較し、等しくなければ1を出力する
- <=** 2つの数値を比較し、より小さいか等しいことを判断する
- ==** 2つの数値を比較し、等しければ1を出力する
- >** 2つの数値を比較し、より大きいことを判断する
- >=** 2つの数値を比較し、より大きいか等しいことを判断する

入力

整数 第1インレット：この数値が第2インレットで受け取った数値以下であれば1を、それ以外の場合は0を出力する。

第2インレット：受け取った数値は内部に格納され、第1インレットに受け取った値と比較される。

実数 <=オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。

bang 第1インレット：現在、格納されている値を元に比較を行う。アーギュメントがない場合、初期値は0に設定される。

リスト 第1インレット：1つ目の数値が2つ目の数値以下の場合は1が、それ以外の場合は0が出力される。

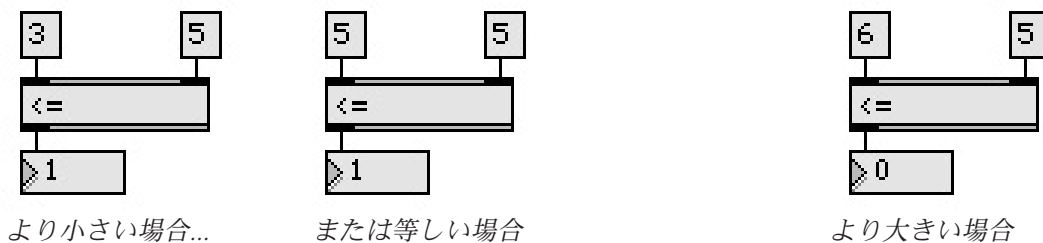
アーギュメント

整数または実数 オプション。初期値を設定し、第1インレットで受け取った数値と比較される。実数のアーギュメントであれば、実数として数値の比較を行う。

出力

整数 第1インレットで受け取った数値が第2インレットの数値以下であれば1が、第1インレットで受け取った数値が第2インレットの数値よりも大きければ0が出力される。

利用例



参照

- != 2つの数値の比較し、等しくなければ1を出力する
- < 2つの数値を比較し、より小さいことを判断する
- == 2つの数値を比較し、等しければ1を出力する
- > 2つの数値を比較し、より大きいことを判断する
- >= 2つの数値を比較し、より大きいか等しいことを判断する

入力

整数 第1インレット：この数値と第2インレットに受け取った数値を比較し、2つの値が等しければ1が、等しくなければ0が出力される。

第2インレット：受け取った数値は内部に格納され、第1インレットに受け取った数値と比較される。

実数 ==オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。

bang 第1インレット：現在、格納されている値を元に比較を行う。アーギュメントがない場合、初期値は0に設定される。

リスト 第1インレット：1つ目の数値と2つ目の数値を比較し、2つの数値が等しければ1が、等しくなければ0が出力される。

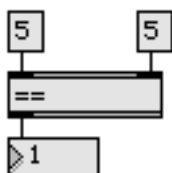
アーギュメント

整数または実数 オプション。初期値を設定し、第1インレットに受け取った数値と比較される。実数のアーギュメントであれば、実数として数値の比較を行う。

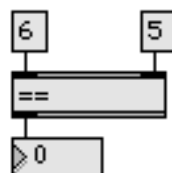
出力

整数 受け取った数値が等しければ1が、等しくなければ0が出力される。

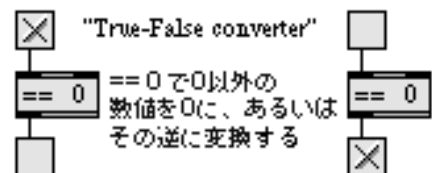
利用例



数値同士は等しい



数値同士は等しくない



論理演算子 notとして== 0を用いる

参照

select
split

特定の入力を選択し、それ以外は通過させる
数値を範囲によって振り分ける

!=

2つの数値の比較し、等しくなければ1を出力する

<

2つの数値を比較し、より小さいことを判断する

<=

2つの数値を比較し、より小さいか等しいことを判断する

>

2つの数値を比較し、より大きいことを判断する

>=

2つの数値を比較し、より大きいか等しいことを判断する

入力

- 整数** 第1インレット：この数値が第2インレットの値より大きければ1が、それ以外の場合は0が出力される。
- 第2インレット：受け取った値は格納され、第1インレットに受け取った数値と比較される。
- 実数** >オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。
- bang** 第1インレット：現在、格納されている値を元に比較を行う。アーギュメントがない場合、初期値は0に設定される。
- リスト** 第1インレット：1つ目の数値が2つ目の数値より大きい場合1が、それ以外の場合は0が出力される。

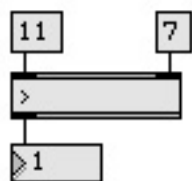
アーギュメント

- 整数または実数** オプション。初期値を設定し、第1インレットに受け取った数値と比較される。実数のアーギュメントであれば、実数として数値の比較を行う。

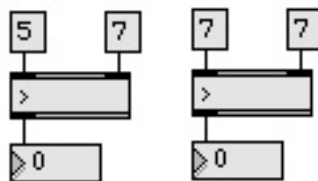
出力

- 整数** 第1インレットの値が第2インレットの値より大きい場合1が、第1インレットの値が第2インレットの値よりも小さいか等しい場合には0が出力される。

利用例



左側の数値の方が大きい



左側の数値は右側に比べて大きくない

参照

- !=** 2つの数値の比較し、等しくなければ1を出力する
- <** 2つの数値を比較し、より小さいことを判断する
- <=** 2つの数値を比較し、より小さいか等しいことを判断する
- ==** 2つの数値を比較し、等しければ1を出力する
- >=** 2つの数値を比較し、より大きいか等しいことを判断する

>=

2つの数値を比較し、より大きいか等しいことを判断する

入力

整数 第1インレット：この数値が第2インレットに受け取った値以上であれば1が、それ以外の場合は0が出力される。

第2インレット：この数値は内部に格納され、第1インレットに受け取った数値と比較される。

実数 >=オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。

bang 第1インレット：現在、格納されている値を元に比較を行う。アーギュメントを持たない場合、初期値は0に設定される。

リスト 第1インレット：1つ目の数値が2つ目の数値以上の場合1が、それ以外の場合は0が出力される。

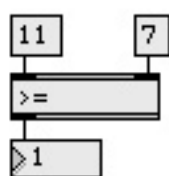
アーギュメント

整数または実数 オプション。比較のための初期値を設定し、第1インレットに受け取った数値と比較される。実数のアーギュメントであれば、実数として数値の比較を行う。

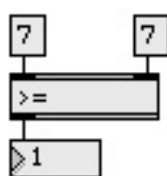
出力

整数 第1インレットの値が第2インレットの値以上なら1が、第1インレットの値が第2インレットの値よりも小ければ0が出力される。

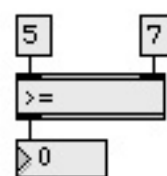
利用例



より大きい場合...



または等しい場合



大きくも等しくもない場合

参照

- != 2つの数値の比較し、等しくなければ1を出力する
- < 2つの数値を比較し、より小さいことを判断する
- <= 2つの数値を比較し、より小さいか等しいことを判断する
- == 2つの数値を比較し、等しければ1を出力する
- > 2つの数値を比較し、より大きいことを判断する

入力

- 整数** 第1インレット：その数値は第2インレットに受け取った数値とバイナリー形式で比較される。出力は両方の数値で1であるビットから構成された数値になる。
- 第2インレット：その数値は第1インレットに受け取った数値との比較のために格納される。
- 実数** 整数に変換される。
- bang** 第1インレット：現在、格納されている値の比較を行う。アーギュメントが設定されていない場合は、**&**オブジェクトは比較のために0を初期値とする。
- リスト** 第1インレット：1つ目と2つ目の数値をビットごとに比較し、両方の数値で1であるビットから構成された数値を出力する。

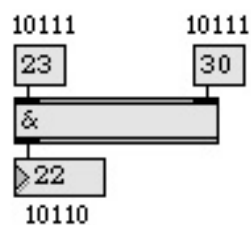
アーギュメント

- 整数** オプション。第1インレットから受け取った数値と比較するための初期値を設定する。

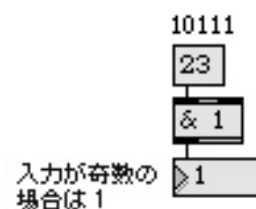
出力

- 整数** インレットに受け取った2つの数値は1ビットずつ比較される。両方の数値でビットが1ならば、出力する数値のビットは1になる。それ以外のビットは0になる。

利用例



0でないビットは
両方の値で共有される



入力が奇数の
場合は1
偶数か奇数か判定する
ためにも使用できる

参照

- &&** 2つの数値がともに0以外であれば、1を出力する
- |** 2つの数値のビット単位でのOR演算
- ||** 2つの数値のどちらか一方が0以外であれば、1を出力する

&&

2つの数値がともに0
以外であれば、1を出力する

入力

整数 2つのインレットに入力された値がどちらも0でない時、1を出力する。2つのインレットのうち、どちらか一方でも0が入力されると、出力される値は0になる。第1インレットに数値を入力する時に、結果が出力される。

実数 整数に変換される。

bang 第1インレット：その時点で格納されている2つの値で処理を行う。アーギュメントを設定しない場合の初期値は0となる。

リスト 第1インレット：1つ目の数値と2つ目の数値のどちらも0でない場合1が、それ以外の場合は0が出力される。

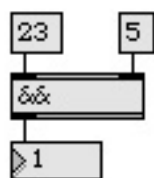
アーギュメント

整数 オプション。アーギュメントを与えるとその値は格納され、第2インレットに入力される値の初期値になる。

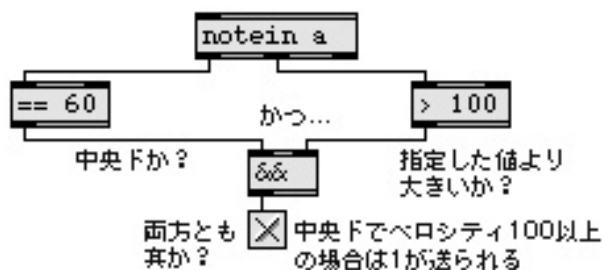
出力

整数 第1インレットに入力された値と第2インレットに入力された値（またはアーギュメントで指定された値）が共に0でない場合、1が出力される。そうでない場合は0が出力される。

利用例



どちらの数値も0ではない



比較演算子と組み合わせて使用した場合

参照

- &** 2つの数値のビット単位でのAND演算
- |** 2つの数値のビット単位でのOR演算
- ||** 2つの数値のどちらか一方が0以外であれば、1を出力する

入力

- 整数** 第1インレット：入力された2つの数値のうち、どちらかの数値で1であるビットから構成された数値を出力する。
- 第2インレット：数値は第1インレットに受け取った数値とのOR演算のために格納される。
- 実数** 整数に変換される。
- bang** 第1インレット：現在、格納されている値を用いて演算を行う。アーギュメントが設定されていない場合は、**I**オブジェクトは0を初期値とする。
- リスト** 第1インレット：1つ目と2つ目の数値をビットごとに結合し、どちらかの数値で1であるビットから構成された数値を出力する。

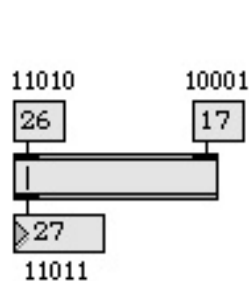
アーギュメント

- 整数** オプション。第1インレットから受け取った数値とOR演算をするための初期値を設定する。

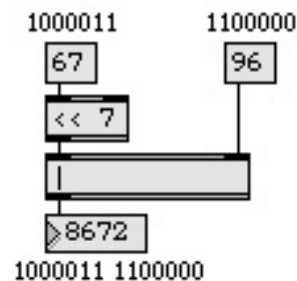
出力

- 整数** インレットに受け取った2つの数値の0でないすべてのビットは結合される。どちらかの数値でビットが1ならば、出力する数値のビットは1になる。それ以外の出力する数値上でのビットは0になる。

利用例



すべての0でないビットは結合される



1つの整数に2つの数値をまとめることに使用できる

参照

- &** 2つの数値のビット単位でのAND演算
- &&** 2つの数値がともに0以外であれば、1を出力する
- ||** 2つの数値のどちらか一方が0以外であれば、1を出力する

入力

- 整数** 両方のインレットで受け取った値のうち、少なくともどちらか一方が0でない時に、1を出力する。2つのインレット両方で0を受け取ると、出力される値は0になる。第1インレットで数値を受け取ると、結果が出力される。
- 実数** 整数に変換される。
- bang** 第1インレット：現在、格納されている値を元に演算を行う。アーギュメントを持たない場合、初期値は0に設定される。
- リスト** 第1インレット：1つ目の数値と2つ目の数値のうち、少なくともどちらか一方が0でない場合1が、それ以外の場合は0が出力される。

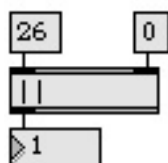
アーギュメント

- 整数** オプション。内部に格納され、第2インレットに入力される値の初期値を設定する。

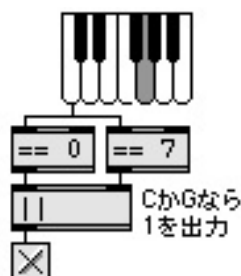
出力

- 整数** 第1インレットで受け取った数値と第2インレットで受け取った数値（またはアーギュメントで指定された数値）のうち、少なくともどちらか一方が0でない場合は1が、そうでない場合は0が出力される。

利用例



2つの数値のうち1つが0でない



比較演算子と組み合わせて使用した場合

参照

- &** 2つの数値のビット単位でのAND演算
- &&** 2つの数値がともに0以外であれば、1を出力する
- |** 2つの数値のビット単位でのOR演算

入力

- 整数** 第1インレット：受け取った数値をバイナリー形式で、指定のビット数分だけ左にシフトする。その結果はアウトレットより出力される。
- 第2インレット：この数値は格納され、第1インレットに受け取った数値を左に何ビットシフトするかを指定する。
- 実数** 整数に変換される。
- bang** 第1インレット：現在、格納されている値を元にビットシフト演算を行う。アーギュメントがない場合、ビットシフト量の初期値は0に設定される。
- リスト** 第1インレット：1つ目の数値が2つ目の数値で指定されたビット数分、左にビットシフトされる。

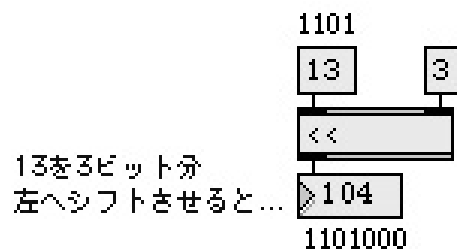
アーギュメント

- 整数** オプション。ビットシフト量の初期値を設定する。

出力

- 整数** 第1インレットの値は指定したビット数分、左にビットシフトされる。ビットシフト量は第2インレットで受け取った値によって指定される。ビットシフト演算された値が出力される。

利用例



2の累乗で乗算したのと同じ効果

参照

- * >> 2つの数値を乗算し、結果を出力する
数値のビットを右へシフトする

入力

- 整数** 第1インレット：受け取った数値をバイナリー形式で、指定のビット数分だけ右にシフトする。その結果はアウトレットより出力される。
- 第2インレット：この数値は格納され、第1インレットに受け取った数値を右に何ビットシフトするかを指定する。
- 実数** 整数に変換される。
- bang** 第1インレット：現在、格納されている値を元にビットシフト演算を行う。アーギュメントがない場合、ビットシフト量の初期値は0に設定される。
- リスト** 第1インレット：1つ目の数値が2つ目の数値で指定されたビット数分、右にビットシフトされる。

アーギュメント

- 整数** オプション。ビットシフト量の初期値を設定する。

出力

- 整数** 第1インレットの値は指定したビット数分、右にビットシフトされる。ビットシフト量は第2インレットで受け取った値によって指定される。ビットシフト演算された値が出力される。

利用例



2の累乗で除算したのと同じ効果

参照

- !/** 数値の除算を行う (ノブジェクトとはインレットが逆)
- <<** 数値のビットを左へシフトする

入力

整数 入力の絶対値を出力する。

実数 **abs**オブジェクトが実数のアーギュメントを持たない限り、整数に変換される。

アーギュメント

整数または実数 オプション。実数のアーギュメントであれば実数での出力となる。

出力

整数 入力の絶対値。

実数 アーギュメントに小数点がある時は、入力の絶対値は実数として出力される。

利用例



出力は自然数



入力を反転させる使用

参照

expr 数式の値を求める

Tutorial 14 Sliders and dials (英文PDF参照)

入力

任意のシンボル シンボルとしてのファイル名かパス。**absolutePath**オブジェクトはファイル名かパスを絶対パスへ変換する。エイリアスも、同じように元のファイルの絶対パスを出力する。

アーギュメント

なし。

出力

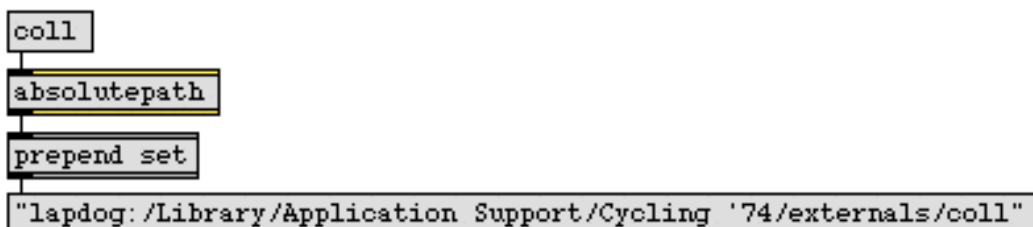
入力されたファイル名かパスが見つければ、出力はその絶対パスとなる。絶対パスはMax 4.1以前のバージョン(コロンを使用)を除き、スラッシュで区切られて表示される。具体的には下記のようなになる。

"C:/Max Folder/extras/mystuff/mypatch.pat"

パスの形式や種類を変更したい場合は、**conformPath**オブジェクトを使用する。

その名前のファイルが見つからなければ、**absolutePath**オブジェクトは**notfound**というシンボルを出力する。

利用例



参照

- dropfile** ファイルをドラッグ&ドロップする領域を設定する
- opendialog** ファイルやフォルダを選択するためのダイアログを開く
- relativePath** 絶対パスを相対パスへ変換する
- savedialog** 保存するファイルの名前を尋ねるダイアログを開く
- strippath** フル・パス名からファイル名を取得する

入力

整数または実数 アーク・コサイン関数への入力。

bang 第1インレット：格納されている数値でアーク・コサイン関数の計算を行なう。アーギュメントがなければ、初期値は0である。

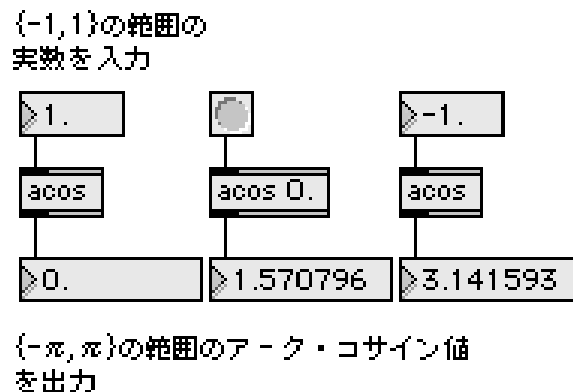
アーギュメント

整数または実数 オプション。アーク・コサイン関数の初期値を設定する。

出力

整数または実数 入力された値のアーク・コサイン値。

利用例



参照

asin	アーク・サイン関数
atan	アーク・タンジェント関数
atan2	アーク・タンジェント関数 (2変数)
cos	コサイン関数
cosh	双曲線コサイン関数
sin	サイン関数
sinh	双曲線サイン関数
tanh	双曲線タンジェント関数

入力

実数または整数 双曲線アーク・コサイン関数への入力。

bang 第1インレット：現在格納されている数値の双曲線アーク・コサイン値を計算する。アーギュメントがない場合は、**acosh**の初期値は0である。

アーギュメント

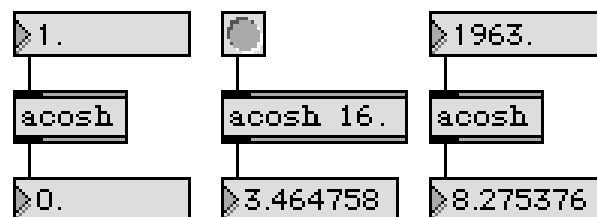
実数または整数 オプション。双曲線アーク・コサイン関数への初期値を設定する。

出力

実数または整数 入力された値のアーク・コサイン値。

利用例

実数を入力



双曲線アーク・コサイン値

参照

acos	アーク・コサイン関数
asin	アーク・サイン関数
asinh	双曲線アーク・サイン関数
atan	アーク・タンジェント関数
atan2	アーク・タンジェント関数 (2変数)
atanh	双曲線アーク・タンジェント関数
cos	コサイン関数
cosh	双曲線コサイン関数
sin	サイン関数
sinh	双曲線サイン関数
tan	タンジェント関数
tanh	双曲線タンジェント関数

入力

- 整数** 第1インレット：**accum**オブジェクトに格納された数値を更新し、更新された数値をアウトレットから出力する。
- 第2インレット：数値を出力せず、入力された数値を格納された値に加算する。
- 第3インレット：数値を出力せず、入力された数値を格納された値に乗算する。
- 実数** 第1インレットおよび第2インレット：アーギュメントに実数を指定していなければ、整数に変換する。
- 第3インレット：格納されている値が整数であっても、実数として乗算を行なう。
- bang** 第1インレット：現在格納されている数値を出力する。
- set** setメッセージに続く数値を**accum**オブジェクトに格納する。出力は行わない。

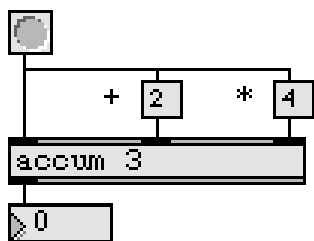
アーギュメント

- 整数あるいは実数** オプション。**accum**オブジェクトに格納する数値の初期値を指定する。初期値に小数点を付けると、数値は実数として格納される。

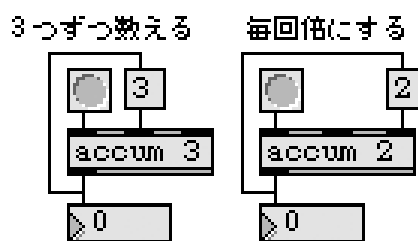
出力

- 整数** **accum**オブジェクトに現在格納されている数値を出力する。
- 実数** アーギュメントに小数点を指定している場合のみ、実数として出力する。

利用例



格納された値を加算する、または格納された値を乗算する



異なる量を増加させるために使用する

参照

counter

bangメッセージをカウントし、回数を出力する

float

実数を格納する

int

整数を格納する

Tutorial 21

Storing numbers (英文PDF参照)

active

パッチ・ウィンドウがアクティブであれば1を、
そうでなければ0を出力

入力

active オブジェクトには、インレットがない。出力はパッチがアクティブあるいはアクティブでない時に自動的にトリガーされる。

アーギュメント

なし。

出力

整数 パッチがアクティブな時は1が出力され、アクティブでない時は0が出力される。

利用例



パッチがアクティブの時に gate を開く、または処理を開始する

参照

closebang loadbang

パッチ・ウィンドウが閉じられた時に **bang** を出力する
パッチが読み込まれた時に自動的に **bang** を出力する

Tutorial 40

Automatic actions (英文PDF参照)

入力

- 整数** 連続する2つの数値を過去に何回受け取ったかを出力する（最初に数値を受け取った場合は、先行する数値はないので、何も起こらない）。
- reset** **anal**オブジェクトに格納された数値の中から、最も最近に受け取った数値を消去する。次に入力された数値は、消された数字の部分に格納され、出力はされない。
- clear** **anal**オブジェクトに格納された数値をすべて消去する。ただし、最後に格納された値は残り、次の過去の数値として使われる。

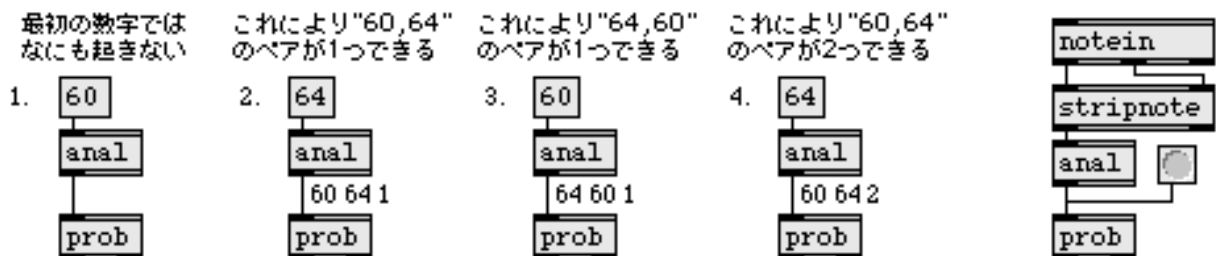
アーギュメント

- 整数** オプション。異なる2つの数値のペアを格納する最大数を指定する。この最大値は1024である。アーギュメントに何も指定しない場合は、**anal**オブジェクトは128の異なった数値のペアを格納する。

出力

- リスト** リストの中の最初の2つの数値は、最後に入力された2つの値である。そして、3番目の数値は最初の2つの数値が、これまでに何回受け取られているかを示す。3つの数値からなるリストは、1つの数値から次の数値へ移行する確率マトリックスを作るために、**prob**オブジェクトに入力して使用するよう設計されている（それはマルコフ連鎖として知られている）。

利用例



2つの数値が発生した相対度数を **anal** から出力し、**prob**オブジェクトに渡して、次の数値へ移行する確率を求める

参照

Histo
prob

受け取った数値のヒストグラムを生成する
重みを付けてランダムな数値の流れを作り出す

append

メッセージの末尾に
アーギュメントを付け加える

入力

set **set**メッセージに続くメッセージを出力せずに、**append**オブジェクトにアーギュメントとして格納する。

その他の任意のメッセージ **append**オブジェクトに格納されたメッセージが、入力されたメッセージの末尾に付け加えられて、出力される。この時、メッセージ同士は空白文字で区切られる。

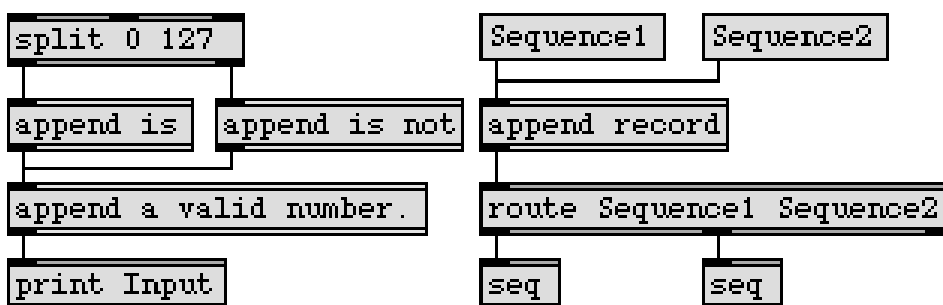
アーギュメント

任意 オプション。入力されるメッセージの末尾に付け加えるメッセージを指定する。

出力

任意 入力されたメッセージの末尾に、**append**オブジェクトに格納されたメッセージを付け加えて出力する。

利用例



シンボルにメッセージを付け加えて、意味のあるメッセージにすることができる

参照

prepend メッセージの前に、あるメッセージを付け加える

Tutorial 25 Managing messages (英文PDF参照)

注記：このオブジェクトは、Max 4.2以降のバージョンでは使用されない

Maxは、**appledvd**オブジェクトによって、Apple DVD Playerアプリケーション（バージョン2.1以降）を制御することができる。このためには、MaxとApple DVD Playerアプリケーションは、ともに起動していなければならない。**appledvd**オブジェクトは、AppleEventによってApple DVD Playerアプリケーションとやり取りをする。これは、Maxでプログラミングをしている時は、DVDPlayerよりMaxの方がFinderでは前面になっているので、DVDのビデオ画像が見えない場合があり、多少不便である。解決法としては、2つのモニターを使うことが最適だろう。また、Apple DVD Playerは、ディスクからオーディオ・データとビデオ・データを展開するのにCPUパワーを多く使うので、DVD再生中にはMaxの処理は制限されてしまう。

ここで、**appledvd**オブジェクトの中で使用される用語のうち、いくつかの簡潔な用語説明を挙げておく。

サブピクチャーとは、DVDに重ねることができるイメージである。サブピクチャーはテキストでもグラフィックでもどちらでも可能である。

タイトルとは、DVD上の完結した作品（映画など）を意味する。DVDによっては、複数のタイトルを持っていることがある。

チャプターは、タイトルの中の1つのセクションである。タイトルにおけるチャプターの数任意であり、DVDの作者に任せられている。

入力

整数	DVDの始まりの位置からの秒数として、特定の時間に再生位置を移動する。
実数	整数に変換される。
bang	第1アウトレットから、現タイトルの始めから現在までの秒数として、再生位置を出力する。
play	現在の位置から再生を始める。
stop	再生を停止する。
pause	再生を一時停止する。
presentation	presentation に続く1か0の数値によって、プレゼンテーション・モードのオンとオフを切り替える。プレゼンテーション・モードは、スクリーンでビデオの上映に似せて、メニューバーを消し、背景を黒くする。
size	DVDイメージのサイズを指定する。サイズの定義として、small, half,

normal, fullの4種類がある。その4つの大きさは、使用するモニタに依存する。

- viewer** **viewer**に続く1か0の数値によって、ビデオ・イメージのオンとオフを切り替える。
- controller** **controller**に続く1か0の数値によって、コントロール・ウィンドウのオンとオフを切り替える。
- mute** **mute**に続く1か0の数値によって、DVDのオーディオ再生のオンとオフを切り替える。
- get** **get**メッセージに続くキーワードによって、DVDに関する情報を得て、第2アウトレットから出力する。

キーワード	説明
disk	DVDがマウントされていれば1を出力する。そうでなければ0を出力する。
mute	オーディオ出力がミュート状態ならば1、そうでなければ0を出力する。
volume	現在のオーディオの音量レベルを出力する（1～10）。
numangles	現在の再生位置で利用可能なカメラ・アングルの数を出力する。
numaudios	現在の再生位置で利用可能なオーディオ・トラックの数を出力する。
numchapters	現在の再生位置で利用可能なチャプターの数出力する。
numsubpictures	現在の再生位置で利用可能なサブピクチャーの数を出力する。
titles	利用可能なタイトルの数を出力する。
angle	現在のカメラ・アングルを出力する。
audio	現在のオーディオ・トラックの数を出力する。
chapter	現在のチャプターの数出力する。

subpicture	現在のサブピクチャーの数を出力する。
title	現在のタイトルの数を出力する。
volume	volume に続く1か0の数値によって、現在のオーディオの音量レベルを設定する。
angle	angle に続く0から360までの数値によって、現在のカメラ・アングルを設定する。
audio	audio に続く1からオーディオ・トラックの最大数までの数値によって、現在のオーディオ・トラックを設定する。
chapter	chapter に続く1からチャプターの最大数までの数値によって、現在のチャプターを設定する。
subpicture	subpicture に続く1からサブ・ピクチャーの最大数までの数値によって、現在のサブピクチャーを設定する。
title	title に続く1からタイトルの最大数までの数値によって、現在のタイトルを設定する。
menu	menu メッセージで、DVDプレーヤをメニューへ移動させる。
titlemenu	titlemenu メッセージで、DVDプレーヤをタイトル・メニューへ移動させる。
return	return メッセージは、キーボードのreturnキーと同じ役割を持つ。
up	up メッセージは、キーボードの上向き矢印キーと同じ役割を持つ（メニューを操作する為に使用する）。
down	down メッセージはキーボードの下向き矢印キーと同じ役割を持つ（メニューを操作する為に使用する）。
left	left メッセージはキーボードの左向き矢印キーと同じ役割を持つ（メニューを操作する為に使用する）。
right	right メッセージはキーボードの右向き矢印キーと同じ役割を持つ（メニューを操作する為に使用する）。
enter	enter メッセージはキーボードのenterキーと同じ役割を持つ（メニュー

を操作する為に使用する)。

open Apple DVD Player が開いていれば、それを最前面にする。

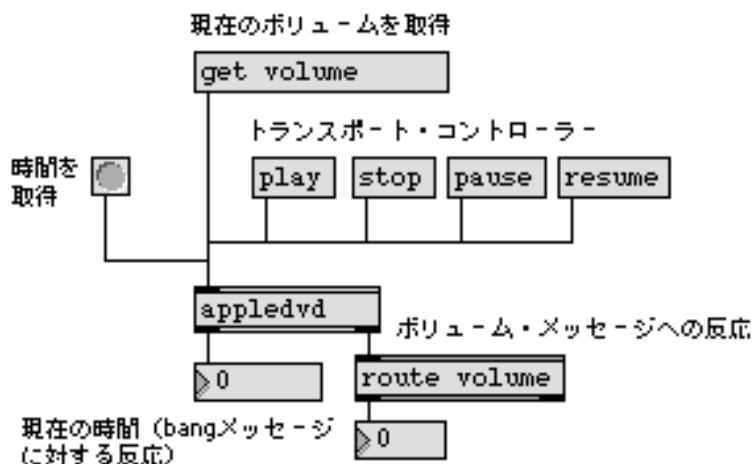
(マウス) マウスで **appledvd** オブジェクトをダブルクリックすれば、**open** メッセージと同じ効果が得られる。

出力

整数 第1アウトレット：**bang**メッセージを受け取ると、現タイトルの始めから現在までの再生時間を出力する。

第2アウトレット **get**メッセージに対応して様々な値が出力される（前の説明を参照のこと）。

利用例



参照

cd
movie
vdp

オーディオCDをコントロールする
QuickTimeムービーをウィンドウ内で再生する
シリアル・ポートを通じて、ビデオディスク・プレーヤーをコントロールする

入力

整数または実数 アーク・サイン関数への入力。

bang 第1インレット：格納されている数値で、アーク・サイン関数の計算を行なう。アーギュメントがなければ、初期値は0である。

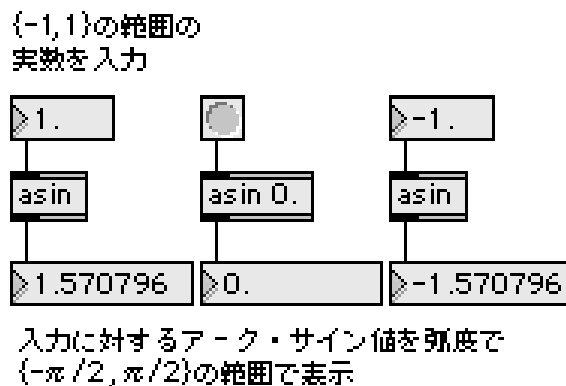
アーギュメント

整数または実数 オプション。アーク・サイン関数の初期値を設定する。

出力

整数または実数 入力された値のアーク・サイン値。

利用例



参照

acos	アークコサイン関数の数値計算を行なう
atan	アークタンジェント関数の数値計算を行なう
atan2	2変数のアークタンジェント関数の数値計算を行なう
cos	コサイン関数の数値計算を行なう
cosh	ハイパーボリック（双曲線）コサイン関数の数値計算を行なう
sin	サイン関数の数値計算を行なう
sinh	ハイパーボリック（双曲線）サイン関数の数値計算を行なう
tanh	ハイパーボリック（双曲線）タンジェント関数の数値計算を行なう

入力

実数または整数 双曲線アーク・サイン関数への入力。

bang 第1インレット：現在格納されている数値の双曲線アーク・サイン値を計算する。アーギュメントがない場合は、**asinh**の初期値は0である。

アーギュメント

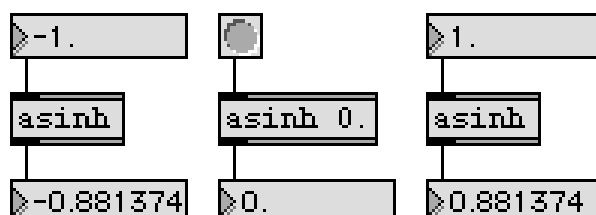
実数または整数 オプション。双曲線アーク・サイン関数への初期値を設定する。

出力

整数または実数 入力された値の双曲線アーク・サイン値。

利用例

実数による入力



入力に対する双曲線アーク・サイン値

参照

acos	アーク・コサイン関数
acosh	双曲線アーク・コサイン関数
asin	アーク・サイン関数
asinh	双曲線アーク・サイン関数
atan	アーク・タンジェント関数
atan2	アーク・タンジェント関数 (2変数)
atanh	双曲線アーク・タンジェント関数
cos	コサイン関数
cosh	双曲線コサイン関数
sin	サイン関数
sinh	双曲線サイン関数
tan	タンジェント関数
tanh	双曲線タンジェント関数

入力

整数または実数 アーク・タンジェント関数への入力。

bang 第1インレット：格納されている数値で、アーク・タンジェント関数の計算を行なう。アーギュメントがなければ、初期値は0である。

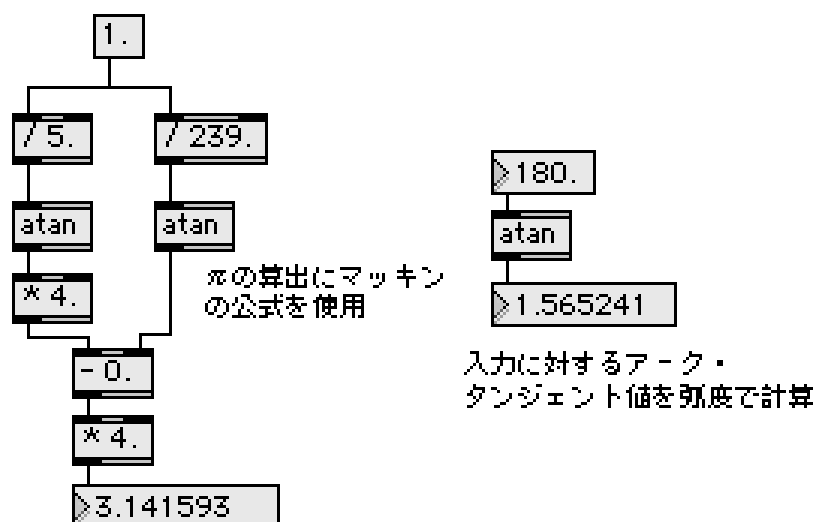
アーギュメント

整数または実数 オプション。アーク・タンジェント関数の初期値を設定する。

出力

整数または実数 入力された値のアーク・タンジェント値。

利用例



参照

acos	アーク・コサイン関数
asin	アーク・サイン関数
atan2	アーク・タンジェント関数 (2変数)
cos	コサイン関数
cosh	双曲線コサイン関数
sin	サイン関数
sinh	双曲線サイン関数
tanh	双曲線タンジェント関数

入力

整数または実数 第1インレット：2変数のアーク・タンジェント関数のうちxの値を入力する。
第2インレット：2変数のアーク・タンジェント関数のうちyの値を入力する。

bang 第1インレット：格納されている数値で、2変数のアーク・タンジェント関数の計算を行なう。アーギュメントがなければ、初期値は0である。

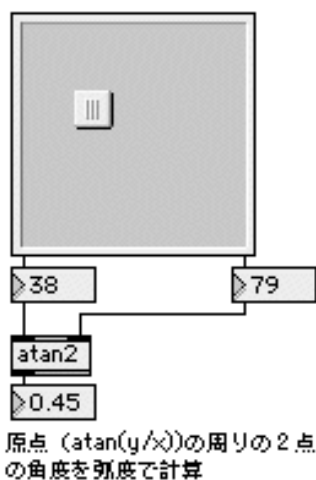
アーギュメント

整数または実数 オプション。2つの数値を与え、2変数のアーク・タンジェント関数の初期値を設定する。

出力

整数または実数 入力された値のアーク・コサイン値。

利用例



参照

acos	アーク・コサイン関数
asin	アーク・サイン関数
atan	アーク・タンジェント関数
cos	コサイン関数
cosh	双曲線コサイン関数
sin	サイン関数
sinh	双曲線サイン関数
tanh	双曲線タンジェント関数

入力

実数または整数 双曲線アーク・タンジェント関数への入力。

bang 第1インレット：現在格納されている数値の双曲線アーク・タンジェント値を計算する。アーギュメントがない場合は、**atanh**への初期値は0である。

アーギュメント

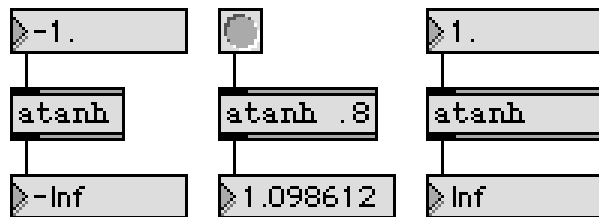
実数または整数 オプション。双曲線アーク・タンジェント関数への初期値を設定する。

出力

整数または実数 入力された値の双曲線アーク・タンジェント値。

利用例

浮動小数点の入力



双曲線アーク・タンジェントは、-1.0から1.0近辺の漸近線となる

参照

acos	アーク・コサイン関数
acosh	双曲線アーク・コサイン関数
asin	アーク・サイン関数
asinh	双曲線アーク・サイン関数
atan	アーク・タンジェント関数
atan2	アーク・タンジェント関数 (2変数)
cos	コサイン関数
cosh	双曲線コサイン関数
sin	サイン関数
sinh	双曲線サイン関数
tan	タンジェント関数
tanh	双曲線タンジェント関数

入力

- 整数** 第1インレット：**bag**オブジェクトに格納されている数値に、入力された数値を追加または削除する。それは、第2インレットに入力された数値に依存する。
- 第2インレット：第1インレットに入力する数値を追加するのか、削除するのかを指定する。0以外は第1インレットに入力する数値を**bag**オブジェクトに追加する。0なら削除する。両方のインレットにどのような数値を入力しても、出力はされない。
- 実数** 整数に変換される。
- bang** 第1インレット：格納されている数値をアウトレットから出力する。
- clear** 第1インレット：**bag**オブジェクトに格納されている数値をすべて削除する。
- list** 第1インレット：2つの数値からなるリストを入力して、リストの2つ目の数値が0以外なら、リストの1つ目の数値は**bag**オブジェクトに追加される。リストの2つ目の数値が0なら、リストの1つ目の数値は削除される。
- send** 第1インレット：**send**メッセージに続く**receive**オブジェクトの名前で、その名前の**receive**オブジェクトすべてに、**bag**オブジェクトに**bang**を入力した時に得られる出力を送信する（**receive**オブジェクトから、**bag**オブジェクトに格納されている数値が出力される）。
- length** 第1インレット：現在**bag**オブジェクトに数値がいくつ格納されているかを出力する。
- cut** 第1インレット：**bag**オブジェクトに最初に格納された数値を出力し、**bag**オブジェクトに格納されている中から削除する。

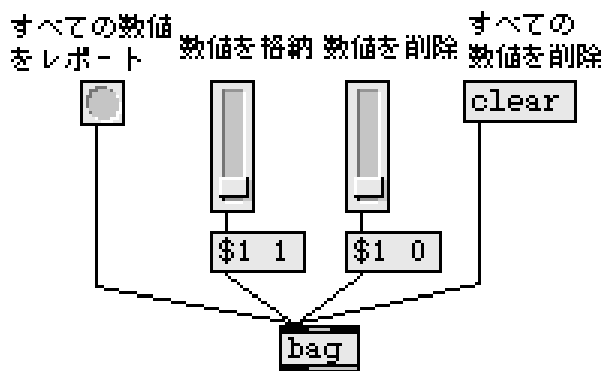
アーギュメント

- 任意のシンボル** オプション。アーギュメントにシンボルを書き込むと、**bag**オブジェクトに同じ数値が何度も格納される。アーギュメントに何も書き込まなければ、同じ数値は1度しか格納されない。このアーギュメントは、必ずしも数値である必要はない。

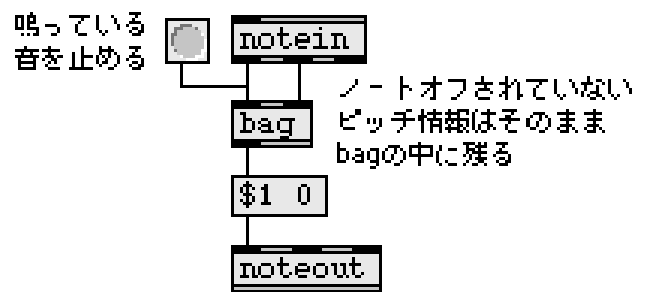
出力

整数 第1インレットに**bang**を受け取れば、**bag**オブジェクトに格納されている数値が1度にアウトレットから出力される。第1インレットに**cut**を受け取れば、**bag**オブジェクトに最初に格納された数値を出力する。第1インレットに**length**を受け取れば、**bag**オブジェクトに格納されている数値の数を出力する。

利用例



数値の集合を格納する



保持されているノートを見つけるために利用

参照

coll
funbuff
offer

様々な種類のメッセージを格納し、編集する
xとyの数値のペアを対にして格納する
x, yを1組にした数値を一時的に格納する

入力

任意 入力によって、右から左の順番にすべてのアウトレットから**bang**が出力される。

アーギュメント

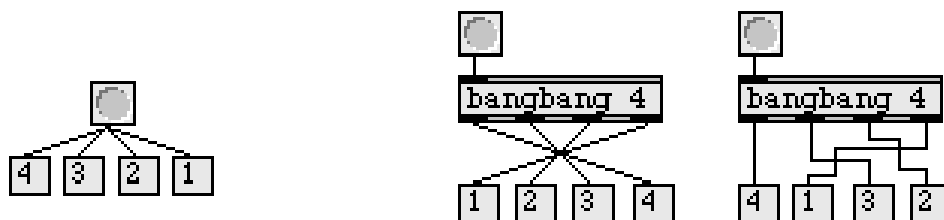
整数 オプション。アウトレットの個数を1つから10個まで指定する。10より大きい数値は、10に設定される。1より小さい数値は、2に設定される。アーギュメントがなければ、2つのアウトレットを持つ。

実数 整数に変換される。

出力

bang メッセージが入力された時、右から左の順番に**bang**がすべてのアウトレットから出力される。

利用例



通常の右から左への順序

bangbang オブジェクトによって順序付けを行なう

参照

button trigger

任意のメッセージによって動作し、**bang**を送信する
入力を順番に従って様々な場所へと出力する

Tutorial 7

Right-to-left order (英文PDF参照)

入力

- (MIDI) MIDI入力デバイスから送信されたMIDIピッチ・ベンド・メッセージを受信する。
- enable** **enable 0**のメッセージはオブジェクトを使用不能にし、受信するMIDIデータを無視するようになる。**pcontrol**オブジェクトやMIDIEnable/Disableボタンによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを再び受信可能にする。
- port** **port**の後にaからzの文字を続けるか、MIDI入力デバイスの名前を続けることによって、ピッチ・ベンド・メッセージを受信するポートを設定する。**port**という単語はオプションで、省略することもできる。
- (マウス) **bendin**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

アーギュメント

- a - z** オプション。ピッチ・ベンド・メッセージを受信するポートを指定する。アーギュメントを指定しない場合、**bendin**はすべてのポートのすべてのチャンネルから受信する。

(MIDI入力デバイス名)

オプション。入力ポートを指定するために、第1アーギュメントにMIDI入力デバイスの名前を使うことができる。

- a - zと整数** 空白文字で区切られた文字と数値の組み合わせは、ピッチ・ベンド・メッセージを受信するポートとMIDIチャンネルを指定する。チャンネル・ナンバーが16より大きい場合は、1から16の範囲に収まるように変換される。

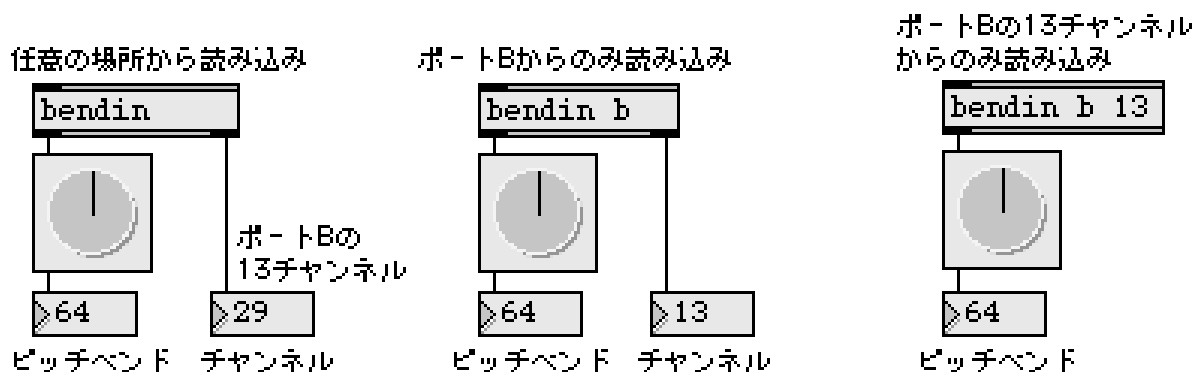
- 整数** 文字と数値の組み合わせの代わりに、数値だけを指定しても良い。チャンネル・ナンバーのアーギュメントの正確な意味は、**MIDI Setup**ダイアログで指定したポートごとのチャンネル・オフセットによって決まる。

出力

- 整数** アーギュメントに特定のチャンネル・ナンバーを指定した場合、アウトレットは1つだけにある。出力は、指定したチャンネルとポートから受け取った、ピッチ・ベンド値である0から127までの数値 (MIDIピッチ・ベンド・メッセージの最上位バイト) を出力する。

アーギュメントにチャンネル・ナンバーを指定しない場合は、第2アウトレットが存在し、そのアウトレットから受信したピッチ・ベンド・メッセージのチャンネル・ナンバーを出力する。

利用例



ピッチ・ベンド・メッセージは、どこでも受信することができ、ポートのみを指定するか、ポートとチャンネルを指定することができる

参照

bendout	MIDIピッチ・ベンド・メッセージを送信する
ctlin	MIDIコントロール・チェンジを受信する
midin	生のMIDIデータを受信する
notein	MIDIノート・メッセージを出力する
rtin	MIDIのリアルタイム・メッセージを出力する
xbendout	MIDIピッチ・ベンド・メッセージ (14ビット) を作成する
xbendin	MIDIピッチ・ベンド・メッセージ (14ビット) を解釈する

Tutorial 16

More MIDI ins and outs (英文PDF参照)

入力

整数 第1インレット：数値は、MIDIピッチ・ベンド・メッセージの値として扱われ、指定されたチャンネルとポートから送信される。数値は0から127の間に制限される。

第2インレット：数値は、ピッチ・ベンド・メッセージを送信するチャンネル・ナンバーとして格納される。

実数 整数に変換される。

リスト 第1インレット：最初の数値はピッチ・ベンド値として、2番目の数値はチャンネル・ナンバーとして、MIDIピッチ・ベンド・メッセージが指定されたチャンネルとポートから送信される。

enable **enable 0**メッセージはオブジェクトを使用不可にし、MIDIデータを送信しないように設定する。MIDI Enable/Disableボタンや**pcontrol**オブジェクトによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを送信可能にする。

port 第1インレット：**port**の後に文字の**a**から**z**の文字を続けるか、MIDI出力デバイスの名前を続けることによって、プログラム・チェンジ・メッセージを送信するポートを設定する。**port**という単語はオプションで、省略することもできる。

(マウス) **bendout**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

アーギュメント

a - z オプション。ピッチ・ベンド・メッセージを送信するポートを指定する。文字のアーギュメントを指定している時に、チャンネル・ナンバーが16より大きければ、1から16の範囲に収まるように変換される。アーギュメントが存在しない場合は、**bendout**は初期状態として、ポート**a**からMIDIチャンネル1として送信する。

a - zと整数 空白で区切られた文字と数値の組み合わせによって、送信するピッチ・ベンド・メッセージのポートとチャンネル・ナンバーを指定する。チャンネル・ナンバーが16より大きければ、1から16の範囲に収まるように変換される。

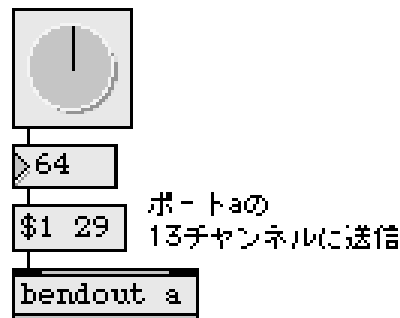
(MIDI出力デバイス名)

オプション。MIDI出力デバイスの名前は、ポートを指定する最初のアーギュメントとして用いることができる。

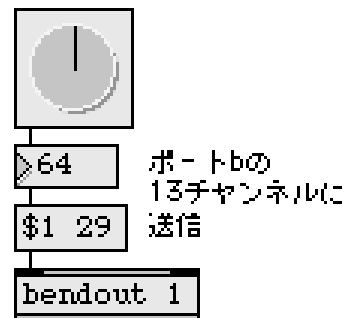
整数 文字と数値の組み合わせの代わりに、数値だけを指定しても良い。チャンネル・ナンバーのアーギュメントの正確な意味は、**MIDI Setup**ダイアログで指定したポートごとのチャンネル・オフセットによって決まる。

出力

(MIDI) アウトレットは存在しない。オブジェクトに指定したMIDI出力ポートから、MIDIピッチ・ベンド・メッセージが送信される。

利用例

文字のアーギュメントにより、1つのポートだけに送信



または、数値によってポートとチャンネルの両方を指定する

参照

bendin MIDIピッチ・ベンド・メッセージを受信する
midout 生のMIDIデータを送信する
xbendin MIDIピッチ・ベンド・メッセージ (14ビット) を解釈する
xbendout MIDIピッチ・ベンド・メッセージ (14ビット) を作成する

Tutorial 16

More MIDI ins and outs (英文PDF参照)

入力

- 任意のメッセージ 任意のインレット：入力はそのインレットに一致している場所に格納されて、入力によって先に格納されたすべてのメッセージをインレットに一致するアウトレットから出力する。特定のインレットにまだ入力がない場合は、それに一致するアウトレットからは**0**を出力する。
- bang** 任意のインレット：直ちに、すべての格納されたメッセージを出力する。
- set** 任意のインレット：**set**に続く任意のメッセージは、出力トリガーとならずにそのインレットに一致する場所に格納される。

アーギュメント

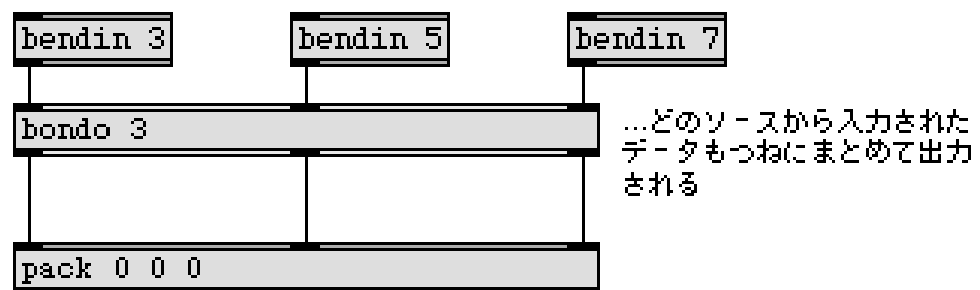
- 整数** オプション。第1アーギュメントはインレットとアウトレットの数を指定する。インレットとアウトレット数の初期値は2。メッセージをアウトレットから出力する前に新たなメッセージが入力される時、第2アーギュメントでは遅らせるための秒数を指定する。

出力

- 任意のメッセージ 何かインレットに格納されたものは、一致するアウトレット・ナンバーから出力される。**bang**がトリガーとなる場合は直ちに出力される。第2アーギュメントに数値がタイプされていてかつメッセージがトリガーとなって出力される場合、出力は第2アーギュメントに指定した秒数によって遅延される。

利用例

3つの異なるMIDIソースから届いたピッチベンドデータは...



bondo は異なるソースから入力したメッセージを同期させることができる

参照

- buddy** 入力されたデータを同期させ、それらを同時に出力する
- onebang** **bang**メッセージの流れをコントロールする
- pack** 数値やシンボルをひとつのリストに結合させる
- thresh** 短時間に受け取った数値をリストとしてまとめる

入力

整数 第1インレット：数値はMIDIノート・オン・メッセージまたはノート・オフ・メッセージ（ベロシティが0であるノート・オン）のノート・ナンバーの値である。ノート・ナンバーは第2インレットでベロシティと組み合わせられる。**Borax**は、すでに保持しているノート・ナンバーに対するノート・オン・メッセージを無視し、またすでにオフにされたノート・ナンバーに対するノート・オフ・メッセージを無視する。ノートが重複しないのであれば、**Borax**はピッチとベロシティ値を他の情報と同様に出力する。

第2インレット：数値は第1インレットに受け取ったノート・ナンバーと組み合わせられてベロシティとして記憶される。

実数 第2インレット：整数に変換される。

リスト 第1インレット：1番目の数値は、ノート・ナンバーとベロシティの組み合わせにおけるノート・ナンバーとして使われ、2番目の数値はベロシティとして記憶される。ノートが重複しないのであれば、**Borax**はノート・ナンバーとベロシティ値を他の情報と同様に出力する。

デルタ 第1インレット：デルタ・タイム（最後のノート・オフから経過した時間）とデルタ・カウント（記録されたデルタ・タイムの数）が出力されるようになる。

bang 第3インレット：**Borax**オブジェクトの受信するすべてのノート情報を削除し、かつ現在保持しているすべてのノートのためのノート・オフを出力することによって**Borax**をリセットし、そして**Borax**のカウンターとクロックを0にセットする。

アーギュメント

なし。

出力

整数 第1アウトレット：**Borax**が受信したそれぞれのノート・オンに対して割り振る一意的な番号で、（最後のリセットから）受信したノート・オンの総数と一致する。この番号はノート・オンが受信された時に出力され、そのノートがオフにされた時には同じ番号が出力される。

第2アウトレット：それぞれのノート・オンに割り振る一意的なボイス番号で、発音されているボイス数と一致する（割り振られている番号のノートがオフになる時はそのボイス番号が出力される）。この番号はノート・オンが受信された時に出力され、そのノートがオフにされた時には同じ番号が出力される。

第3アウトレット：ノート・オンまたはノート・オフが入力されるごとに**Borax**がホールドしているノート数を出力する。

第4アウトレット：ノート・オンまたはノート・オフのノート・ナンバーが出力される。

第5アウトレット：ノート・オンまたはノート・オフのベロシティが出力される。

第6アウトレット：ノート・オフを受信した時、ノート・オフされたノートの総数が出力される。

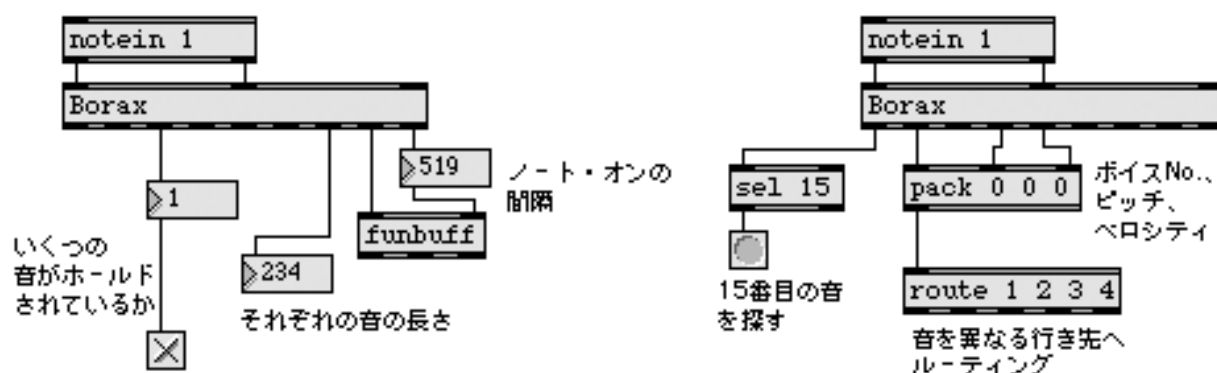
第7アウトレット：ノート・オフを受信した時、そのノートのデュレーションがミリ秒で出力される。

第8アウトレット：デルタ・タイムが記録されるごとに、デルタ・タイムのカウント数が出力される。

第9アウトレット：ノート・オンを受信するき、デルタ・タイム（直前のノート・オンから経過した時間）をミリ秒で出力する。第1インレットでのデルタ・メッセージとは同じ出力となる。

第3インレットに入力された**bang**は現在保持しているすべてのノートに対してノート・オフを行う。このノート・オフは、実際にノート・オフの情報が入力されたかのように同じ出力をトリガーする。

利用例



Borax は通過していくノートに関する広範囲な情報を提示する

参照

midiparse
poly

生のMIDIデータを解釈する
 各ノートにボイス番号をつける



入力

- すべて **bpatcher**オブジェクトのインレットの数は、そのサブパッチ・ウィンドウに含まれる**Inlet**オブジェクトの数によって決まる。**bpatcher**で使用されているパッチが**Inlet**オブジェクトを含むならば、それらは**bpatcher**オブジェクト・ボックス内に左から右へ順番にインレットとして現れる。
- offset** **bpatcher**で使われているサブパッチが、**Inlet**オブジェクトのうちの一つに接続している**thispatcher**オブジェクトを含むならば、対応する**bpatcher**のインレットで受け取られた**offset**メッセージによって、サブパッチの表示を変化することができる。**offset**の後に2つの整数が続かなければならず、subpatchの左上の角が**bpatcher**の範囲内で水平に、そして垂直にオフセットされるピクセル数を指定しなければならない。このように、一つの**bpatcher**で別々なsubpatchの表示が示せれるように使うことができる。部分的にオブジェクト・ボックスの範囲外にあるサブパッチのユーザーインターフェイスは、それらのインレットで受けたメッセージに応じて、(**bpatcher**の境界線の範囲外を)完全に再描画する。よって、完全に**bpatcher**オブジェクト・ボックスの範囲外と範囲内の両方に存在しているサブパッチのユーザーインターフェイスを提示することができる。
- border** **bpatcher**で使われているサブパッチが、**Inlet**オブジェクトのうちの一つに接続している**thispatcher**オブジェクトを含むならば、そのインレットに0以外の数を伴う**border**というメッセージによって、**bpatcher**の周囲に黒色の境界線を描けるようになる。**border 0**というメッセージは**bpatcher**の境界線を消す (デフォルト状態)。
- (マウス) **bpatcher**を含んでいるウィンドウがロックされている (もしくはcommandキー (Macintosh) またはCtrlキー (Windows) を押している状態) で、**bpatcher**オブジェクトの範囲内をマウスでクリックする時、ボックス内のパッチを制御することができる。
- shift+commandキー (Macintosh) もしくはshift+Ctrlキー (Windows) を押しながら**bpatcher**をクリックしてマウス・ドラッグすることによってボックス内のパッチの可視部分の左上角を動かすことができる。パッチ・ウィンドウのアシスタンス・エリアではオフセットのピクセル値を表示する。**bpatcher**のインスペクター・ウィンドウにおいて *Enable Drag-Scrolling* にチェックが入っていないならば、この機能は使用できない。
- bpatcher**を、controlキーを押しながらクリック (Macintosh) もしくは右クリック (Windows) した場合、コンテキスト・メニューによって、そのウィンドウ内のボックス内部に含まれるパッチのファイルを開いて



現在含まれているパッチを変更することができる。

インスペクター

bpatcherオブジェクトの振る舞いは、そのインスペクターによって編集することができる。Windowsメニューから**Show Floating Inspector**を選択しフローティング・インスペクターを可能にした場合に、**bpatcher**を選択するとフローティング・ウィンドウとして**bpatcher**のインスペクターを表示する。オブジェクトを選択し、Objectメニューの**Get Info...**を選ぶことによって、そのインスペクターを表示することができる。

bpatcherのインスペクターでは以下の属性を設定できる。

*Offset*では、表示させるものの左上の角が**bpatcher**領域の左上の角から垂直と水平にオフセットされる数値を指定する。デフォルトでは、表示させるものの左上の角は**bpatcher**の左上の角になっている（すなわち、オフセットは0,0）。このオフセットはナンバー・ボックスで新規にピクセル値を入力することによって変更することができる。デフォルトはオフセットなし（すなわち、水平0、垂直0）。

*Offset*ナンバーボックスは、サブパッチの左上の角が**bpatcher**オブジェクトの表示領域内で水平、垂直にオフセットされる数値を指定するために使用する。デフォルト値は水平、垂直両方とも0である。

*Border*チェックボックスをチェックすると黒色の境界線が**bpatcher**の周りに表示される。デフォルト表示はチェックなし（境界線なし）。

*Embed Patcher in Parent*チェックボックスは、別々のファイルで保存されているサブパッチではなく、（パッチ・オブジェクトと同様に）メイン・パッチの一部としてサブパッチを保存し、埋め込むことができる。デフォルトはチェックなし（サブパッチは別々のファイルとして保存される）。

*Enable Drag-Scrolling*チェックボックスをチェックすれば、Shift+Commandキー (Macintosh) あるいはShift+controlキー (Windows) を押しながら**bpatcher**をクリックしてマウスをドラッグすることにより、ボックス内にある可視部分の左上の角を動かすことが可能になる。デフォルトはチェックなし（ドラッグ・スクロールは使用できない）。

*Patcher File*オプションは、*Open*ボタンをクリックすることによって、使用される**bpatcher**のパッチ・ファイルを選択する。現在のファイル名はボタンの左側のテキスト・ボックスに表示される。また、このボックスにそのファイル名を入力するか、このボックスにFinderからファイルのアイコンをドラッグすることによってもパッチ・ファイルの



選択ができる。

*Arguments to Patcher*では、メイン・パッチとともに保存されるパッチに対してのアーギュメントを設定する。

*Revert*ボタンはインスペクターを開いてからのオブジェクトの設定値の変更をすべて元に戻す。また、インスペクターを開いている時に、Editメニューから**Undo Inspector Changes**を選ぶことによって、インスペクターウィンドウを開く前のオブジェクトの状態に戻すこともできる。

アーギュメント

なし。

出力

bpatcherが使用するパッチが**outlet**オブジェクトを持つならば、左から右への順序で**bpatcher**オブジェクト・ボックスのアウトレットとして現れる。

利用例



サブパッチの内容を表示する

このパッチの内容をウィンドウに表示することができ...

..*offset*メッセージを使って小さな**bpatcher**内容にできる

参照

patcher
pcontrol
thispatcher

パッチの中でサブ・パッチを作る
パッチ内のサブ・ウィンドウを開いたり、閉じたりする
パッチにメッセージを送る

Tutorial 27
Tutorial 28

Your object (英文PDF参照)
Your argument (英文PDF参照)

入力

- 整数** **Bucket**に現在格納されている数値が出力され、各数値は右のアウトレットに移動し、次に数値を受け取った時に第1アウトレットから出力する数値として格納される。
- 実数** 整数に変換される。
- リスト** リストの中の最初の数値だけが使われる。
- bang** すべての格納された数値が出力されるが、それらの数値は移動しない。
- freeze** **Bucket**の出力を保留するが、新しく入ってくる数値は内部において移動しながら格納し続ける。
- thaw** **Bucket**の出力を再開する。
- roll** 任意の数値が続く**roll**というメッセージによって、**Bucket**が一番右に格納された数値をインプットとして使うようになる。従って、すべて格納された数値は右に移動して出力され、そして一番右にあった数値は一番左にあたかもインレットで受け取ったように格納される。
- L2R** インレットに数値を受け取る時にはいつでも、**Bucket**が左から右へ数値を移動し格納するように設定する（デフォルト）。
- R2L** インレットに数値を受け取る時にはいつでも、**Bucket**が右から左へ数値を移動し格納するように設定する。一番右のアウトレットに入力された数値があてがわれる。
- set** 数値が続く**set**というメッセージは、各アウトレットにその数値を送り、各アウトレットから送信される次の値として格納される。
- l2r** **L2R**と同じ。
- r2l** **R2L**と同じ。

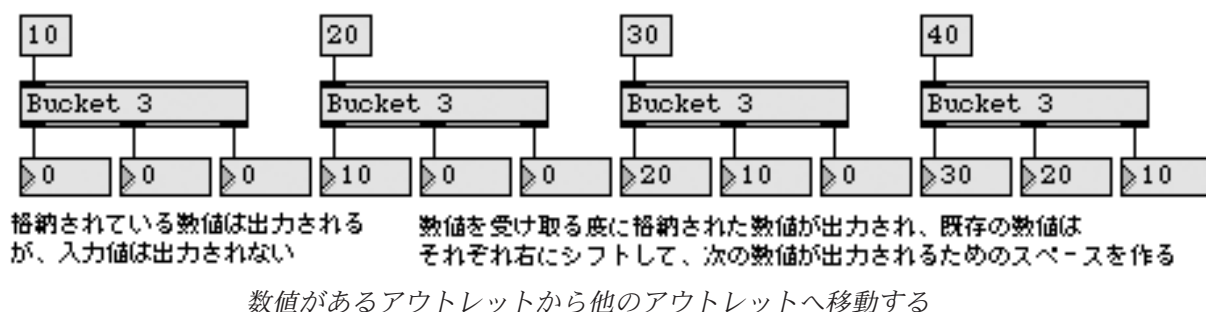
アーギュメント

- 整数** オプション。アウトレット数を設定する。アーギュメントがない場合は1つのアウトレットとなる。

出力

整数 数値が受け取られると、その数値はすぐに送信されないが、**Bucket**に格納された数値が出力される。数値はすべて右のアウトレットへ移動し、受け取った数値は一番左のアウトレットに格納される。

利用例



参照

cycle
Decode
gate
spray

独立したアウトレットからデータの流れを出力する
1から0を特定のアウトレットから出力する
入力を特定のアウトレットから出力する
整数を、アウトレットの番号に従って分配する

入力

任意のメッセージ 任意のインレット：データをすべてのインレットに入力すると、そのインレットに対応するアウトレットからデータが出力される。そして、すべてのインレットに再びデータが入力されるまで、出力を待つ。

clear 第1インレット：**buddy**オブジェクトに格納されているデータを消去する。

bang 任意のインレット：数値の0と同じである。

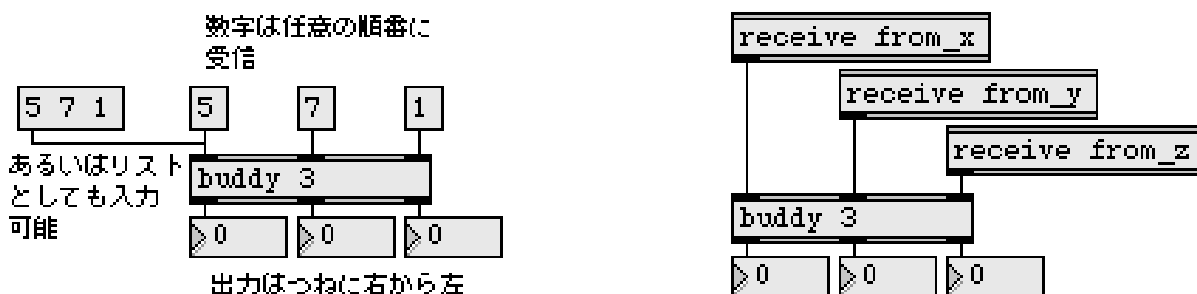
アーギュメント

整数 オプション。インレット（およびアウトレット）の数を指定する。アーギュメントの指定をしない場合は、インレットとアウトレットは2つずつ作成される。

出力

任意のメッセージ すべてのインレットにデータを受け取れば、右から左の順番で、対応するアウトレットからデータが出力される。

利用例



参照

bondo
onebang
pack
swap
thresh
unpack

一群のメッセージを同期する
bangメッセージの流れをコントロールする
数値やシンボルをひとつのリストに結合させる
2つの数値の一連の順番を逆にする
短時間に受け取った数値をリストとしてまとめる
リストを個々のメッセージに分割する



入力

color 後に0から15までの数字が続く**color**というメッセージ、またはObjectメニューの**Color**コマンドから利用可能なオブジェクト・カラーのうちの一つによって、**button**内の円の色を設定する。**button**が**bang**を送信する時は常に黄色が点滅する。

任意のメッセージ 任意のメッセージをインレットに受け取ると、**button**は短く点滅しアウトレットから**bang**を出力する。

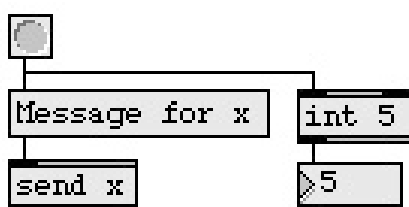
アーギュメント

なし。

出力

bang マウスクリックまたは任意のメッセージによって、**button**が点滅し**bang**を出力する。**button**の上でのマウスクリックは同様の効果を持つ。

利用例



他のメッセージをトリガーし、処理を行わせる



他のメッセージを **bang** に変換

参照

bangbang 順番に複数の**bang**を出力する
matrixctrl マトリックス状のスイッチ・コントロール
pictctrl ピクチャーを用いたコントロール
trigger 入力を順番に従って様々な場所へと出力する
ubutton 透明なボタン、**bang**を出力する

Tutorial 2 **bang** means “Do it!” (英文PDF参照)

入力

- 整数** 入力された順に数値を格納する。
- リスト** すべての整数によるリストの数値は左から右へ順番に格納される。実数やシンボルは無視される。
- clear** **capture**オブジェクトに格納されている内容を消去する。
- count** 格納されている数値の個数をMaxウィンドウに表示する
- dump** **capture**オブジェクトの内容を順番に出力する。
- open** **capture**オブジェクトをダブルクリックするのと同様で、**capture**オブジェクトに格納されている数値が表示されているテキスト・ウィンドウが開く。
- wclose** テキスト・ウィンドウを閉じる。
- write** **write**メッセージに続くシンボルによって、**capture**オブジェクトの内容がテキストファイルで保存される。保存されたファイルは、Maxアプリケーションと同じフォルダに保存される。シンボルはファイル名として使用する。シンボルにフルパス名（例、“**C:/Max Folder/extras/mystuff/mypatch.pat**”）など、何も書かれていない場合は、標準ファイル保存ダイアログが開く。

アーギュメント

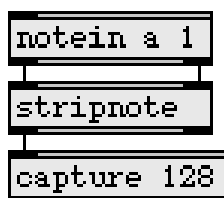
- 整数** オプション。第1アーギュメントで、数値を格納する個数の最大数を設定する。アーギュメントに何も設定していない状態では、512個まで格納できる。入力が最大数を越えた時に、最も過去に格納された数値が消去され、代わりに新しく入力された数値が格納される。
- xまたはm** オプション。第2アーギュメントが**x**の時には、すべての数値は16進数でテキスト・ウィンドウに表示される。第2アーギュメントが**m**ならば、128以下は10進数、128以上は16進数で表示される。アーギュメントに何も指定しなければ、すべての数値は10進数で表示される。

出力

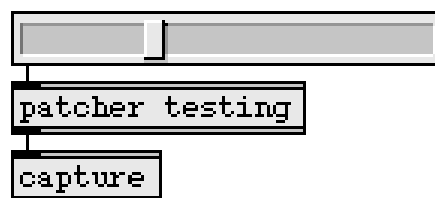
- 整数** 格納された内容が**dump**メッセージによって、第1アウトレットから順番に出力される。
- パッチ・ウィンドウがロックの状態では**capture**オブジェクトをダブル

クリックすると、テキスト・ウィンドウが開き格納されている数値が表示される。テキスト・ウィンドウは内容を変更しないが、値をテーブルなど、別のファイルにペーストする時に役立つ。また、**capture** オブジェクトは、テキスト・ウィンドウが開いている間も数値を格納し続けている。

利用例



数値を集めて、テーブルにペーストする・・・



・・・または、何が起きているのか確認する

参照

Text

数値をテキスト・ファイルとしてフォーマットする

Tutorial 34

Managing raw MIDI data (英文PDF参照)

入力

実数 第1インレット：周波数領域での値の実数部。これは振幅と位相の極座標に変換される。

第2インレット：周波数領域での値の虚数部。これは振幅と位相の極座標に変換される。

アーギュメント

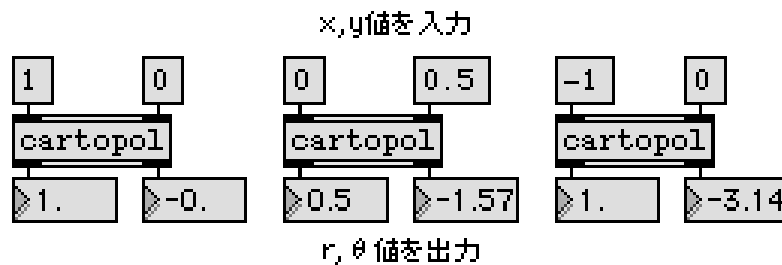
なし。

出力

実数 第1アウトレット：入力された値の極座標における振幅。

第2アウトレット：入力された値の極座標における位相。第1アウトレットだけが接続されている場合は、位相の計算は行なわれない。

利用例



極座標を直交座標に変換する

参照

atan2

アーク・タンジェント関数 (2変数)

lcd

パッチ・ウィンドウ内に画像を描画する

poltoCAR

極座標を直交座標に変換する

pow

累乗を計算する

注記：このオブジェクトは、Max 4.2以降のバージョンでは使用されない

入力

- 整数** 0でCDドライブを停止する。1から始まるCDのトラック番号で、そのトラックの始めから再生を始める。
- 実数** 浮動小数点で表される秒数から再生を始める。値は**cd**オブジェクトによって、分、秒、ブロックに置き換えられる。
- リスト** 3つの数字のリストは、分、秒、およびブロック（75分の1秒）を指定し、CDドライブは指定された位置から再生を始める。
- bang** **cd**オブジェクトの左から3つの分のアウトレットからCDの現在の時間位置を出力する。
- drive** **drive**メッセージに続く数値によって、マルチ・ディスクCDチェンジャー機能の付いたCDドライブに対して、その番号のドライブに切り替える。ドライブ番号は1から始まる。
- eject** 現在挿入されているCDをドライブから取り出す。**eject**メッセージは、FinderでCDアイコンをゴミ箱にドラッグすることと同じである。
- endstop** 設定された停止ポイントを消去する。
- ff** 早送り再生を行う。アーギュメントで何も指定されていなければ、現在の位置から早送り再生をし、1つの数値が指定されていれば、そのトラック番号の始めから再生する。3つの数値からなるリストが指定されている場合は、分/秒/ブロックで指定した部分から再生する。
- fstop** 浮動小数点で表される秒数の位置に停止位置を設定する。値は分、秒、およびブロックに置き換えられる。
- init** CDの内容一覧をMaxウィンドウに表示する。
- pause** CDドライブを一時停止させる。
- resume** CDドライブが一時停止の状態であれば、停止した位置から再生を始める。
- rewind** CDを巻き戻しで再生する。アーギュメントに何も指定していない場合は、再生している位置から巻き戻し再生を始める。1つの数値が指定されている場合は、そのトラックの始めから巻き戻し再生を始める。3つの数値からなるリストで指定されている場合は、分/秒/ブロックで指定した部分から巻き戻し再生をする。

- search** **search**に続く3つの数値（分、秒、ブロック）からなるリストで時間の位置を検索する。検索するだけで、再生はされない。**search**というメッセージなしで、3つの数値から成るリストを送れば、その位置から再生される。**search**メッセージを用いれば、それを送った後のコマンドに対するCDの時間的な応答性が僅かながら改善されることになる。
- select** **select**メッセージに続く、数字または**first**から**sixth**までのメッセージによって、使用するCDドライブを指定する。CDドライブを数値によって指定する時は（例：**select 1**）、第1アーギュメントとして、CDドライブのデバイスIDを指定する。CDドライブが1つだけの場合は、IDとして0から7までのどの数値を指定しても構わない。0を指定すると、ATAPIドライブが指定され、1から7まではSCSIドライブが指定される。任意である第2アーギュメントは、選択されたCDドライブのバッファ・サイズを設定する。
- stop** アーギュメントがない場合は、CDドライブを停止させる。3つの数値のリストが後に続く場合は、指定された分、秒、ブロックの位置で自動的に停止するように設定し、CDの再生が指定された位置に達すると、自動的に停止する。**endstop**によって停止位置を削除するまで、自動停止位置より先の部分は、コンピューターを再起動しても再生されないことに注意すること。
- toc** CDのインデックスの内容を、リストとして第4アウトレットから出力する。リストは直接**coll**オブジェクトに格納できる形式になっている。リストは、トラック番号に続いて分、秒、ブロックの順に続く。出力される最後のリストには、CDが終わる時間が出力される。**toc**メッセージは、CDのトラック番号と現在位置を表示するパッチを作るために利用できる。
- trackdurs** CDのトラックの長さを設定する。これは、第2アウトレットからトラック番号とその長さ（秒単位の実数）のリストで出力される。
- volume** **volume**に続く0から255までの数値で、左右両方のチャンネルの音量を設定する。0を指定すると消音になり、255で最大音量になる。**volume**の後に、同じ数値範囲の2つの数値が続く場合は、1つ目の数値は左チャンネルの音量を、2つ目の数値は右チャンネルの音量を設定する。

アーギュメント

- 整数** **cd**オブジェクトは2つのアーギュメントを持ち、いずれも任意である（2番目のアーギュメントを指定するためには、両方のアーギュメントを指定しなければならない）。最初のアーギュメントは、CDドライブのデバイスIDである。1つのドライブしかない場合には、0~7のどの数値を指定しても構わない。0はATAPIドライブで、1から7まではSCSIドライブである。2番目のアーギュメントは、CDが再生されてい

る間、**cd**オブジェクトがどれぐらいの頻度でCDの時間位置を出力するかを指定する。0を指定すると、**bang**を送るまで、**cd**オブジェクトは時間位置を出力しない。

出力

cdオブジェクトが作られる時に、Maxウィンドウに、現在のCDのインデックス（それぞれのトラックの始まりの分、秒、ブロック）を表示する。この表示は、分、秒、ブロックでCDの時間位置を指定する時のガイドとして使用することができる。

また、上で述べたように、**cd**オブジェクトの2番目のアーギュメントが0（初期値）ならば、時間位置は出力されない。2番目のアーギュメントが指定されれば、指定された時間間隔で現在の時間位置が出力される。

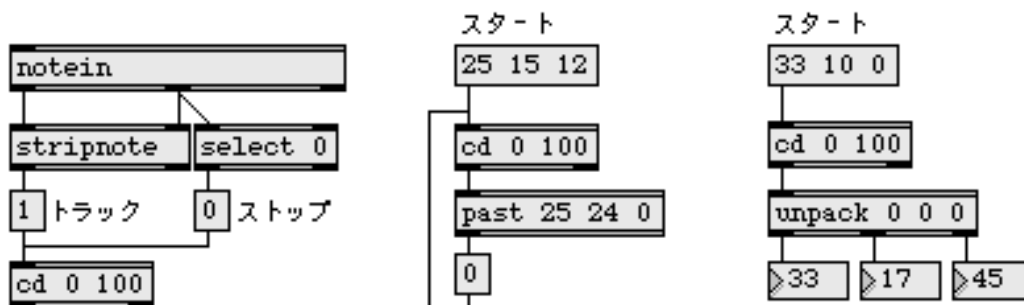
整数 第1アウトレット：再生されているCDの分を出力する。

第2アウトレット：再生されているCDの秒を出力する。

第3アウトレット：再生されているCDのブロックを出力する。1秒は75ブロックである。

リスト 第4アウトレット：一連のリストは**toc**メッセージに対応した出力である。各リストは、トラック番号に続いて、トラックが始まる分、秒、ブロックの値である。

利用例



ノートオンが入ることにより、8秒63ブロック再生後に停止するトラック1を制御する

CDトラックの現在の時間位置を表示する

参照

appledvd
past
vdv

Apple DVD Playerアプリケーションを制御する

数値がある値を超えた時に出力する

シリアル・ポートを通じて、ビデオディスク・プレーヤーをコントロールする

入力

- 整数または実数** **change**オブジェクトに格納されている数値と、入力された数値が異なる場合に、その数値が出力され、格納されている数値と入れ替える。
- set** **set**メッセージに続く数値を入力すると、**change**オブジェクトに格納されている値を出力せずに、入れ替える。
- mode** **mode**メッセージに続く+を入力すると、入力された値が以前に入力された値よりも大きい場合だけ、第1アウトレットから**1**が出力される。この状態では、**change**オブジェクトは他の入力を受け付けない。また、**mode**メッセージに続く-を入力すると、入力された値が以前に入力された値よりも小さい場合だけ、第1アウトレットから**-1**が出力される。この状態でも、**change**オブジェクトは他の入力を受け付けない。**mode**メッセージだけであれば、**change**オブジェクトに格納されている数値と入力された数値が異なる場合にだけ、その数値が出力されるようになる。これはデフォルトのモードである。

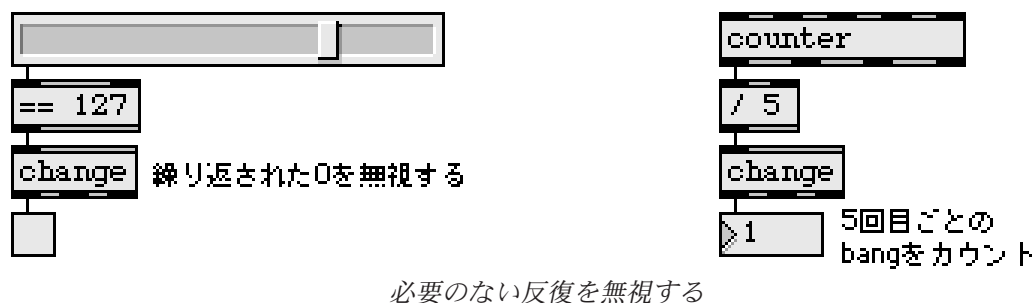
アーギュメント

- 整数または実数** オプション。**change**オブジェクトで比較するための初期値を指定する。このアーギュメントを指定しないならば、初期値は0である。
- symbol** 第2アーギュメントとして+または-を指定すると、**change**オブジェクトはそれぞれのモードに切り替わる。その後の**mode**メッセージでも、モードを切り替えることができる。

出力

- 整数** 第1アウトレット：入力された数値が格納されている数値と異なる場合に、その数値が出力される。
- 第2アウトレット：格納されている値が0で、入力した値が0でない場合には**1**、それ以外は何も出力されない。
- 第3アウトレット：格納されている値が0以外で、入力した値が0の場合には**1**、それ以外は何も出力されない。

利用例



参照

Peak	数値が直前の値より上回る時に出力する
TogEdge	0と0以外の数値の変化を出力する
Trough	数値が以前の値より小さければ、それを出力する
!=	2つの数値の比較し、等しくなければ1を出力する

Tutorial 15 Making decisions with comparisons (英文PDF参照)

入力

- 整数または実数** 第1インレット：入力する数値がアーギュメントで指定された最小値と最大値の範囲内で抑制され出力される。実数を入力したら、そのまま実数として出力される。
- 第2インレット：**clip**オブジェクトに指定する最小値を入力する。
- 第3インレット：**clip**オブジェクトに指定する最大値を入力する。
- リスト** リスト内のそれぞれの数は、アーギュメントで指定された最小値と最大値の範囲内ならば、そのまま出力され、範囲外だったら、強制的に指定した最小値、最大値に変換され出力される。
- set** **set**メッセージにつづく二つの数値で、**clip**オブジェクトに指定する最小値と最大値を指定する。

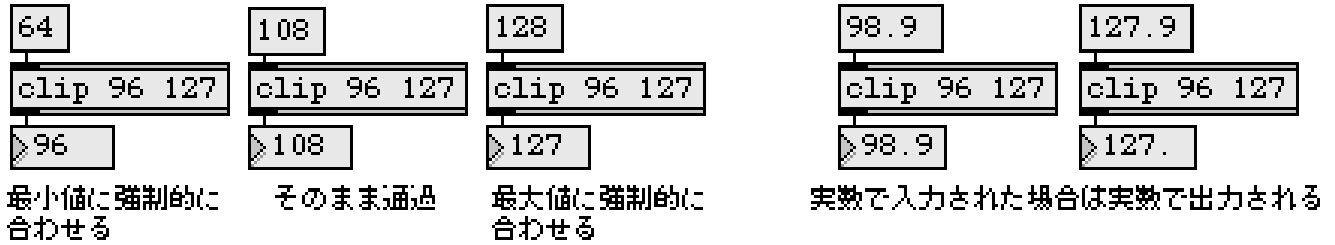
アーギュメント

- 整数または実数** オプション。最初の数値は最小値を、2番目の数値は最大値を指定し、アウトレットから出力する前に、受け取った数値をこれらの範囲に強制的に変更する。アーギュメントとして1つの数値のみを指定した場合は、その数値を最小値と最大値として扱う。アーギュメントを指定しなければ、最小値と最大値はともに0となる。

出力

- 整数** 整数がインレットに入力された時に、指定されたアーギュメントの範囲に変更された数値が出力される。入力が指定した最小値と最大値の範囲内であれば、そのまま数値は出力される。最小値以下なら、最小値が出力される。最大値以上なら、最大値が出力される。
- 実数** 実数がインレットに入力された時にも、指定された最小値と最大値の中の範囲に変更された数値が出力され、実数で出力される。
- リスト** リストがインレットに入力された時は、リストに含まれる数値のそれぞれが、指定された最小値と最大値の範囲に変更され、リストとして出力される。

利用例



数は指定された範囲内に保たれる

参照

maximum

数値のリストの中で最も大きい値を出力する

minimum

数値のリストの中で最も小さい値を出力する

split

数値を範囲によって振り分ける

<

2つの数値を比較し、より小さいことを判断する

<=

2つの数値を比較し、より小さいか等しいことを判断する

>

2つの数値を比較し、より大きいことを判断する

>=

2つの数値を比較し、より大きいか等しいことを判断する

入力

整数または実数 第1インレット：0以外の数値を入力すれば、**clocker**オブジェクトが動作する。**clocker**オブジェクトが動作すると、時間が一定の間隔で出力され、0を入力すると止まる。**clocker**オブジェクトの動作中に0以外の数値が入力されれば、いったん時間が0にリセットされ、その時点からの時間が一定の間隔で出力される。

第2インレット：出力する時間間隔をミリ秒単位で入力する。ここに入力された時間間隔で、**clocker**オブジェクトは経過時間を出力する。第2インレットに数値を入力しても、次の**clocker**オブジェクトの時間が出力されるまでは、時間間隔は変更されない。

bang 第1インレット：**clocker**オブジェクトを動作させる。

stop 第1インレット：**clocker**オブジェクトを停止させる。

clock 既存の**setclock**オブジェクトの名前が後に続いた**clock**メッセージによって、**clocker**オブジェクトにMax内部のミリセカンド・クロックでなく、**setclock**オブジェクトによってコントロールされるように設定する。**clock**メッセージだけでは、Maxの通常のミリセカンド・クロックを使用する設定に戻す。

reset 時計を停止や再開せずに、経過時間を0にリセットする。**clocker**オブジェクトは同じ時間間隔で、新しい経過時間を出力し続ける。**clocker**オブジェクトが動作していない場合は、経過時間は常に0にリセットされているので、このメッセージは無意味である。

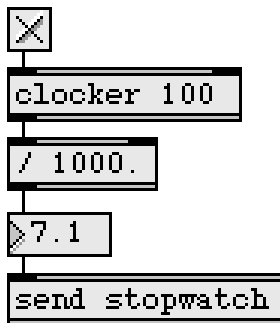
アーギュメント

整数 第2アーギュメントが1の場合には、**clocker**オブジェクトはOMSのタイミングによってコントロールされる（OMSがインストールされていない場合は、MIDIマネージャーを用いる）。これは他のOMS互換アプリケーションとMaxを連動させるために役立つ。第2アーギュメントが0または存在しないならば、**clocker**オブジェクトはMaxの内部ミリセカンド・クロックを使用する。

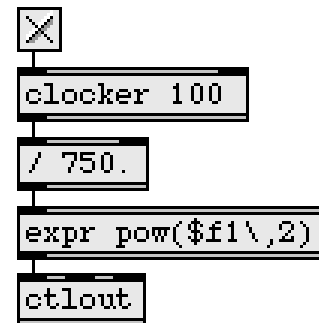
出力

整数 **clocker**オブジェクトを動作させると、その時点からの経過時間がミリ秒単位で出力される。最初の出力は常に0であり、**clocker**オブジェクトが動作すると、ただちに出力される。

利用例



経過時間を得る



時間の関数として数値を発生させる

参照

metro
setclock

一定の時間間隔で**bang**メッセージを出力する
タイミング関連のオブジェクトのクロック速度をリモート・コントロールする

tempo
timer

メトロノームのようなテンポで数値を出力する
2つのイベント間の経過時間を出力する

Tutorial 31

Using timers (英文PDF参照)

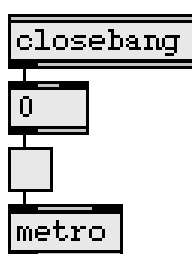
入力

インレットは存在しない。パッチ・ウィンドウが閉じた時に出力が発生する。

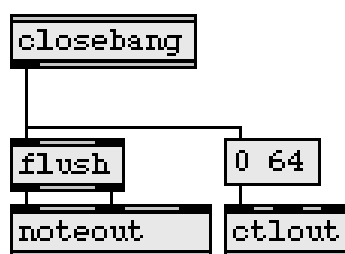
出力

bang パッチ・ウィンドウを閉じると、自動的に**bang**が出力される。

利用例



ウィンドウが閉じられた時に、
動作のプロセスを止める



ウィンドウが閉じられた時に、
音とサスティン・ペダルを切る

参照

active
button
loadbang

パッチ・ウィンドウがアクティブであれば1を、そうでなければ0を出力
任意のメッセージによって動作し、**bang**を送信する
パッチが読み込まれた時に自動的に**bang**を出力する

Tutorial 40

Automatic actions (英文PDF参照)

入力

- リスト** 1番目の数値をアドレス (**coll**内の格納場所) として、その数値に続くリスト (**coll**は構成要素が250までのリストを格納できる) を格納する。アドレスは常に整数で与える。
- 整数** **coll**に格納されているメッセージのアドレスを検索する。そのアドレスに何らかのメッセージが格納されている場合は、そのメッセージが第1アウトレットから出力される。
- 実数** 単独でアドレスの指定として受け取った場合は、整数に変換される。格納されるメッセージの一部として入力した場合は、実数のまま格納される。
- bang** 効果は**next**メッセージと同様である。
- (Get info..)** **coll**オブジェクトは格納されている内容を、その**coll**オブジェクトを含むパッチの一部として保存することができる。そのためには、パッチ・ウィンドウがアンロック状態の時、**coll**オブジェクトを選択し、Objectメニューから**Get Info..**を選択し、*Save coll with patcher*にチェックを入れれば良い。
- assoc** **assoc**はシンボルと数値をアーギュメントに持つ。その数値のアドレスが既に存在している場合、このシンボルと数値によって、シンボルとアドレスを関連付ける。関連付けを行うと、**coll**が受け取ったシンボルはアドレスに変換されて解釈され、シンボルによっても関連づけられたアドレスのメッセージを出力させることができる。0以外の数値アドレス1つにつき、最大で1つのシンボルを関連付けることができる。0のアドレスにはシンボルを関連付けることはできない。(注記：特定のシンボルが既にアドレスとして使用されている、または関連付けられている場合に、このメッセージを入力すると、それまでアドレスに関連付けられていたシンボルの情報は消えてしまう。)
- clear** 格納されているすべての情報を消去する。
- deassoc** **deassoc**はシンボルと数値をアーギュメントに持ち、指定したシンボルとアドレスの関連付けを解除する。解除されたシンボルは**coll**において何の意味もなさなくなる。
- delete** 機能は**remove**メッセージと類似している。ただし、指定したアドレスが数値である場合、それより大きいすべてのアドレスの値を1ずつ減らす。
- dump** 第2アウトレットから格納しているメッセージのアドレスを、第1アウトレットからはメッセージを、格納されている順にすべて出力する。**dump**の処理が終わると、第4アウトレットから**bang**が出力される。

- end** アドレスのポインター (**goto**、**next**、**prev**メッセージでも使われる) を、**coll**内の最後のアドレスに設定する。
- filetype** **filetype**はシンボルをアーギュメントに持ち、**coll**オブジェクトが読み書きするファイル・タイプを指定する。ファイル・タイプは標準的な4つの文字からなる文字列で指定する (例えば、**filetype ffoo**)。アーギュメントを持たない**filetype**メッセージを受け取ると、ファイルタイプの設定は初期値であるMax binaryとテキストファイルに戻される。
- flags** 通常、**coll**の内容はパッチャーウィンドウを閉じる時に、パッチとともに保存されない。**flags 1 0**というメッセージを入力すると、**coll**オブジェクトの内容はその**coll**オブジェクトを含むパッチの一部として保存されるようになる。**flags 0 0**というメッセージを入力すると、**coll**オブジェクトの内容は保存されなくなる。
- goto** **goto**は数値またはシンボルをアーギュメントとして持ち、ポインターを数値またはシンボルで指定したアドレスに設定する。指定したアドレスが存在しない場合は、ポインターは先頭のアドレスに設定される。標準の初期設定では、ポインターは先頭のアドレスに設定されている。
- insert** **insert**は数値とメッセージをアーギュメントとして持ち、数値によって指定されたアドレスにメッセージを挿入する。必要であれば、指定した数値以上のすべてのアドレスの値に1を加算する。
- length** **coll**に格納されているメッセージの個数を第1アウトレットから出力する。**length**メッセージは**grab**オブジェクトと組み合わせて使用すると、うまく使うことができる。
- max** **coll**に格納されている単独の数値 (リストやシンボルは含まない) の中で、最大の値を第1アウトレットから出力する。**max**メッセージは**grab**オブジェクトと組み合わせて使用すると、うまく使うことができる。
- merge** **merge**はアドレスとメッセージをアーギュメントとして持ち、既に指定したアドレスに格納されているメッセージの後にメッセージを追加する。アドレスが存在していない場合は、新たに作成される。
- min** **coll**に格納されている、単独の数値 (リストやシンボルは含まない) の中で、最小の値を第1アウトレットから出力する。**min**メッセージは**grab**オブジェクトと組み合わせて使用すると、うまく使うことができる。
- next** ポインターで指定されているアドレスを第3アウトレットから出力し、そのアドレスに格納されているメッセージを第1アウトレットから出力し、ポインターを次のアドレスに進める。アドレスが数値ではなくシンボルである場合、第3アウトレットからは0が出力される。ポインターが最後のアドレスまで来ると、次は最初のアドレスに戻る。(注

記：数値のアドレスは昇順で格納される。シンボルのアドレスは、すべての数値アドレスの後に、**coll**に追加された順番で格納される。) **next**メッセージの直前に**prev**メッセージを受け取れば、**next**メッセージは直前に出力される値のアドレスよりも、1つ大きいアドレスに格納されている値を出力することになる。

- nstore** **nstore**は数値とシンボル（またはシンボルと数値）、その後に他のメッセージが続く形のアーギュメントを持ち、指定した数値のアドレスにメッセージを格納する。そして同時に、指定したシンボルが指定したアドレスに関連付けられる。（これは、整数のアドレスにメッセージを格納し、その後、**assoc**メッセージを使ってシンボルとこのアドレスを関連付ける場合と同じ機能である。）
- nsub** **nsub**はアドレス、要素の順番、その他に数値またはシンボルを続ける形でアーギュメントを持ち、指定したアドレスに格納されているメッセージの中の一つの要素を置き換える。（例：**nsub pgms 4 7**はアドレス**pgms**に格納されたメッセージの内、4番目の要素を7に置き換える。）数値とシンボルは、この方法で置換することができる。
- nth** **nth**はアドレスと数値をアーギュメントに持ち、指定したアドレスのメッセージからn番目（数値で指定される）の要素を取り出し、第1アウトレットから出力する。（例：**nth pgms 4**は**pgms**というアドレスの格納されているメッセージから4番目の要素を出力する。）
- open** **coll**オブジェクトのテキスト編集ウィンドウを開く。また、このウィンドウは一番手前に開く。
- prev** **next**と同様の出力をするが、ポインタのアドレスは増加するのではなく減少する。ポインタが最初のアドレスにある時は、次は最後のアドレスに移動する。**prev**メッセージの直前に**next**メッセージを受け取ると、**prev**メッセージは直前に出力されたアドレスよりも1つ小さいアドレスに格納されている値を出力することになる。
- read** アーギュメントのない**read**メッセージを受け取ると、標準ファイル・ダイアログが開き、**coll**に読み込むファイルを選択することができる。**read**にアーギュメントとしてシンボルで表すファイル名が続くと、そのファイルを探し出して**coll**に読み込む。
- readagain** 直前に読み込んだファイルを再び読み込む。以前に一度も**read**または**readagain**メッセージを**coll**が受け取っていない場合は、**readagain**メッセージは**read**メッセージとして扱われ、標準ファイル・ダイアログが表示される。
- refer** **refer**には他の**coll**オブジェクトの名前が続き、**coll**オブジェクトが受け取るメッセージは、その名前のオブジェクトのデータに対して実行されるようになる。

ファイルの読み込みやインレットからのメッセージ格納だけでなく、タイプ入力によっても**coll**にデータを格納することができる。**coll**オブジェクトの上にマウス・カーソルを置いてダブルクリックすると、編集ウィンドウが開き、**coll**の内容がテキストとして表示される。このウィンドウ上でデータを修正をすることができる。

手作業での格納データの編集や、他のファイルからの読み込みをするためには、**coll**オブジェクトに格納されるテキスト・データの正しいフォーマットを知っている必要がある。各々のメッセージは**coll**オブジェクト内で別々の行に格納される。それぞれの行のフォーマットは、アドレス（整数またはシンボル）、アドレスに関連付けられたシンボル（アドレスが整数の場合）、カンマ（アドレスとアドレスに格納されているデータを分けるため）、データ（どんなものでも構わない）、そして各行の終わりにセミコロンが置かれる。次のような行があるとすると、

```
3 reset, set 4.7;
```

3はアドレスの数値、**reset**はアドレスに関連付けられたシンボル、そして**set 4.7**がメッセージとして格納される。

100, 200, 300, 400の値をそれぞれアドレスの**1, 2, 3, 4**に格納する場合は、以下の通りである。

```
1, 100;
2, 200;
3, 300;
4, 400;
```

remove **remove**は数値またはシンボルをアーギュメントに持ち、指定したアドレスと含まれるデータを削除する。

renumber **coll**オブジェクト内のデータに連続なアドレスを昇順で関連付ける。アーギュメントとしてアドレスの開始番号を指定する。以下は**coll**に**renumber 1**メッセージを送る前と送った後の例である。

処理前	処理後
4, apple;	1, apple;
6, banana;	2, banana;
3, cherry;	3, cherry;
9, durian;	4, durian;

sort **sort**メッセージは2つのアーギュメントを持つ。1つ目のアーギュメントが-1である場合、**coll**内の項目は昇順に並び替えられる。1つ目のアーギュメントが1の場合、**coll**内の項目は降順に並び替えられる。

2つ目のアーギュメントは並び替えに、どのデータを使用するかを指定する。2つ目のアーギュメントが-1である場合、アドレスの数値（またはシンボル）が使用される。2つ目のアーギュメントがないか、または0である場合、データの最初の項目が使用される。2つ目のアーギュメントが1以上の場合、データの2つ目（以降）の項目が使用される。

store **store**はなんらかの（多くの場合は1単語の）シンボルとメッセージをアーギュメントに持ち、シンボルをアドレスとしてメッセージを格納する。（例：**store triad 0 4 7**はリスト**0 4 7**を**triad**というアドレスに格納する。）

sub **nsub**と同様の機能である。ただし、要素の置き換え後に、指定したアドレスに格納されたメッセージが出力される。

swap **swap**メッセージは、アドレスを示す2つの数値またはシンボルをアーギュメントとして持ち、それぞれのアドレスに格納されたデータを入れ替える。例えば、**coll**の内容が以下の通りである場合、

```
1, 400;
2, 700;
```

ここで、**coll**が**swap 1 2**メッセージを受け取ると、以下のように変化する。

```
1, 700;
2, 400;
```

subsym データにアドレスとして関連付けられたシンボルを変更する。**subsym**の1つ目のアーギュメントは新しくアドレスとして使用されるシンボル、2つ目のアーギュメントは入れ替えられるシンボルである。例えば、**coll**の内容が以下の通りである場合、

```
jill, 40 50 60
```

ここで、**coll**が**subsym jack jill**メッセージを受け取ると、以下のように変化する。

```
jack, 40 50 60
```

シンボル **coll**に格納されているメッセージのアドレスを参照する。このシンボルのアドレスにメッセージが格納されている場合、そのメッセージは第1アウトレットから出力される。シンボルは、必須ではないが、単語**symbol**に続けて使用されることもある。

wclose **coll**オブジェクトのテキスト編集ウィンドウを閉じる。

write 標準ファイル保存ダイアログを表示し、独立したファイルとして**coll**

の内容を保存することができる。**write**にアーギュメントとしてシンボルを与えると、与えたシンボルをファイル名として直ちに**coll**の内容が保存される。

writeagain 直前に**coll**の内容を保存したファイルに上書き保存する。まだ一度も**write**メッセージや**writeagain**メッセージでファイルへの保存を行っていない場合、**writeagain**は**write**メッセージとして扱われ、ファイル保存ダイアログが表示される。

インスペクター

collオブジェクトの振舞はインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**coll**オブジェクトを選択することで、**panel**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**coll**オブジェクトを選び、オブジェクト・メニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

*Save coll with patcher*にチェックを入れると、**coll**オブジェクトの内容はその**coll**オブジェクトを含むパッチの一部として保存されるようになる。

Revertボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

アーギュメント

任意のシンボル オプション。パッチが読み込まれた時に、自動的に**coll**オブジェクトに読み込まれるファイルの名前を指定する。読み込まれるファイルの内容は**coll**で扱える適切なフォーマットでなければならない。同じ名前を持つすべての**coll**オブジェクトはどれも同じデータを共有する。実際のファイルを必要としない場合でも、設定したファイル名は複数の**coll**オブジェクトでデータを共有するための識別子として使用することができる。

出力

任意のメッセージ **coll**に格納されているメッセージが第1アウトレットから出力される。メッセージがシンボル1つのみである場合、シンボルの前に**symbol**が付加されて出力される。

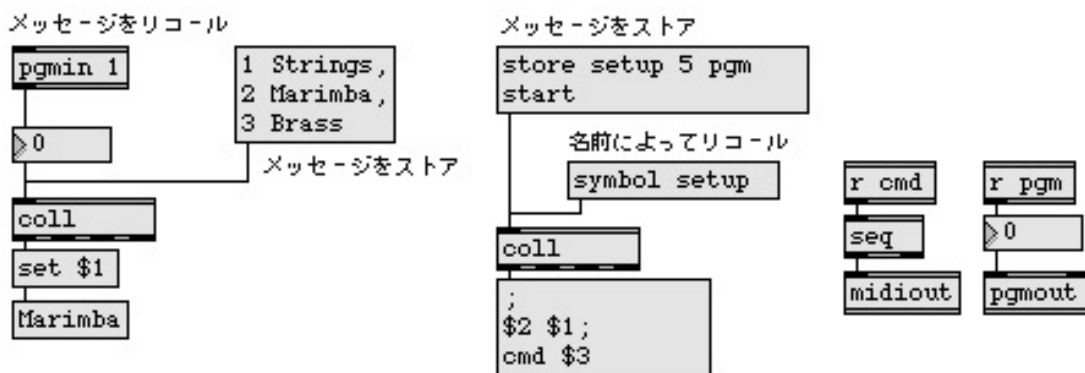
整数 第1アウトレット：**length**メッセージを受け取ると、**coll**に格納されているメッセージの数が出力される。

整数またはシンボル 第2アウトレット：**bang**, **dump**, **next**, **prev**, **sub**のいずれかによって第1アウトレットからメッセージが出力される時に、アドレスが出力される。

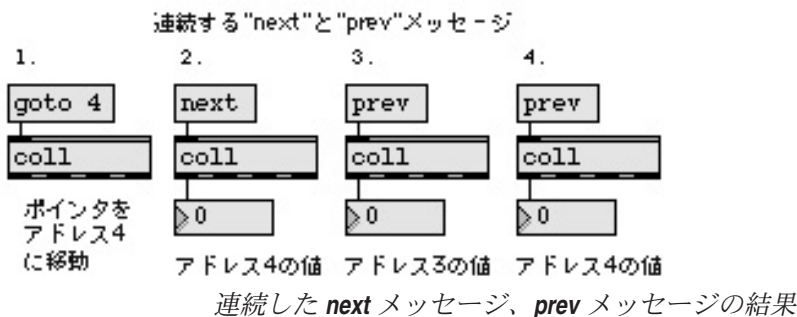
bang 第3アウトレット：ファイルの読み込みや、データのファイルへの保存が完了すると、**bang**が出力される。

第4アウトレット：**dump**メッセージに対し、第1アウトレット、第2アウトレットから、格納されたすべてのアドレスとメッセージの出力が完了すると、**bang**が出力される。

利用例



複雑なメッセージが1つの数値やシンボルで呼び出される



参照

bag 数値の集合を格納する
table 数値の配列をグラフィカルに編集し、格納する
funbuff x と y の数値のペアを対にして格納する

Tutorial 37 Data Structures (英文PDF参照)

colorpickerオブジェクトは、Mac OSのモーダル・ダイアログを使い、複数の異なるカラー・システムで色を選択しする。カラー・システムの種類は、CMYK、HLS、HSV、HTML、RGB、クレヨンモードである。

入力

(マウス) **colorpicker**オブジェクトをダブルクリックすると、モーダル・ダイアログが開く。パッチがロックされていない状態では、commandキー (Macintosh) あるいはcontrolキー (Windows) を押した状態でダブルクリックすると、モーダル・ダイアログが開く。

bang **colorpicker**オブジェクトをダブルクリックする事と同じである。

list 0から255までの3つの数値のリストで、モーダル・ダイアログが開かれる時のデフォルトのRGBカラーを設定する。

setprompt **setprompt**メッセージに続く文字列で、モーダル・ダイアログのラベルを設定する。変更を行った後にダイアログボックスを開いた時点で、この設定が反映される。

出力

リスト モーダル・ダイアログを開き、色を選択した後に、OKボタンをクリックすると、選択した色のRGB値のリストが出力される。モーダル・ダイアログで、Cancelボタンをクリックした場合は、メッセージは出力されない。

利用例



色を表示する、あるいは選択されたRGB値を得る

参照

panel
swatch

色をついた背景領域
RGBカラーの選択と表示のためのカラー・スウォッチ



入力

commentオブジェクトはインレットがなく、入力もない。パッチ・ウィンドウがエディット・モードになっている時に、**comment**オブジェクトのボックスにテキストをキーボード入力できる。パッチ・ウィンドウがロックされている時は、**comment**オブジェクトのボックスの破線の外枠は見えなくなり、書き込んだテキストだけが見える状態になる。フォントやボックスのサイズを変えることによって、**comment**オブジェクトの外観を変更することができる。

commentオブジェクトのフォントとサイズは、Fontメニューで変更することができる。

インスペクター

commentオブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**comment**オブジェクトを選択することで、**comment**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**comment**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

commentオブジェクトのインスペクターでは、以下の設定ができる。

*Two-bite Compatible*オプションをチェックすることによって、日本語や中国語などの2バイトの文字を**comment**オブジェクトに表示することができる（デフォルトではチェックされていない）。

*Color*オプションではカラー・ピッカーやRGBの値によって、**comment**オブジェクトのテキストの色を変えられる。デフォルトのテキストの色は黒（0,0,0）である。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

アーギュメント

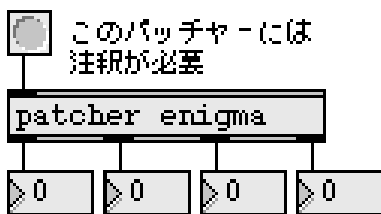
なし。



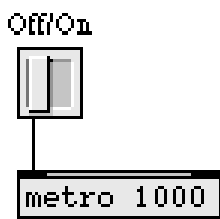
出力

comment オブジェクトにはアウトレットはなく、出力もない。このオブジェクトを利用しても、パッチの機能に影響することはない。

利用例



説明



ラベル

機能を示す
(**ubutton** オブジェクトに覆われている)

参照

ubutton

透明なボタン、**bang**を出力する

Tutorial 5

toggle and comment (英文PDF参照)

注記：このオブジェクトは、Max 4.1以前のバージョンでは使用できない

入力

任意のシンボル シンボルとしてのファイル名かパス。**conformpath**オブジェクトは、パスをある**pathstyle** (例、セパレータとしてコロンあるいはスラッシュを使用するパス形式) および**pathtype** (絶対パス、相対パス、起動ディスクからの相対パス、Cycling 74フォルダを基準とする相対パスのいずれかのパスの種類) から、別の形式および種類へ変換する。これは**absolutepath**および**relativepath**両オブジェクトの機能を包括するものである。

pathstyle **pathstyle**に続けてパス形式を指定すると、出力パス名を指定した形式で表示する。選択できるパス形式には以下のようなものがある。

colon **colon**のパス形式を指定すると、オブジェクト間でパス名をやり取りする際に、コロン(:)をパスの区切りとして使用する。これは、Macintosh版のMax 4.2以前のバージョンで使用されていたパス形式である。

(注記：Macintoshでの標準パス形式はコロンを使用したものなので、native_macというパス形式は用意されていない。)

max デフォルト。**max**のパス形式を指定すると、現在使用しているバージョンのMaxが使用しているパス形式を選択して、オブジェクト間でパス名をやり取りする際に選択する。

native **native**のパス形式を指定すると、現在使用しているOSでの標準形式をパス形式として使用する。

(注記：nativeのパス形式を使用すると、絶対パスのみが使用するパスの種類として選択できる。)

native_win **native_win**のパス形式を指定すると、Windowsでの標準パス形式 (バックスラッシュあるいは日本語環境では¥がパスの区切りとなる) がパス形式として使用される。

注記：**native_win**のパス形式を表示用以外に使用することは推奨しない。Max/MSPでは、バックスラッシュ (\) は特殊文字の特別な役割を無効にするために使用されるため、メッセージボックスや**sprintf**、**coll**などのテキストを解析するオブジェクトで混在させて使用すると問題が生じる場合がある。

slash	slash のパス形式を指定すると、オブジェクト間でパス名をやり取りする際に、スラッシュ (/) をパスの区切りとして使用する。
pathtype	pathtype に続けてパスの種類を指定すると、出力パス名を指定した種類のフォーマットで表示する。選択できるパスの種類には以下のようなものがある。
absolute	absolute を指定すると、フォルダやファイルの出力パス名を絶対パスに変換してシンボルとして出力する。
boot	boot を指定すると、フォルダやファイルの出力パス名を起動ディスクを基準とした相対パスに変換してシンボルとして出力する。もしファイルが起動ディスクから辿れない場合は、 conformpath オブジェクトは第2アウトレットから0を出力し、第1アウトレットからの出力パス名は変換されずにそのまま出力される。
C74	C74 を指定すると、フォルダやファイルの出力パス名をCycling 74フォルダを基準とした相対パスに変換してシンボルとして出力する。もしファイルがCycling 74フォルダから辿れない場合は、 conformpath オブジェクトは第2アウトレットから0を出力し、第1アウトレットからの出力パス名は変換されずにそのまま出力される。
ignore	デフォルト。 ignore を指定すると、フォルダやファイルの出力パス名を変換せずにそのまま出力する。
relative	relative を指定すると、フォルダやファイルの出力パス名をMaxのアプリケーションフォルダを基準とした相対パスに変換してシンボルとして出力する。もしファイルがアプリケーションフォルダから辿れない場合は、 conformpath オブジェクトは第2アウトレットから0を出力し、第1アウトレットからの出力パス名は変換されずにそのまま出力される。

アーギュメント

シンボル オプション。出力に使用されるパスの種類をシンボルで指定することができる。指定できるパスの種類には以下のようなものがある。

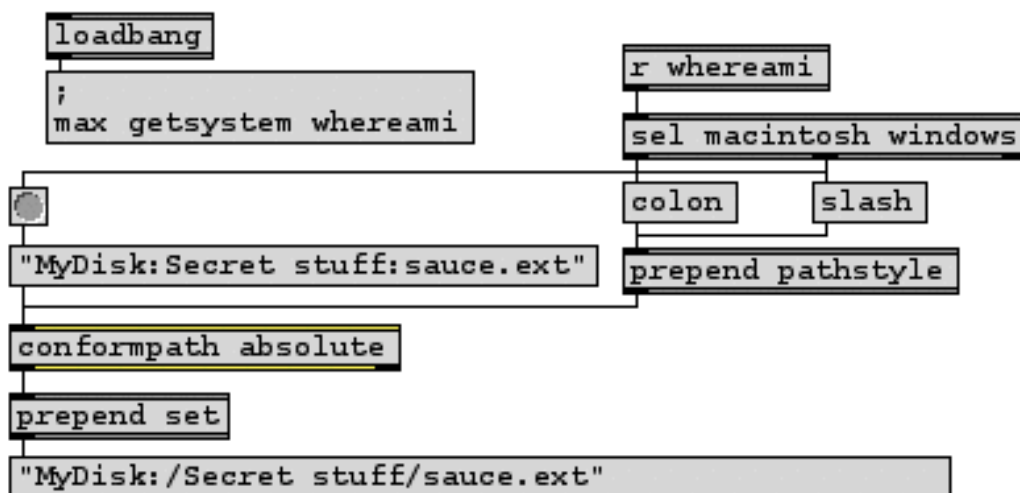
absolute	absolute を指定すると、フォルダやファイルの出力パス名を絶対パスに変換してシンボルとして出力する。
boot	boot を指定すると、フォルダやファイルの出力パス名を起動ディスクを基準とした相対パスに変換してシンボルとして出力する。
C74	C74 を指定すると、フォルダやファイルの出力パス名をCycling 74フォルダを基準とした相対パスに変換してシンボルとして出力する。
ignore	ignore を指定すると、フォルダやファイルの出力パス名を変換せずにそのまま出力する。
relative	relative 指定すると、フォルダやファイルの出力パス名をMaxのアプリケーションフォルダを基準とした相対パスに変換してシンボルとして出力する。
シンボル	オプション。出力に使用されるパス形式を指定することができる。指定できるパス形式には以下のようなものがある。
colon	colon のパス形式を指定すると、オブジェクト間でパス名をやり取りする際に、コロン (:) をパスの区切りとして使用する（詳細は入力の項を参照）。
max	max のパス形式を指定すると、現在使用しているバージョンのMaxが使用しているパス形式を選択して、オブジェクト間でパス名をやり取りする際に選択する。
native	native のパス形式を指定すると、現在使用しているOSでの標準形式をパス形式として使用する（詳細は入力の項を参照）。
native_win	native_win のパス形式を指定すると、Windowsでの標準パス形式がパス形式として使用される（詳細は入力の項を参照）。 注記： native_win のパス形式は、表示用以外に使用することは推奨しない。
slash	slash のパス形式を指定すると、オブジェクト間でパス名をやり取りする際に、スラッシュ (/) をパスの区切りとして使用する。

出力

シンボル 指定されたパス形式および種類で、フォルダまたはファイルのパス名を出力する。

整数 第2アウトレット：ファイルまたはフォルダのパス名が、指定された形式および種類で出力された場合は1が出力される。もし、(相対パスを辿ることができない場合など)指定された通りに出力されなかった場合は、0が出力される。

利用例



getsystemメッセージを使用してプラットフォーム間のパス名の表記を自動的に変換する

参照

- absolute** ファイル名を絶対パスに変換する
- opendialog** ファイルやフォルダを選択するためのダイアログを開く
- relative** 絶対パスを相対パスへ変換する
- savedialog** 保存するファイルの名前を尋ねるダイアログを開く
- strippath** フル・パス名からファイル名を取得する

入力

実数 コサイン関数への入力。

bang **COS**オブジェクトに現在格納されている数値でコサイン関数を計算する。アーギュメントがなければ、**COS**オブジェクトは初期値として0を持つ。

アーギュメント

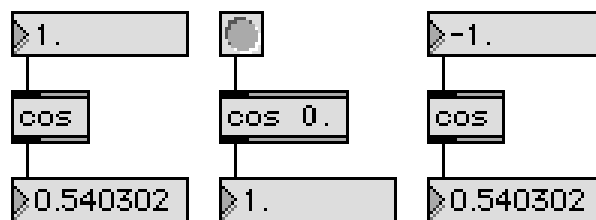
なし。

出力

実数 入力された値のコサイン値。

利用例

実数を入力



入力に対するコサイン値を出力

参照

acos	アーク・コサイン関数
asin	アーク・サイン関数
atan	アーク・タンジェント関数
atan2	アーク・タンジェント関数 (2変数)
cosh	双曲線コサイン関数
sin	サイン関数
sinh	双曲線サイン関数
tanh	双曲線タンジェント関数

入力

整数または実数 双曲線コサイン関数への入力。

bang **cosh**オブジェクトに現在格納されている数値で双曲線コサインの計算を行なう。アーギュメントがなければ、**cosh**オブジェクトは初期値として0を持つ。

アーギュメント

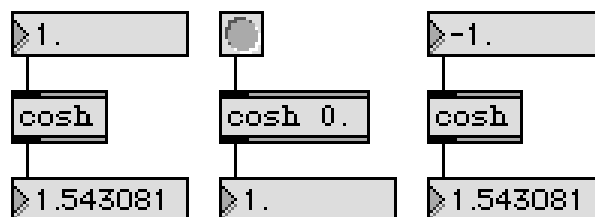
なし。

出力

整数または実数 入力された値の双曲線コサイン値。

利用例

実数を入力



入力に対する双曲線コサイン値を出力

参照

acos	アーク・コサイン関数
asin	アーク・サイン関数
atan	アーク・タンジェント関数
atan2	アーク・タンジェント関数 (2変数)
cos	コサイン関数
sin	サイン関数
sinh	双曲線サイン関数
tanh	双曲線タンジェント関数

入力

bang 第1インレット：**bang**を入力し、現在のカウントされた数値を出力する。

第2インレット：カウントの増減方向を変更する。

第3インレット：指定された最小値に**counter**オブジェクトをリセットする。リセット後の数値は、次の第1インレットへの**bang**メッセージによって出力される。

第4インレット：指定された最小値に**counter**オブジェクトをリセットして、すぐにその値を出力する。

第5インレット：指定された最大値に**counter**オブジェクトをリセットする。リセット後の数値は、次の第1インレットへの**bang**メッセージによって出力される。

整数 第1インレット：**bang**と同じ効果である。

第2インレット：カウントの増減方向を設定する。0なら最小値からカウント・アップする。1なら最大値からカウント・ダウンする。2なら最小値から最大値へカウント・アップした後、最大値から最小値へカウント・ダウンすることを繰り返す。

第3インレット：入力された数値によって、**counter**オブジェクトから出力される次の値を設定する。入力された数値が最小値より小さい値の場合は、最小値は入力された値に設定される。最大値は変わらない。入力された数値が最大値より大きい値の場合は、最大値は入力された値に設定されるが、最小値も最大値と同じ値になる。

第4インレット：入力された数値によって、**counter**オブジェクトから出力される数値を設定して、すぐにその数値を出力する。入力された数値が最小値より小さい値の場合は、最小値は入力された値に設定される。最大値は変わらない。入力された数値が最大値より大きい値の場合は、最大値は入力された値に設定されるが、最小値も最大値と同じ値になる。

第5インレット：**counter**オブジェクトによって出力される最大値を設定する。この数値が現在の最小値より小さければ、最大値は最小値よりも1大きな数値に設定される。

- 実数** 第1インレット：**bang**と同じ効果である。
- set** 第1インレット：**set**メッセージに続く数値を、**counter**オブジェクトの値として設定する。設定後の数値は、次の第1インレットへの**bang**メッセージによって出力される。
- jam** 第1インレット：**jam**メッセージに続く数値を。**counter**オブジェクトの値として設定し、すぐにその値を出力する。
- goto** 第1インレット：**set**と同じ効果である。
- up** 第1インレット：最小値からカウント・アップする。
- down** 第1インレット：最大値からカウント・ダウンする。
- updown** 第1インレット：最小値から最大値へのカウント・アップと、最大値から最小値へのカウント・ダウンを交互に繰り返す。
- inc** 第1インレット：**counter**オブジェクトの値を増加して出力する。これは、通常を増減方向の設定に関係なく行われる。
- dec** 第1インレット：**counter**オブジェクトの値を減少させて出力する。これは、通常を増減方向の設定に関係なく行われる。
- next** 第1インレット：**bang**メッセージと同じ効果である。
- min** 第1インレット：**min**メッセージに続く数値を、**counter**オブジェクトの最小値として設定する。数値が現在の最大値より大きい値ならば、最小値は最大値と同じ値に設定される。
- max** 第1インレット：**max**メッセージに続く数値を、**counter**オブジェクトの最大値として設定する。数値が現在の最小値より小さければ、最大値は最小値よりも1大きな数値に設定される。
- carrybang** 第1インレット：出力される値が最大値に達した時に、**bang**メッセージを第3アウトレットから出力するように設定する。また、出力される値が最小値に達した時に、**bang**メッセージを第2アウトレットから出力するように設定する。(デフォルトでは、**counter**オブジェクトは**bang**の代わりに1を出力する)
- carryint** 以前に受け取った**carrybang**メッセージの効果をもとに戻す。**counter**オブジェクトが最小値と最大値に達した時に、**bang**の代わりに1と0を第2アウトレットと第3アウトレットから出力するよう、**counter**オブジェクトをリセットする。

アーギュメント

整数 オプション。アーギュメントが1つだけならば、**counter**オブジェクトの最大値の初期値を設定する。アーギュメントが2つならば、最初の数値は最小値の初期値を設定し、2番目の数値は最大値の初期値を設定する。アーギュメントが3つならば、最初の数値は**counter**オブジェクトの増減方向を設定し、2番目の数値は最小値を設定し、3番目の数値は最大値を設定する。アーギュメントがないならば、最小値は0で最大値はない。

出力

整数 第1アウトレット：**bang**が入力された時や数値を第1インレットから受け取った時に、現在のカウントされている数値が、指定されている最小値と最大値の範囲内で出力される。**counter**オブジェクトの増減方向がカウント・アップまたはカウント・アップ&ダウンの場合は、最小値からカウントを始める。増減方向がカウント・ダウンの場合には、最大値からカウントが始まる。

第2アウトレット：**counter**オブジェクトの数値が減少し、最小値に達した時に、**1**が出力される。最小値でない場合には、**0**が出力される。

第3アウトレット：**counter**オブジェクトの数値が増加し、最大値に達した時に、**1**が出力される。最大値でない場合には、**0**が出力される。

第4アウトレット：**counter**オブジェクトの数値が最大値に達した回数をカウントする。最大値に達する度に、数値は出力される。

bang 第2アウトレット：**carrybang**が第1インレットに入力された場合に、**counter**オブジェクトの数値が減少し、最小値に達した時に、**bang**が出力される（デフォルトでは**1**が出力される）。最小値でない場合には、何も出力されない。

第3アウトレット：**carrybang**が第1インレットに入力された場合に、**counter**オブジェクトの数値が増加し、最大値に達した時に、**bang**が出力される（デフォルトでは**1**が出力される）。最大値でない場合には、何も出力されない。

利用例



イベントの回数を数える、または、連続するループを作成する

参照

tempo

メトロノームのようなテンポで数値を出力する

Tutorial 31

Using timers (英文PDF参照)

入力

- (MIDI) **ctlin**オブジェクトはMIDI入力デバイスからMIDIコントロール・チェンジ・メッセージの入力を受信する。
- port** **port**メッセージに続く**a**から**z**までの文字、またはMIDI入力デバイス名によって、どのポートからMIDIコントロール・チェンジ・メッセージを受信するかを設定する。**port**メッセージは任意であり、省略することもできる。
- set** **set**メッセージに続く**0**から**127**の数値によって、**ctlin**オブジェクトが受信するコントロール・ナンバーを特定する。このメッセージは、特定のコントロール・ナンバーが元々アーギュメントによって指定されている場合にだけ有効である。**ctlin**オブジェクトにアーギュメントが何も指定されていなければ、このメッセージは無視される。
- enable** **enable 0**というメッセージを入力すると、入力後に受け付けるMIDIメッセージを無視して、**ctlin**オブジェクトの動作を止める。0以外の任意の数値が続く場合は、パッチ・ウィンドウのMIDIイネーブル・ボックスや**pcontrol**オブジェクトによって動作を止めていても、再び**ctlin**オブジェクトは動作するようになる。
- (マウス) **ctlin**オブジェクトをダブルクリックすると、OMS入力デバイスをリストから選択するダイアログが開く。

アーギュメント

- a~z** オプション。受信されるコントロール・メッセージを受け取る1つのMIDIポートを指定する。アーギュメントに何もなければ、**ctlin**オブジェクトは、すべてのポートから受信する。

(MIDI入力デバイス名)

オプション。最初のアーギュメントとして、MIDI入力デバイスの名前を指定することもできる。

- 整数** MIDIポートを指定するアーギュメントに続くアーギュメントは、**ctlin**が認識する1つのコントロール・ナンバーを指定する。アーギュメントとしてコントロール・ナンバーを指定していない場合や、負の数を指定した場合は、**ctlin**オブジェクトはすべてのコントロール・ナンバーを認識する。1つのコントロール・ナンバーを指定する場合は、コントロール・ナンバーを出力するアウトレットは不要になるので、そのアウトレットは作られない。

コントロール・ナンバーのアーギュメントに続くアーギュメントは、コントロール・メッセージを受信するMIDIチャンネルを指定する。MIDIチャンネルのアーギュメントを指定しなければ、**ctlin**オブジェク

トはすべてのチャンネルからのコントロール・メッセージを受け取る。このアーギュメントを使用するには、コントロール・ナンバーをアーギュメントとして指定する必要がある。コントロール・ナンバーを指定しないでチャンネルを指定する場合は、コントロール・ナンバーに-1を指定する。MIDIチャンネルがアーギュメントとして指定されれば、チャンネルを出力するアウトレットは作られない。MIDIポートがアーギュメントとして指定されていて、MIDIチャンネルの数値が16以上の場合は、1-16の範囲にMIDIチャンネルが変換される。MIDIポートがアーギュメントとして指定されていないければ、MIDIチャンネルは文字と数字の組み合わせによって指定できる。MIDIチャンネルのアーギュメントの正確な意味は、**MIDI Setup**ダイアログによって各ポートに指定されるチャンネル・オフセットに従って決定される。

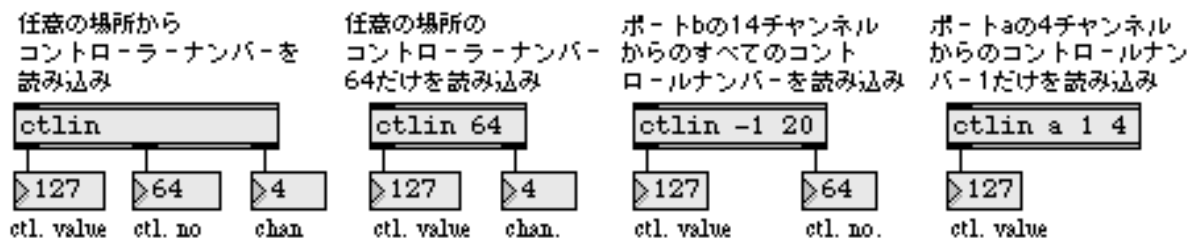
出力

整数 第1アウトレット：受信されたMIDIコントロール・チェンジ・メッセージのコントロール値が出力される。

コントロール・ナンバーがアーギュメントとして指定されていないければ、コントロール・ナンバーが第2アウトレットから出力される。

MIDIチャンネルがアーギュメントとして指定されていないければ、MIDIチャンネルが第3アウトレットから出力される。

利用例



様々な方法でコントロール・メッセージをフィルターにかけることができる

参照

- bendin** MIDIピッチ・バンド・メッセージを受信する
- ctlout** MIDIコントロール・チェンジを送信する
- midiiin** 生のMIDIデータを受信する
- notein** MIDIノート・メッセージを出力する
- rtin** MIDIのリアルタイム・メッセージを出力する
- xbendin** MIDIピッチ・バンド・メッセージ (14ビット) を解釈する

入力

整数 第1インレット：入力される数値は、コントロール値として認識され、コントロール・チェンジ・メッセージとして、**ctlout**オブジェクトによって送信される。数値は0から127までである。

第2インレット：入力される数値は、**ctlout**オブジェクトが送信するコントロール・チェンジ・メッセージの、コントロール・ナンバーとして格納される。数値は0から127までである。

第3インレット：入力される数値は、コントロール・チェンジ・メッセージを送信する時のMIDIチャンネルとして格納される。

実数 整数に変換される。

リスト 第1インレット：リストの左から1番目の数値はコントロール値、2番目の数値はコントロール・ナンバー、3番目はMIDIチャンネルである。**ctlout**オブジェクトは、それらの数値に従って、MIDIコントロール・チェンジ・メッセージを送信する。

enable **enable 0**メッセージは、MIDIメッセージの送信を止め、**ctlout**オブジェクトの動作を止める。**enable**に続く0以外の数値が続く**enable**メッセージは、パッチ・ウィンドウのMIDIイネーブル・ボックスや**pcontrol**オブジェクトによってMIDI送受信を止めていても、**ctlout**オブジェクトは再び動作するようになる。

port **port**に続く**a~z**までの文字または、MIDI出力デバイス名によってMIDIコントロール・チェンジ・メッセージを送信するために使用するポートを設定する。**port**という単語は任意であり、省略できる。

(マウス) **ctlout**オブジェクトをダブルクリックすることによって、OMSデバイスをリストから選択するダイアログが開く。

アーギュメント

a~z オプション。MIDIコントロール・チェンジ・メッセージを送るためのポートを指定する。アーギュメントに何も指定していない場合は、ポートaのチャンネル1としてメッセージを送信する。アーギュメントを指定している場合で、第3インレットに入力されるMIDIチャンネルの値が16以上の時には、1から16までの値に置き換えられる。

(MIDI出力デバイス名)

オプション。最初のアーギュメントとして、MIDI出力デバイス名を指定することもできる。

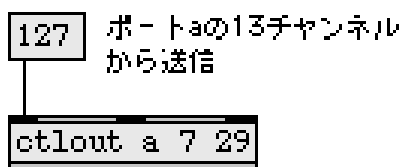
整数 任意のポートを指定するアーギュメントに続いて、次のアーギュメントは、**ctlout**オブジェクトによって送信されるコントロール・ナンバーの初期値である。コントロール・ナンバーは自動的に0~127の間に置き換えられる。何も指定しなければ、コントロール・ナンバーは1である。

コントロール・ナンバーに続くアーギュメントは、コントロール・チェンジ・メッセージを送るMIDIチャンネルの初期値である。アーギュメントに何も指定しなければ、デフォルトではチャンネル1である。このアーギュメントを指定するためには、コントロール・ナンバーが先に指定されていなければならない。ポートがアーギュメントで指定されていれば、チャンネル・ナンバーは、1から16までの範囲に置き換えられる。ポートがアーギュメントで指定されていなければ、チャンネル・ナンバーは、ポートとチャンネルの両方を指定する。チャンネル・ナンバーの正確な意味は、**MIDI Setup**ダイアログで、各ポートに指定されるチャンネル・オフセットに従って決定される。

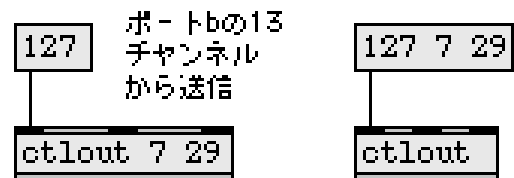
出力

(MIDI) アウトレットは存在しない。出力は、直接MIDI出力ポートに送られるMIDIコントロール・チェンジ・メッセージである。

利用例



文字のアーギュメントによって、1つのポートだけに送られる



別の方法で、数値だけでポートとチャンネルの両方を指定する

参照

- bendout** MIDIピッチ・ベンド・メッセージを送信する
- ctlin** MIDIコントロール・チェンジを受信する
- midout** 生のMIDIデータを送信する
- noteout** MIDIノート・メッセージを送信する
- xbendout** MIDIピッチ・ベンド・メッセージ (14ビット) を作成する

Tutorial 16 More MIDI ins and outs (英文PDF参照)

入力

任意のメッセージ 連続的に異なるアウトレットに送られる整数、実数、またはシンボルの流れ。

- set** **set**に続く数値によって、次の入力を送られるアウトレットを指定する。サイクル・モードでは、アウトレットは**0**から始まる番号で指定する。実際に存在しないアウトレットの番号が指定された場合には、メッセージは無視される（このメッセージは、**cycle**オブジェクトがイベント・センシティブ・モードである場合には、何の効果もない。その場合、各メッセージは常に第1アウトレットから順に出力される）。
- thresh** **thresh**メッセージに続く数値で、出力モードを切り替える。これは、第2アーギュメントと同じ働きである。0以外の数値では、入力される一連の数値の出力は、常に第1アウトレットから始まる（イベント・センシティブ・モード）。**0**の場合は、入力される一連の数値は、前の入力の最後の出力があったアウトレットの次のアウトレットから出力される（サイクル・モード）。

アーギュメント

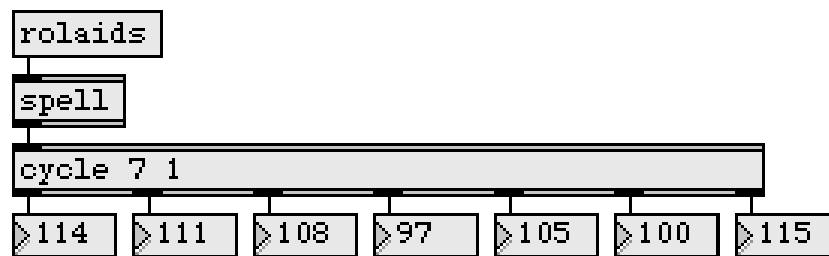
- 整数** オプション。第1アーギュメントはアウトレットの数を設定する。第1アーギュメントに何も指定しなければ、アウトレットは1つである。第2アーギュメントは、出力モードを切り替える。数値が0以外の場合は、入力される一連の数値の出力は、常に第1アウトレットから始まるイベント・センシティブ・モードであり、**0**の場合は、入力される一連の数値は、前の入力の最後の出力があったアウトレットの次のアウトレットから出力されるサイクル・モードである。別々のイベントの例としては、時間的な遅れがある複数のメッセージや、連続したマウス・クリックやMIDIイベントによって発生するメッセージがある。また、メッセージ・ボックス中にコンマで区切られたメッセージの流れは、単一のイベントとして認識される。第2アーギュメントに何も指定しないか、アーギュメントが**0**である場合は、値はすべてのアウトレットを循環して出力される。

出力

任意のメッセージ すべてのアウトレット：サイクル・モードの場合は、一連の数値の出力は、前の出力のアウトレットの次のアウトレットから始まる。**cycle**オブジェクトの出力が、最も右のアウトレットに達した場合は、その次は第1アウトレットから出力される。

イベント・センシティブ・モードの場合は、あらゆるメッセージは常に第1アウトレットから順に出力される。

利用例



ASCIIコードを得るために、**cycle** オブジェクトを利用する

参照

**Bucket
counter
spell
spray**

順番にアウトレットからアウトレットへ数値を渡す
bangメッセージをカウントし、回数を入力する
入力をASCIIコードに変換する
整数を、アウトレットの番号に従って分配する

入力

- date** 左のアウトレットから現在の日付をリスト（月/日/年）の形で出力する。
- ticks** 第3アウトレットから現在のTicks値（システム起動後の60分の1秒単位の経過時間）を出力する。
- time** 第2アウトレットから現在の時間をリスト（時/分/秒）の形で出力する。

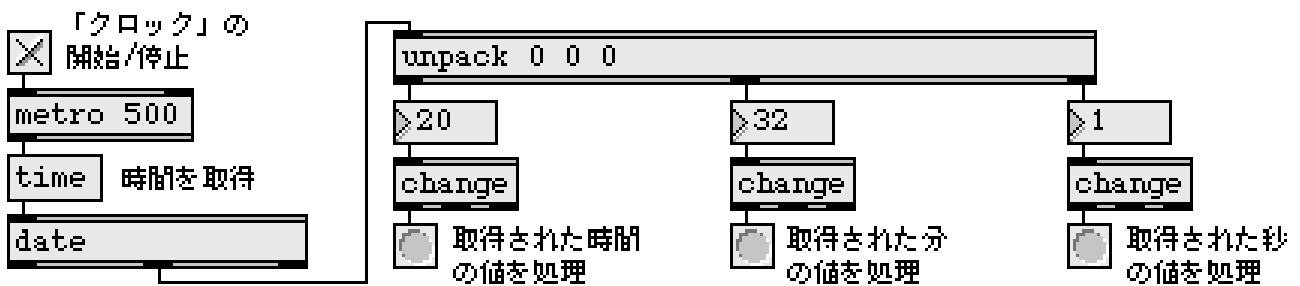
アーギュメント

なし。

出力

- リスト** 第1アウトレット：**date**メッセージを受け取る時、**date**オブジェクトはリストの形で現在の日付を送信する。
- リスト** 第2アウトレット：**time**メッセージを受け取る時、**date**オブジェクトはリストの形で現在の時間を送信する。
- 整数** 第3アウトレット：**ticks**メッセージを受け取る時、**date**オブジェクトは現在のTicksの値を送信する。

利用例



ゆっくりと変化してゆく作品のために、**date**はイベントをトリガーしてゆく時計のように利用できる

参照

clocker timer

一定の間隔で経過時間を出力する
2つのイベント間の経過時間を出力する

入力

bang 第1インレット：1または0をランダムに選択して出力する。

整数 第1インレット：**bang**と同じ働きをする。

第2インレット：シード値を与えることで0と1との特定の（再現可能な）シーケンスから成る疑似ランダムを出力する。数値0はシステム起動時から経過した時間（予想不可能な数）をシードとして使い、予想できない0と1のランダムな出力をする。

アーギュメント

整数 オプション。シード値を書き込むことで0と1との特定の（再現可能な）シーケンスの疑似ランダムを出力する。アーギュメントがなければ、システム起動時から経過した時間（予想不可能な数）をシードとして使い、予想できない0と1のランダムな出力をする。

出力

整数 ランダムに選択された1または0。シード値を与える時、最初はランダムではないパターンが出力されていると思われるかもしれない。しかし多く反復を繰り返してみる時、その流れは予測不能なものになり、また1と0の出力される割合は均一に保たれる。

利用例



コイン投げをシュミレート。オンとオフをランダムに切り替える

参照

drunk
random
toggle
urn

ある範囲を移動するようなランダムな数値を出力する
ランダムな数値を発生させる
オンとオフ (1と0) を切り替える
重複しないランダムな数値のジェネレーター

Decodeオブジェクトは階層的なスイッチボードとして機能する。第3インレットはマスター・スイッチであり、すべてのアウトレットのスイッチを切る（0を出力）。第2インレットはサブマスター・スイッチであり、マスター・スイッチがすべてのアウトレットのスイッチを切っていない時に限り、すべてのアウトレットのスイッチを入れる（1を出力）。第1インレットはサブマスター・スイッチやマスター・スイッチが働いていない時に限り、ひとつのアウトレットを独占的にオンにすることができる。

入力

整数 第1インレット：受け取った数値に対応したアウトレットのスイッチをオンにし、それ以外はオフにする。（オフのアウトレットがオンになる時、1を送信する。逆に、オンのアウトレットがオフになる時は、0を出力する。）アウトレットは第1アウトレットを0とした数値によって参照されるので、第1インレットに受け取る数値は必然的に0から始まってアウトレットの数から1を引いた数に制限されることになる。

第2インレット：すべてのアウトレットがオフになっていない時に限り、0以外の数値を受け取るとすべてのアウトレットに1を送ってスイッチをオンにする。0を受け取ると**Decode**オブジェクトは前にオンになっていたアウトレット以外をすべてオフにする。

第3インレット：0以外の数値を受け取るとすべてのアウトレットに0を送って、すべてのアウトレットをオフにする。このやり方で一度すべてのアウトレットがオフになると、第3インレットが0を受け取るまでどのアウトレットもオンにすることはできない。0を受け取ると、**Decode**オブジェクトはオフにしたアウトレットをオンにし直す。

実数 整数に変換される。

アーギュメント

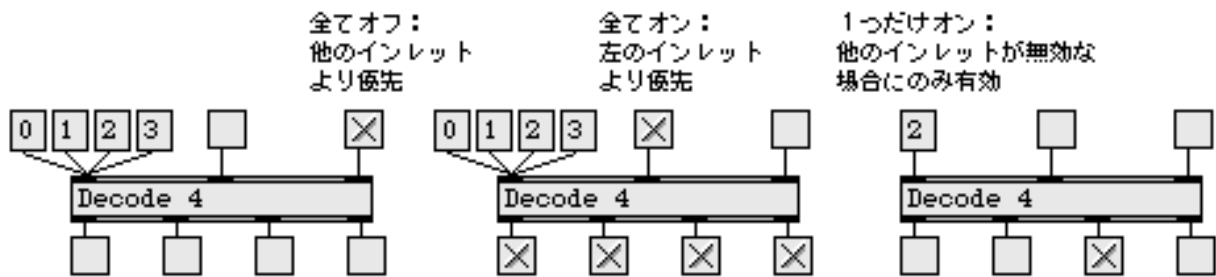
整数 オプション。アウトレットの数を設定する。デフォルトではアウトレットは1つ。

実数 整数に変換される。

出力

整数 オフだったアウトレットをオンにする時、そのアウトレットから1を出力する。オンだったアウトレットをオフにする時、そのアウトレットから0を出力する。第1アウトレットは最初オンである。

利用例



Decode オブジェクトは階層的なオン/オフスイッチである

参照

**Bucket
gate
toggle**

順番にアウトレットからアウトレットへ数値を渡す
入力を特定のアウトレットから出力する
オンとオフ（1と0）を切り替える

入力

任意のメッセージ インレットに受け取るメッセージがMIDIオブジェクト (**notein**など) やタイミング・オブジェクト (**metro**や**seq**など) によって出力され、オーバードライブの設定がオンになっている場合、Maxは通常そのメッセージに対して優先権を与えるので、Maxの処理としてのタイミングは (Maxウィンドウへ表示する時のように) 不正確なものになってしまう。**defer**オブジェクトはメッセージからこの特別な優先権を取り除き、メッセージをもっとタイミングの正確なものにする。これは音楽的リズムの妨げになってしまうメッセージから優先権を解除したり、オーバードライブの設定をオンにしてもうまく機能しないメッセージに対して有効である。

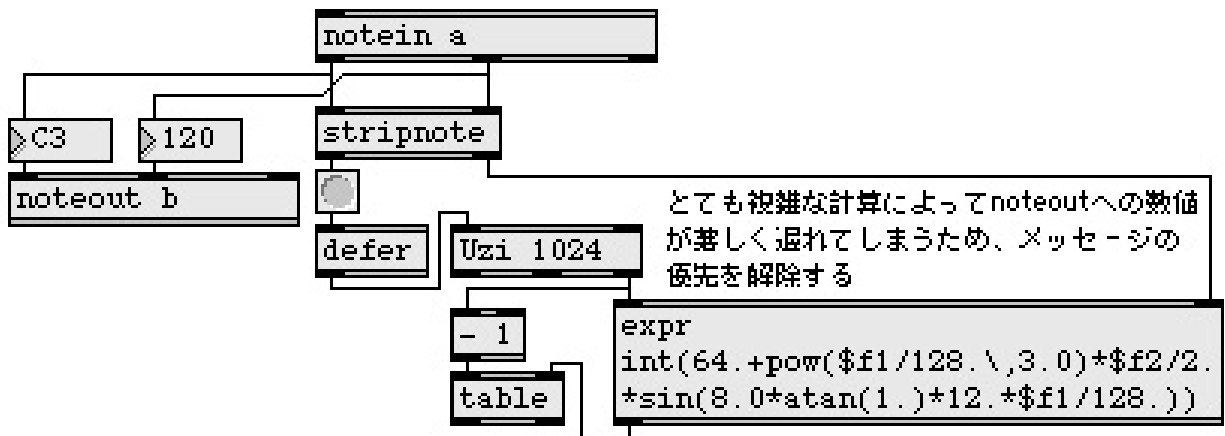
アーギュメント

なし。

出力

任意のメッセージ 入力と同じ。

利用例



MIDIやタイミング・メッセージに与えられるオーバードライブの優先権は、**defer**によって変更することができる

参照

Uzi 指定した数の**bang**を出力する

入力

bang 第1インレット：**bang**は一定のミリ秒遅らされてアウトレットへ送信される。

stop 第1インレット：現在**delay**オブジェクト内で出力を待っている**bang**の出力を止める。

整数または実数 第1インレット：遅らせたい時間をミリ秒単位で設定する。その時間だけ遅れて**bang**が出力されることになる。

整数または実数 第2インレット：第1インレットが受け取った**bang**メッセージをどれだけ遅らせるかを指定するミリ秒単位の数値として格納する。第2インレットが受け取った数値（ミリ秒）は、次から第1インレットに入ってくる**bang**がどれだけ遅れるかという時間を変更するもので、現在**delay**オブジェクト内で出力を待っている**bang**に対しては時間の変更はしない。

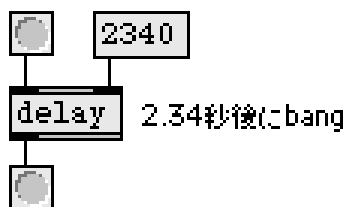
アーギュメント

整数または実数 第1インレットへ入ってくる**bang**を遅らせるための数値（ミリ秒）の初期値を設定する。このアーギュメントがなければ、初期値は0（ミリ秒）となる。

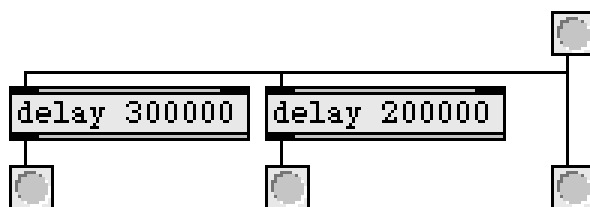
出力

bang 第1インレットへの**bang**は第2インレットで指定された数値（ミリ秒）だけ遅らされてアウトレットから出力される。**delay**オブジェクトが一度に扱える**bang**は1つのみである。もし、**bang**がすでに**delay**オブジェクト内にある状態で、新しい**bang**が第1インレットに入ってくれば、最初の**bang**は失われることになる。

利用例



bangメッセージは一定の時間遅らされて出力される



特定の時間に分けられたトリガーとして利用できる

参照

pipe 数値またはリストを遅らせて出力する

Tutorial 22 Delay lines (英文PDF参照)

入力

record 第1インレット：インレットに受け取る数値の記録を始める。それは、グラフィック・スコアに記録されるノート・イベントのパラメーターとして扱われる。第1インレットに数値を受け取る度に、イベントのオンセット（直前のイベントからの経過時間）として記録される。

整数 **record**メッセージを受けた後に、すべての受け取った数値はノート・イベントのパラメーターとして扱われる。

第1インレット：直前に記録されたイベントからのミリ秒単位でのデルタ・タイム（ディレイ）。これは、オンセットの間隔と解釈される。つまり、ノートとノートの開始の間の時間であり、イベントが記録されたリズムを事実上決定することになる。これは、イベントが記録された実際の時間であるわけでは必ずしもない。**detonate**はイベントを受け取った（マイナスではない）デルタ・タイムとして理解する。

第2インレット：数値は音のノート・ナンバー（ピッチ）として扱われる。ノート・ナンバーを受け取っていない状態では、デフォルトとして**60**が使われる。

第3インレット：ノートのベロシティ。ベロシティが0であればノート・オフとして見なされるが、この場合は、イベントは同じキーである前の音の最後として扱われる。ベロシティの数値を受け取っていない状態では、デフォルトとして**64**となる。

第4インレット：ノート・オフの代わりに、ノート・オンのイベントの一部として、音の長さが与えられる。長さの数値を受け取っていない場合、または、音の最後としてノート・オフを受け取っていない場合は、デフォルトとして**10**ミリ秒の長さが用いられる。

第5インレット：ノート・イベントが記録されるトラックの番号。**detonate**ではオーバーダブ録音はできないが、それぞれの録音された音は、内部的に異なるトラックに記録するために、トラック番号によって区別される。トラック番号を受け取っていない状態では、ノートは常に第**1**トラックに記録される。

第6インレット：ノートのMIDIチャンネル。チャンネルが指定されていない場合は、常に第**1**チャンネルとして記録される。

第7インレット：何らかの目的のために使われる特別な数値で、ノート・イベントにさらに付け加えることができる。この数値は追加のイベントのパラメーターとして、または、ノートと同期するコントロール数値として用いることができる。このインレットに数値を受け取ら

ない場合は、デフォルトとして**0**が常に記録される。
第8インレット：2つめの特別な数値。

録音中に**detonate**が第1インレットに数値を受け取る場合、オンセット同士の間隔（直前のイベントから経過した時間）として数値を扱い、もっとも近い時間内に他のインレットに受け取った数値を合わせて、ひとつのノート・イベントとしてそれらを一緒に録音する。ほとんどのMaxのオブジェクトがそうであるように、他のインレットに受け取った数値は、第1インレットに数値を受け取った時にトリガーされることになるノート・イベントとして記録される。

detonateが**follow**のメッセージ（下記を参照のこと）を受け取っている時は、第2インレットに続いて入れられる数値はノート・ナンバー（ピッチ）として扱われる。その数値が、スコアにおける現時点でのノート（または、最も近いノート）のノート・ナンバーと同じ場合、デルタ・タイム以外の記録されたノートの情報を出力する。

detonateが録音でもなければスコアを追う状態でもない場合は、第1インレットの数値は、**nth**メッセージと同じ意味を持っている（下記を参照のこと）。

- float** 常に**int**に変えられる
- list** リストの最初の数値はデルタ・タイムとして使われる。他の数値は、それぞれ左から右へと他のインレットに受け取った場合と同じように扱われる。
- start** 最初のデルタ・タイムを単に出力して、スコアの再生を始める。スコアの再生が開始されてからは、**next**メッセージは次のイベントの情報を出力するために用いられる。
- next** スコアの再生が**start**メッセージによって始められた後は、**next**はスコアにおける現在のノートの（デルタ・タイムを除く）イベント情報を出力し、次のノートのためにデルタ・タイムを出力する。そのデルタ・タイムは、**detonate**へ次の**next**メッセージを送るためのディレイ・タイムとして使われる。さらに続く音がすでにない最後のノートに対して**next**を受け取った場合は、スコアの最後という合図として、第1アウトレットから**-1**の数値が出力される。スコアが再生されていない時に、**next**メッセージを受け取った場合は、**detonate**は単にMaxウィンドウに**not playing**のメッセージを表示する。
- nth** **nth**の後に数値が続く場合は、スコアに示されたイベントのノートの情報を出力する（イベント番号は0より始まる）。イベントのデルタ・タイムでは、イベントの開始時間が第1アウトレットから出力される。

- clear** **detonate**の内容をすべて消去する
- follow** スコアに記録されたイベントと入力されたノート・ナンバーの情報を比較しながら、**detonate**にスコアの読み手のような振る舞いを行う。ノート・ナンバーを第2インレット（ピッチ）に受け取った時、**detonate**はデルタ・タイム以外のイベントとして録音された情報を出力する。
- followat** ノート・ナンバー、ベロシティ、MIDIチャンネル番号が後に続く**followat**によって、**detonate**は、録音されたスコアの中から、これらの属性のノート・イベントを探し出す。指定されたノートが見つかった場合は、すぐ次のイベントからスコア・フォローイング（楽譜での位置を追い掛けること）を始める。見つからない場合は、Maxウィンドウに**detonate:note not found**と単に表示する。
- startat** ノート・ナンバー、ベロシティ、MIDIチャンネル番号が続く**startat**によって、**detonate**は、録音されたスコアの中から、これらの属性のノート・イベントを**detonate**に探し出す。指定されたノートが見つかった場合は、次のイベントのデルタ・タイムを送り出す。そして、後に送られる**next**メッセージは次のイベントを参照することになる。見つからない場合は、Maxウィンドウに**detonate:note not found**と単に表示する。
- stop** 録音、再生、フォローイングの状態を停止する。**record**、**start**、**follow**のいずれかの状態から、直接他の状態へ変える時に、一旦**detonate**を止める必要はない。
- mute** 特別に指定された音のイベントに対してミュートができる。イベント・パラメーターの番号、パラメーターの数値、「ミュート」または「ミュートしない」を指定するために1または0の数値を、**mute**の後に指定しなければならない。イベントのパラメーターは、デルタ・タイムなら0、ピッチなら1といった数値である。例えば、メッセージで**mute 4 10 1**というメッセージなら、MIDIチャンネル10（チャンネルはパラメーター番号は4である）のノート情報を送り出さないことによって、音をミュートする。後に、メッセージの**mute 4 10 0**によってミュートを解除することができる。
- unmute** イベントのパラメーター番号とパラメーター数値が後に続く、**unmute**は同じ基準で以前にミュートされたものを解除する。例えば、**unmute 4 10**は**mute 4 10 0**と同じ意味である。
- unmuteall** 以前に出されたすべての**mute**メッセージによる指定を解除する。
- params** **params**の後には、3つの数字が続く。受け取ったノート・ナンバーが、スコアでのその時点でのノート・ナンバーと一致しない時、**params**は

スコア・フォローイングを自ら調節する。最初の数字は、スコアにおいて、次に進行する前に、幾つ間違いを許すかを指定する。第2番目の数字は、多くの間違いが起こった場合、スコアでどれだけのミリ秒を飛び越して進行するかを指定する。第3番目の数字は、0と指定されていないければ、受け取ったノート・ナンバーがオクターブ高すぎるか低すぎる場合、自動に修正して数値を扱うことを指定する。例えば、メッセージの **params 3 1000 1** は、1秒間に、3つの連続する間違い（オクターブの置き換えは合っているとみなす）を許容して、再び続けることを意味する。デフォルトでは、**detonate** は、200ミリ秒の間に、2つまでの間違いを許し、録音した音をオクターブの置き換えで一致させることをしないと設定される。

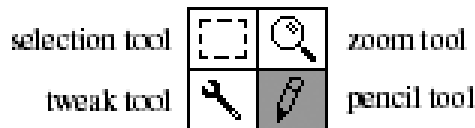
write スタンダードMIDIファイルとして **detonate** の中身を保存するためにダイアログを開く。**write** は最大2つまでの数字をその後に加えることができる。最初の数字が0でない場合は、ファイルは小節、拍子、テンポ数の単位ではなく、ミリ秒の時間単位として記録される。この数字が0であるか、省略した場合は、拍子の単位として記録される。第2番目の数字はMIDIファイルのフォーマットの選択で、0ならすべてのノートが単一のトラックに納められ、0でなければ、トラック・パラメータによってノートをトラックごとに分けたマルチ・トラックのフォーマットとなる。**detonate** の中身は、パッチが保存される時に、パッチの一部として保存される。

read **detonate** のスコアの中身として、スタンダードMIDIファイルを読み込むためのダイアログを開く。Maxのサーチ・パスに、MIDIファイルの名前が**read**の後に指定している場合は、ダイアログを開くことなく、直接そのファイルを読み込む。**read**のメッセージは数字を後に加えることもできる。その数字が0でない場合は、小節、拍子、テンポ数の単位ではなく、ミリ秒の時間単位としてファイルを解釈する。その数字が0であるか、省略した場合は、時間は小節、拍子として読み込まれる。

export **write**と同じ。

import **read**と同じ。

(マウス) ロックされたパッチでの**detonate**をダブルクリックすれば、ノートのイベントのグラフィック表示をするためのエディター・ウィンドウが開く。エディター・ウィンドウは様々な方法でイベント情報を表示することができる。そして、ノートを編集するために、あるいは、新たなノートを加えるために小さなパレットが用意されている。



pencil toolにより新たなノートを描くことができる。ノートのイベントの開始時間はグラフのx軸で常に表される。描かれたノートの初期値のパラメーターはエディター・ウィンドウの上部のナンバー・ボックスに表される。なお、そのナンバー・ボックスをドラッグすることによって数値を変えることができる。y軸にあてがわれる機能や、描かれた音の長さを変更することができる。デフォルトでは、y軸は音高であり、ノートの横の長さは音の長さとして表される。

selection toolによりすでに記録されているノートを選択でき、ノートの中央をクリックすることにより垂直にドラッグでき、ノートの左側をクリックすることで水平にドラッグできる。ノートの右側をドラッグすれば、ノートの長さを長くしたり短くすることができる。選択されたノートのパラメーターは、エディター・ウィンドウの上部のナンバー・ボックスでも変えることができる。

tweak toolは**selection tool**と同じような働きをするが、より細かい解像度でドラッグによる調整を行うことができる。**zoom tool**でグラフをクリックすれば、さらに正確な編集ができるようにグラフの部分を拡大することができる。**zoom tool**でグラフをoptionキー(Macintosh)あるいはAltキー(Windows)を押しながらクリックすれば、グラフを縮小することができる。

アーギュメント

シンボル **detonate**のグラフィック・エディター・ウィンドウのタイトル・バーに表示する名前を与える。同じ名前を持つ**detonate**オブジェクトは同じイベントのデータを共有する。それらは、同じ名前を持つ**detonate**のタイムライン・エディターによって、同じイベント・データを共有する。

出力

detonateが**start**メッセージ、または、**startat**メッセージを第1インレットに受け取った時、ノート・イベントの開始時間 (**startat**の場合では、ノートが見つかったすぐ後のノート) のデルタ・タイムを送り出す。その後、**detonate**が**next**のメッセージを毎回受け取るごとに、そのイベントにおける他のノートのデータと、スコアを進行させるための次のイベントのデルタ・タイムを送る。従って、第1アウトレットから出力される数字は、その時間だけ遅らせて次の**next**メッセージを送り込むことによって、再生するリズムをコントロールするために用いられる。

detonateの第1インレットに**nth**のメッセージを受け取る時（または、止まっている時に数字を受け取れば）、それをインデックス・ナンバー（最初のノートのイベントをインデックス・ナンバー0として）として扱い、指定されたイベントでのすべてのノートのデータを出力する。第1アウトレットからは、ノートのデルタ・タイムを出力するのではなく、ノートの開始時間、つまり、スコアの始めからの時間を出力する。

detonateが第1インレットに**follow**、または、**followat**のメッセージを受け取った後、スコアでその時点のノート（または、そのすぐ後のふたつのノートのいずれかのノート）のノート・ナンバーが、第2インレットで受け取った数値と一致する場合は、デルタ・タイムを除く、一致するノートのすべてのデータが送り出される。

整数 第1アウトレット：**start**、**startat**、または、その後続く**next**メッセージを第1インレットに受け取る時、次のノートのイベントのデルタ・タイムが出力される。スコアでの最後のイベントが**next**メッセージにより演奏された時、そして、その後続く音がない時に、特定のデルタ・タイムとして**-1**が出力され、最後の音がすでに演奏されたことを伝える。

nthメッセージを第1インレットに受け取る時（または、**detonate**が止まっている場合は整数を受け取る時）、特定のノートの開始時間が送り出される。

第2アウトレット：**nth**メッセージ、**detonate**が停止中の整数、再生中の**next**メッセージ、あるいは、スコアを追う状態で一致したノート・ナンバーに従って、ノート・バンバーが出力される。

第3アウトレット：ノートのベロシティ。

第4アウトレット：ノートの長さ。

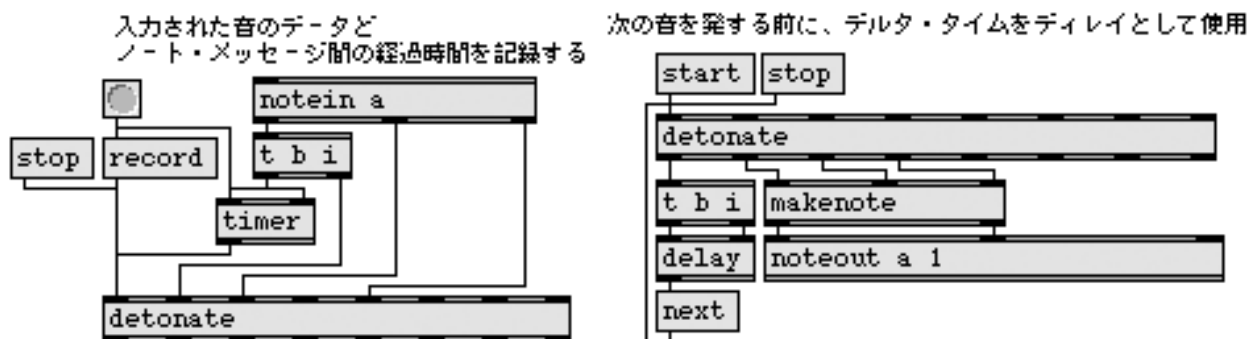
第5アウトレット：ノートのMIDIチャンネル。

第6アウトレット：ノートのトラック番号。

第7アウトレット：ノートに関連して追加する数値。

右アウトレット：ノートに関連して追加する第2番目の数値。

利用例



リズムとしてノートを再生するためのデルタ・タイムがノートのイベントとして記録される

参照

follow
seq
timeline

ライブなものと同録音されたものを比較する
MIDIの録音と再生のためのシーケンサー
Maxメッセージを時間軸で扱うスコア



入力

- 整数** インレットに入力された数値は**dial**の表示を変えながら、アウトレットから出てゆく。オプションで**dial**は数値がアウトレットに出力される前に、その数値にいくらかの数を掛けたり、オフセットを追加することができる。
- また**dial**は直接マウスでクリックやドラッグすることにより数値を出力することができる。
- 実数** 整数に変換される。
- bang** **dial**内に直前に記憶されていた数値を出力する。
- color** 0から15の数値を伴った**color**メッセージは**dial**の中央の円の部分の色を設定する。また、これはオブジェクトメニューのカラーコマンドを使っても行える。
- min** 数値を伴った**min**メッセージは**dial**オブジェクトが数値を出力する前に加える数量を設定する。デフォルト値は**0**である。
- mult** 数値を伴った**mult**メッセージは、乗数を指定する。**dial**オブジェクトの数値はアウトレットへ出力される前に、この乗数によってかけ算される。この計算はオフセット値を加える前に行われる。デフォルト値は**1**である。
- set** 数値を伴った**set**メッセージは出力することなしに**dial**の表示を変える。
- size** 数値を伴った**size**メッセージは**dial**オブジェクトの範囲を設定する。デフォルト値は**128**。**size**を**1**に設定すると**dial**は目に見えるように表示できなくなる（1つの数値しか表示しないことになるので）。1より小さい**size**は**2**に設定される。

インスペクター

インスペクターを使うことにより**dial**オブジェクトの振る舞いを表示、編集することができる。Windowsメニューにある**Show Floating Inspector**を選択してフローティング・インスペクターを使うなら、任意の**dial**オブジェクトを選択してそのインスペクターをフローティング・インスペクター内に表示させることができる。オブジェクトを選択し、Objectメニューから**Get Info...**を選択してもインスペクターを表示できる。



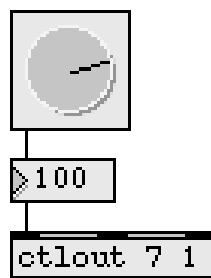
dialオブジェクトのインスペクターにより**dial**の範囲値を設定できる。インレットに受け取った数値は自動的に0から指定された値より1小さい数の範囲に限定される。デフォルトの範囲値は128。乗算の後の数に足されるオフセット値を指定することもできる。デフォルトのオフセット値は0。**dial**オブジェクトのインスペクターは乗数も指定できる。**dial**オブジェクトの数値は出力される前に、この乗数によって掛けられる。この掛け算はオフセットが加えられる前に行われる。デフォルトの乗数は1。

*Revert*ボタンはインスペクターを開いてから変更したオブジェクトの設定をすべて元に戻す。また、インスペクターが開いている状態でも、Editメニューから**Undo Inspector Changes**を選択することにより、インスペクターを開く前の状態に復帰させることもできる。

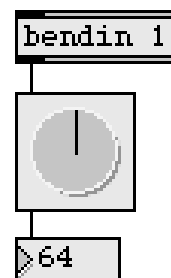
出力

整数 インレットに受け取った数値、またはマウスでクリックかドラッグすることで生じた数値に、まず設定された乗数によって掛け算が行われ、その後その数にオフセットが加えられ、アウトレットから出力される。

利用例



画面上をドラッグすることで数値を出力する...



または数値を通過させて視覚化するために使う

参照

hslider
pictctrl
pictslider
rslider
slider
uslider

画面上のスライダーを動かすことによって数値を出力する
ピクチャーを用いたコントロール
ピクチャーを用いたスライダー・コントロール
数値の範囲を変更し、それを表示する
画面上でスライダーを動かすことによって数値を出力する
画面上でスライダーを動かすことによって数値を出力する

入力

シンボル 第1インレット：任意の言葉を伴った**symbol**メッセージはユーザーがテキストを入力するためのダイアログ・ボックスを開く。**symbol**の後に続く言葉が、デフォルトのテキストとして表示される。1単語以上の言葉をデフォルトの言葉として表示させたいなら、一連の言葉をシングル・クォーテーション（'、option+[、shift+option+[で得られる）で囲むか、またはスペースの前にバックスラッシュ（\）を置く必要がある。

bang 第1インレット：以前に入力したテキストをデフォルトにしてダイアログ・ボックスを表示する。

整数 第1インレット：シンボルと同じ。

第2インレット：数値0はユーザーがダイアログ・ボックスに入力したものは何でもその言葉の前に**symbol**という言葉をつけて出力するように**dialog**オブジェクトを設定する。0以外の数値は入力されたテキストを、それが単語から始まっているが、数値から始まるリストであろうが、そのままテキストとして出力するように**dialog**オブジェクトを設定する。何も数値を受け取っていないなら、デフォルトとして**0**が設定されている。

アーギュメント

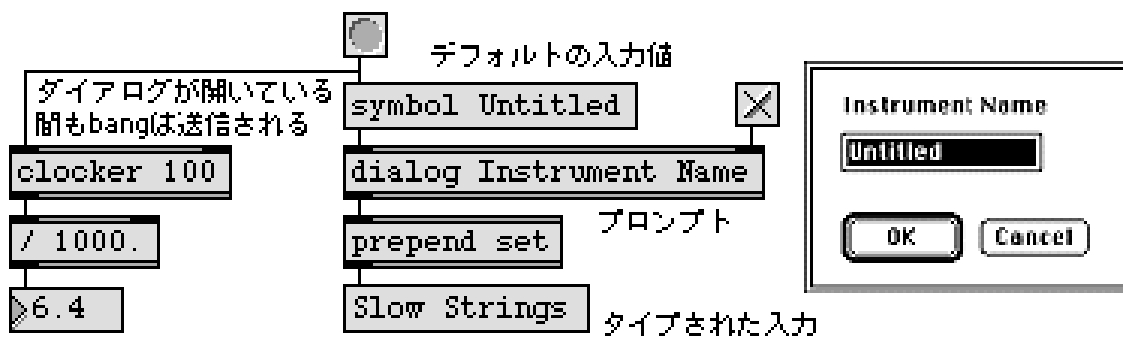
任意のメッセージ オプション。ダイアログ・ボックス内のテキスト入力する場所の上に表示する言葉を設定する。

出力

シンボル ユーザーがOKを押せば、**dialog**オブジェクトは入力されたテキスト（数値や複数の言葉であっても）からシンボルを作り、前に**symbol**という言葉をつけて出力する。0以外の数値が第2インレットに入力されるなら、入力されたメッセージは（前に**symbol**という言葉をつけずに）そのまま出力される。このメッセージは前に**set**という言葉をつけてメッセージ・ボックスに送れば、そこに表示できる。ユーザーがキャンセルを押せば、何も出力されない。

ユーザーがダイアログ・ボックスにテキストを入力している間もパッチは常に動いているので、開いているダイアログ・ボックスにメッセージを書き込んでいる時に、そのままシンボルとして得るようなことはできない。

利用例



入力されたメッセージはOKボタンをクリックした時に出力される
ダイアログ・ボックスが開いている間も他の処理は続いている

dialog オブジェクトによって
ダイアログ・ボックスが開く

参照

message
opendialog
savedialog
sprintf

任意のメッセージを送る
ファイルやフォルダを選択するためのダイアログを開く
保存するファイルの名前を尋ねるダイアログを開く
単語や数値のメッセージをフォーマットする



入力

(ドラッグ) ロック状態のパッチ・ウインドウ内にある**dropfile**オブジェクトへFinderからファイルのアイコンをドラッグしてくると、**dropfile**オブジェクトはそのファイル形式を調べる。そのファイルが受け取り可能なファイルなら、**dropfile**オブジェクトの輪郭はハイライトされる。**dropfile**オブジェクト内にマウスカーソルがある状態でマウスボタンを放すと、**dropfile**オブジェクトはそのファイル形式とフル・パス名をアウトレットから出力する。

types 4個までの文字から成るタイプを表す記号を1つ、またはそれ以上伴った**types**メッセージは**dropfile**オブジェクトが受け付けるファイル・タイプを設定する。マッキントッシュ上のファイルに対応するタイプを表す記号の例を挙げれば、**TEXT**はテキスト・ファイル、**maxb**はMaxのバイナリー形式になっているパッチ・ファイル、そして**AIFF**はAIFF形式のオーディオファイルである。アーギュメントを伴わない**types**メッセージは、すべてのファイル形式を受け付けるように設定するが、これはデフォルトの設定である。

border 1または0を伴った**border**メッセージは、**dropfile**オブジェクト自身の境界線を表示するかどうかを設定する。デフォルトでは境界線はない。

アーギュメント

なし。

出力

シンボル 第1アウトレットからの出力：**dropfile**内に受け入れ可能なファイル・アイコンがドラッグされてきて、マウスのボタンがそこで放される時、そのファイルのフル・パス名が単一のシンボルとして出力される。絶対パスはMax 4.1以前のバージョン(コロンを使用)を除き、スラッシュで区切られて表示される。具体的には下記のようなになる。

“C:/Max Folder/extras/mystuff/mypatch.pat”

パスの形式や種類を変更したい場合は、**conformpath**オブジェクトを使用する。

フォルダのエイリアスをドラッグ&ドロップした場合は、元のフォルダの位置が解析され、その絶対パスが表示される。

dropfileオブジェクトを、ファイル名をアーギュメントに伴った**read**メッセージを受け付けることのできる他のオブジェクトと組み合わせ



て、ファイルの情報を読み込めるように使いたい場合、以下の例で見られるように、**prepend read**オブジェクトを**dropfile**オブジェクトとファイルを読み込むオブジェクトの間に置けばよい。

任意のシンボル 第2アウトレット：受け入れ可能なファイルのタイプを表す記号が第2アウトレットから出力される。

利用例

ファイルのドロップを受け付ける
ファイルタイプを指定



dropfileに重ねられたファイルは、
collに読み込まれる

参照

absolutepath
relativepath
strippath
opendialog

ファイル名を絶対パスに変換する
絶対パスを相対パスへ変換する
フル・パス名からファイル名を取得する
ファイルやフォルダを選択するためのダイアログを開く

入力

- bang** 第1インレット：**drunk**オブジェクトは直前に格納した数値を基準にして、それよりも大きいか小さい数値になるようにランダムに進行する。格納される数値は常に更新されながら、数値が出力される。
- 整数** 第1インレット：格納されている数値と入れ替わり、アウトレットから出力される。
- 第2インレット：**drunk**オブジェクトが出力できる最大値として記憶される。(注記：0より小さい数値が指定された場合、最大値は0になる。)
- 第3インレット：第1インレットに**bang**メッセージが送られた反応として得られる数値の変化量を設定する。この数値の絶対値より小さい数値でいつも（大きい方へ）進んでゆく。
- 実数** 整数に変換される。
- リスト** 第1インレット：リストの中の2番目の数値は**drunk**オブジェクトから出力される数値の最大値を設定し、(もしあるならば) 3番目の数値は変化量を設定する。そして、1番目の数値はオブジェクト内に格納されている数値と入れ替わり、アウトレットから出力される。
- set** 第1インレット：数値を伴った**set**メッセージは、その数値をアウトレットから出力することなしに、**drunk**オブジェクト内にその数値を格納する。格納される数値の初期値は、**drunk**オブジェクトが出力する数値全体の中心の数（最大値の半分の値）として設定される。
- seed** 第1インレット：数値を伴った**seed**メッセージは、**drunk**オブジェクトのランダム・ジェネレータに対してシードを与える。これにより特定の（再現可能な）疑似ランダムな数値を結果として得ることができる。数値0はシステム起動時から経過した時間（予測不可能な数）をシードとして使い、予測できない数値結果が得られる。この予測不可能なシードは**drunk**オブジェクトが生成された時にデフォルトで使われる。

アーギュメント

- 整数** オプション。最初のアーギュメントは**drunk**オブジェクトが出力する最大値の初期値として設定する。第2のアーギュメントは**drunk**オブジェクトが行うランダムな進行の幅を制限する初期値を設定する。これにより変化量の絶対値は常にこの制限値の絶対値より小さいものとなる。アーギュメントが与えられていない時、最大値は128、変化量の制限は2（1のみの幅で増減）として設定される。

出力

整数 アウトレットから出力される数値は、自動的に0から指定された最大値までの値に限定され、最大の変化量よりも小さい範囲内で直前に格納された値とは異なる数値になる。

利用例



数値は全体の範囲内において少しずつふらふらと進む

参照

decide
random
urn

オンあるいはオフ (1あるいは0) をランダムに選択する
ランダムな数値を発生させる
重複しないランダムな数値のジェネレーター

入力

- bang** **dump**メッセージと同じ。2つの項目から成るリストを出力する。配列のインデックスとそのインデックスにおける数値から成るこのリストは、**env**オブジェクトの水平方向と垂直方向の位置に対応する。
- 実数** 整数に変換される。
- set** 配列のインデックスの数値と、そのインデックスに格納される数値を伴った**set**メッセージは、そのインデックスに対する数値の位置を再描画するが、この時にアウトレットからは何も出力されない。
- embed** 0以外の数値を伴った**embed**メッセージによって、**env**オブジェクトの置かれているパッチの一部としてスクリプト・ファイルを保存するようになる。その後パッチを保存しておくだけで、**env**オブジェクトはスクリプト・ファイルを探す必要がなくなる。**embed 0**メッセージが入力されると、**env**オブジェクトはパッチが閉じられる時に、スクリプト・ファイルの内容を保存しないようになる。どちらの場合でも、**embed**メッセージの効果を反映させるためには、メッセージを入力した後にパッチを保存しなくてはならない。
- open** **env**オブジェクトのウィンドウを表示する。ウィンドウは最前面に表示される。パッチがロック状態の時に**env**オブジェクトをダブル・クリックをしても、同じ効果が得られる。
- wclose** **env**オブジェクトのウィンドウを閉じる。

envオブジェクトはスクリプト定義できるユーザー・インターフェースであり、シンセサイザー・パッチ・エディターにおけるエンベロープ編集に適している。

このオブジェクトは2つの顔を持っている。1つはエンベロープの表示・編集を**env**オブジェクト自身のウィンドウで行う場合であり、もう1つは**envi** (“envy”と発音される) オブジェクトという、パッチ・ウィンドウの中でエンベロープを確認できるユーザー・インターフェースを使用する場合である。別の言い方をすれば、このどちらのオブジェクトも一般的には、この**env**オブジェクトの説明を参照すれば良い。

envオブジェクトはスクリプトによって定義される。それはエンベロープ内における位置と、そこに割り当てられるデータ値から成るテキスト・ファイルである。スクリプトの読み込みが成功すると（言い換えると、シンタックス・エラーが生じなかった時）、ユーザーは**env**ウィンドウに表示されるデータ・ポイントを変えることができる。**env**オブジェクトはそのスクリプト・ファイルを保存して、次回にパッチ内へ読み込むことができる。

アーギュメント

シンボル **env**オブジェクトはオプションとしてアーギュメントを持つことができ、それはスクリプト・ファイルを示すシンボルである。スクリプト・ファイルはエンベロープの動作と外観を定義するものである。

envオブジェクトはユーザー・インターフェースのオブジェクトであるため、いわゆるアーギュメントは持つことができない。しかし、**env**と**envi**オブジェクトはどちらも、そのオブジェクトを含むパッチ・ファイルの中に、最後に読み込んだスクリプト・ファイルの名前を保存しておくことができる。

新しいスクリプト・ファイルは**read**メッセージによって開くことができる。また、**envi**オブジェクトを選択し、Objectメニューの**Get Info...**を選んで標準ファイル・ダイアログを開き、スクリプト・ファイルを選ぶこともできる。

エンベロープの構成

エンベロープは階層的に配置されたスクリプト・メッセージによって定義される。**env**オブジェクトと**envi**オブジェクトは、どちらもスクリプト・ファイルという同一のファイル形式を使用する。

各**env**オブジェクトは1つのウィンドウ (**envi**オブジェクトは、厳密にはパッチ・ウィンドウのボックス)、グループを示す数値と、それらを論理的に結ぶポイントを持っている。各ポイントは水平方向と垂直方向、あるいはどちらか一方、から成り立ち、各方向は1つ以上の表示尺度を持つことができる。これはエンベロープ・ウィンドウ内に表示される内部データ値に対応する。

スクリプト・メッセージ

スクリプト・ファイルの形式は**lib**オブジェクトのスクリプトと似ている。**#E**の後に、(**group**または**point**といった) キーワード・メッセージと、そのメッセージのアーギュメントを伴ったものである。例として下記のスクリプトの例の部分参照のこと。

windowメッセージ

envオブジェクト全体と、その表示に適応されるパラメーターを定義する。

シンボル 1. エンベロープ・ウィンドウのタイトル (**envi**オブジェクトには適応されない)。タイトル内に空白文字を使うためには、シングル・スマート・クォート (‘、option+]、option+shift+) で得られる) で囲む必要がある。

整数 2. 水平方向のサイズ。ピクセル単位でウィンドウの (**envi**オブジェクトの場合はボックスの) サイズを決める。ウィンドウにスクロール・

バーを表示させるためには15ピクセル以上が必要である。

- 整数 3. 垂直方向のサイズ。
- 整数 4. グループの数。各グループは後に置かれるグループ・メッセージによって定義される（下記を参照のこと）。
- 整数 5. エンベロープを定義するデータ量の数値。
- 整数 6. 左余白。ウィンドウ（またはボックス）の左端から、エンベロープやテキストが表示される領域までの距離をピクセル単位で表す。
- 整数 7. 下の余白。ウィンドウ（またはボックス）の下から、エンベロープが表示される領域までの距離をピクセル単位で表す。
- 整数 8. 上の余白。ウィンドウ（またはボックス）の上端から、エンベロープが表示される領域までの距離。コメントの部分を考慮すべきなので、20ピクセル以上が望ましい。

groupメッセージ

ポイントを論理的に結ぶグループを定義する。これは一般に1つのエンベロープと考えられるが、**env**オブジェクトは1つのウィンドウに任意の数のグループを持つことができる。

- 整数 1. グループ・ナンバー。定義されるグループを決定する（1から始まる数値として指定する）。
- シンボル 2. グループの名前。エンベロープのディスプレイ部内に表示される特定のパラメーター名や数値に先行して表示される。グループの名前が不要の場合は、**none**というシンボルを使うことができる。
- 整数 3. グループ内のポイントの名前。これらは下記のポイント・メッセージを定義する。
- 整数 4. 表示。1の時はこのグループが表示され、0の時は非表示となる。
- 整数 5. ディスプレイ・フラグ。1の時は、ドラッグされているパラメーターの名前とポイントの数値だけを表示する。0の時は、ドラッグされているポイントの所属するグループの、すべてのパラメーターの名前と数値を表示する。他のディスプレイ・フラグは後で定義される。
- 整数 6. (オプション) 色。1から255の数値を、アプリケーションで使われるパレットのインデックスとして使う（16色を使う場合、ID9998または9997のplttリソースとして格納されている）。

pointメッセージ

エンベロープ内でのポイントのアピアランスを定義する。

- 整数** 1. 定義されるポイントのナンバー。各グループの最初のポイントはナンバー1である。
- 整数** 2. ポイントの中央に置かれる、円形あるいは正方形のボタンのサイズ（ピクセル単位）。
- 整数** 3. ボタン・フラグ。ビット（言い換えれば、0または1の数値）で表され、0の時にはボタンは四角になり、1の時はボタンは円形になる。1ビット（0または2の数値）では、1の時にボタンは立体的になり、0の時はボタンは透明になる。2～6ビットでは、白黒パターンのインデックスを指定する。一般的な白黒パターンのインデックスを知るために、ResEditを使ってシステム・ファイルを調べ、ID 1のPAT#リソースを参照すれば良い。
- 整数** 4. ポイントからの線。このポイントのナンバーより小さい数値であるべきだが、0以外の数値が与えられると、このポイントと線で結ばれる他のポイントを指定することになる。この結びつきは表示上のみのものである。ポイント間での論理的な依存関係を作るためには、下記の**horiz**と**vert**メッセージを使用する。

horizメッセージとvertメッセージ

これらのメッセージは、各ポイントの2つの方向上の外観を定義する。エンベロープの仕様を決める多くの要点はこれらのメッセージの中に含まれる。1つの方向に固定したい場合は、これらのメッセージによって、特定のポイントごとに方向を定義する必要はなくなる。**horiz**と**vert**メッセージのアーギュメントは、どこに記入するかを除けば、まったく同じものである。

- シンボル** 1. パラメーターの名前。この名前（例、Rate 1）は、一定の方向で動くポイントと関連付けられる。ポイントと関連付けられるパラメーター名がない場合は、**none**を使うことができる。
- 整数** 2. データ・インデックス。このインデックスは、パラメーターの数値と対応するデータ値の配列（0から始まる）の中にある。この方向と関連付けられるデータがない時は、-1を使用する（この方法は、エンベロープ内で1つ以上のポイントに複数の方向が与えられる時には、一般的ではない）。このデータ・インデックスと数値を含んだリストが**env**オブジェクトに送られると、ポイントはそれに応じて動く。

すべてのデータ値は整数で格納される。スケールの表示やテーブルを

定義することで、パラメーターのコメント表示において実数のポイント・ナンバーを表示することができる（下記のスケール・メッセージを参照のこと）。

整数 3. パラメーターの最小値。

整数 4. パラメーターの最大値。

整数 5. パラメーターの初期値。

整数 6. パラメーターの増加量。現在のところサポート外であるため、1が望ましい。

シンボル 7. 単位名。パラメーターの単位（例、msはミリ秒、%はパーセント）。YAMAHAのシンセサイザーに使われる単位のように、どのパラメーターの単位とも結びつかない場合は**none**を使う。

2つのポイントが水平方向または垂直方向で互いに結ばれるということは、1つのポイントの変化が他のポイントにも連結されることを指す。連結は、ナンバーの大きなポイントが小さなポイントと連結される、と表現される。絶対的な連結と相対的な連結の、2つの連結方法がある。絶対的な連結とは、あるポイントのスクリーン上での変化が、他のポイントへ正確に反映されることを指す。相対的な連結は、水平方向ではもっとも一般的に使われるが、スクリーン内の任意のポイントの位置は他のポイントとの距離に関連して配置される。下記スクリプトの例の2番目の例に見られる一般的なエンベロープでは、ポイント2がポイント1と相対的に水平方向で連結され、ポイント3がポイント2と相対的に水平方向で連結されている（故にポイント1とも連結される）、そしてポイント4はポイント3と相対的に水平方向で連結されている。ポイント1が左右に動くのを許すならば（この例のために、初期的な遅れがエンベロープにある場合）、すべてのポイントがその動きに従って動くことになる。垂直方向に連結されるポイントはない。DX7のエンベロープでは最後が0にならないものもあるが、そうしたものは慣習的に、ポイント1と4を絶対的に垂直方向で連結する。水平方向のポイントと垂直方向のポイントを連結することはできない。

整数 8. 絶対的な連結をするポイント。このポイント・ナンバーは絶対的に連結されるポイントを指し（連結するポイント・ナンバーより小さな数値でなくてはならない）、この連結されるポイントは連結するポイントと、同じ水平方向か垂直方向で表示される。どのポイントとも連結させない時は0を使う。

整数 9. 固定。このポイントをスクリーン上で特定の位置に留めたい時は、1を使う。それ以外は0を使う。エンベロープにおける最初の（最も左の）ポイントで、水平方向か垂直方向で使うのが好ましいだろう。

- 整数** 10. 相対的な連結をするポイント。このポイント・ナンバーは、ある方向において相対的に連結されるポイントを指し（連結するポイント・ナンバーより小さな数値でなくてはならない）、このポイントの位置は（数値に依存するが）、水平方向か垂直方向に連結されるポイントとのオフセットとなる。ある方向において他のポイントと連結しない時は0を使用する（一般的に、垂直方向には適している）。
- 整数** 11. 正の方向。ポイントの増加方向を設定する。垂直方向においては、0はカーソルがスクリーンの上に向かって動くことが数値の増加と見なされ、1はカーソルがスクリーンの下へ向かって動くことが数値の増加と見なされる。水平方向においては、0はカーソルが左に動くことが数値の増加と見なされ、1はカーソルが右に動くことが数値の増加と見なされる。
- 整数** 12. 適応範囲のサイズ。パラメーターを配置する割合を、どれほどのピクセル数で表示するかを決定する。ガーデン・バラエティー・エンベロープでは、垂直方向の空間をほとんど全部、エンベロープの垂直方向のために使うのが一般的で、以下のような決まり文句を使って定義する：<ウィンドウの垂直方向のサイズ> - <コメント部分の高さ> - <上部の余白> - <下部の余白>
水平方向は、エンベロープ内のポイント総数から決まる空間と、ユーザーがエンベロープを最大に伸ばす時に、どれほどの移動を許すか、ということから決定される。

scaleメッセージ

エンベロープ内に格納されるデータである内部数値（整数）と、それらを実数として表示する間での転換を定義する。エンベロープのパラメーターが物理的な数量を表す時、制作者はよく割合の係数を使用する。scaleメッセージを使って、格納されている数値を他の整数や実数に転換する数学的な記述を指定することができる。転換された数値はウィンドウのコメント部分に表示される。

scaleメッセージは**expr**オブジェクトのアーギュメントの書式で表現するか、または内部データの数値を（シンボルも含む）数値のリストで表すことができる。

各方向は任意のスケールを持つことができ、それらは指定された範囲をかなり超えるものでも良い。データ値に適応される縮尺がない場合、ウィンドウのコメントは内部データ値を表示する。エンベロープの最小値がoffと表示されるようなものを使いたい時、0がoffという言葉に置き換えられ、他の数値は変化しないスケールを使えば良い。

整数 1. 最小値。スケールが適応される最も小さな値。

整数 2. 最大値。スケールが適応される最も大きな値。

整数 3. 小数点以下の数字。表示される実数の小数点以下の数。

シンボル 4. isまたはtableという言葉。次に続くものを数学的な表現として解釈するか、数値のテーブルとして解釈するかを決定する。

5. 追加のデータ。式の利用：**\$i1**はウィンドウに表示される内部データを表す。例：

\$i1 * .07; 係数を使用して内部数値を掛け算する。

\$i1 - 1; 内部数値から1を引く。

(\$i1-1) *.07; 複合的な式。

100 - \$i1; 内部数値の位置を逆にする。

テーブルの利用：数値のリストであり、空白文字で区切られた連続する内部データから成る。テーブルは240までの要素を含むことができる。追加の**scale**メッセージを使って大きなテーブルを使うことができる。例：

table off 10 20 30 40;

これは、最小値が「off」と表示され、次が10、その次が20、というようになる。

scaleメッセージを使った他の例：

#E scale 0 0 0 table Off; （最小値を「off」という言葉で表す）

#E scale 1 10 2 is \$i1 * .04; （.04を追加の数値とし、浮動小数点数2数を表示する）

phaseメッセージ

このメッセージは、前もって定義する垂直方向のポイントに付く符号を指定する。エンベロープのパラメーターが負の値になるか、位相で表される分けられたデータ（0は負、1は正）になる。**phase**メッセージは**vert**メッセージのすぐ次に置かれる。

commentメッセージ

このメッセージはスクリプトの中で**comment**で始まる。これはシングル・ラインとセミコロンで仕切られる。

endメッセージ

このメッセージはスクリプトの最後に置かれる。これは**env**オブジェクトを再設定し、必要であればウィンドウあるいはボックスの表示を変化させる。アーギュメントはない。

スクリプトの例

下記のスクリプトは独立した4つのグループから成るエンベロープを定義しているが、これはかつてタップ・エディターとして使われたものである。ポイントの水平位置はディスプレイを決定し、垂直位置は元のシグナルを繰り返す割合を決定する。スクリプトの後に図を示す。

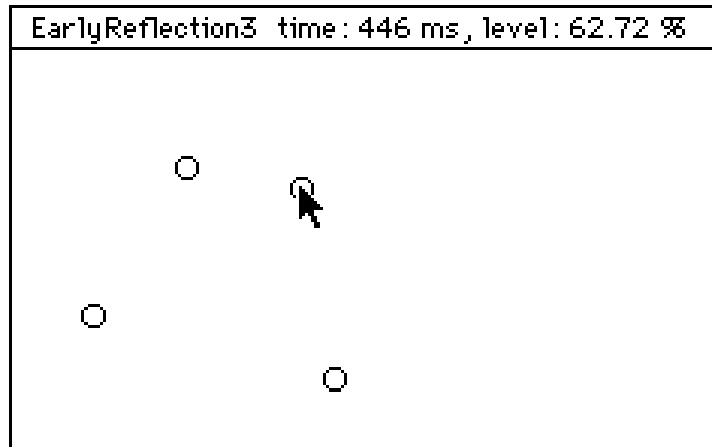
```
#E window ERFEnv 400 148 4 96 8 8 24;
#E group 1 EarlyReflection1 1 1 1;
#E point 1 8 1 0;
#E horiz time 0 1 500 1 1 ms 0 0 0 0 100;
#E vert level 1 0 1024 0 1 % 0 0 0 0 100;
#E scale 0 1024 2 is $i1 * .0977;

#E group 2 EarlyReflection2 1 1 1;
#E point 1 8 1 0;
#E horiz time 2 1 500 1 1 ms 0 0 0 0 100;
#E vert level 3 0 1024 0 1 % 0 0 0 0 100;
#E scale 0 1024 2 is $i1 * .0977;

#E group 3 EarlyReflection3 1 1 1;
#E point 1 8 1 0;
#E horiz time 4 1 500 1 1 ms 0 0 0 0 100;
#E vert level 5 0 1024 0 1 % 0 0 0 0 100;
#E scale 0 1024 2 is $i1 * .0977;

#E group 4 EarlyReflection4 1 1 1;
#E point 1 8 1 0;
#E horiz time 6 1 500 1 1 ms 0 0 0 0 100;
#E vert level 7 0 1024 0 1 % 0 0 0 0 100;
#E scale 0 1024 2 is $i1 * .0977;

#E end;
```



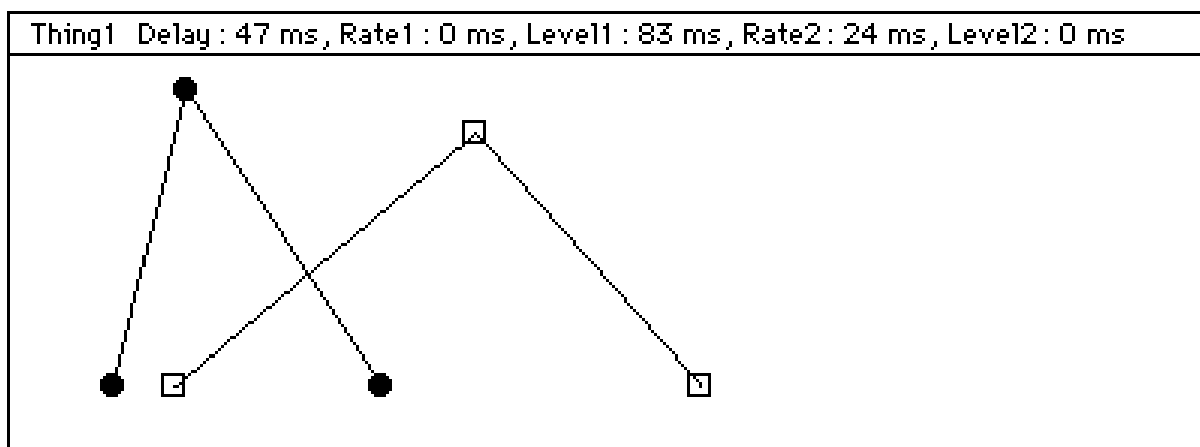
スクリプトの例1を反映したオブジェクトの図

下記のスクリプトは、3つのポイントから成る伝統的なシンセサイザーの音量エンベロープを、2つのグループとして定義する。最初のポイントは垂直方向において確定されているが、水平方向には動かせる。他の2つのポイントはどちらの方向にも動かすことができ、3つのポイントは線で結ばれている。スクリプトの後に図を示す。

```
#E window TestEnv 400 148 2 10 8 8 24;

#E group 1 Thing1 3 1 0;
#E point 1 8 0 0;
#E horiz Delay 0 0 99 0 1 ms 0 0 0 0 100;
#E vert none -1 0 99 0 0 none 1 0 0 0 100;
#E point 2 8 0 1;
#E horiz Rate1 1 0 99 50 1 ms 0 0 1 1 100;
#E vert Level1 2 0 99 50 1 ms 0 0 0 0 100;
#E point 3 8 0 2;
#E horiz Rate2 3 0 99 50 1 ms 0 0 2 1 100;
#E vert Level2 4 0 99 50 1 ms 0 0 0 0 100;

#E group 2 Thing2 3 1 0;
#E point 1 8 3 0;
#E horiz Delay 5 0 99 0 1 ms 0 0 0 0 100;
#E vert none -1 0 99 0 0 none 1 0 0 0 100;
#E point 2 8 3 1;
#E horiz Rate1 6 0 99 50 1 ms 0 0 1 1 100;
#E vert Level1 7 0 99 50 1 ms 0 0 0 0 100;
#E point 3 8 3 2;
#E horiz Rate2 8 0 99 50 1 ms 0 0 2 1 100;
#E vert Level2 9 0 99 50 1 ms 0 0 0 0 100;
#E end;
```

スクリプトの例2を反映したオブジェクトの図

メッセージの入力

データとして任意の数値を持つことができるので、**env**オブジェクトはただ1つのインレットしか持たない。エンベロープ・データは配列として格納される。スクリプト・ファイルは配列のインデックスをどのようにエンベロープ内の水平位置、垂直位置に置くかを決定する。

list **env**オブジェクトに入力されたリストは、データ・ポイントの新たな数値として格納される。リストの最初の数値は位置（配列のインデックス）を決め、2番目の数値はその位置に格納されるデータ量を決める。**env**オブジェクトはスクリプト・ファイルによって決められた各データの最小値、最大値に従って、入力される数値の範囲を制限する。

funnelオブジェクトは、そのインレットのひとつに入力される数値を使って、最初の項目は入力されたインレットの位置に対応する数値で、2番目の数値はそのインレットに入力された数値から成るリストを出力する。これは**env**オブジェクトに必要なリストを用意するために設計されている。

整数 0から配列のインデックスの最大値までの整数が入力されると、**env**オブジェクトはそのインデックスの数値とそこに格納されたデータ量から成るリストを出力する。

show グループ・ナンバーを伴った**show**メッセージは、そのグループを表示する。2つの数値を伴う時、最初の数値から2番目の数値までにあるグループの範囲を表示する。

hide グループ・ナンバーを伴った**hide**メッセージは、そのグループを非表示にする。2つの数値を伴う時、最初の数値から2番目の数値までにあるグループの範囲を非表示にする。

- open** **env**オブジェクトのウィンドウを開き、最前面に表示する。**envi**オブジェクトには適応されない。
- read** 標準ファイル・ダイアログを開き、ユーザーに新しいスクリプト・ファイルを選択させる。
- dump** エンベロープの現在のデータをすべて、2つの項目から成るリストで連続して出力する。最初の数値はデータのインデックスで2番目の数値はデータの値である。

出力

- リスト** マウス・ボタンが離された時、またはインレットに数値が入力された時、**env**オブジェクトはアウトレットから2つの数値から成るリストを出力する。最初の数は配列のインデックスであり、2番目の数はインデックスにおける新たな数値である。新たに修正された数値のみが出力される。**env**オブジェクトのインレットに**dump**メッセージが入力されると、リスト形式ですべてのデータが出力される。

sprayオブジェクトはリストを受け取ると、そのリストの最初の数値に対応するアウトレットから2番目の数値を出力する。これは**env**オブジェクトのアウトレットから出力されたリストを、独立したアウトレットへ分配し、ナンバー・ボックスや**libto**オブジェクトのようなエディターに表示させるために設計されている。

エンベロープ・ウィンドウまたはエンベロープ・ボックスの使用

エンベロープの表示は水平線で2つの領域に分かれている。上の領域は15ピクセル分の空間でユーザーが変更したポイントの名前と数値が、9ポイントのGenevaフォントで表示される。下の領域は実際のポイントのグループが表示され、これらは線で結ばれている場合とそうでない場合がある。

envオブジェクトのウィンドウ（あるいは**envi**オブジェクトのボックス）を使うのは簡単である。表示されているポイントをクリックすれば良い。キーが押されていない状態では、データ量は動いたピクセル数に対応して増加する。どれほどの変化量になるかは、各方向におけるパラメーターの範囲（最大値と最小値の違いによって異なる）に対する範囲サイズのアーギュメントの割合によって決定される。例えば、上記の最初のスクリプト例を見ると、そこには1024のデータと、100の範囲サイズがある。この場合、1ピクセル分カーソルを動かすと数値は $1024/100$ 、あるいは10ほど変化することになる。

shiftキーを押した状態では、ポイントをドラッグする動きは、カーソルが早く動かないように制限される。shiftキーを放せば、この制限はいつでも解除される。

commandキー (Macintosh) あるいはCtrlキー (Windows) を押した状態では、マウスの移動はファイン・モードになる。パラメーターの範囲サイズの割合に関係なく、データはマウスを1ピクセル動かすごとに1ずつ変化するようになる。

マウスをドラッグ中にcommandキー (Macintosh) あるいはCtrlキー (Windows) を押すか離すことで、ファイン・モードを有効にするか無効にするかを、即座に切り替えることができる。

参照

envi	パッチ・ウィンドウに置く、スクリプトによって定義できるエンベロープ
funbuff	x と y の数値のペアを対にして格納する
funnel	受け取ったインレットの番号を付け加えて出力する
line	ある値から異なる値へと、補完しながら数値を連続して出力する
multiSlider	複数のスライダーとスクロール表示
spray	整数を、アウトレットの番号に従って分配する



enviオブジェクトは**env**オブジェクトのパッチ・ウィンドウ版である。**env**オブジェクトでの説明は、**envi**オブジェクトでも同様である。

入力

整数 **error**オブジェクトはエラーを検知し、**Max**メッセージとしてそれらを出力する。0以外の数値を受け取れば、**error**オブジェクトはMaxエラーの検出を開始する。**error**オブジェクトはどのような出力においても検知を行う。0を受け取れば、検出を停止する。

実数 整数に変換する。

アーギュメント

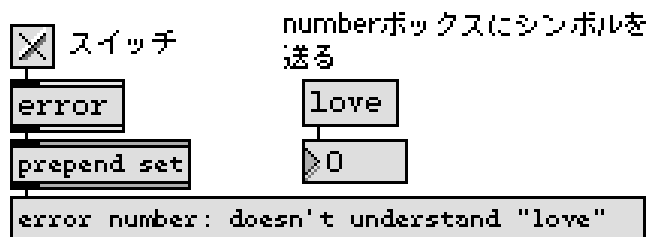
なし。

出力

シンボル **error**オブジェクトが検出した任意のパッチのオブジェクトから生じるMaxエラーは、シンボル**error**の後に続けて、**error**オブジェクトのアウトレットから出力される。特定のエラーの確認ができるように、個々の単語としてメッセージが出力される。

始めに付く**error**メッセージを出力から分離したい場合は、**route**オブジェクト（アーギュメントに**error**を与える）を使用する。**error**オブジェクトの出力にメッセージを表示させたい場合は、**route**オブジェクト（アーギュメントに**error**）とエラー・メッセージを表示するオブジェクトの間に、**prepend**オブジェクト（アーギュメントに**set**）を入れる。

利用例



エラー・メッセージの検出

参照

print

Maxウィンドウに任意のメッセージを出力する

入力

- 整数** 各インレットで受け取る数値は、それに関連する**\$i**または**\$f**アーギュメントの場所に格納される。例：どの場所にあっても、第2インレットの数値は**\$i2**または**\$f2**アーギュメントの場所に格納される。
- 実数** 各インレットの数値は、それに関連する**\$f**または**\$i**アーギュメントの場所に格納される。**\$i**アーギュメントにおいては、数値は切り捨てとなる。
- symbol** **symbol**の後に**table**名が続くメッセージは、**table**に格納されている値にアクセスするために、そのインレットに関連する**\$s**アーギュメントの場所に格納される。
- bang** 第1インレット：現在格納されている値を使って数式の値を求める。
- リスト** 第1インレット：それぞれが異なったインレットから入力されたかのようにリストのアイテムが扱われ、式の値が求められる。リストの内容がインレットの数より少ないアイテム数であるならば、残りのインレットでは直前に受け取った値が使われる。
- 第1インレットでの上記のメッセージはいずれも計算を行い結果を出力する。置き変わることになるアーギュメントの数値を受け取っていない場合は、その数値を0として数式を評価する。
- インレット数は、置き変わることになるアーギュメントがいくつあるかで確定される。インレットの最大数は9個。
- set** 一つ以上の数値が続く**set**メッセージは、異なったインレットにそれぞれ入力されたかのように、それらの数値を処理し、格納されている数値を新しい数値で取り換える。しかし、計算は行われず、何もアウトレットからは出力されない。メッセージの数値がインレットの数より少ない場合、残りのインレットで格納されている値は変更されない。

アーギュメント

必須。式は、+または*のような算術演算子、<or>のような比較、**min()**や**pow()**のようなC関数、インレットに入力される整数、実数、シンボルを置き換えるアーギュメント (**\$i**, **\$f**, **\$s**) といった数値によって成り立つ。

整数または実数 数値は数式において定数として使うことができる。

\$iまたは\$f 変数のアーギュメントは、**\$i**、**\$f**を指し、インレット番号を指定する(例：**\$i2**)。そのアーギュメントは指定されたインレットに入力された数値に置き変わる。

\$s インレット番号とアーギュメントの**\$s**は、**table**を参照する名前として置き変わる。アーギュメントの直後には、**table**において指定するアドレスを鉤括弧でかこった数値が続かなければならない (例：**\$s2[7]**または**\$s3 [\$i1]**)。

その他 **expr**で理解される算術演算子は次の通りである：**+**、**-**、*****、**/**、**%**。他の演算子では、**~** (ビット補数)、**^** (ビット排他和)、**&**、**&&**、**|**、**||**、**!** (論理否定)。

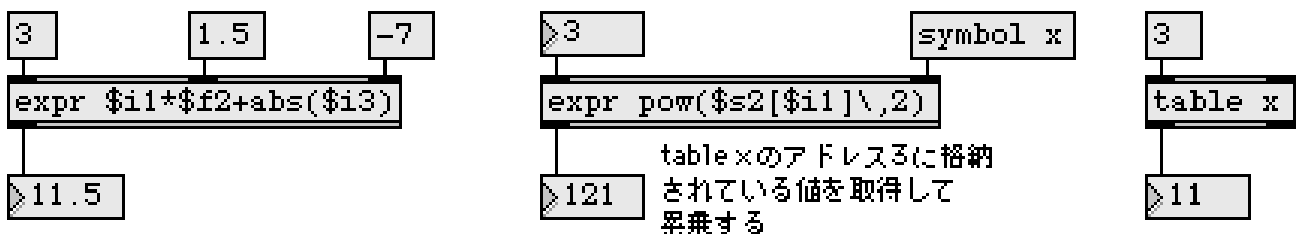
多くのC言語関数が**expr**で使用できる。関数の直後に、括弧でかこった関数に必要な任意のアーギュメントが続かなければならない。アーギュメントの中で関数がカンマを必要とするならば、Maxが誤認識しないように、カンマの前にバックスラッシュ (****) を入れなければならない。例えば、**pow (\$i1, 2)**のようにする。

exprで用いることができるC言語関数は次の通りである：**abs**、**min**、**max**、**sin**、**cos**、**tan**、**asin**、**acos**、**atan**、**atan2**、**sinh**、**cosh**、**tanh**、**int** (整数に変換)、**float** (実数に変換)、**pow**、**sqrt**、**fact** (階乗)、**exp** (eを低数とするべき乗)、**log10** (常用対数)、**ln**または**log** (自然対数)、**random**。max-startupフォルダにエクスターナル・コード・リソースを追加することによって、関数を追加することができる。

出力

整数または実数 出力は数式を評価して得られた結果である。

利用例



他のオブジェクトでは利用できない関数や多くの計算を一つのオブジェクトとしてまとめることができる

参照

if if/then/else形式の条件文
vexpr リスト同士の算術式を評価する

Tutorial 38 expr and if (英文PDF参照)

入力

シンボルとしての絶対パス名。絶対パスはMax 4.1以前のバージョン(コロンを使用)を除き、スラッシュで区切られて表示される。具体的には下記のようなになる。

“C:/Max Folder/extras/mystuff/mypatch.pat”

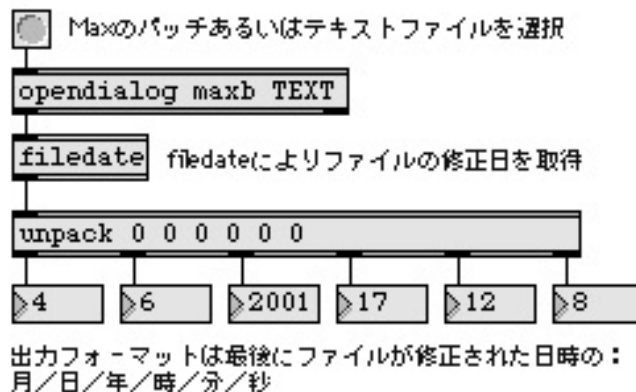
アーギュメント

なし。

出力

リスト ファイルが最後に変更された日時をリストとして出力する。

利用例



filedate はいつファイルが変更されたか示している

参照

- date** 現在の日付と時刻を出力する
- filein** ファイルからバイナリー・データを読み出す
- filepath** 現在のMaxサーチ・パスの情報を出力する
- folder** 指定したフォルダー内のファイル・リストを表示する
- opendialog** ファイルやフォルダを選択するためのダイアログを開く

入力

整数 バイナリー・ファイルの中のバイト・オフセットを指定して、ファイルからそこに記憶されているデータを出力する。

第1インレット：ファイルの中でバイト数でのオフセットで指定されたバイトが、第1アウトレットから出力される。

第2インレット：ファイルの中でバイト数でのオフセットで指定された16ビット長のワードが、符号無し整数（short型16ビット）として第1アウトレットから出力される。

第3インレット：ファイルの中でバイト数でのオフセットで指定された32ビット長のワードが、符号無し整数（long型32ビット）として第1アウトレットから出力される。

リスト 第1インレット：リストの2つ目の数値を第2インレットが受け取り、（存在するならば）3つ目の数値を第3インレットが受け取る。リストの1つ目の数値は第1インレットが受け取る。アウトプットは対応する順序で第1アウトレットから出力される。

read メモリに読み込むファイルを選択するために標準ファイル・ダイアログを表示する。**read**に続くファイル名がMaxのサーチ・パスの中で見つかった場合は、そのファイルは自動的にメモリに読み込まれる。

spool **int**メッセージを受信した時は必ずディスクにアクセスし、ファイルを選択するために標準ファイル・ダイアログを表示する。**spool**に続くファイル名がMaxのサーチ・パスの中で見つかった場合は、次のアクセスのために自動的にファイルが指定される。ファイルにアクセスする方法は、より少ないメモリで済むが、**int**メッセージに応じてすぐに割り込んで出力することはできない。

fclose 読み込まれたファイルを閉じる。その後は**filein**は**int**または**list**メッセージに対して応答しなくなる。

アーギュメント

シンボル オプション。パッチが読み込まれた時に、自動的にファイルを**filein**オブジェクトに読み込むためのファイル名を指定する。ファイル名の後に2つ目のアーギュメントとして**spool**が与えられた場合は、ファイルはメモリに読み込まれず、ディスクからアクセスされる。

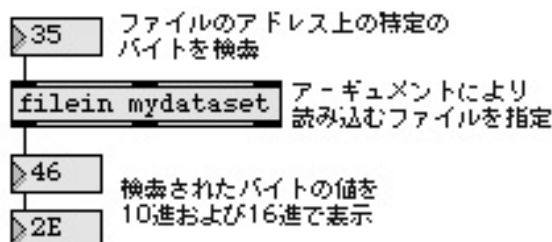
出力

整数 第1アウトレット：インプットから入力された整数によって指定された場所に記憶されている8、16、または32ビットを符号無し整数で出力する

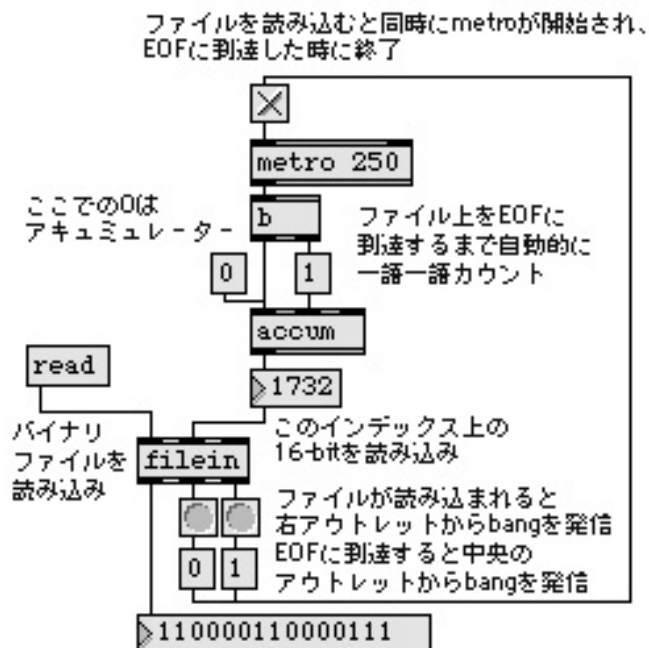
bang 第2アウトレット：ファイルのバイト数よりも大きな数値を受け取った時、ファイルの終わりに到達したこと（EOF）を示して、**bang**が出力される。

第3アウトレット：読み込みやスプールなどの処理が完了したことを示す。この**bang**はファイルがアクセスに成功したことで、**filein**が整数メッセージを受け取る準備ができたことを示す。

利用例



相当するバイナリー・ファイルからデータを読み出す



8、16、32ビットでファイルの内容を出力している

参照

Text

数値をテキスト・ファイルとしてフォーマットする

入力

任意のメッセージ シンボルとしてのサーチ・パス名。パス名には、スラッシュ、コロン、バックスラッシュのいずれかが区切り文字として使用される。パス名は具体的には以下のようなになる。

“drive:/folder/filename.ext” (絶対パス)

“./mypatches/steaksauce.ext” (相対パス)

bang **bang**は現在保存されているパス名をリストとして出力する。

append **append**に続くフォルダを指定するシンボルを、パスのリストに追加する。(しかし、初期設定ファイルには保存されない。)

set **set**に続くMaxのサーチ・パスの型の名前 (**search**、**startup**、**help**、**action**、**default**)によって、現在のサーチ・パスを指定された型に設定する。

revert **revert**はパス名を最後にMaxの初期設定ファイルに保存された状態にリセットする。

clear 現在指定されたパスを消去する。

アーギュメント

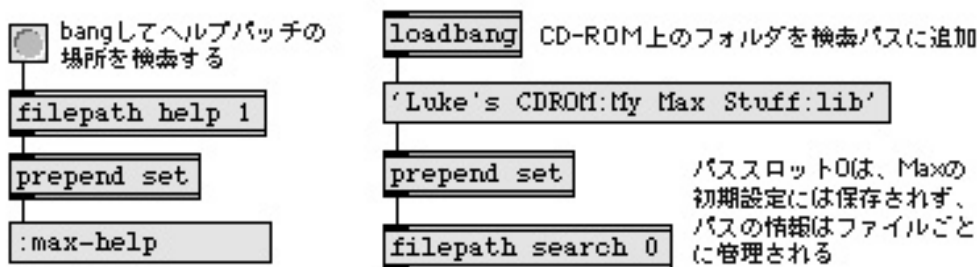
symbol 必須。Maxのサーチ・パスの型の中から一つ指定する。(**search**、**startup**、**help**、**action**、**default**)

int オプション。ゼロより大きい数は、初期設定ファイルの中のスロットを指定する。アーギュメントが0または数値以外のものが与えられた場合パスは初期設定ファイルには保存されない。あるパッチの一時的なサーチ・パスを作るためにこの特徴を使うことができる。action、help、そしてstartupパスには一つのスロットしかない。サーチ・パスは最大で256スロットまで作り出すことができる (通常は約8である)。デフォルトのパスは初期設定ファイルに保存されない。

出力

シンボル **bang**によって現在記憶されているパス名を出力する。

利用例



filepath をサーチ・パスの確認のために、またはあるパッチの一時的なサーチ・パス・スロットの設定のために使用する

参照

filedate

ファイルの最終修正日を出力する

filepath

現在のMaxサーチ・パスの情報を出力する

folder

指定したフォルダー内のファイル・リストを表示する

opendialog

ファイルやフォルダを選択するためのダイアログを開く

入力

- 実数** 第1インレット：現在格納されている値を入力される数値で置き換えて、アウトレットに出力する。
- 第2インレット：出力せずに入力される数値で格納されている値を置き換える。
- bang** 第1インレット：格納されている値をアウトレットから出力する。
- set** 第1インレット：出力せずに**set**に続く数値で格納されている値を置き換える。
- send** 第1インレット：**send**に続く**receive**オブジェクトの名前によって、**float**オブジェクトに格納されている数値をその名前を持つすべての**receive**オブジェクトに送信する。**float**オブジェクトのアウトレットからは出力しない。
- int** 実数に変換する。

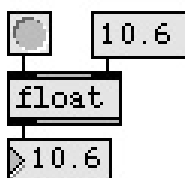
アーギュメント

- float** オプション。**float**オブジェクトに格納する初期値を設定する。アーギュメントがない場合の初期値は0.0に設定される。**float**オブジェクトにおいて、**float**という単語なしでアーギュメント単独の場合は**float**オブジェクトを作成して、初期値を設定するもう一つの方法となる。

出力

- 実数** 実数で格納された数値は単精度浮動小数点数として**float**オブジェクトに格納される。整数部分の精度を増すために、数値の小数部分の精度が減少するような実数の格納方法が用いられているので、パッチ・ファイルに保存された実数は再度ファイルを開いた時に僅かに変わることがある。

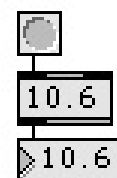
利用例



格納している値を出力する



格納している値と置き換えて出力する



初期値が与えられている

参照

int 整数を格納する
pv パッチとそのサブ・パッチ内に限定して、変数を共有する
value 他のオブジェクトと格納されている値を共有する

Tutorial 21 Data Structures (英文PDF参照)

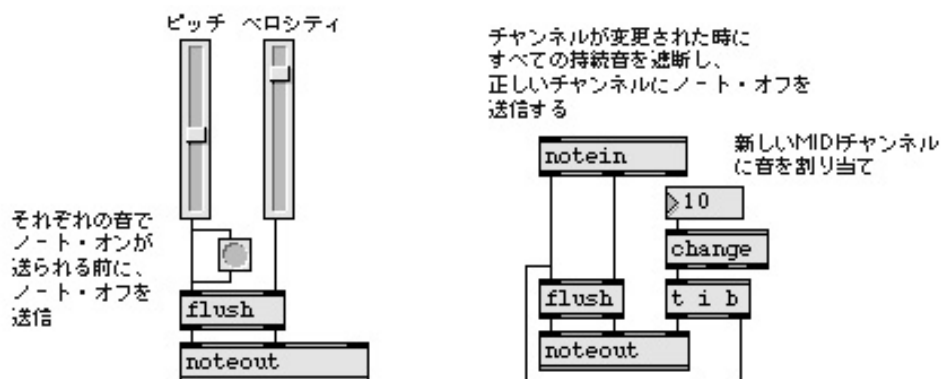
入力

- int** 第1インレット：数値はノート・ナンバーとベロシティのペアにおけるノート・ナンバーの値として扱われてノートが出力される。
- 第2インレット：数値は第1インレットで受け取った数値とペアになるベロシティとして格納される。
- list** 数値は整数でなくてはならない。ノート・ナンバーとベロシティのペアで最初の数値はノート・ナンバーとして扱われる、そして2番目の数値はベロシティとして扱われ、数値はそのままアウトレットから出力される。
- bang** 第1インレット：ノート・オフではなく、ノート・オンとして受信したすべてのノート・ナンバーに対して、第1アウトレットから出力されるノート・ナンバーの値に続いて第2アウトレットから**0**を出力することによって即座にノート・オフを出力する。
- clear** **flush**オブジェクトによって保持されているすべての数値を、ノート・オフを出力せずに消去する。アーギュメントなし。

出力

- int** 第1アウトレット：ノート・オンかノート・オフのノート・ナンバーの値を出力する。
- 第2アウトレット：ノート・オンかノート・オフのベロシティの値を出力する。**flush**オブジェクトは通過するノートの情報を取得し続ける。インレットに**bang**を受け取った時、ノート・オンとして保持しているすべてのノートに対してノート・オフ・メッセージを出力する。

利用例



保持されたノートにノート・オフを与えることによって、すべてのノートが鳴り終わるのを確認しよう

参照

bag	数値の集合を格納する
Borax	ノートオンとノートオフに関する情報を知らせる
makenote	ノート・オンに続けてノート・オフ・メッセージを生成する
midiflush	保持しているノート・オンに対するノート・オフを、生のMIDIデータとして出力する
offer	x, yを1組にした数値を一時的に格納する
stripnote	ノート・オフ・メッセージを取り除き、ノート・オン・メッセージのみを出力する
sustain	ノート・オフ・メッセージを保持して、コマンドによりそれを出力する

入力

bang 指定したフォルダ内にある指定したタイプのファイル名を得る。それらの名前はメッセージ・ボックスや**umenu**オブジェクトに格納するために出力することができる。

シンボル フォルダのパス名を指定する。そして、そのフォルダの中身が**umenu**またはメッセージ・ボックスに格納できるように出力される。パス名には、スラッシュ、コロン、バックスラッシュのいずれかが区切り文字として使用される。パス名は具体的には以下のようなになる。

“drive:/folder/filename.ext” (絶対パス)

“./mypatches/steaksauce.ext” (相対パス)

パス名に空白文字が含まれる場合には、**folder**がそのパス名が一つのアーギュメントであることがわかるように、パス名をダブル・クォート (") で囲まなければならない。あるいは、**folder**がその空白文字を特殊文字としてみなさないように、空白文字の前にバックスラッシュ (\) を置くこともできる。

types 1個以上4個までの文字が続く**types**というメッセージによって、**folder**オブジェクトが指定されたフォルダの中で探すファイルのファイル・タイプを設定できる。例えば、Macintoshファイルのファイル・タイプとしては、テキスト・ファイルを指す**TEXT**、Maxバイナリー形式のパッチ・ファイルである**maxb**、AIFFオーディオフォーマットである**AIFF**などがある。アーギュメントなしの**types**メッセージはすべてのファイル・タイプに対応し、これがデフォルトである。

整数 **bang**と同じ。

アーギュメント

シンボル オプション。マウントされたボリュームでフォルダの完全なパス名を指定する。

出力

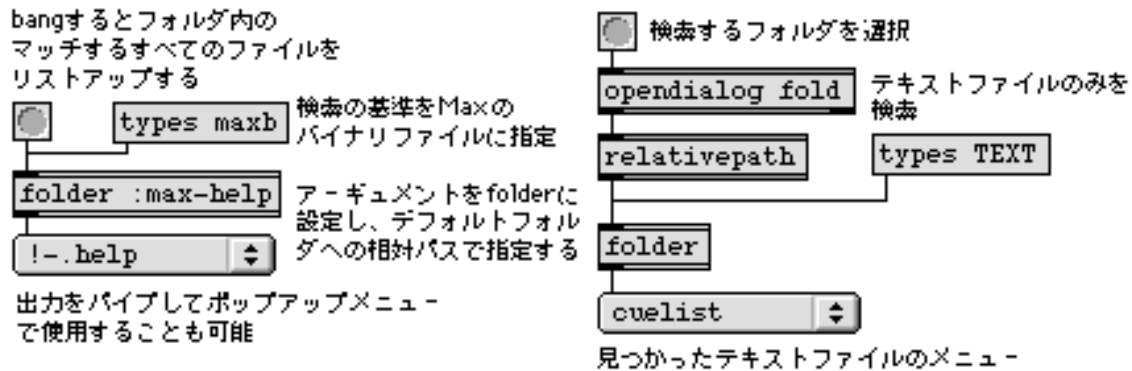
clear 第1アウトレット：インレットで**bang**またはパス名を受信した時、第1アウトレットから出力する最初のメッセージは**clear**である。これは、メッセージ・ボックスまたは**umenu**オブジェクトがすでに受け取っている内容を消去するためである。

append 第1アウトレット：**clear**メッセージに続いてすぐに、指定したフォル

ダの各ファイル名がアルファベット順に、**append**メッセージに続けて出力される。

整数 第2アウトレット：パス名または**bang**が入力されると、そのフォルダーにあるファイル数が第2アウトレットから出力される。

利用例



フォルダーからファイル名を読み出し、ポップアップ・メニューで呼び出す

参照

filein
filepath
opendialog
pcontrol

ファイルからバイナリー・データを読み出す
現在のMaxサーチ・パスの情報を出力する
ファイルやフォルダを選択するためのダイアログを開く
パッチ内のサブ・ウィンドウを開いたり、閉じたりする

入力

- record** インレットで受け取る整数の記録を開始する。
- bang** **follow**に記録されたシーケンスを再生する。
- start** **start**のみのメッセージは**bang**と同じ効果を持つ。数値が続く**start**メッセージでは、記録されたシーケンスを数値によって決められたテンポで再生する。**start 1024**というメッセージでは本来のテンポを示す。数値が**512**であれば**follow**は、記録された本来の速度から半分の速さで再生し、**2048**では本来の速度の倍の速さで再生する。
- follow** **follow**メッセージは**follow**オブジェクトが**seq**オブジェクトと異なる大きな特徴である。この動作は、**follow**はスコア・リーダーのようであり、先に記録されたものとライブなものを比較する。
- 数値が続く**follow**とメッセージで、**follow**に記録された数値と入力される数値を、指定されたインデックス（これは保存されたシーケンスの指定されたイベントを指定する）から比較を開始する。**follow**オブジェクトが追従（**follow**）中の場合、**follow**に記録された数値と一致する数値を受け取った場合、その数値のインデックスを出力する。
- たとえ入力される数値が記録されたシーケンスに完全に一致しなくても、**follow**オブジェクトは寛大なスコア・リーダーとして追従していく。数値がシーケンスの次の数値または2番目以降の数値に一致しないものを受け取った場合、**follow**は何もしない。多くとも2音先のシーケンスの数値と一致するものを受け取った場合は、**follow**は演奏者が単に1音か2音を間違えたと仮定して、一致する先の数値まで移動する。
- stop** 記録、再生、追従を停止する。記録から再生へ、あるいは追従から記録へのように直接切り替えることができ、その場合は**stop**メッセージを送る必要はない。
- next** 比較対象となっている数値とそのインデックスを出力し、一つ先に数値を進める。
- append** 既存のシーケンスを消去せずに、記録されたシーケンスの終わりから記録を開始する。
- 整数** 記録中であれば、インレットで受け取る数値はシーケンスとして記録される。数値は**seq**オブジェクトと同様に、MIDIメッセージ（**midiformat**または**midiiin**から）のバイトでなければならない。しかし、**follow**は個々の整数を記録する方法が**seq**とは異なる。すなわち、**follow**では一つの音高値としてノートを記録することができる。演奏

が完全なMIDIメッセージとして記録されているか、単なるノート・オンの音高として記録されているかに関わらず、**follow**はノート・オンの音高値に従って的確に演奏を追従することができる。

followが追従している時、そのインレットで受け取る数値はシーケンスで記録されている数値と比較される。数値がシーケンスの数値と一致するものを受け取った時、**follow**はその数値のインデックスを出力する。

- float** 整数に変換する。
- delay** 数値が続く**delay**というメッセージでは、記録されたシーケンスの最初のイベントの開始時間をミリ秒単位で設定する。
- hook** 実数が続く**hook**というメッセージでは、記録されたシーケンスのすべてのイベント時間をその実数分掛け合わせる。例えば、数値が2.0であるならば、すべてのイベント時間は2倍になり、シーケンス再生は半速となる。シーケンスが再生中であっても、この乗算を実行することができる。
- write** 標準ファイル保存ダイアログが開き、**follow**のシーケンスをファイルとして保存する。
- read** アーギュメントなしの**read**というメッセージでは、標準ファイル・ダイアログが開き、**follow**にシーケンス・ファイルを読み込む。**read**の後にシンボルによるファイル名がアーギュメントとして続く場合は、そのファイルを探して**follow**に読み込む。
- print** 記録されたシーケンスの最初の数個のイベントをMaxウィンドウに表示する。
- dump** 標準ファイル・ダイアログが開き、過去に記録されたシーケンスやスタンダードMIDIファイルをテキストとして開くことができ、その内容が新しい名称未設定のテキスト・ウィンドウに表示される。実際には、これは**follow**オブジェクトに直接の影響を及ぼさず、シーケンスを見るか編集を行うだけである。変更を実行させるためにはファイルとして保存し、**read**メッセージによってそのファイルを**follow**に読み込ませなければならない。

アーギュメント

- 任意のシンボル オプション。このアーギュメントは過去に記録したシーケンス名であり、パッチを開いた時に自動的に**follow**内部に読み込まれる。

出力

整数 第1アウトレット：**follow**が追従している場合と、保存されたシーケンスにおいて次の数値（またはその後の2つの数値のうちの一つ）と一致する数値を受け取った場合、一致する数値のインデックスを出力する。**next**メッセージを受け取れば、次の数値のインデックスが同様に出力される。

第2アウトレット：**bang**または**start**メッセージを受け取った時に、記録された数値が再生されて出力される。追従している時や**next**メッセージを受け取った時は、記録されたシーケンスの次の数値が出力される。

利用例



記録されたノートに一致するノートによって処理が進行する、あるいはノートを進ませることができる

参照

seq
detonate

MIDIの録音と再生のためのシーケンサー
ノート・イベントのためのグラフィック・スコア

Tutorial 35

seq and follow (英文PDF参照)

入力

任意 **forward**オブジェクトに現在参照されている名前を持つすべての**receive**オブジェクトに任意のメッセージを送信する。

send **send**には、**receive**オブジェクトの名前が続き、**forward**オブジェクトに受け取られる任意のメッセージの行き先を設定する。このメッセージの行き先を変えられる機能が、**send**オブジェクトと**forward**オブジェクトの違いである。

アーギュメント

任意のシンボル オプション。メッセージを受け取る**receive**オブジェクトの名前を設定する。この名前は**send**メッセージによって後で変更できる。

出力

任意 アウトレットはない。**send**以外の**forward**オブジェクトに受け取られたメッセージは、**receive**オブジェクトが他のパッチにあったとしても、同じ名前を持つそれぞれの**receive**オブジェクトから出力される。

利用例



forwardオブジェクトを使い、複数のオブジェクトに一度にメッセージを送信する

二つの**send**オブジェクトを使うのと同じことである

messageボックスを用いて同じ処理を行うことができる

参照

message 任意のメッセージを送る
receive パッチ・コードなしでメッセージを受信する
route 選択的に入力を特定のアウトレットから出力する
send パッチ・コードなしでメッセージを送信する
value 他のオブジェクトと格納されている値を共有する



入力

- (マウス) ロックされていないパッチ内では、command + shiftキー (Macintosh) あるいはCtrl + shiftキー (Windows) を押しながら **fpic** オブジェクト上をドラッグすることで、画像のオフセット位置を変えることができる。画像の現在のオフセット位置はドラッグしたパッチ・ウィンドウのアシスタンス・エリアに表示される。
- autofit** **autofit**に0でない数値が続く場合、**fpic** オブジェクトの表示ウィンドウにフィットするように画像を拡大縮小する。
- link** **link**には、ファイル名を指定するシンボルが続き、その画像が既に他の**fpic** オブジェクトに読み込まれていないかをチェックする。他の**fpic** オブジェクトが既にメモリに読み込んでいれば、**fpic** オブジェクトはその以前に読み込まれた画像を参照することでメモリを節約する。
- matrix** **matrix**には、9つの実数が続き、現在読み込まれているファイルをマトリックス (行列) による変型を実行した後にメモリに読み直す。変型は、QuickTimeによる、1つの座標空間 (元の画像) からもう1つの座標空間 (元の画像の拡大縮小、回転、または変換された画像) へのマッピングと同じものが使用される。

マトリックスによる変型は9つの行列の要素から成り立っている。

$$\begin{array}{ccc} a & b & u \\ c & d & v \\ t_x & t_y & w \end{array}$$

u と v が0で、 w が1ならば、以下の変型の式に従う。

$$\begin{aligned} x' &= a * x + c * y + t_x; \\ y' &= b * x + d * y + t_y \end{aligned}$$

以下の式は拡大縮小/回転に用いられる。

$$\begin{aligned} a &= xscale * \cos(\theta) \\ b &= yscale * \sin(\theta) \\ c &= xscale * (-\sin(\theta)) \\ d &= yscale * \cos(\theta) \end{aligned}$$

マトリックス変型についての詳しい情報は、以下のAppleのQuickTimeデベロッパー・ドキュメントを参照のこと。

<http://developer.apple.com/techpubs/quicktime/qtdevdocs/INMAC/QT/iqMovieToolbox.c.htm#18006>



- noscale** **noscale**は画像の拡大縮小を無効にする。
- offset** **offset**には、2つの数値が続き、**fpic**オブジェクトのボックスの左上の隅からの、画像の左上の隅の水平、垂直のオフセット位置をピクセル単位で指定する。デフォルトでは画像の左上の隅のピクセルは、**fpic**オブジェクトの左上の隅（0,0のオフセット位置）に位置している。わずかに異なる連続した**offset**メッセージでは、画像を**fpic**オブジェクトの中で動かすことができ、また、それにより**fpic**オブジェクトは大きな画像の任意の部分を表示できる。（**offset**メッセージを使う時には、イメージをなめらかに動かすために、元のイメージは消去されない。画像が背景を完全に覆わない空白の背景を含むならば、これはあまり良くない表示になってしまう。）
- pict** **pict**にはMaxサーチ・パス内の画像ファイル名が続き、そのファイルを開き、画像を表示する。既に画像が表示されていた場合は置き換える。**fpic**オブジェクトはPICTファイルや、QuickTimeがインストールされていれば、QuickTimeでサポートされている他のファイル形式の画像も受け付ける。
- read** **read**には、ファイル名を指定するシンボルが続き、Maxのサーチ・パス内でQuickTimeの画像ファイルを検索し、存在していればそれを開き、**fpic**オブジェクトの画像表示領域内に表示する。ファイル名が任意のスペース、また特殊文字を含む場合は、名前はダブル・クォート（"）で囲うか、それぞれの特殊文字をバックスラッシュ（\）に続けなければならない。**read**というメッセージだけの場合は、標準ファイル・ダイアログを開き、QuickTimeでサポートされている画像ファイル形式のファイルを表示する。
- readany** **readany**にはファイル名を指定するシンボルが続き、**read**メッセージと同じ処理を行う。ただし、標準ファイル・ダイアログには、現在サポートされているファイル形式以外のものも表示する。
- rect** **rect**には、画像を表示する矩形の領域を指定する4つの数字が続き、画像ファイルをハードディスクからメモリへと読み直し、表示する。はじめの2つの数値は画像表示ウィンドウ内のオフセット位置を指定し、次の2つの数値は矩形の幅と高さをピクセル単位で指定する。
- scalemode** **scalemode**には0から3の範囲の数値が続き、**fpic**オブジェクトの拡大縮小のモードを設定する。

モードが0に設定されていれば、拡大縮小は適用されない。画像はメモリに読み込まれた状態のまま表示される。

また、モードが1に設定されていれば、拡大縮小はQuickTimeのマトリ



ックスによる変型を用いて実行される（詳しい情報は**matrix**メッセージを参照のこと）。画像は、現在の、またはデフォルトのマトリックス変形の設定に従って、拡大縮小、また回転される。**matrix**の設定値は**fpic**オブジェクトのインスペクターか、**matrix**メッセージを使うことで変更できる。

モードが2に設定されていれば、矩形の拡大縮小が適用される（詳しい情報は**rect**メッセージを参照のこと）。画像は現在の、またはデフォルトの**rect**メッセージの設定に従って読み込まれ、表示される。

モードが3に設定されていれば、画像のサイズを自動的に調整する。画像は現在表示されているウィンドウにフィットするように拡大縮小される。

storage **storage**には、高さ、幅をピクセル単位で指定する2つの数値が続き、画像の一部のみを読み込むことでメモリ・リソースを節約する。

注記：どちらかのアーギュメントが0ならば、**fpic**オブジェクトはその値に制限されない。

time **time**には、QuickTimeタイム・ユニットを指定する数値が続き、QuickTimeムービーから特定のフレームを読み込み、表示する。通常はQuickTimeムービーは1秒あたり、600ユニットのレートで表示される。デフォルト値は0（フレーム1）である。

インスペクター

fpicオブジェクトの振舞はインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**fpic**オブジェクトを選択することで、**fpic**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**fpic**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

fpicオブジェクトのインスペクターは以下の属性を設定できる。

*Picture Offset*は、**fpic**オブジェクトの表示ボックスの左上の隅からの、画像の左上の隅のピクセルの水平、垂直のオフセット位置を指定する。デフォルトでは画像の左上の隅のピクセルは、**fpic**オブジェクトの左上の隅（0,0のオフセット位置）に位置している。このオフセット位置は、ナンバー・ボックスに新しいピクセル位置の値を入れることで変更できる。デフォルトではオフセットされない（つまり、水平位置0、垂直位置0）。

*Time Offset Model*はQuickTimeのタイム・ユニットのフレームオフセッ



ト位置を指定でき、ムービーから特定のフレームを画像として読み込む。デフォルト値は0（すなわちフレーム1）である。

*Scaling Mode*ポップアップ・メニューは**fpic**オブジェクトで使用される拡大縮小の形式を選択する。それには、4つの拡大縮小のモードが用意されている。*None*（デフォルト）では画像の拡大縮小は適用されない。*Matrix*を選択すると、パッチ・ウィンドウが開き、画像を拡大縮小、回転させるためのマトリックスの値を指定できる。マトリックスの値を指定していなければ、デフォルト値が適用される。*Rectangular*を選択すると、同じようにパッチ・ウィンドウが開き、表示ウィンドウ内の相対的な矩形の位置、幅、高さをピクセル単位で指定できる（デフォルト値は0に設定されている）。*Auto-Fit*を選択すれば、表示範囲に適合するように画像は自動的に拡大縮小される。

*Internal Storage*はメモリに画像の一部のみを読み込むことで、メモリを節約できる。範囲は高さ、幅をピクセル単位で指定する。注記：どちらかの値が0ならば、**fpic**オブジェクトはその値に制限されない。

*Picture File*はOpenボタンを押すことで、**fpic**オブジェクトに読み込む画像ファイルを選択する。現在読み込まれているファイル名はボタンの左のテキスト・ボックスに表示される。そのボックスにファイル名を入力するか、Finderファイルのアイコンをボックスにドラッグすることでも選択することができる。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

アーギュメント

(Get Info...) パッチ・ウィンドウ内に**fpic**オブジェクトを置いた後で、それが選択されている時に、Objectメニューから**Get Info...**コマンドを選択する。この操作によって、**fpic**オブジェクトのインスペクター・ウィンドウを開き、**fpic**オブジェクトのボックス内に表示する画像ファイルを選択することができる。画像は100パーセントのサイズで表示され、その後、**fpic**オブジェクトのボックスは画像サイズに合うように、手動でサイズを変更することができる。**fpic**オブジェクトが画像サイズより小さければ、それに従って、画像の右下の部分は切り取られる。

fpicオブジェクトはパッチ・ウィンドウ内に単純に画像を表示する。同じ効果はEditメニューから**Paste Picture**を選択することでも可能だが、それはパッチ・ファイル内に画像ファイルを含むことになり、読



み込みや保存時にファイルの操作を遅くすることになる。**fpic**オブジェクトはハードディスク上の画像ファイルを参照するだけである。**fpic**オブジェクトを使う他の利点は、同じ画像ファイルが多数のパッチ・ウィンドウから参照される場合に、個々に保存されるよりもディスクスペースやメモリの使用を減らせることである。ただし、次にパッチを開いた時に外部の画像ファイルを自動的に表示させるためには、それをMaxのサーチ・パス内に置かれなければならない。

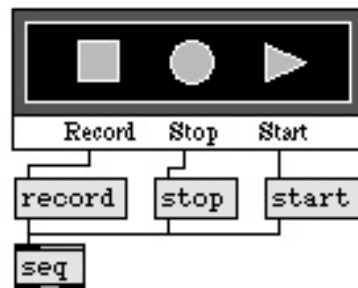
出力

なし。

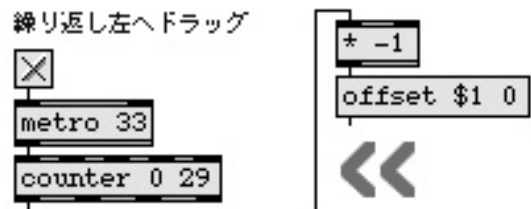
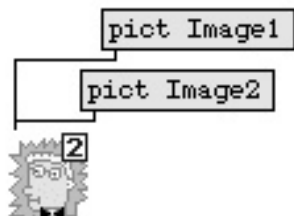
利用例



パッチ内に画像を置く
(純然たる美観として)



...または、**uabutton** オブジェクトを上に置くことで処理を行う



画像を切り替えることでスライドショーをする、また、そのオフセット位置を変化させることで画像を動かす

参照

imovie
lcd
matrixctrl
panel
pictctrl
pictslider
uabutton

パッチ・ウィンドウ内でQuickTimeムービーを再生する
パッチ・ウィンドウ内に画像を描画する
マトリックス状のスイッチ・コントロール
色のついた背景領域
ピクチャーを用いたコントロール
ピクチャーを用いたスライダー・コントロール
透明なボタン、**bang**を出力する

入力

- bang** 第1インレット：現在設定しているスクリーン座標、表示モード、表示色によって矩形を表示する。
- 整数** 第1インレット：矩形を描く左座標を設定し、形状を描く。
- 第2インレット：矩形を描く上座標を設定する。
- 第3インレット：矩形を描く右座標を設定する。
- 第4インレット：矩形を描く下座標を設定する。
- 第5インレット：矩形を描く表示モードを設定する（**oval**オブジェクトの表示モードの定数リストを参照のこと）。
- 第6インレット：グラフィック・ウィンドウの現在のパレットに従って、フレームの表示色のインデックス番号を設定する。モニターが白黒モードである時は、0以外のインデックス番号では黒であり、0は白である。
- frgb** 第1インレット：**frgb**の後に続く0から255までの3つの数値によって、次に描かれるフレームの表示色のRGB値を設定する。
- priority** 第1インレット：**priority**の後に続く0以上の数値によって、グラフィック・ウィンドウ内の**frame**オブジェクトを重ね合わせる優先順位を設定する。優先順位の低いオブジェクトは、優先順位の高いオブジェクトの後ろに描かれる。

アーギュメント

- 任意のシンボル** 必須。**frame**の最初のアーギュメントは、矩形が描かれるグラフィック・ウィンドウの名称でなければならない。**frame**オブジェクトを編集している時にグラフィック・ウィンドウがある必要はないが、名称が一致するウィンドウが開かれるまでは矩形は描かれない。
- 整数** **frame**の重ねる優先順位の初期値を設定する。優先順位が指定されていない場合の初期値は3である。

出力

(ビジュアル) **frame**オブジェクトと関連付けられたグラフィック・ウィンドウが表示されている場合、第1インレットに**bang**メッセージか数値を受け取れば、ウィンドウに矩形が描かれる。そして、先に描かれている矩形があれば、それは消去される。

利用例

ovalまたは**rect**の利用例を参照のこと。**frame**は**oval**、**rect**、**ring**の例に直接置き換えることが可能。

参照

graphic
lcd
oval
rect
ring

スプライトによるグラフィックを描くためのウィンドウ
パッチ・ウィンドウ内に画像を描画する
グラフィック・ウィンドウに塗り潰した楕円形を描く
グラフィック・ウィンドウに塗り潰した矩形を描く
グラフィック・ウィンドウに楕円形の枠を描く

入力

シンボル **fromsymbol** オブジェクトは入力にシンボルを受け付け、数値のリスト、またはシンボルの内容に対応するメッセージを出力する。**fromsymbol** オブジェクトは数字によって構成されたシンボル(例えば `'3.5 5 6.5 20'`) を解析することや、1つのシンボルを個々のメッセージに分割するのに役に立つ。

アーギュメント

なし。

出力

メッセージ、リスト、数値 元のシンボルの内容を解析した結果に従って、数値のリスト、またはメッセージを出力する。

利用例



参照

sprintf
tosymbol
zl

単語や数値のメッセージをフォーマットする
メッセージ、数値、リストを1つのシンボルに変換する
多目的なリスト処理

入力

- 実数** 第1インレット：数値は第2アウトレットから出力され、第2インレットの数値は第1アウトレットから出力される。
- 第2インレット：数値は第1アウトレットから出力するために格納される。そして、それは数値を第1インレットで受け取った時に出力する。
- 整数** アーギュメントに実数があれば、数値は実数に変換される。アーギュメントに整数がある場合、またはアーギュメントがない場合は、第2インレットで受け取った数値は整数として格納される。
- リスト** リストは**fswap**に格納される。最初の数値は第2アウトレットから出力され、2番目の数値は第1アウトレットから出力される。
- bang** 現在 **fswap**に格納されている数値を交換して出力する。

アーギュメント

- 整数または実数** オプション。第1アウトレットから出力する数値の初期値を設定する。アーギュメントがない場合は、初期値は0。整数のアーギュメントであるか、またはアーギュメントがない場合は、第1アウトレットから整数が出力される（第2アウトレットから出力される数値は常に実数である）。

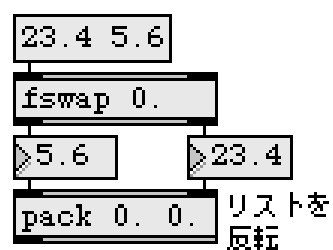
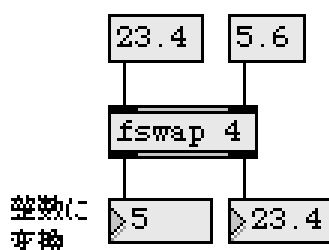
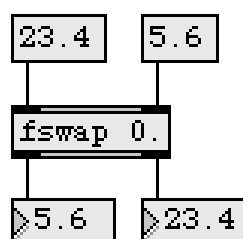
出力

- 整数** 第1インレットに数値を受け取った時、それぞれのインレットでの数値は反対側のアウトレットから出力される。整数のアーギュメントまたはアーギュメントがない場合は、第1アウトレットから整数が出力される。
- 実数** 第2アウトレットから出力される数値は常に実数である。実数のアーギュメントがある場合のみ、第1アウトレットから実数が出力される。

fswap

左右のインレットに受け取った
2つの実数を交換して出力する

利用例



受け取った数値は逆順に出力される

参照

pack
swap
unpack

数値やシンボルをひとつのリストに結合させる
2つの数値の一連の順番を逆にする
リストを個々のメッセージに分割する

入力

- リスト** 第1インレット：**funbuff**オブジェクトに格納される x と y のペアのうち、リストの1つ目の数値は x の値、2つ目の数値は y の値となる。 x の値が既に**funbuff**に格納された x の値と同じである場合、以前に格納されたペアは新しいペアによって上書きされる。
- 整数** 第1インレット：この値は x 、 y のペアの x の値として扱われる。 y の値が第2インレットに受け取られていた場合、その2つの値を対として**funbuff**オブジェクトに格納する。それ以外の場合には、**funbuff**オブジェクトに格納された対応する y の値を第1アウトレットから出力する。
- 格納されている x の値で、受け取った数値と一致するものがない場合は、**funbuff**オブジェクトは受け取った数値の、次のより小さい x の値を用い、対応する y の値を出力する。
- 第2インレット：この値は、次に第1インレットに受け取られた x の値とペアとなり、**funbuff**オブジェクトに格納される。
- clear** **funbuff**オブジェクトの内容を消去する。
- embed** **embed**には0でない値が続き、**funbuff**オブジェクト内のデータをパッチ内に保存する。デフォルトの設定では**funbuff**オブジェクトのデータはパッチ内には保存されない。
- goto** **goto**には1つの数値が続き、その値によって指定される x の値（インデックス）へのポインターを設定する。続く**next**メッセージでその x の値によって特定される y の値を返す。
- min** **funbuff**オブジェクト内に現在格納されている y の最小値を第1アウトレットから出力する。
- max** **funbuff**オブジェクト内に現在格納されている y の最大値を第1アウトレットから出力する。
- next** ポインターによって示される x の値（ポインターが格納されていない x の値を示していた場合、次のより大きな x の値）を検索し、第1アウトレットから対応する y の値を出力する。また、**funbuff**オブジェクトはその x の値と、以前にポインターによって指定された x の値との差を計算し、第2アウトレットから出力し、**goto**ポインターを次のより大きな x の値へ再設定する。
- write** 標準ファイル保存ダイアログ・ボックスを呼び出し、**funbuff**オブジェクトの内容を別ファイルへと保存することができる。**write**の後にシンボルが続く場合は、そのシンボルをファイル名として、**funbuff**オブジェクトの内容をファイルに保存する。

- read** 標準ファイル・ダイアログを呼び出し、 x 、 y の値を入れたファイルを **funbuff** オブジェクトに読み込むことができる。 **read** の後にシンボルが続く場合は、Maxはファイル・サーチ・パス内のファイルをその名前で検索し、直接**funbuff**オブジェクト内に読み込む。 **funbuff** オブジェクト用のファイルの形式は、この後に記述する。
- bang** 第1インレット：Maxウィンドウ内に、**funbuff** オブジェクトの内容についての情報、すなわち、いくつの要素を格納しているか、格納されている値のうち x と y の最大値、最小値、その領域と範囲（ x 、 y 軸それぞれの最大値と最小値）を出力する。
- 実数** どちらかのインレット：整数に変換される。
- set** 第1インレット：**set**には、1つ、また複数のスペースで区切られた数値のペアが続き、それぞれのペアを**funbuff**オブジェクトに格納する。
- delete** 第1インレット：**delete**には2つの数値が続き、**funbuff**オブジェクト内で一致する x 、 y のペアを探し、存在すればそれを削除する。**delete**に1つの数値だけが続き場合は、 x の値のみを調べ、それがあればそのペアを削除する。
- dump** 第1インレット：すべての格納されているペアを第1アウトレット、第2アウトレットから連続して出力する。この場合のみ、 y の値は第2アウトレットから、 x の値は第1アウトレットから出力されることになる。ペアは、 x の値を元に昇順で出力される。
- interp** 第1インレット：**interp**には1つの数値が続き、その値を x の値として用い、その前後の x の値によってその位置を補完し、その位置に対応する y の値を同じように前後の y の値から算出し、第1アウトレットから出力する。受け取られた数値が既に格納されている x 、 y のペアの x の値と同じであれば、それに対応した y の値を出力する。また、受け取られた数値が**funbuff**オブジェクトに格納された x の値の最大値または最小値を超える場合は、 x の最大値または最小値に対応する y の値を出力する。**funbuff**オブジェクトが空の場合は、0が出力される。

アーギュメント

任意のシンボル オプション。パッチを開いた時に**funbuff**オブジェクト内に読み込むファイルの名前を指定する。1つの**funbuff**オブジェクトの内容の変更は、他の同じ名前をもつ**funbuff**オブジェクトに影響することはない。

funbuffオブジェクトのためのファイルは、テキスト・エディター・ウィンドウを用いて作ることもできる。テキストは**funbuff**という言葉から始まり、交互に x と y の値を指定するスペースで区切られた数値のリストが続く。ファイルとして保存された**funbuff**オブジェクトはFileメニューの**Open as Text...**を選択することでテキストとして表示、編集ができる。テキスト内の数値は他のソース、例えば**capture**オブジェク

トや、他のワード・プロセッサなどのプログラムからもペーストすることができる。

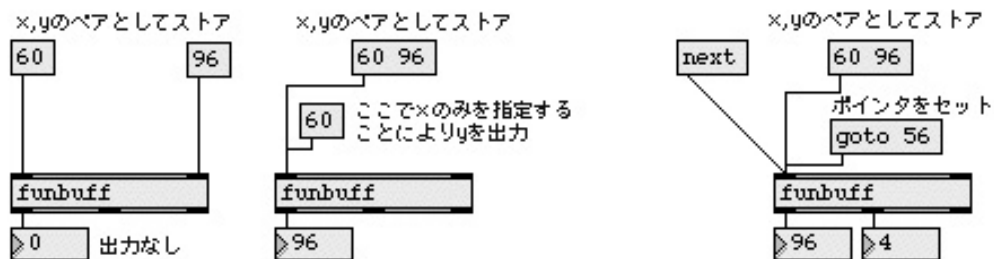
出力

整数 第1アウトレット：第1インレットから x の値を受け取った時に、対応した y の値を出力する（一致する x の値がなければ、次のより小さい x の値に対応した y の値を出力する）。**next**メッセージを第1インレットに受け取った時には、そのポインターに示される x の値に対応した y の値を出力する（一致する x の値がなければ、次のより大きな x の値に対応した y の値を出力する）。

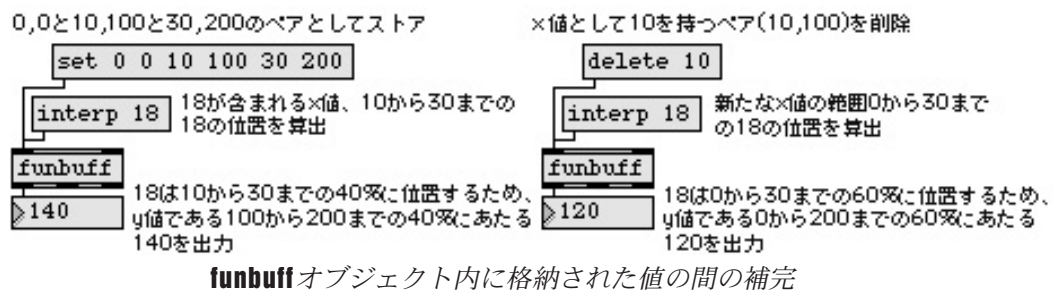
第2アウトレット：第1インレットに**next**メッセージを受け取った時、以前にポインターによって示されていた x の値と、現在示されている x の値との差を出力する。その後、ポインターは次の x の値に設定される。

bang 第3アウトレット：ポインターが**funbuff**オブジェクトの最後に達した時には、**next**メッセージによって値は出力されなくなり、最後に達したことを知らせる**bang**が出力される。

利用例



ペアやリストは x 、 y のペアとして扱われる。 x の値のみや、**next**メッセージは y の値を出力する



参照

coll
envi
line
table

様々な種類のメッセージを格納し、編集する
パッチ・ウィンドウに置く、スクリプトによって定義できるエンベロープ
ある値から異なる値へと、補完しながら数値を連続して出力する
数値の配列をグラフィカルに編集し、格納する

入力

- 整数** 任意のインレット：インレットの番号と入力された数値がリストとして出力される。
- 実数** 整数に変換される。
- リスト** 任意のインレット：リストを受け取ったインレットの番号がそのリストに加えられ、新しいリストとして出力される。リスト内の実数は整数に変換されることはない。リストは整数、実数とシンボルを含むことができる。(ただしリスト内の最初にシンボルを使うことはできない。)
- bang** 任意のインレット：**bang**メッセージを受け取ったインレットの番号と、そのインレット内に格納されている数値（もっとも最後に受け取られた数値）の、2つの項目からなるリストを出力する。

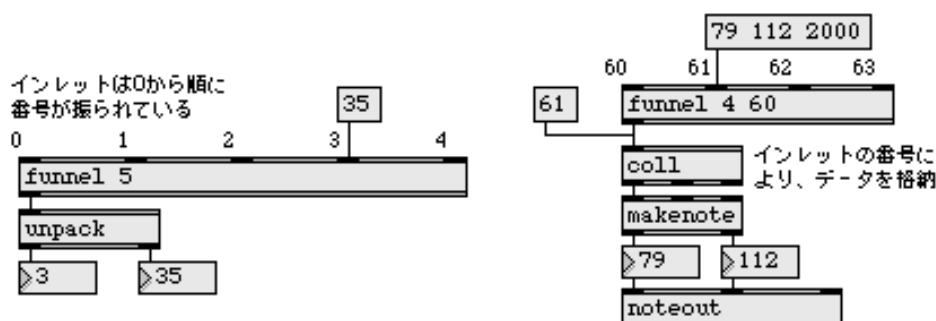
アーギュメント

- 整数** オプション。最初のアーギュメントは**funnel**オブジェクトのインレットの数を設定する。アーギュメントがなければ**funnel**オブジェクトのインレットは2つになる。2番目のアーギュメントは最初のインレットに対するオフセット番号を指定する。2番目のアーギュメントがない時は、インレットの番号は0から始まることになる。

出力

- リスト** 数値またはリストを任意のインレットに受け取れば、**funnel**オブジェクトはそのインレットに対応する番号を追加したリストを出力する。**funnel**オブジェクトは、**env**オブジェクトまたは**envi**オブジェクトへ入力される多くの数値を漏斗（funnel）のように扱えるよう設計されているが、**coll**オブジェクト、**funbuff**オブジェクトや**table**オブジェクトといったほかのオブジェクトとの連結でも有用なものになる。

利用例



funnel オブジェクトを使って入ってきた数値に番号を付加する、または数値を**coll** オブジェクトへ格納する

参照

env
envi
spray

スクリプト定義できるエンベロープ・エデッター
パッチ・ウィンドウに置く、スクリプトによって定義できるエンベロープ
整数を、アウトレットの番号に従って分配する

入力

整数 第1インレット：第2インレットに入力されるすべてのメッセージが出力されるアウトレットを指定する。この数値自信は出力を引き起こさない。

実数 第1インレット：整数に変換される。

bang 第1インレット：第1アウトレットから現在開いているアウトレットの番号を出力する。すべてのアウトレットが閉じていれば0を出力する。このメッセージは**grab**オブジェクトと連結して使うために設けられている。

任意のメッセージ 第2インレット：すべてのメッセージは、第1インレットに入力された数値によって開いたアウトレットへ出力される。

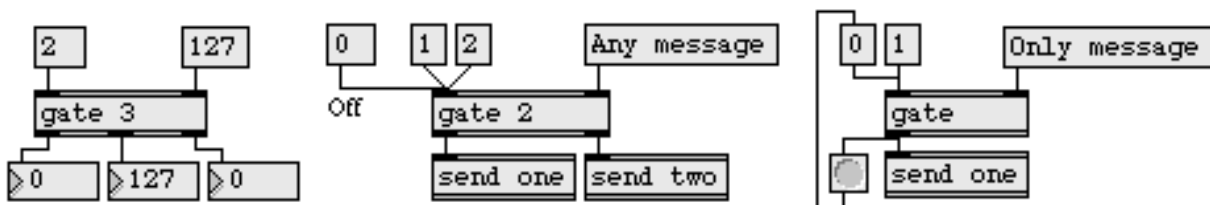
アーギュメント

整数 オプション。アウトレットの数を指定する。1から10に限定される。アーギュメントがない場合、アウトレットは1つだけとなる。

出力

任意のメッセージ 第2インレットに受け取られたメッセージは、第1インレットに入力された数値によって指定されたアウトレットへ出力される。0が第1インレットに入っているか、またはアウトレットの番号がまだ受け取られていない時、すべてのメッセージは出力されず無視される。0より小さい数値が第1インレットに入っている時、メッセージは最も左のアウトレットから出力される。存在するアウトレットの数より大きい数が第1インレットに入っている時、メッセージは最も右のアウトレットから出力される。

利用例



メッセージは指定されたアウトレットから出力される

入力された後、アウトレットを閉じる

参照

Ggate
Gswitch
onebang
route
send
switch

2つのうち1つのアウトレットから数値を出力する
 2つのうち1つのインレットから数値を受け取る
bangメッセージの流れをコントロールする
 選択的に入力を特定のアウトレットから出力する
 パッチ・コードなしでメッセージを送信する
 特定のインレットに入力されたメッセージを出力する

Tutorial 17

Gates and switches (英文PDF参照)

入力

様々なメッセージ **gestalt** オブジェクトは4文字以内のMac OSのゲシュタルト・セレクターで定められているシンボルを受け取る。例えば**sysv**はシステム・バージョン、**qtim**はQuickTimeのバージョンのことである。ゲシュタルト・セレクターの完全なリストはアップル・デベロッパー・ドキュメント (<http://developer.apple.com>) を参照のこと。Mac OS上では、**gestalt** オブジェクトはセレクターが結果を得るのにMacintoshのゲシュタルト機能を使う。Windows上ではこの機能はエミュレーションで動作するため若干結果が異なるが、それでも役立つ情報になるものと思われる。その結果やエラー・コードはアウトレットから出力される。

アーギュメント

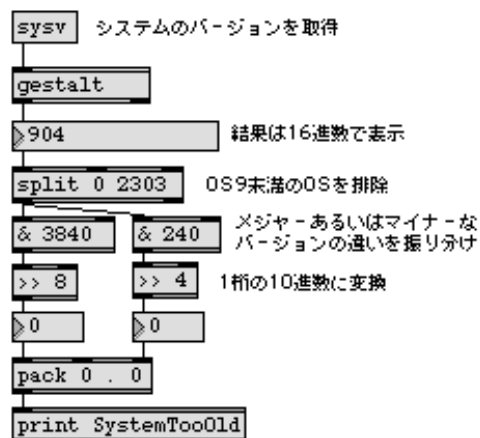
なし。

出力

整数 第1アウトレット：オブジェクトがセレクターで得た結果にエラーが無ければ第1アウトレットから結果が出力される。2進数または16進数での表示や、ビット・シフトと**&**オペレーターを使うことで結果を解釈しやすくなるだろう。

第2アウトレット：セレクターによる結果にエラーがある時、そのエラー・コードは第2アウトレットから出力される。エラー・コードの完全なリストはアップル・デベロッパー・ドキュメントを参照のこと。入力したセレクターが未定義の時、**-1**が出力される。エラーがない時は**0**が出力される。

利用例



gestalt オブジェクトは使っているシステムについての情報やハードウェアの特徴についての情報を知られてくれる

参照

screensize

モニターの大きさを出力する

Inside Macintosh



入力

整数 第1インレット：2つのアウトレットのうちどちらか1つのアウトレットを指定して開く。**0**は第1アウトレットを指定し、0以外の数値は第2アウトレットを指定する。**Ggate**オブジェクトの矢印は開いているアウトレットを指している。

bang 第1インレット：矢印の指す方向が別のアウトレットの方向に向けられる。マウスで**Ggate**オブジェクトをクリックしても同じ効果が得られる。

実数 第1インレット：整数に変換される。

任意のメッセージ 第2インレット：すべてのメッセージは開いているアウトレットに出力される。

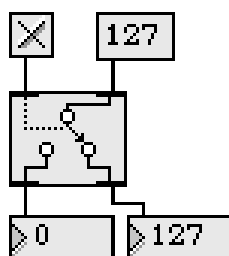
アーギュメント

なし。

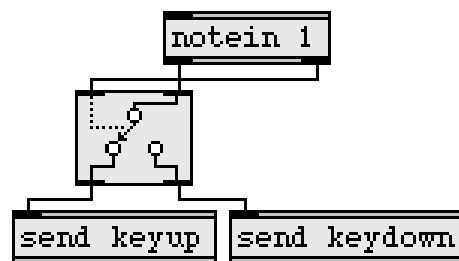
出力

任意のメッセージ 第2インレットに受け取られたメッセージは2つのうち1つのアウトレットから出力される。第1インレットに**0**が入っている時、入ってくるメッセージは第1アウトレットから出力される。0以外の数値が第1インレットに入っている時は、メッセージは第2アウトレットから出力される。

利用例



2つのアウトレットのうち1つを指定する



どのような比較も判断基準として用いることができる



参照

gate	入力を特定のアウトレットから出力する
Gswitch	2つのうち1つのインレットから数値を受け取る
onebang	bang メッセージの流れをコントロールする
pictctrl	ピクチャーを用いたコントロール
route	選択的に入力を特定のアウトレットから出力する
send	パッチ・コードなしでメッセージを送信する
switch	特定のインレットに入力されたメッセージを出力する

Tutorial 17

Gates and switches (英文PDF参照)

入力

- 任意** メッセージは一番右のアウトレットから出力される。あるいは、第2アーギュメントがある場合は、メッセージは第2アーギュメントで指定した名前の**receive**オブジェクトに送られ、離れたオブジェクトからのメッセージを取得することができる。
- set** **grab**に**receive**オブジェクトの名前を指定する第2アーギュメントが与えられている場合は、**set**に続くシンボルによって、**grab**がメッセージを取得する異なる**receive**オブジェクトの名前を指定する。

アーギュメント

- 整数** オプション。最初のアーギュメントは第1アウトレットの次に追加されるアウトレットの数を設定する。アーギュメントがない場合は、追加されるアウトレットは1個である。
- シンボル** オプション。シンボルが第2アーギュメントとして存在するならば、インレットで受け取ったメッセージは、一番右のアウトレットから出力される代わりに、シンボルによって名前が指定されたすべての**receive**オブジェクトに出力される。この場合、第2アーギュメントが存在しない場合に、通常入力されるメッセージを出力する一番右のアウトレットは、存在しない。

出力

- 任意** 一番右のアウトレットの出力：一番右のアウトレットは、他のオブジェクトの第1インレットに接続されていなければならない。インレットで受け取るメッセージは、一番右のアウトレットから接続しているすべてのオブジェクトの第1インレットに出力される。それらのオブジェクトのアウトレットから出力されるものは、すべて**grab**に取得されることになる。

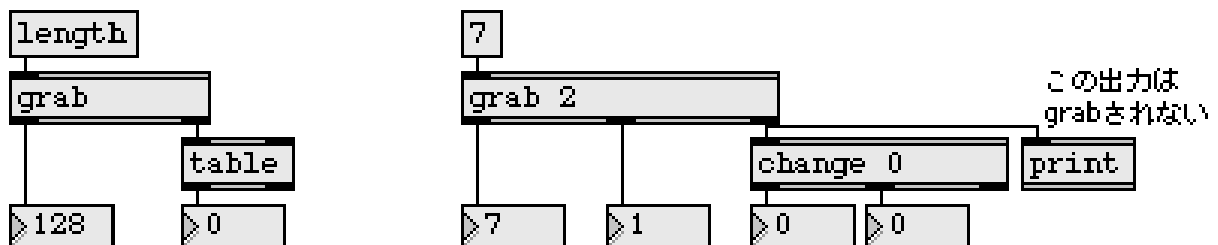
他のアウトレットの出力：**grab**からのメッセージに応じて、第2アウトレットに繋がれたオブジェクトのアウトレットから本来出力されるものが、その代わりに**grab**オブジェクトのアウトレットから出力される。また、他のオブジェクトの第1アウトレットから出力されるものが、**grab**の第1アウトレットから出力される。注記：**grab**は他のオブジェクトのアウトレットから送られる出力の処理結果を取得することができる。MIDIメッセージの送信やMaxウィンドウへの出力のような、他のタイプの出力については、**grab**はその出力を取得することができない。また、**grab**は、例えば、**seq**、**metro**、**clocker**のようなタイミング・オブジェクトの出力も取得することはできない。さらに、パッチ・オブジェクトのインレットへ**grab**の第2アウトレット

トを接続している時は、サブ・パッチの出力を取得することができない。それは単にサブ・パッチの中のインレット・オブジェクトの出力を取得するだけである。それはその入力と全く同じものである。ただし、**grab**は、**grab**の第2アーギュメントで指定した名前の**receive**オブジェクトを通して、離れたオブジェクトと通信することができる。

第2アーギュメントが存在するならば、インレットに受け取るメッセージは直接一番右のアウトレットから出力される代わりに、アーギュメントによって指定された**receive**オブジェクトに出力される。そのような**receive**オブジェクトは、他のオブジェクトの第1インレットだけに接続されていなければならない。第2アーギュメントがある場合は、他のオブジェクトの出力を取得するために、一番右のアウトレットは現れない。

注記：**grab**が多数の同じ名前の**receive**オブジェクトを経由して、他の離れたオブジェクトに繋がっている場合、**grab**が他のオブジェクトと通信する順序が明確でないので、それらの出力が**grab**オブジェクトの他のアウトレットから出力される順序は予測できない。

利用例



アウトレットから出力する前に「つかみとる」ことによって、オブジェクトの出力を取得する



grab は **set** メッセージで指定した **receive** オブジェクトを通して、通信することができる

参照

preset table

他のオブジェクトの設定を格納し、再現する
数値の配列をグラフィカルに編集し、格納する

入力

open **graphic**オブジェクトに関連付けられたグラフィック・ウィンドウが表示される。またウィンドウは一番手前に表示される。ロックされたパッチで**graphic**オブジェクトをダブルクリックしても同様である。

wclose **graphic**オブジェクトに関連付けられたウィンドウを非表示にする。

アーギュメント

シンボル オプション。**graphic**オブジェクトのウィンドウを特定する。図形やアニメーションがどのウィンドウに描かれるかをMaxに伝えるためにシンボルを使用する。アーギュメントがない場合は、ウィンドウ名は *Graphics-1* となる（以降は、グラフィック・ウィンドウは順番に連続する番号が付けられる）。

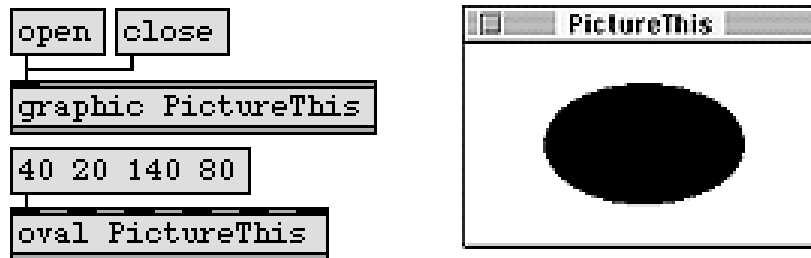
整数 オプション。**graphic**オブジェクトの名前の後に続く4つの座標によって、画面上でのウィンドウの表示領域を指定する。数値は順に表示領域の左端、上端、右端、下端を、画面上の座標で表す。座標のアーギュメントを指定せずに**graphic**オブジェクトを含むパッチを保存する場合、現在表示しているウィンドウの場所が保存されることに注意しよう。ドラッグしたか否かに関係なく、パッチが開かれるたびにオブジェクト・ウィンドウを特定の位置に表示させたい場合には、座標のアーギュメントが有用である。

オプション。**graphic**オブジェクトの名前の後に続き、4つの座標の前に0以外の数値があれば、グラフィック・ウィンドウのタイトル・バーが表示されない。タイトル・バーのないグラフィック・ウィンドウはその上でcommand+クリック (Macintosh) あるいはCtrl+クリック (Windows) でドラッグすることができる。

出力

なし。他のオブジェクトが**graphic**オブジェクトのウィンドウに図形を描く。

利用例



graphic オブジェクトはグラフィック・オブジェクトの出力のためにウィンドウを作成する
ウィンドウは右下隅をドラッグすることにより、ウィンドウ・サイズを変更することができる

参照

frame

グラフィック・ウィンドウに矩形の枠を描く

lcd

パッチ・ウィンドウ内に画像を描画する

oval

グラフィック・ウィンドウに塗り潰した楕円形を描く

pics

グラフィック・ウィンドウでのアニメーション

pict

グラフィック・ウィンドウでピクチャーを描く

rect

グラフィック・ウィンドウに塗り潰した矩形を描く

ring

グラフィック・ウィンドウに楕円形の枠を描く

Tutorial 42

Graphics (英文PDF参照)



入力

整数 第1インレット：2つのインレットのうち1つのインレットを指定して開く。0は第2インレットを、0以外の数値は第3インレットを指定する。**Gswitch**オブジェクトの矢印は開いているインレットを指している。

bang 矢印の指す方向が別のインレットの方向に向けられる。マウスで**Gswitch**オブジェクトをクリックしても同じ効果が得られる。

実数 第1インレット：整数に変換される。

任意のメッセージ 第2インレットまたは第3インレット：開いているインレットに受け取られたメッセージはアウトレットに出力される。一方、他のインレットに入ってきたメッセージは無視される。

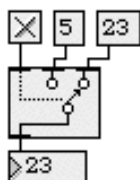
アーギュメント

なし。

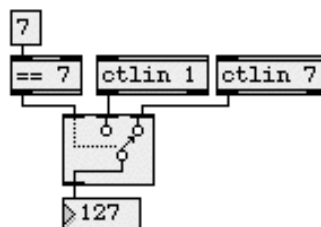
出力

任意のメッセージ 第1インレットに0が入っている時、第2インレットに受け取られたすべてのメッセージがアウトレットに出力される。この時、第3インレットに受け取られるメッセージは無視される。0以外の数値が第1インレットに入っている時、第2インレットに受け取られるメッセージは無視され、第3インレットに受け取られるすべてのメッセージはアウトレットに出力される。

利用例



2つのインレットのうち1つを指定する



どのような比較も判断基準として用いることができる

参照

gate	入力を特定のアウトレットから出力する
Ggate	2つのうち1つのアウトレットから数値を出力する
pictctrl	ピクチャーを用いたコントロール
receive	パッチ・コードなしでメッセージを受信する
route	選択的に入力を特定のアウトレットから出力する
switch	特定のインレットに入力されたメッセージを出力する

入力

(マウス) カーソルが**hint**オブジェクトの矩形範囲内に移動した時、指定された遅延の後、その矩形の下側に、テキスト・メッセージが色の付いた領域として表示される。

(フォント・メニュー) **hint**オブジェクトの外観は、Fontメニューから異なるフォントやサイズを選ぶことによって変えることができる。

delay **delay**に続く数値によって、ヒントが表示されるまでの遅延時間をミリ秒で設定する。デフォルト値は1000 (1秒) である。

interval **interval**に続く数値によって、マウスの位置がチェックされる間隔をミリ秒で設定する。デフォルト値は100である。

brgb **brgb**に続く0から255までの3つの数値によって、**hint**オブジェクトの背景色をRGB値で設定する。デフォルト値は白 (**brgb 255 255 255**) である。

frgb **frgb**に続く0から255の3つの数値によって、**hint**オブジェクトのテキスト色をRGB値で設定する。デフォルト値は黒 (**frgb 0 0 0**) である。

set **set**に続く任意のメッセージによって、**hint**に格納されているメッセージを入れ替える。**delay**メッセージで指定した時間間隔の後に、マウスが**hint**オブジェクト上にある時、このメッセージが表示される。

updatemode **updatemode**に続く0または1によって、**hint**オブジェクトのアップデート・モードを切り換える。アップデート・モードを有効にすることは (**updatemode 1**)、**hint**の下方にあるパッチ・ウィンドウのすべてを消去し、描き直すことを意味する。ヒントを表示している間に、その外観が変化することになる何かの領域 (例えば、ナンバー・ボックスやスライダ) 上で、ヒント・メッセージを表示するならば、このモードを使われなければならない。**updatemode 0**のメッセージは、最初に描かれた時にヒントの下にある画面に戻すことになる。デフォルト値は1である。

インスペクター

hintオブジェクトの振る舞いは、そのインスペクターによって編集することができる。Windowsメニューから**Show Floating Inspector**を選び、フローティング・インスペクターを有効にすれば、**hint**を選択した時に、フローティング・ウィンドウで**hint**のインスペクターを表示する。オブジェクトを選択し、Objectメニューの**Get Info...**を選ぶことによって、そのインスペクターを表示できる。

hintインスペクターでは以下の属性を設定できる：

マウスが**hint**オブジェクトによって指定した範囲に位置する時に表示させたいテキストを、*Set Hint Text*ボックスに入力することができる。

*Pop-up Delay*では、ヒントが現れるまでの遅延時間をミリ秒で設定することができる。デフォルト値は1000（1秒）である。

*Check Interval*では、マウスの位置をチェックする時間間隔をミリ秒単位で設定することができる。デフォルト値は100である。

*Redraw Behind Hint*チェック・ボックスがチェックされている時、**hint**の下方にあるパッチ・ウィンドウのすべてが、消去されて描き直される。ヒントを表示している間に、その外観が変化することになる何かの領域（例えば、ナンバー・ボックスやスライダー）上で、ヒント・メッセージを表示するならば、このモードを使われなければならない。デフォルトではチェックされている。

*Color*オプションでは、カラー・ピッカーかRGB値によってヒントのメッセージや背景色を変えることができる。*Message*はメッセージ表示の色を設定し（デフォルト値は0 0 0）、*Background*はヒントが現れるメッセージ領域の色を設定する（デフォルト値は221 221 221）。

*Revert*ボタンは、インスペクターを表示させてからオブジェクトの設定の変更を、すべて元に戻す。また、インスペクターを開いている時に、Editメニューから**Undo Inspector Changes**を選ぶことによって、インスペクター・ウィンドウを開く前のオブジェクトの状態に戻すこともできる。

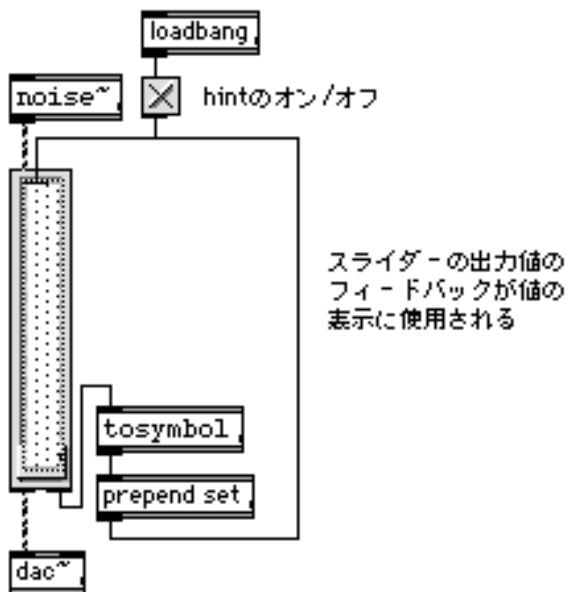
アーギュメント

なし。

出力

メッセージ **hint**オブジェクトに格納されているメッセージ。

利用例



UIオブジェクトに関する補助的な情報を表示する

参照

comment
umenu

注記またはラベル
コマンドを表示し、出力するポップアップ・メニュー

入力

整数 第1インレット：第1インレットに0から127の数値がそれぞれ何回受け取られたかの総数を**Histo**オブジェクトは蓄えてゆく。数値を受け取ると、**Histo**オブジェクトはそれをカウントに入れ、その数値を受け取った回数を第2アウトレットから出力する。そして、その数値そのものは第1アウトレットに出力する。0から127の範囲から外れた数値は無視される。

第2インレット：**Histo**オブジェクトがその数をカウントに入れないということ以外は第1インレットと同じ。

clear **Histo**オブジェクト内のメモリーを消去して、新しいヒストグラムとして始める。

bang 第1インレット：第1インレットが最後に受け取った数値を用いて**Histo**オブジェクトはその数値を何回受け取っているかを第2アウトレットから出力し、その数値そのものを第1アウトレットから出力する。何の数値も第1インレットが受け取ってない時は、2つのアウトレットどちらにも**0**を出力する。

アーギュメント

なし。

出力

整数 第1アウトレット：インレットに受け取った数値を出力する。

第2アウトレット：数値が受け取られた回数が出力される。

利用例



受け取った数値のヒストグラムを記憶する。**table** オブジェクト内に表示

参照

anal prob table

受け取った2つの数値をペアとして、ヒストグラムを作る
重みを付けてランダムな数値の流れを作り出す
数値の配列をグラフィカルに編集し、格納する

Tutorial 33

Probability tables (英文PDF参照)



入力

整数 インレットで受け取る数値は **hslider**によって視覚的に表示され、アウトレットから出力される。オプションとしてアウトレットの出力値を送る前に、数値を何倍かしたり、オフセットを加えることができる。

hsliderはマウスでドラッグするかクリックすることに反応して、数値を出力する。

実数 整数に変換される。

bang **hslider**に格納されている現在の数値を出力する。

color **color**に続く0から15のメッセージによって、または、Objectメニューの **Color**コマンドのオブジェクト色によって **hslider**の中心部の色を設定することができる。

local **local**に続く0以外のメッセージによって、(デフォルトの) マウス・クリックによるオブジェクトの反応を設定する。**local 0**のメッセージでは、マウスによるオブジェクトの反応を無効にする。すなわち、**hslider**はインレットからの入力値だけに反応し、すべてのマウス・クリックを無視する。

min **min**に続く数値のメッセージによって、アウトレットから出力する前の**hslider**の数値に加える値を設定する。デフォルト値は0である。

mult **mult**に続く数値のメッセージによって、乗算する値を指定する。アウトレットから出力される前にこの乗算する値によって**hslider**オブジェクトの入力値は掛け合わされる。乗算は、オフセット値を加える前の数値に対して行われる。デフォルト値は1である。

resolution **resolution**に続く数値のメッセージによって、サンプリング間隔をミリ秒で設定する。この数値は、表示を更新する時間間隔を決めるとともに、**hslider**オブジェクトのアウトレットから数値を出力する時間間隔も決めることになる。

set **set**に続く数値のメッセージによって、出力を行わずに、**hslider**による表示値を設定する。

size **size**に続く数値のメッセージによって、**hslider**オブジェクトの範囲を設定する。デフォルト値は128である。**size**を1に設定することは(1つの値を表示するだけなので)、視覚的な**hslider**の動作は無効になる。1よりも小さい数値を指定した場合は、**2**を設定したものと見なされる。



インスペクター

hsliderオブジェクトの振る舞いは、そのインスペクターによって編集することができる。Windowsメニューから**Show Floating Inspector**を選択しフローティング・インスペクターを可能にした場合に、**hslider**を選択するとフローティング・ウィンドウとして**hslider**のインスペクターを表示する。オブジェクトを選択し、Objectメニューの**Get Info...**を選ぶことによっても、そのインスペクターを表示することができる。

インレットで受け取る数値は、0からSlider Rangeの値よりも1小さい数値までに自動的に制限される。デフォルト値は128である。乗算の後において、一定の数値に加えるOffsetを指定することができる。デフォルトのオフセット値は0である。また、**hslider**のインスペクターでは数値を乗算するMultiplierも指定でき、**hslider**オブジェクトの値がアウトレットから出力される前に掛け合わされる。Multiplierの乗算はOffsetによる加算の前に行われる。Multiplierのデフォルト値は1である。

Revertボタンはインスペクターを開いてからのオブジェクトの設定値の変更をすべて元に戻す。また、インスペクターを開いている時に、Editメニューから**Undo Inspector Changes**を選ぶことによって、インスペクター・ウィンドウを開く前のオブジェクトの状態に戻すこともできる。

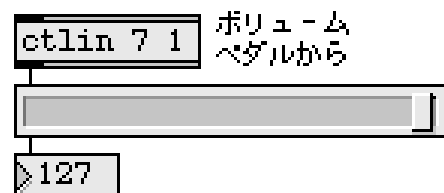
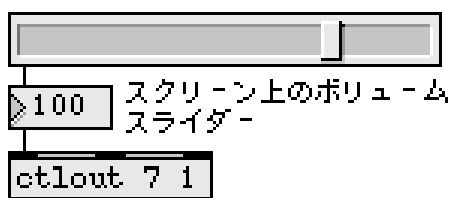
アーギュメント

なし。

出力

整数 インレットで受け取る数値、またはマウスで**hslider**上でのドラッグやクリックの操作によって、最初にMultiplierによって乗算し、Offsetによって加算を行った上で、アウトレットから出力する。

利用例



hslider上をドラッグすることで数値を出力する または数値を通過させながら、その数値を出力する



参照

kslider	画面上のキーボードから数値を出力する
pictctrl	ピクチャーを用いたコントロール
pictslider	ピクチャーを用いたスライダー・コントロール
rslider	数値の範囲を変更し、それを表示する
slider	画面上でスライダーを動かすことによって数値を出力する
uslider	画面上でスライダーを動かすことによって数値を出力する

Tutorial 14

Sliders and dials (英文PDF参照)

入力

- 整数** 各インレットからの数値は、それに関連付けられる**\$i**または**\$f**アーギュメントに置き換えられて格納される（例：左から2番目のインレットへの数値は**\$i2**または**\$f2**アーギュメントに置き換えられて格納される）。
- 実数** 各インレットからの数値は、それに関連付けられる**\$f**または**\$i**アーギュメントに置き換えられて格納される。**\$i**アーギュメントの場合、実数の小数点以下は切り捨てられる。
- symbol** 第1インレット：**symbol**に続くシンボルのメッセージによって、**\$s1**アーギュメントに置き換えられて格納される。
- bang** 第1インレット：現在格納されている値を使って条件文を実行する。
- 上記のいずれかのメッセージを第1インレットに受け取ると、条件文を実行し、結果を出力する。まだ受け取っていないインレットでは、デフォルト値として0が用いられる。
- インレットの数は、何個の変数アーギュメントが指定されているかで決定される。インレット数の最大は9個である。
- リスト** 第1インレット：異なるインレットで受け取ったかのようにそれぞれのリストの要素が扱われ、条件文が実行される。リストの要素が存在するインレット数より少ない場合は、残っているインレットではそれ以前に受け取った値が使用される。
- set** 第1インレット：**set**に続く1個以上の数値によって、それらの数値がそれぞれ異なるインレットに受け取られるかどうかを判断し、格納されている値を新しい値に置き換える。しかし、条件文は実行されず、何もアウトレットからは出力されない。インレット数より少ないメッセージであれば、残りのインレットでは置き換えは行わない。

アーギュメント

必須。**if**オブジェクトのためのアーギュメントは**expr**と同じ構文を使う条件文によって始まる。詳細は**expr**オブジェクトの記述を参照のこと。**then**に続く条件文は、以下に説明するメッセージ文が続く。メッセージ文の後には、オプションとして、2番目のメッセージ文を与える**else**がある。

ifは条件にあたる場所を実行し、結果が0以外であるならば、**then**以降のメッセージ文を実行する。結果が0であれば、**else**に続く2番目のメッセージ文が実行される（**else**や2番目のメッセージ文が指定されていない場合は除く）。

- then, else** メッセージ文は、以下の違いを含むが、メッセージ・ボックスへの入力と類似している。
- \$i1, \$f1, \$s1** 変数アーギュメントとして**\$1**の代わりに**\$i1, \$f1, \$s1**を使用する。
- send** コンマやセミコロンは使用してはいけない。**receive**オブジェクトと同じ名前の**send**に続くメッセージ文によって、離れた**receive**オブジェクトにメッセージを送信することができる。
- out2** メッセージ文での**out2**というキーワードによって、2番目のアウトレットを作成する。**out2**がメッセージ文に先行する場合は、このメッセージは第1アウトレットの代わりに第2アウトレットから出力される。

出力

- 任意** アーギュメントの**then**または**else**部分の結果がアウトレットから出力される。**out2**がアーギュメントとして存在するならば、2番目のアウトレットが存在し、**out2**以降のメッセージが第2アウトレットから出力される。**send**がアーギュメントとして存在するならば (**receive**オブジェクトと同じ名前が**send**に続くこと)、それに続くメッセージは**receive**オブジェクトに送信される。

利用例



複雑な比較やその結果は、1個のオブジェクトによって作成することができる

参照

- !=** 2つの数値の比較し、等しくなければ1を出力する
- <** 2つの数値を比較し、より小さいことを判断する
- <=** 2つの数値を比較し、より小さいか等しいことを判断する
- ==** 2つの数値を比較し、等しければ1を出力する
- >** 2つの数値を比較し、より大きいことを判断する
- >=** 2つの数値を比較し、より大きいか等しいことを判断する
- expr** 数式の値を求める
- select** 特定の入力を選択し、それ以外は通過させる



入力

(movieを参照) **movie**オブジェクトが認識するメッセージは**imovie**オブジェクトも同様に認識する。

border このオブジェクトはデフォルトで黒色の境界線がムービーの周りに描かれる。**border 0**メッセージはこの境界線を消去し、**border 1**メッセージは境界線を再描画する。

アーギュメント

(Get Info...) オプション。このオブジェクトを（パッチャーがロックされていない状態で）選択し、Objectメニューから**Get Info...** を選ぶと標準ファイル・ダイアログが開かれ、QuickTimeムービーを選択できる。この選択されたムービーはその後パッチが開かれる時に、自動的に読み込まれるようになる。**imovie**オブジェクトが自動的にムービーを見つけるためには、ムービーはMaxのファイル・サーチ・パス（Optionメニューの**File Preferences...**で設定できる）内に置かれていなければならない。

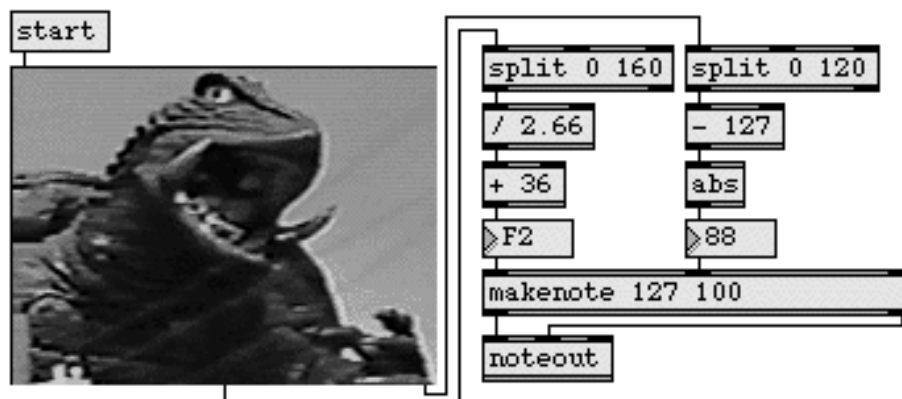
出力

整数 第1アウトレット：**length**メッセージを受け取った時、ムービーの最終時間位置を出力する。**time**メッセージでは、現在の時間を出力する。**start**メッセージでは、**0**を出力する。

第2アウトレット：ムービー内でマウスをクリックかドラッグした時、ムービーの左端からのマウスの水平位置が出力される。

第3アウトレット：ムービー内でマウスをクリックかドラッグした時、ムービーの上端からのマウスの垂直位置が出力される。

利用例



パッチ内にムービーを表示することができ、またムービー内のマウスの動きも知ることができる

**参照****lcd
movie
playbar**

パッチ・ウィンドウ内に画像を描画する
QuickTimeムービーをウィンドウ内で再生する
QuickTimeムービー再生のためのコントローラー



入力

整数 **IncDec**オブジェクトのインレットに入力される数値は、このオブジェクトの上半分または下半分をクリックして数値を増加または減少させるために用いる数量を設定する。この数値はアウトレットから出力はされない。**IncDec**オブジェクトはナンバー・ボックス、ダイアル、その他多くのスライダー類といったユーザー・インターフェースと共に使えるように設計されている。

(マウス) マウス・クリックによって格納されている数値を増加、または減少させ、アウトレットから出力する。(増加、減少はクリックする矢印の方向によって決まる)

(Fontメニュー) **IncDec**オブジェクトの大きさは、オブジェクトを選択しFontメニュー内の異なるフォントかサイズを選択することで変更することができる。

アーギュメント

なし。

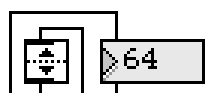
出力

整数 **IncDec**オブジェクトの上半分をクリックすると、**IncDec**オブジェクトは最後にインレットに受け取った数値かあるいは最後に出力した数値より1大きな数値を出力する。マウス・ボタンを押したままにしておくこと次第に速度をあげながら出力し続ける。

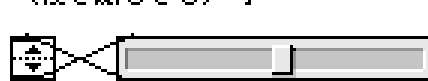
IncDecオブジェクトの下半分のボタンの動作も数値が減少するという以外は同様である。

利用例

ナンバー・ボックスでの標準的な使用法



パッチコードを×状につないでも(後で隠しても)可



IncDecオブジェクトはナンバー・ボックスや **hslider** との連携に適している

参照

counter
number box
hslider
umenu
uslider

bangメッセージをカウントし、回数を入力する数値を表示し、出力する
 画面上のスライダーを動かすことによって数値を出力する
 コマンドを表示し、出力するポップアップ・メニュー
 画面上でスライダーを動かすことによって数値を出力する



入力

(パッチャー) パッチャー内の各**inlet**オブジェクトは、そのパッチが他のパッチの中に（オブジェクトかサブ・パッチとして）使われる時に、オブジェクト・ボックスのインレットとして現れる。このようなインレットにメッセージが送られると、そのメッセージはサブ・パッチ内の**inlet**オブジェクトに受け取られることになる。

インスペクター

インスペクターを使用して**inlet**オブジェクトとして表示するアシスタンス・メッセージを指定し、編集することができる。Windowsメニューの**Show Floating Inspector**を選択しフローティング・インスペクターが使える状態にある時、任意の**inlet**オブジェクトのインスペクターをフローティング・ウインドウに表示することができる。**inlet**オブジェクトを選択してObjectメニューから**Get Info...**を選んでも、同様にインスペクターを表示できる。

*Describe Inlet*エリアにテキストを入力することで、アシスタンス・メッセージの内容を指定できる。

Revert ボタンを押すとインスペクターを開いてから変更したオブジェクトの設定すべてがもとに戻る。また、インスペクターが開いている時に（テキストを入力し、returnキーを押した後）、Editメニューの**Undo Inspector Changes**を選ぶことで、オブジェクトの設定をインスペクターを開く前の状態に戻すこともできる。

アーギュメント

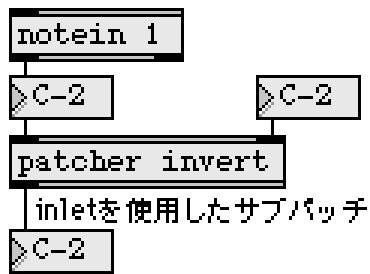
なし。

出力

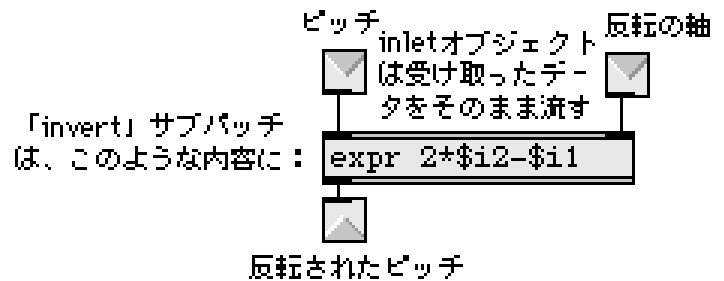
任意のメッセージ サブ・パッチ内の**inlet**オブジェクトは、そのサブ・パッチを含むパッチからパッチ・コードを通じて受け取った任意のメッセージを出力する。



利用例



サブ・パッチのインレットは...



サブ・パッチ内の **inlet** オブジェクトに対応する

参照

bpatcher
outlet
pcontrol
receive
send

ボックス内に埋め込まれた可視型のサブ・パッチ
パッチの外へメッセージを出力する
パッチ内のサブ・ウィンドウを開いたり、閉じたりする
パッチ・コードなしでメッセージを受信する
パッチ・コードなしでメッセージを送信する

Tutorial 26

The **patcher** object (英文PDF参照)

入力

- 整数** 第1インレット：現在格納している数値を入力された数値で置き換える、さらにその数値をアウトレットから出力する。
- 第2インレット：出力することなしに格納している数値を置き換える。
- 実数** 整数に変換される。
- bang** 第1インレット：格納している数値を出力する。
- set** 第1インレット：数値を後に伴った**set**メッセージは、出力することなしに格納している数値を置き換える。
- send** 第1インレット：**receive**オブジェクトの名前を伴った**send**メッセージは、名前的一致するすべての**receive**オブジェクトへ、**int**オブジェクト内に格納している数値を送信する。この時、**int**オブジェクトのアウトレットからは出力しない。

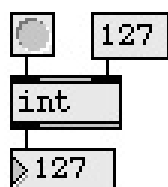
アーギュメント

- 整数** オプション。**int**オブジェクトへ格納する初期値を設定する。アーギュメントがなければデフォルト値は0となる。**int**という言葉を使わずにアーギュメントのみを入力しても、その数値をデフォルト値とする**int**オブジェクトを作成することになる。
- 実数** 整数に変換される。

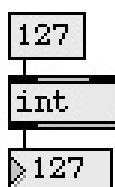
出力

- 整数** **int**オブジェクトに格納される（そして出力される）数値はlong型（32ビット）整数である。

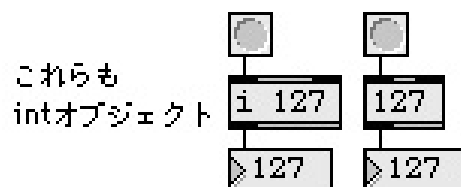
利用例



格納した数値を出力する



数値を格納しなおし、出力する



これらも
intオブジェクト

デフォルト値が与えられる

参照

float
pv
value

実数を格納する
パッチとそのサブ・パッチ内に限定して、変数を共有する
他のオブジェクトと格納されている値を共有する

Tutorial 21

Storing numbers (英文PDF参照)

入力

リスト リストの各数値が、順番に出力される。

整数または実数 そのままアウトレットに出力される。

bang 最後に受け取られた数値、またはリストを順番に出力する。

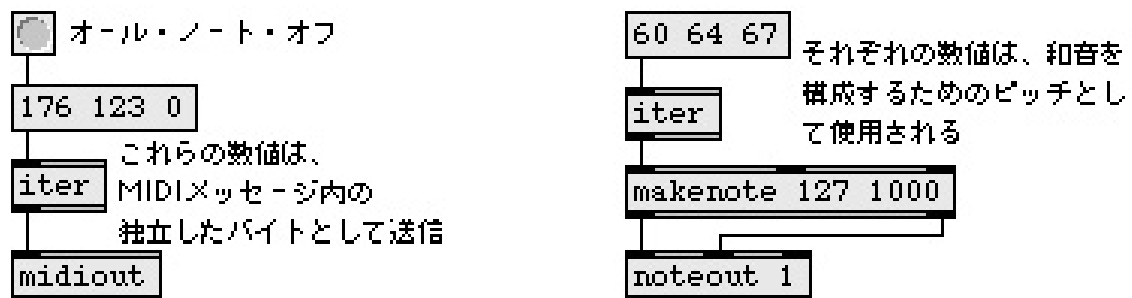
アーギュメント

なし。

出力

整数 インレットに受け取られた数値は個別に連続して出力される。

利用例



リスト内の数値は **iter** オブジェクトから個別に連続して出力される

参照

cycle 独立したアウトレットからデータの流れを出力する
thresh 短時間に受け取った数値をリストとしてまとめる
unpack リストを個々のメッセージに分割する
zl 多目的なリスト処理

Tutorial 30 Number groups (英文PDF参照)

入力

(keyboard) **key**への入力は、直接コンピューター・キーボードから入力される。インレットはない。

アーギュメント

なし。

出力

整数 コンピューター・キーボードのキーを押す度に、数値が出力される（キーを押さえ続けても、繰り返して出力されることはない）。

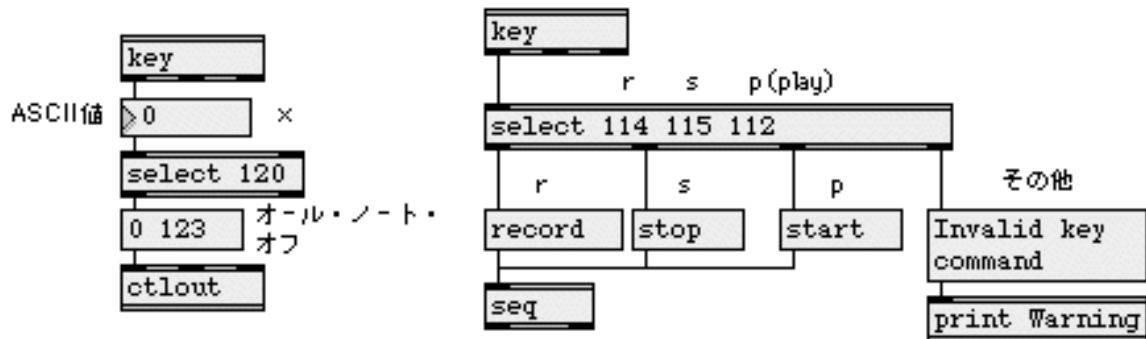
第1アウトレット：押されたキーのアスキー値を出力する。

第2アウトレット：押されたキーのキー・コードを出力する。

第3アウトレット：それぞれのモディファイア・キーは、この出力値と&オブジェクトによってトグル・スイッチを作ることができる。第3アウトレットから出力される数値は、以下のリストのようなアーギュメントを含む&オブジェクトによってトグル・スイッチを実現できる。

モディファイア・キー	出力値	トグル
キーイベント	128	& 128
(Windowsではマウスボタンが押されていない場合は0が、Macintoshでは常に0が出力される)		
Ctrl (Windows)	384	& 256
(システムが使用しているため情報は出力されない)		
Command (Macintosh)	384	& 256
(システムが使用しているため情報は出力されない)		
Shift	640	& 512
Caps Lock (on)	1152	& 1024
Alt (Windows)	2176	& 2048
(システムが使用しているため情報は出力されない)		
Option (Macintosh)	2176	& 2048
右クリック (Windows)	4224	& 4096
Control	4224	& 4096

利用例



コンピューター・キーボードで押されたキーの情報を、メッセージのトリガーとして使うことができる

参照

keyup	コンピューター・キーボードのキーを離した時の情報を入力する
numkey	コンピューター・キーボードで入力された数値を解釈する
spell	入力をASCIIコードに変換する
sprintf	単語や数値のメッセージをフォーマットする

Tutorial 20

Using the computer keyboard (英文PDF参照)

入力

(keyboard) **keyup**への入力は、直接コンピューター・キーボードから入力される。インレットはない。

アーギュメント

なし。

出力

整数 コンピューター・キーボードのキーを離す度に、数値が出力される（一番最初にキーが押され時は、何も出力されない）。

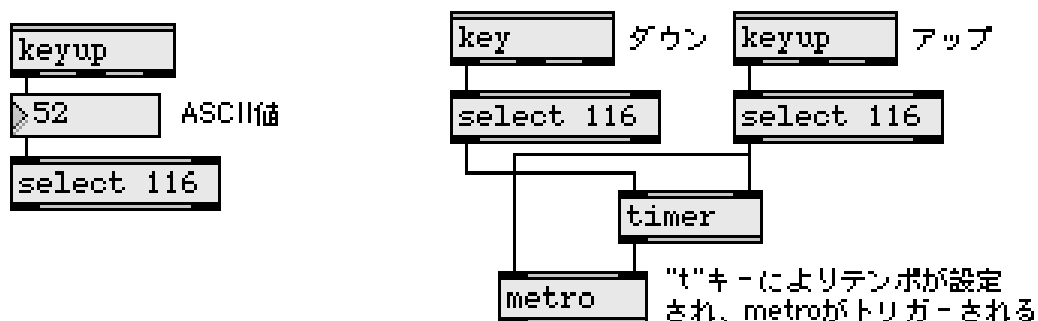
第1アウトレット：離されたキーのアスキー値を出力する。

第2アウトレット：離されたキーのキー・コードを出力する。

第3アウトレット：それぞれのモディファイア・キーは、この出力値と**&**オブジェクトによってトグル・スイッチを作ることができる。第3アウトレットから出力される数値は、以下のリストのようなアーギュメントを含む**&**オブジェクトによってトグル・スイッチを実現できる。

モディファイア・キー	出力値	トグル
キーイベント	128	& 128
(Windowsではマウスボタンが押されていない場合は0が、Macintoshでは常に0が出力される)		
Ctrl (Windows)	384	& 256
(システムが使用しているため情報は出力されない)		
Command (Macintosh)	384	& 256
(システムが使用しているため情報は出力されない)		
Shift	640	& 512
Caps Lock (on)	1152	& 1024
Alt (Windows)	2176	& 2048
(システムが使用しているため情報は出力されない)		
Option (Macintosh)	2176	& 2048
右クリック (Windows)	4224	& 4096
Control	4224	& 4096

利用例



キーを離した時に、アスキー値を出力する キーが押されていた時間を知るためにkeyを使用している

参照

key	コンピューター・キーボードのキー入力を出力する
MouseState	マウスの状態や位置を出力する
numkey	コンピューター・キーボードで入力された数値を解釈する
spell	入力をASCIIコードに変換する
sprintf	単語や数値のメッセージをフォーマットする

Tutorial 20

Using the computer keyboard (英文PDF参照)



入力

整数 第1インレット：**kslider**の表示範囲内であるならば、インレットで受け取られた数値は**kslider**上にハイライトで表示される。**kslider**が保持しているベロシティ値（1から127）を第2アウトレットから出力した後、受け取った数値を第1アウトレットから出力する。

第2インレット：第2インレットで受け取った数値は、出力は行わずに、ベロシティ値の出力値を設定する。

(マウス) マウスでクリックまたはドラッグした時に、**kslider**は数値を出力する。ベロシティ値は各キーの垂直上のマウスの位置によって設定される。垂直方向で上の部分ほど、ベロシティ値は大きくなる。最大のベロシティ値は127である。

ksliderがポリフォニック・モードであるならば、1回目はノート・オン、2回目はノート・オフとなるので、キーを2回クリックする必要がある。

ksliderの最も右端をクリックすると、見えているキーの中でもっとも音が高いC音のノート・ナンバーを出力する。

実数 整数に変換される。

bang 第1インレット：現在格納されているベロシティ値とノート・ナンバーを出力する。

clear **clear**メッセージはキーボード上で現在ハイライトされているノートを元に戻す。この時、出力は起こらない。

color **color**に続く0から15の数値のメッセージによって、またはObjectメニューの**Color**のサブメニューで利用できる色の1つによって、キーボード上でハイライトされる色を設定することができる。

frgb 第1インレット：**frgb**に続く3つの数値（0から255）のメッセージによって、キーボード上でハイライトされる色をRGB値で設定することができる（デフォルト値は128 128 128）。

mode 第1インレット：**mode**に続く0か1のメッセージによって、**kslider**がモノフォニックかポリフォニックかを設定する。デフォルトでは**mode 0**でモノフォニック・モードである。モノフォニックでは表示されているキーを1回だけ選択することができる。**mode 1**ではポリフォニック・モードになる。ポリフォニック・モードでは、MIDIキーボードを



押さえ続けているかのように、**kslider**はノート・オフされるまでノート・オンの状態を続ける。キーはベロシティ0のノート・オン・メッセージを**kslider**オブジェクトに送ることによって解除する。

- offset** 第1インレット：**offset**に続く数値のメッセージによって、**kslider**オブジェクトのオクターブのオフセット値を設定する。**kslider**のキーボードはMIDIキーボードのレンジの一番低いオクターブ（c-2）からのノートを出力する。**offset 5**メッセージは**kslider**オブジェクトの一番左のキーがC3であることを表す。デフォルト値は3である。
- range** 第1インレット：**range**に続く数値のメッセージによって、**kslider**オブジェクトのオクターブ幅を設定する。デフォルト値は4である。
- set** 第1インレット：**set**に続く数値のメッセージによって、出力することなく、**kslider**で表示される値を変更する。
- size** 第1インレット：**size**に続く0か1のメッセージによってキーボード表示のサイズを設定する。デフォルトである**size 0**は大きいサイズのキーボードを表示し、**size 1**では小さいキーボードを表示する。

インスペクター

ksliderオブジェクトの振る舞いは、そのインスペクターによって編集することができる。Windowsメニューから**Show Floating Inspector**を選択しフローティング・インスペクターを可能にすれば、**kslider**を選択するとフローティング・ウィンドウで**kslider**のインスペクターを表示する。オブジェクトを選択し、Objectメニューの**Get Info...**を選ぶことによって、そのインスペクターを表示することができる。

ksliderインスペクターでは、**kslider**オブジェクトのオクターブ幅である *Slider Range* の値を入力できる（デフォルト値は4）。*Offset* の値（デフォルト値は3）は、最も低いCであるc-2から数えて、表示されているキーボードの最も低いノートのオクターブの値を指定する。*Keyboard Size* ボタンはキーボードのサイズを選択し、*Keyboard Mode* ボタンではモノフォニックかポリフォニックかを選択する。デフォルトの色は128 128 128である。

Revert ボタンはインスペクターを開いてからのオブジェクトの設定値の変更をすべて元に戻す。また、インスペクターを開いている時に、Editメニューから**Undo Inspector Changes**を選ぶことによって、インスペクターウィンドウを開く前のオブジェクトの状態に戻すこともできる。



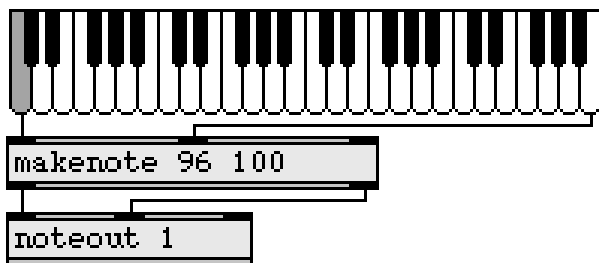
アーギュメント

なし。

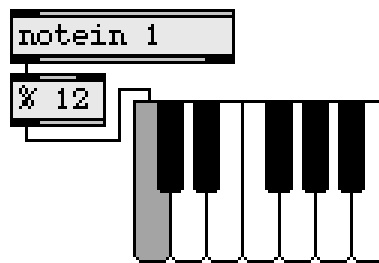
出力

整数 インレットに数値を受け取るか、オブジェクトをクリックまたはドラッグする時、**kslider**は現在のベロシティ値を第2アウトレットから出力した後、（表示されている）ノート・ナンバーの値を第1アウトレットから出力する。

利用例



キーボードをクリックすることによって出力を行う



入力されたノート・ナンバーを表示する

参照

hslider	画面上のスライダーを動かすことによって数値を出力する
makenote	ノート・オンに続けてノート・オフ・メッセージを生成する
notein	MIDIノート・メッセージを出力する
noteout	MIDIノート・メッセージを送信する
pictslider	ピクチャーを用いたスライダー・コントロール
rslider	数値の範囲を変更し、それを表示する
slider	画面上でスライダーを動かすことによって数値を出力する
uslider	画面上でスライダーを動かすことによって数値を出力する

Tutorial 14

Slider and dials (英文PDF参照)



Max4.0以降では、**lcd**オブジェクトのすべての描画コマンドは小文字になっている。互換性を持たせるために、過去の大文字から始まる古い形式のメッセージも依然として使用できる。**lineto**でも**LineTo**でも使用することができる。

入力

- (マウス) (local 0メッセージによってこの機能が無効になっていない場合) マウスを用いて**lcd**オブジェクト内に自由に描画できる。マウスは現在設定されているペンと色によって描画され、マウスの位置情報はアウトレットから出力される。
- ascii** **ascii**には0から255までの数値が続き、現在のペンの位置にASCII値に対応する文字が描かれ、ペンの位置はその文字の右側に移動される。0から255の範囲を超えた数値は剰余演算により、その範囲に制限される。
- border** **border 1**メッセージは**lcd**オブジェクトの表示領域の周囲に枠線を引くように設定する。デフォルト値は1である。**border 0**メッセージはその機能をオフにする。
- brgb** **brgb**には0から255までの3つの数値が続き、**lcd**オブジェクトの背景色をRGB値で指定する。
- clear** **lcd**オブジェクトの内容を消去する。
- clearpicts** **lcd**オブジェクトのすべての名前の付けられたピクチャーを削除する。
- clearregions** **lcd**オブジェクトのすべての名前の付けられたリージョンを削除する。
- clearsprites** **lcd**オブジェクトのすべての名前の付けられたスプライトを削除する。
- clipoval** **clipoval**には矩形の左、上、右、下の隅を指定する4つの整数が続き、その矩形に内接する楕円形領域内で、**lcd**オブジェクトの描画を行うようにクリッピングする。これらの整数は、**lcd**オブジェクトの表示領域の左上の隅からの相対位置としてピクセル単位で指定する。
- clippoly** **clippoly**には254個までの整数のアーギュメントが続き、それらの整数をx/yのペアとして設定される多角形領域内で、**lcd**オブジェクトの描画を行うようにクリッピングする。それぞれのx/yのペアは、**lcd**オブジェクトの表示領域の左上の隅からの相対位置としてピクセル単位で指定する。
- cliprect** **cliprect**には矩形の左、上、右、下の位置を指定する4つの整数のアーギュメントが続き、その矩形領域内で、**lcd**オブジェクトの描画を行う



- ようにクリッピングする。それぞれの整数は、**lcd**オブジェクトの表示領域の左上の隅からの相対位置としてピクセル単位で指定する。
- cliprgn** **cliprgn**には1つのシンボルが続き、その名前のリージョンで表される領域内で、**lcd**オブジェクトの描画を行うようにクリッピングする。
- cliproundrect** **cliproundrect**には矩形の左、上、右、下の位置と、角を水平および垂直に丸める量をピクセル単位で指定する6つの整数が続き、その丸められた矩形領域内で、**lcd**オブジェクトの描画を行うようにクリッピングする。矩形の位置を表す数値は、**lcd**オブジェクトの表示領域の左上の隅からの相対位置としてピクセル単位で指定する。
- closeregion** **closeregion**にはそのリージョンに名前をつけるためのシンボルのアーギュメントが続き、リージョンの定義を終了させ、定義されたリージョンをそのシンボルに関連づける。**closeregion**メッセージの後には、再び、描画コマンドが機能するようになる。
- closesprite** **closesprite**にはそのスプライトに名前をつけるためのシンボルのアーギュメントが続き、スプライト・コマンドの記録を終了させ、定義されたスプライトをそのシンボルに関連づける。**closesprite**メッセージの後には、再び、描画コマンドが機能するようになる。
- color** **color**には0から255までの数値が続き、続く操作で**lcd**オブジェクトに描画される色をMaxのカラー・パレットから指定する。0から255の範囲を超えた値は剰余演算により、その範囲に制限される。
- deletepict** **deletepict**には1つのシンボルが続き、その名前が付けられているピクチャーを削除する。
- deleteregion** **deleteregion**には1つのシンボルが続き、その名前が付けられているリージョンを削除する。
- deletesprite** **deletesprite**には1つのシンボルが続き、その名前が付けられているスプライトを削除する。
- drawpict** **drawpict**には1つのシンボルが続き、その名前が付けられているピクチャーを描画する。オプションとして、画像を拡大縮小して描画するデスティネーション領域の矩形や、描画に用いる領域を指定するソース領域の矩形を設定することができる。このためには、それぞれ4つの数値を続け、それらの矩形は、左、上、幅、高さの値をピクセル単位で指定する。デスティネーション領域の矩形は**lcd**オブジェクトの表示領域の左上の隅からの相対位置、ソース領域の矩形はピクチャーの左上の隅からの相対位置で指定する。指定されなければ、それらの矩形は両方ともピクチャーと同じサイズに設定される。



drawsprite	drawsprite には1つのシンボルが続き、その名前が付けられているスプライトを描画する。オプションとして、スプライトを描画する時の水平、垂直オフセット位置を指定する数値のペアを続けることができる。
enablesprites	enablesprites 1 メッセージはスプライトの描画を有効にする。 enablesprites 0 メッセージはこの機能をオフにする（デフォルト）。スプライトが有効になっている時には、 lcd オブジェクトはよりメモリを消費する。
font	font には2つの数値が続き、 write メッセージか、 ascii メッセージを受け取った時に描画する文字のフォントIDとフォント・サイズを指定する。大部分のフォントID番号は、どのようなフォントがシステムフォルダのフォント・フォルダに存在しているかに依存するため、fontメッセージの結果はコンピューターによって異なる場合があるので注意すること。
framearc	弧の輪郭のみが描画されること以外は、 paintarc メッセージと同じである。
frameoval	楕円の輪郭のみが描画されること以外は、 paintoval メッセージと同じである。
framepoly	多角形の輪郭のみが描画されること以外は、 paintpoly メッセージと同じである。
framerect	矩形の輪郭のみが描画されること以外は、 paintrect メッセージと同じである。
framergn	リージョンの輪郭のみが描画されること以外は、 paintrgn メッセージと同じである。
framroundrect	丸められた矩形の輪郭のみが描画されること以外は、 paintroundrect メッセージと同じである。
frgb	frgb には0から255までの3つの数値が続き、 lcd オブジェクトの描画色をRGB値で指定する。
getpenloc	getpenloc メッセージは、 penloc とそれに続く2つの数値を伴うメッセージを、 lcd オブジェクトの第4アウトレットから出力する。数値は lcd オブジェクトの表示領域の左上の隅からの相対的な座標を表す。1つ目の数値はその隅からの右に何ピクセルか、2つ目の数値はその隅から下に何ピクセルかを表している。
getpixel	getpixel には、ピクセルの位置を、 lcd オブジェクトの表示領域の左上の隅からの相対的な座標で、指定する2つの数値が続き。このメッセージを受け取れば、 pixel と0から255までの3つの数値を伴うメッセー



ジを、**lcd**オブジェクトの第4アウトレットから出力する。その数値は指定された位置のピクセルのRGB値を表す。指定されたピクセルが**lcd**オブジェクトの範囲外であれば、**getpixel**メッセージは**pixel 0 0 0**というメッセージを出力する。

- hidesprite** **lcd**オブジェクト内の名前が付けられたスプライトの描画をオフにする。
- idle** **idle 1**メッセージは、**lcd**オブジェクト上のマウスの位置を常に出力するように設定する。マウス位置の座標はマウスが動く度に、第2アウトレットから2つの項目を持つリストとして出力される。その数値は、**lcd**オブジェクトの表示領域の左上の隅からの相対座標を表す。1つ目の数値はその隅から右にいくつかを、2つ目の数値はその隅から下にいくつかをピクセル単位で表す。**idle 0**メッセージでその機能はオフになる。デフォルト値は0である。
- line** **line**には、現在のマウスの位置からの相対的な水平と垂直のオフセット位置をピクセル単位で示す2つの整数のアーギュメントが続き、現在のペンの位置から、そのオフセット位置で示される座標まで直線を描く。そして、ペンの位置をその位置に移動させる。正の値のアーギュメントは直線を右か下へ、負の値のアーギュメントは上か左へ描画する。
- linesegment** **linesegment**には、線分の両端の位置を指定する四つの整数のアーギュメントが続き、直線を描画する。その数値は開始点及び終了点の水平と垂直のオフセット位置を示し、**lcd**オブジェクトの表示領域の左上の隅からの相対位置で表す。オプションとして、色も続けて指定することができる。アーギュメントに1つの整数を加えれば、**color**メッセージと同じように、Maxのカラーパレットから色を指定できる。アーギュメントに3つの整数を加えれば、**frgb**メッセージと同じように、RGB値で色を指定できる。
- lineto** **lineto**には、水平、垂直それぞれの終端を指定する2つの整数が続き、現在のペンの位置から、そのアーギュメントによって指定された位置まで線を描画する。
- local** **local 0**メッセージは**lcd**オブジェクト上でのマウスによる描画をオフにする。**local 1**メッセージで再び、この機能はオンにできる。どちらの場合でも、**lcd**オブジェクトはその表示領域上をドラッグするマウスの位置を出力する。
- move** ペンの位置を、現在の位置から右、または下にピクセル単位で移動する。**move**には、**lcd**オブジェクトの表示領域の左上の隅からの相対的な水平と垂直のオフセット位置をピクセル単位で指定する2つの整数のアーギュメントが続く。負の値のアーギュメントは左、または上にペンの位置を移動させる。



- moveto** 次の描画が行われるペンの位置を設定する。**moveto**メッセージは**lcd**オブジェクトの表示領域の左上の隅からの相対的な水平、垂直オフセット位置をピクセル単位で指定する2つの整数のアーギュメントを指定する。
- noclip** 指定されているクリッピング領域をすべて解除する。
- onscreen** **onscreen 1**メッセージはオンスクリーン・ウィンドウを描画に使うことにより、メモリの節約をする機能をオンにする。**onscreen 0**メッセージはこの機能をオフにする。オンスクリーン・モードを使わない場合、**lcd**オブジェクトはより多くメモリを消費する。しかし、**lcd**オブジェクトが隠された場合でも、内容は記憶されており、消去されない。
- paintarc** **paintarc**には、その弧が描かれることになる楕円の内接する左、上、右、下の矩形の隅、そして弧のはじめと終わりを角度で指定する6つの整数のアーギュメントが続き、その弧を描画する。矩形の隅は**lcd**オブジェクトの表示領域の左上の隅からの相対位置としてピクセル単位で指定する。任意で色も続けて指定することができる。アーギュメントに1つ整数を加えれば、**color**メッセージと同じように、Maxのカラーパレットから色を指定できる。アーギュメントに3つの整数を加えれば、**frgb**メッセージと同じように、RGB値で色を指定できる。
- paintpoly** **paintpoly**には254個までの整数のアーギュメントが続き、**lcd**オブジェクト上に描画される多角形を定義する x/y のペアを指定する。それぞれの x/y のペアは**lcd**オブジェクトの表示領域の左上の隅からの相対位置をピクセル単位で指定する。オプションとして、最初のペアと同じ位置を示す x/y のペアを指定した後に続けて、色も指定できる。アーギュメントに1つ整数を加えれば、**color**メッセージと同じように、Maxのカラーパレットから色を指定できる。アーギュメントに3つの整数を加えれば、**frgb**メッセージと同じように、RGB値で色を指定できる。
- paintrect** **paintrect**には矩形の左、上、右、下の位置を指定する4つの整数のアーギュメントが続き、矩形を描画する。それぞれの隅は**lcd**オブジェクトの表示領域の左上の隅からの相対位置をピクセル単位で指定する。オプションとして、色も続けて指定することができる。アーギュメントに1つの整数を加えれば、**color**メッセージと同じように、Maxのカラーパレットから色を指定できる。アーギュメントに3つの整数を加えれば、**frgb**メッセージと同じように、RGB値で色を指定できる。
- paintrgn** **paintrgn**には1つのシンボルが続き、その名前のついたリージョンを塗りつぶして描画する。オプションとして、整数のペアのアーギュメントを続けることで、リージョンを描画する水平、垂直オフセット位置を指定できる。さらに、アーギュメントに1つの整数を追加すれば、**color**メッセージと同じようにMaxのカラーパレットから、描画色を指定できる。アーギュメントに3つの整数を追加すれば、**frgb**メッセージと同じように、RGB値で描画色を指定できる。



- paintroundrect** **paintroundrect**には矩形の左、上、右、下の位置と、角を水平および垂直に丸める量をピクセル単位で指定する6つの整数が続き、角を丸めた矩形を描画する。それぞれの隅は**lcd**オブジェクトの表示領域の左上の隅からの相対的な位置をピクセル単位で指定する。アーギュメントに1つの整数を追加すれば、**color**メッセージと同じようにMaxのカラーパレットから、描画色を指定できる。アーギュメントに3つの整数を追加すれば、**frgb**メッセージと同じように、RGB値で描画色を指定できる。
- penmode** **penmode**には0から7までのうち、1つの整数が続き、続く描画コマンドによる描画の転送モードを設定する。それぞれの描画モードについてはInside MacintoshのQuickDrawに関するドキュメンテーションを参照のこと。
- pensize** **pensize**には、必ず1つの整数のアーギュメントが続き、描画に用いるペンのサイズをピクセル単位で指定する。
- readpict** **readpict**には、その後の描画に必要なピクチャーの名前とファイル名のアーギュメントが続き、その画像ファイルをハードディスクからメモリに読み込む。この時に付けた名前は、その後の**drawpict**メッセージか**tilepict**メッセージによって**lcd**オブジェクトに描画される。**readpict**メッセージを受けると、**lcd**オブジェクトは第4アウトレットから、**pict**という単語に続き、その画像ファイルの名前を示すシンボルと、画像の幅と高さを示す2つの整数からなるメッセージを出力する。読み込みが失敗した場合は、**pict <画像に付けられた名前> error**というメッセージが第4アウトレットから出力される。
- recordregion** リージョンとして保存される描画コマンドの記録を開始する。記録の間は、描画コマンドは**lcd**オブジェクトの描画領域には描画されない。
- reset** **lcd**オブジェクトに描画されている内容を消去し、ペンの設定をデフォルト値に戻す。**reset**メッセージは以下の続くメッセージと等しい。
- ```
clear
pensize 1
penmode 0
frgb 0 0 0 (black)
brgb 255 255 255 (white)
moveto 0 0
```
- scrollrect** **scrollrect**には、スクロールさせる矩形の左、上、右、下の位置と、xおよびy方向にどれだけスクロールさせるかをピクセル単位で指定する6つの整数のアーギュメントが続き、**lcd**オブジェクト内の指定した矩形部分をスクロールさせる。
- size** **lcd**オブジェクトのサイズを変更する。**size**には新しいサイズをピクセル単位で指定する2つの整数のアーギュメントを続ける。





- tilepict** **tilepict**にはピクチャーの名前が続き、ピクチャーをタイル状に敷き詰めて矩形を塗りつぶす。オプションとして、アーギュメントとして4つの整数を付け加えることで、ピクチャーを敷き詰める矩形（デスティネーション矩形）を指定できる。さらに4つの整数を付け加えることで、元のピクチャーのどの部分をこの操作で使うか（ソース矩形）が指定できる。それらの矩形は左、上、右、下の位置をピクセル単位で指定する。デスティネーション矩形は**lcd**オブジェクトの左上の隅からの相対的な位置で、ソース矩形は画像の左上の隅からの相対的な位置で指定する。これらを指定しなければ、デスティネーション矩形は**lcd**オブジェクトと同じサイズに、ソース矩形は画像と同じサイズに設定される。
- write** **write**には任意のシンボルが続き、現在のペンの位置からそのシンボルを描き、描いたテキストの最後の位置にペンを移動させる。
- writepict** **writepict**には1つの任意のファイル名が続き、現在の**lcd**オブジェクトに表示されている内容をPICTファイルに書き出す。ファイル名を指定しなければ、標準ファイル保存ダイアログが開き、ファイル名とそのPICTファイルを書き出す場所を指定できる。

## インスペクター

**lcd**オブジェクトの振舞はインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**lcd**オブジェクトを選択することで、**lcd**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**lcd**オブジェクトを選び、オブジェクト・メニューから**Get Info...**を選択することでもインスペクターを表示することができる。

**lcd**オブジェクトのサイズは、*Width*、および*Height*のナンバー・ボックスにピクセル単位で入力することで、設定することができる。**lcd**オブジェクトのサイズのデフォルトは高さ128ピクセル、幅128ピクセルである。

*Local Mousing Mode*をチェックをすることで、**lcd**オブジェクトの表示領域にマウスで描画することができるようになる。デフォルトではこの機能は有効になっている。

*Draw Border*のチェックボックスはデフォルトでは有効になっている。チェックを入れることで、**lcd**オブジェクトの表示領域のまわりに枠線を表示する。

*Respond to idle Mousing*オプションにチェックを入れることで、**lcd**オブジェクト上でアイドル中のマウスの位置を出力することを指定する。デフォルトではこの機能は無効になっている。



*Onscreen Mode*オプションにチェックを入れることで、オンスクリーン・ウィンドウを描画に使うことにより、メモリの節約をする機能をオンにする。オンスクリーン・モードを使わない場合、**lcd**オブジェクトはより多くメモリを消費する。しかし、**lcd**オブジェクトが隠された場合でも、内容は記憶されており、消去されない。デフォルトではこの機能は無効になっている。

*Enable Sprites*オプションはスプライトの描画を有効にする。デフォルトではこの機能は無効になっている。スプライトを有効にすると、**lcd**オブジェクトはよりメモリを消費する。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

なし。

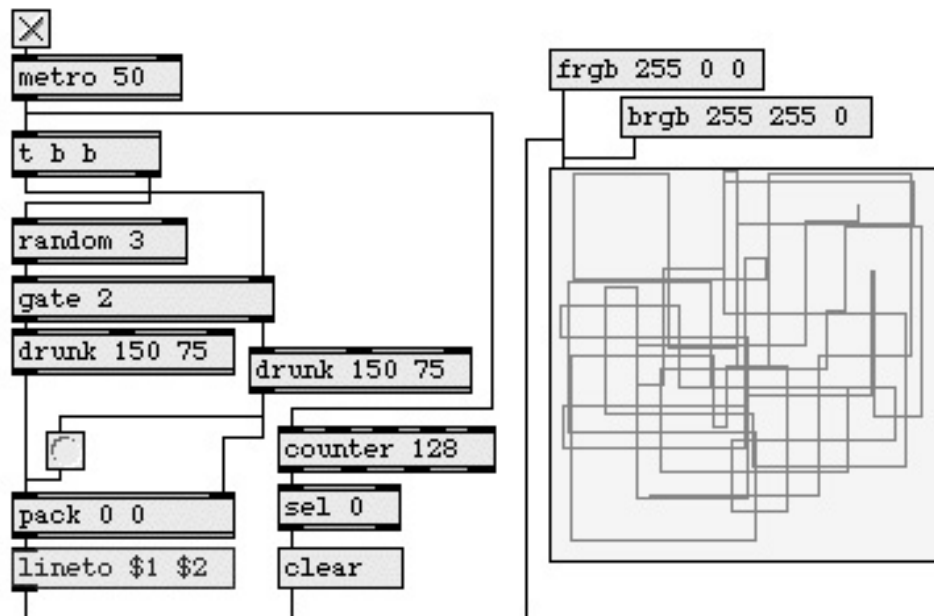
## 出力

- リスト 第1アウトレット：**lcd**オブジェクトの表示領域をクリックやドラッグした時に、そのマウスの位置の座標を、2つの項目からなるリストとしてアウトレットから出力する。その数値は**lcd**オブジェクト表示領域の左上の隅からの相対的な座標を表す。1つ目の数値は、その隅から右に何ピクセルか、2つ目の数値はその隅から下に何ピクセルかを示す。
- 整数 第3アウトレット：マウスのボタンが押されていたら、1というメッセージが第2アウトレットから出力される。それ以外では0というメッセージが出力される。
- リスト 第2アウトレット：**lcd**オブジェクトの表示領域をクリックやドラッグした時に、そのマウスの位置の座標を、2つの項目からなるリストとしてアウトレットから出力する。その数値は**lcd**オブジェクト表示領域の左上の隅からの相対的な座標を表す。1つ目の数値は、その隅から右に何ピクセルか、2つ目の数値はその隅から下に何ピクセルかを示す。
- リスト 第1アウトレット：**lcd**オブジェクト上でマウスのボタンを押しながらドラッグすることで描画を行った時に、そのマウスの位置の座標を、2つの項目からなるリストとしてアウトレットから出力する。その数値は**lcd**オブジェクト表示領域の左上の隅からの相対的な座標を表す。1つ目の数値は、その隅から右に何ピクセルか、2つ目の数値はその隅から下に何ピクセルかを示す。



- リスト** 第4アウトレット：**idle**メッセージか、インスペクターの*Respond to Idle Mousing*オプションによって、マウスのアイドル・モードが有効ならば、**lcd**オブジェクトの表示領域上にマウスが位置している時、そのマウスの位置の座標を第3アウトレットから出力する。
- update** 第4アウトレット：**update**は**lcd**オブジェクトがMaxアプリケーションから**update**メッセージ（**lcd**オブジェクト自身を描き直す）を受け取る度に出力される。これは**lcd**オブジェクトが**on screen**モードの時のみ出力される。
- penloc** 第4アウトレット：**getpenloc**メッセージに対して、**lcd**オブジェクトは、**penloc**から始まり、**lcd**オブジェクトの表示領域の左上の隅からの相対的なペンの位置の座標を表す2つの数値からなるメッセージを出力する。1つ目の数値は、その隅から右に何ピクセルか、2つ目の数値はその隅から下に何ピクセルかを示す。

## 利用例



lcd オブジェクトを用いて一筆書きをする

## 参照

**frame**  
**graphic**  
**MouseState**  
**oval**  
**panel**  
**rect**  
**ring**

グラフィック・ウィンドウに矩形の枠を描く  
スプライトによるグラフィックを描くためのウィンドウ  
マウスの状態や位置を出力する  
グラフィック・ウィンドウに塗り潰した楕円形を描く  
色のついた背景領域  
グラフィック・ウィンドウに塗り潰した矩形を描く  
グラフィック・ウィンドウに楕円形の枠を描く



## 入力

- 整数** 受け取った数値が0ならば、**led**オブジェクトはより暗い色を表示し、**0**を出力する。受け取った数値が**0**以外なら、**led**オブジェクトはより明るい色を表示し、**1**を出力する。
- 実数** 整数に変換される。
- bang** **led**オブジェクトを素早くオンにし、またオフにすることで点滅させ、**0**を出力する。また、**led**オブジェクトをクリックすることで、明るい色と暗い色とを切り替え、それぞれ**1**、**0**を出力する。
- blinktime** 第1インレット：**blinktime**には1つの数値が続き、クリックされた時や**bang**を受け取った時に、**led**オブジェクトを点滅させる間隔をミリ秒単位で指定する。
- pict** 第1インレット：**pict**には0から4までの整数が続き、**led**オブジェクトで使用される色を変更する。改造した**led**オブジェクトでは、**pict**メッセージはこのオブジェクトで使用する画像を変更することができる。
- set** **set**には0でない数値が続き、**led**オブジェクトを明るい色の状態にするが、数値の出力は行わない。**set 0**メッセージは**led**オブジェクトを暗い色の状態にするが、同じく数値の出力は行わない。
- toggle** **led**オブジェクトを暗い色から明るい色に切り替え、アウトレットから**1**を、またはその逆に、明るい色から暗い色へ切り替え、アウトレットから**0**を出力する。

## インスペクター

**led**オブジェクトの振舞はインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**led**オブジェクトを選択することで、**led**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**led**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**led**オブジェクトのインスペクターでは、以下の属性を変更できる。

*LED Pict*オプションでは、**led**オブジェクトの表示を赤（デフォルト）、緑、青、黄色、黒と白の5つの色から選択できる。

*Flash Time*はクリックされた時や**bang**を受け取った時に、**led**オブジェ



クトを点滅させる間隔をミリ秒単位で指定する。デフォルト値は150である。

Revertボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

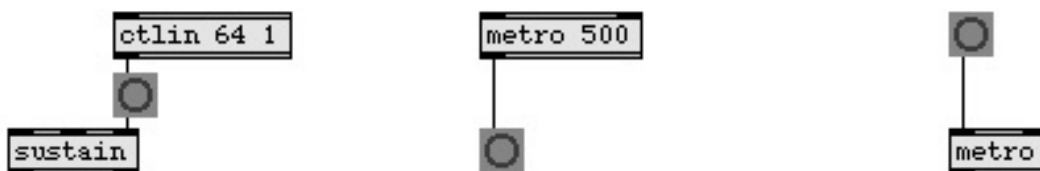
## アーギュメント

なし。

## 出力

**整数** **led** オブジェクトが明るい色の時は**1**を、暗い時には**0**を出力する。インレットに**bang**を受け取った時には、**led** オブジェクトをオンにし、またオフにすることで点滅させ、アウトレットから**0**を出力する。

## 利用例



オン/オフを表示する。点滅することで処理を表示する  
または **toggle** オブジェクトとして使用する

## 参照

**button**  
**pictctrl**  
**TogEdge**  
**toggle**

任意のメッセージによって動作し、**bang**を送信する  
ピクチャーを用いたコントロール  
0と0以外の数値の変化を出力する  
オンとオフ (**1**と**0**) を切り替える

Tutorial 40

Automatic actions (英文PDF参照)



## 入力

- リスト** リストの1つ目の数値は目標となる値を指定し、2つ目の数値は全体でかかる時間をミリ秒単位で指定する。その時間内に、現在格納されている値から目標となる値までリニアに等間隔で数値を出力する。
- 整数、または実数** 第1インレット：数値は、第2インレットに受け取られた数値で指定された時間をかけて到達する目標値となる。以前に目標となる値を受け取ってから、特に時間が設定されていなければ、時間は0と見なされ、ただちに目標となる値を出力する。
- 注記：**line**オブジェクトの出力の形式は、オブジェクトの1つ目のアーギュメントによって設定される（アーギュメントの項を参照のこと）。  
第2インレット：この数値は、目標となる値まで到達するのにかかる時間をミリ秒単位で指定する。  
第3インレット：この数値は、等間隔に連続して出力される数値の間隔を指定する。
- clock** **clock**には既に存在している**setclock**オブジェクトの名前が続き、Max内部のミリ秒クロックではなく、その**setclock**オブジェクトによってコントロールされるように、**line**オブジェクトを設定する。**clock**だけのメッセージなら、Max内部のミリ秒クロックによって、**line**オブジェクトがコントロールされることになる。
- stop** 第1インレット：新しい目標となる値が受け取られるまで、**line**オブジェクトからの出力を停止する。
- set** 第1インレット：**set**には1つの数値が続き、次に目標となる数値を受け取った時の開始値を設定する。**set**メッセージを受け取った時に、**line**オブジェクトが処理を行っていれば、それを停止する。

## アーギュメント

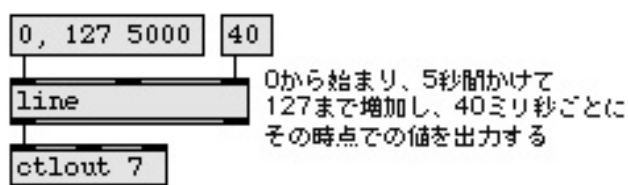
- 整数、または実数** オプション。1つ目のアーギュメントはこのオブジェクトの出力の形式を設定する。そのアーギュメントが整数ならば、**line**オブジェクトは整数を出力し、実数ならば、実数を出力する。また、1つ目のアーギュメントは、**line**オブジェクトに格納される初期値も設定する。アーギュメントが設定されていなければ、初期値は0で、出力の形式は整数になる。2つ目のアーギュメントは、値を出力する時間間隔（何ミリ秒ごとに値を出力するか）の初期値を設定する。その間隔が設定されていなければ、**line**オブジェクトは20ミリ秒ごとに数値を出力する。最小の間隔は1ミリ秒である。1より小さい数値は、20と見なされる。

## 出力

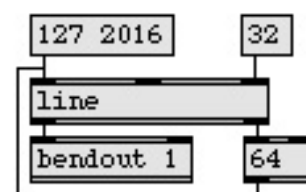
**整数** 第1アウトレット：数値は、目標となる値までリニアに補完されながら等間隔に出力される。**line**オブジェクトの処理が完全に終わらないうちに、新しい目標となる値と時間が設定された場合は、数値の不連続をさけるため、**line**オブジェクトは最も最近に出力した値から新しい処理を開始する。時間が設定されずに、第1インレットに数値を受け取った場合は、時間を0とみなし、その値を直ちに出力する。

**bang** 第2アウトレット：**line**オブジェクトが、その目標となる値に到達した時に、**bang**を出力する。  
注記：実際には、ちょうど指定された時間（時間引く間隔）で目標となる値に到達する。

## 利用例



リニアに値が出力され...



終了した時に **bang** が出力される

## 参照

**envi**

パッチ・ウィンドウに置く、スクリプトによって定義できるエンベロープ

**funbuff**

$x$ と $y$ の数値のペアを対にして格納する

**setclock**

タイミング関連のオブジェクトのクロック速度をリモート・コントロールする

**Uzi**

指定した数の**bang**を出力する

Tutorial 31

Using timers (英文PDF参照)



## 入力

インレットは持たない。ファイルを開いた時、またはパッチが他のファイルの一部である場合は、そのパッチを開いた時に自動的に**bang**が出力される。

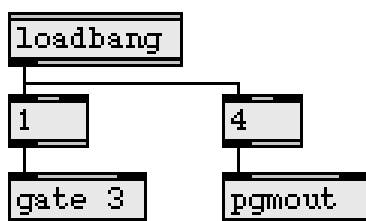
## アーギュメント

なし。

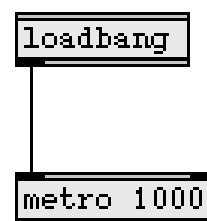
## 出力

**bang** パッチが読み込まれた時に自動的に**bang**が出力される。また、パッチがロックされた状態で**loadbang**オブジェクトをダブルクリックする、または同じパッチ内の**thispatcher**オブジェクトに**loadbang**メッセージを送ることで、**loadbang**オブジェクトから**bang**を出力させることができる。command+shiftキー (Macintosh) あるいはCtrl+Shiftキー (Windows) を押しながらパッチを開くことで、**loadbang**から**bang**が出力されるのを避けることができる。

## 利用例



パッチを読み込んだ時に初期値がセットされるようにする



または処理を自動的に開始させる

## 参照

|                    |                                     |
|--------------------|-------------------------------------|
| <b>active</b>      | パッチ・ウィンドウがアクティブであれば1を、そうでなければ0を出力   |
| <b>button</b>      | 任意のメッセージによって動作し、 <b>bang</b> を送信する  |
| <b>closebang</b>   | パッチ・ウィンドウが閉じられた時に <b>bang</b> を出力する |
| <b>thispatcher</b> | パッチにメッセージを送る                        |

Tutorial 40      Automatic actions (英文PDF参照)

## 入力

- 整数** 第1インレット：MIDIノート・オン・メッセージのノート・ナンバーとして扱われる。これはベロシティの数値と組になり、アウトレットから出力される。一定時間の後、そのノート・ナンバーに対するノート・オフ・メッセージ（ベロシティが0のノート・オン・メッセージ）が出力される。
- 第2インレット：第1インレットに入力されるノート・ナンバーの値と組になって出力されるベロシティの値を格納する。
- 第3インレット：**makenote**オブジェクトがノート・オフ・メッセージを出力するまでに待つ時間であるデューレーション（ミリ秒）の値を格納する。
- 実数** 整数に変換される。
- リスト** リスト内の2番目の数値はベロシティとして扱われ、第2アウトレットから出力される。1番目の数値はノート・ナンバーとして扱われ第1アウトレットから出力される。これに対するノート・オフ・メッセージは、後に出力される。
- stop** **makenote**オブジェクトに蓄えられているすべてのノート・ナンバーに対して、即座にノート・オフを出力する。
- clear** **makenote**オブジェクト内に蓄えられているすべてのノート情報を、ノート・オフを出力せずに消去する。

## アーギュメント

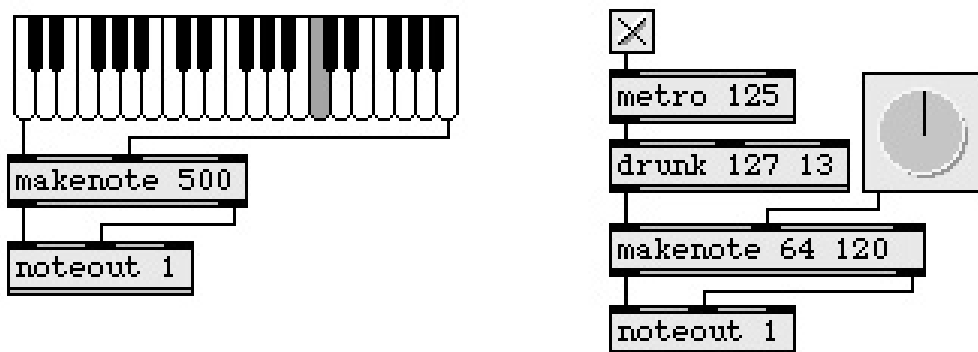
- 整数** オプション。アーギュメントの最初の数値は入力されたノート・ナンバーと組になって出力されるベロシティのデフォルト値を設定する。アーギュメントがなければ、ベロシティのデフォルト値は0である。
- アーギュメントの2番目の数値はデューレーション（ノート・オフを出力するまでの時間）のデフォルト値をミリ秒単位で設定する。もしこのアーギュメントがなければ、ノート・オフはノート・オンに続いてただちに出力される。
- 実数** 整数に変換される。

## 出力

**整数** 第1アウトレット：第1インレットに入力された数値を第2アウトレットのベロシティと一緒に、ただちに出力する。デューレーションで設定された一定間隔をあけて、同じ数値がベロシティ0と組になって出力される。

第2アウトレット：第2インレットに入力された数値はベロシティとして扱われ、ノート・ナンバーの値が第1アウトレットから出力されるのと同時に出力される。デューレーションで設定された一定間隔をあけて、同じノート・ナンバーの数値とともに0が出力される。

## 利用例



ノート・オンに対してノート・オフを Max 内で作り出して送り出す

## 参照

|                  |                                       |
|------------------|---------------------------------------|
| <b>flush</b>     | 保持しているノートについてノート・オフを作る                |
| <b>midiout</b>   | 生のMIDIデータを送信する                        |
| <b>noteout</b>   | MIDIノート・メッセージを送信する                    |
| <b>stripnote</b> | ノート・オフ・メッセージを取り除き、ノート・オン・メッセージのみを出力する |
| <b>xnoteout</b>  | ノート・オフ・ベロシティを持つMIDIノート・メッセージを作成する     |

Tutorial 13      Managing note data (英文PDF参照)

## 入力

- 整数** 数値がアーギュメントと同じ順序で一致すれば、リストとして出力する。
- clear** **match**オブジェクトがそれまで受け取ったすべての数値を消去する。
- set** リストを伴う**set**メッセージは、**match**オブジェクトが新たに判断する数値の並びを指定する。

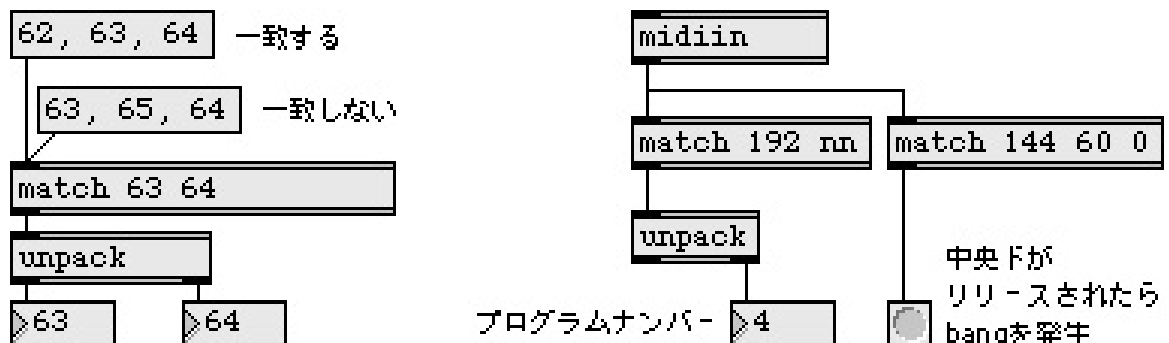
## アーギュメント

- リスト** 必須。このアーギュメントは**match**オブジェクトが同じ並びを探す数値を指定する。**nn**はワイルド・カードとして扱われ、どの数値でも一致すると見なす。

## 出力

- リスト** インレットに入力される数値はアーギュメントと比べられる。それらが同じ順序で一致すれば、リストとして出力する。

## 利用例



数値は同じ数、同じ順序でなくてはならない

## 参照

**iter**  
**pack**  
**select**

リストを数値ごとに分解する  
数値やシンボルをひとつのリストに結合させる  
特定の入力を選択し、それ以外は通過させる



**matrixctrl**は、セル (cell) と呼ばれるスイッチのようなコントロールを、長方形のマトリックス状に並べたユーザー・インターフェースのためのオブジェクトである。**matrixctrl**での、すべてのセルは同じ表示方法と機能を持つ。デフォルトでは、セルは2つの状態、つまり、オンとオフの状態を持つ。複数の状態を持つセルを作ることもできる。セルをクリックすることによって、1つずつ状態を増やすことになる。セルが最後の状態に達した後で、さらにもう一度クリックすることによって、0の状態に戻る。つまり、2つのみの状態のセルでは、マウスでクリックされる度に、スイッチがオンとオフになることを繰り返す。

**matrixctrl**は、MSPのオブジェクト**matrix~**をコントロールするために本来作られた。しかし、スイッチのグループとして、または、視覚的に表示するグループとして、そして、ドラム・マシンのようなシーケンサーとしてのユーザー・インターフェースに活用することができる。

## 入力

(マウス) セルをマウスでクリックすることで、1つずつ数値が増加する。最大限可能な状態に辿り着いた後は、**matrixctrl**の数値は0に戻る。幾つかのセルにまだがってドラッグすることによって、最初にクリックしたセルでの同じ数値を、すべてに設定する。shiftキーを押しながら、複数のセルにまたがってドラッグすることで、縦または横のラインのみをドラッグすることができる。

**bang** **bang**は、それぞれセルの状態を、以下の形式の3つの数値のリストとして出力する。

縦位置 横位置 値

**list** 整数のリストは、縦位置、横位置、値の形式で、セルを設定する。3つの値の組を複数用いることによって、一度に多くのセルを設定することができる。セルの位置は、左上が0から始まる。おのおののセルにおける値は0から始まり、オブジェクトのインスペクターによって設定された最大値から1を引いた値までの範囲である。数値の代わりに、シンボルの**inc**と**dec**を用いれば、セルの値を1つずつ増加、または、減少させることができる。リストによってセルの状態を変えれば、そのリストは**matrixctrl**から出力される。

**set** 上記のように、リストが続く**set**は、そのリストを出力することなく、**matrixctrl**内の状態のみを変える。

**active** 0、または、1が続く**active**は、マウスに反応するか否かかを設定する。デフォルトでは、**matrixctrl**はマウス・クリックに反応する。

**bkgndpicture** ファイル名を指定するシンボルが続く**bkgndpicture**は、マトリックスのバックグラウンドの画像となるグラフィック・ファイルを指定する。**matrixctrl**では、PICTファイルを使うことができる。また、



QuickTimeをインストールしている場合は、QuickTime Appendixに挙げられている、他の画像ファイル・フォーマットも使うことができる。ファイル名として使われるシンボルは、Maxのサーチ・パスにあるファイル名であるか、ファイルの完全なフル・パス名（例えば、“MyDisk:/Documents/UIPictures/CoolBkgnd.pct”）でなければならない。

**cellpicture** ファイル名を指定するシンボルが後に続く**cellpicture**は、マトリックスの画像となるグラフィック・ファイルを指定する。**matrixctrl**では、PICTファイルを使うことができる。また、QuickTimeをインストールしている場合は、QuickTime Appendixに挙げられている、他の画像ファイル・フォーマットも使うことができる。ファイル名として使われるシンボルは、Maxのサーチ・パスにあるファイル名であるか、ファイルの完全なフル・パス名（例えば、“MyDisk:/Documents/UIPictures/CoolBkgnd.pct”）でなければならない。

**clickedimage** 0でない数値が後に続く**clickedimage**は、セルがクリックされた時に表示する追加のイメージを、グラフィック・ファイルが持っていることを指定する。

**clickvalue** 数値が後に続く**clickvalue**は、クリックの数値のモードを変える。0、または、正の数が続く**clickvalue**メッセージの場合は、クリックすればセルの値を指定した数値に変化する設定になる。負の数が**clickvalue**に続く場合は、クリックすればセルの値が増加する、**matrixctrl**オブジェクトのデフォルトの設定に戻る。**clickvalue**メッセージによって、異なる数値に対応する連なったイメージを含む画像のファイルを作ることによって、マトリックスのエディターを作ることができる。例えば、連なった画像を順番にクリックすることによって、セルの画像がペロシティ、ノートなどを反映するように変えることができる。

**disable** 縦横の位置を指定した数値の組のリストが後に続く**disable**は、指定したセルをマウス・クリックに反応しないように設定する。**disable**メッセージは、少なくとも一組以上の数値を必要とする。しかし、複数のセルを作動しないようにするために、一度に多くの数値を付け加えることもできる（例えば、**disable 0 0 3 4 9 12**）。作動しないように設定されたセルは、マウスのクリックに反応しないが、それらの値をメッセージによって設定することはできる。

**enable** 縦横の位置を指定した数値の組のリストが後に続く**enable**は、指定したセルをマウス・クリックに反応するように設定する。**enable**メッセージは、少なくとも一組以上の数値を必要とする。しかし、複数のセルを作動するようにするために、多くの数値を並べることもできる（例えば、**enable 1 1 1 2 2 2**）。

**getrow** 数値が後に続く**getrow**は、第2アウトレットから指定した横の列のセルの値を出力する。





|                          |                                                                                                                                                  |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>getcolumn</b>         | 数値が後に続く <b>getcolumn</b> は、第2アウトレットから指定した縦の列のセルの値を出力する。                                                                                          |
| <b>horizontalmargin</b>  | 数値が後に続く <b>horizontalmargin</b> は、最も外側のセルと <b>matrixctrl</b> オブジェクトの境界線の間をの水平方向の間隔を、ピクセル単位で設定する。                                                 |
| <b>horizontalspacing</b> | 数値が後に続く <b>horizontalspacing</b> は、 <b>matrixctrl</b> オブジェクトのセルの間をの水平方向の間隔を、ピクセル単位で設定する。                                                         |
| <b>imagemask</b>         | 0でない数値が後に続く <b>imagemask</b> は、 <b>matrixctrl</b> のセルのグラフィック・ファイルが、イメージ・マスクとして用いる、横の列の画像を追加で持つことを指定する。                                           |
| <b>inactiveimage</b>     | 0でない数値が後に続く <b>inactiveimage</b> は、 <b>matrixctrl</b> のセルのグラフィック・ファイルが、動作をしない状態 ( <b>active 0</b> のメッセージで設定する)として用いる、横の列の画像を追加で持つことを指定する。        |
| <b>invisiblebkgnd</b>    | 0でない数値が後に続く <b>invisiblebkgnd</b> は、 <b>matrixctrl</b> に背景の画像なしで表示するように指定する。セルは下に置かれているMaxオブジェクトの上に重ねて描かれる。 <b>invisiblebkgnd 0</b> はこの機能を無効にする。 |
| <b>one/row</b>           | 0でない数値が後に続く <b>one/row</b> は、横の列の中で1つのセルのみが0でない状態になるよう設定する。1つの横の列で、あるセルを0でない状態に設定すれば、他の0でないセルは0の状態に変わる。 <b>one/row 0</b> は、この機能を解除する。            |
| <b>one/column</b>        | 0でない数値が後に続く <b>one/column</b> は、縦の列の中で1つのセルのみが0でない状態になるよう設定する。1つの縦の列で、あるセルを0でない状態に設定すれば、他の0でないセルは0の状態に変わる。 <b>one/column 0</b> は、この機能を解除する。      |
| <b>one/matrix</b>        | 0でない数値が後に続く <b>one/matrix</b> は、全体の中で1つのセルのみが0でない状態になるよう設定する。マトリックスにおいて、あるセルを0でない状態に設定すれば、他の0でないセルは0の状態に変わる。 <b>one/matrix 0</b> は、この機能を解除させる。   |
| <b>range</b>             | 整数が続く <b>range</b> は、それぞれのセルが持つ状態の数を設定することができる。最小でも2 (0と1の状態) の数値を設定しなければならない。                                                                   |
| <b>soundfx</b>           | 0、または1が続く <b>soundfx</b> は、MacOS 8.5またはそれ以降で利用できる、アピランスのサウンドを使うように設定する。 <b>soundfx 0</b> は、サウンドをオフにする。                                           |
| <b>verticalmargin</b>    | 数値が後に続く <b>verticalmargin</b> は、最も外側にあるセルと <b>matrixctrl</b> オブジェクトの境界線の間をの垂直方向の間隔を、ピクセル単位で設定する。                                                 |





**vertialspacing** 数値が後に続く**vertialspacing**は、**matrixctrl**オブジェクトのセルの間の垂直方向の間隔を、ピクセル単位で設定する。

## インスペクター

**matrixctrl**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**matrixctrl**オブジェクトを選択することで、**matrixctrl**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**matrixctrl**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

*Cell Spacing*のナンバー・ボックスは、**matrixctrl**オブジェクトのセルの横と縦の間隔を、ピクセル単位で設定する。

*Margin*のナンバー・ボックスは、外側のセルとオブジェクトの境界線との縦と横の間隔を、ピクセル単位で設定する。

*Has Clicked Images*のオプションをチェックすれば、ユーザーがクリックしたりドラッグする時に、セルが異なる表示になるように、グラフィックス・ファイルが異なる画像フレームの集まりを持つことを指定する。

*Has Inactive Images*のチェックボックスは、セルが動作しない状態を表示できるように、グラフィックス・ファイルが追加の画像を持つことを指定する。セルが動作しない状態を表示しない場合は、このチェックボックスをチェックしない。

セルを描く時に、セルのグラフィックス・ファイルで画像のマスクを使いたい場合は、*Has Image Mask*のオプションを選択する。マスクは、長方形ではない形でセルを作るために用いられる。セルの画像がクリックや動作停止状態の異なる画像を持つならば、これらについてもマスクを用意しなければならない。

*Invisible Background*ボックスを選択することによって、マトリックスの背景に何も表示しないように指定する。セルは、下に置かれているオブジェクトの上に、浮かび上がるように表示される。

*Audible Feedback*のオプションが選択されている場合は、クリックされた時に、Mac OSのアピアランス・マネージャーのサウンド効果によって、音が鳴る。



*One Per Column*、*One Per Row*、*One Per Matrix*のチェックボックスは、**matrixctrl**オブジェクトの動作を決定する。それぞれが選択されている場合は、横の列、縦の列、そして全体につき1つのみのセルが、0ではない状態を持つように設定する。あるセルを0ではない状態にすれば、0ではないセルは0の状態に変わる。

*Cell Value Range*は、それぞれのセルが持つことができる状態の数を設定できる。最小でも2（0と1の状態）が設定されなければならない。

*Cell Picture File*と*Background Picture File*は、Openボタンをクリックすることによって、マトリックスのセルと背景のためのグラフィックス・ファイルを選択することができる。PICTファイルを選択することができるが、QuickTimeがインストールされている場合は、QuickTime Appendixに挙げているファイル・フォーマットの画像も選択できる。現時点のファイルの名前は、ボタンの左のテキスト・ボックスに表示される。このボックスにファイル名を入力するか、Finderからファイルのアイコンをドラッグすることによっても、設定をすることができる。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

なし

## ピクチャー・ファイルのフォーマット

背景のピクチャーのファイルは、MacintoshのPICTファイルを用いることができる。QuickTimeがインストールされている場合は、QuickTime Appendixにあるファイル・フォーマットも用いることができる。**matrixctrl**が、選択されたピクチャーより大きい場合は、オブジェクト全体を埋めるようにピクチャーが自動的にコピーされる。セルのピクチャーのファイルは、下記のフォーマットである。



|                                  |                                  |     |                                    |
|----------------------------------|----------------------------------|-----|------------------------------------|
| Not Clicked<br>value = 0         | Not Clicked<br>value = 1         | ... | Not Clicked<br>value = n-1         |
| Clicked<br>value = 0             | Clicked<br>value = 1             | ... | Clicked<br>value = n-1             |
| Inactive<br>value = 0            | Inactive<br>value = 1            | ... | Inactive<br>value = n-1            |
| Not-clicked<br>Mask<br>value = 0 | Not-clicked<br>Mask<br>value = 1 | ... | Not-clicked<br>Mask<br>value = n-1 |
| Clicked<br>Mask<br>value = 0     | Clicked<br>Mask<br>value = 1     | ... | Clicked<br>Mask<br>value = n-1     |
| Inactive<br>Mask<br>value = 0    | Inactive<br>Mask<br>value = 1    | ... | Inactive<br>Mask<br>value = n-1    |

ピクチャーはマトリックス状の画像として構成される。それぞれの画像は同じ縦横の大きさを持つ。それぞれの縦の列の画像は1つのセルの状態を表示する。セルは少なくとも2つの状態を持たなければならないので、画像も少なくとも2つの縦の列を持たなければならない。

画像の最初の横の列は通常の（または、選択されていない）セルの表示として使われる。画像の最初の横の列は必須である。それに続く横の列はオプションである。2番目の横の列は、クリックされた時に表示される画像である。クリックされた時に、これらの画像によってセルが表示される。マウスが離された時、セルの表示は通常の画像へと戻る。**matrixctrl**が使用停止状態の時、例えば**active 0**のメッセージを受け取った時、画像の3番目の横の列が用いられる。

長方形でない形でセルを作るために、画像のマスクを用いることができる。これらのマスクは、ピクチャー・ファイルの一番下の横の列である。セルの画像にマスクを用いたい場合は、すべての状態についてマスクとして用意しておかなければならない。つまり、画像のそれぞれの横の列は、対応するマスクの横の列を持つ。Maxのピクチャーを用いたコントロールにおけるマスクと同じように、黒いピクセルは画像が描かれる領域を指定し、白いピクセルは表示されない領域を指定する。

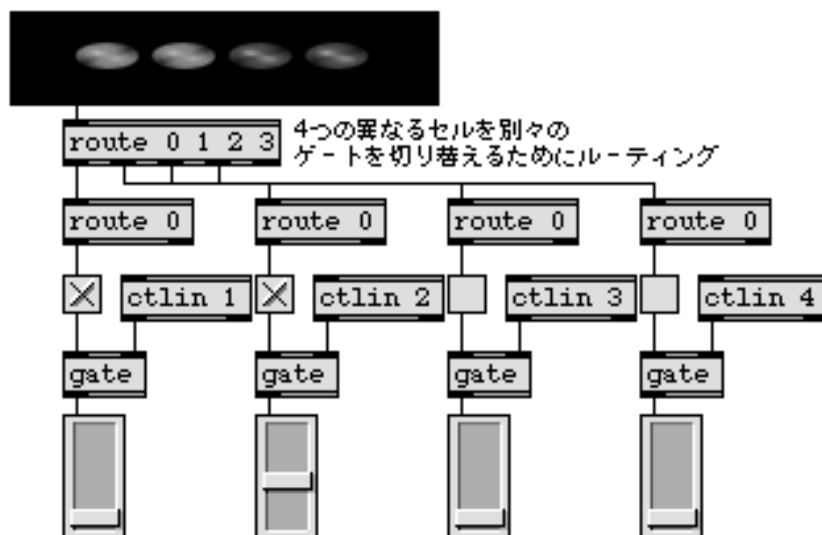


## 出力

**リスト** マウス・クリックに反応してセルの状態が変わる時、**matrixctrl**オブジェクトの第1アウトレットからリストが出力される。リストは横の列、縦の列、クリックされたセルの値から成り立っている。オブジェクトの第1インレットにリストを送ることによって、個々のセルの状態を設定することもできる。縦の列と横の列は、マトリックスの左上の隅を0として番号付けられる。

インレットに受け取った数値はアーギュメントと比較される。数値と順序が一致すれば、それらはリストとしてアウトレットから出力される。

## 利用例



**matrixctrl** は複数の **gate**、または、**switch** を同時に操作するために使うことができる

## 参照

|                   |                            |
|-------------------|----------------------------|
| <b>dial</b>       | ダイアル操作によって数値を出力する          |
| <b>hslider</b>    | スライダー操作によって数値を出力する         |
| <b>kslider</b>    | 画面上のキーボードから数値を出力する         |
| <b>pictctrl</b>   | ピクチャーを用いたコントロール            |
| <b>pictslider</b> | ピクチャーを用いたスライダー・コントロール      |
| <b>rslider</b>    | 数値の範囲を変更し、それを表示する          |
| <b>slider</b>     | 画面上でスライダーを動かすことによって数値を出力する |
| <b>ubutton</b>    | 透明なボタン、 <b>bang</b> を出力する  |
| <b>uslider</b>    | 画面上でスライダーを動かすことによって数値を出力する |

## 入力

- 整数** 第1インレット：入力された数値が、現在**maximum**オブジェクトに格納されている値よりも大きい場合は、入力された値を出力する。そうでない場合は、格納されている値を出力する。
- 第2インレット：この数値は格納され、それより後に第1インレットから入力された値と比較される。
- 実数** アーギュメントが実数でない場合は、整数に変換される。アーギュメントが実数の場合は、すべての数値は実数として扱われる。
- リスト** 第1インレット：リスト内のすべての数値が互いに比較され、そのうち最も大きい値が、アウトレットから出力される。また、リストのうち2番目に大きい値が、**maximum**オブジェクトに格納される。**maximum**オブジェクトで扱えるリストの最大要素数は、256である。
- bang** 第1インレット：最後に出力した数値を、再びアウトレットから出力する。

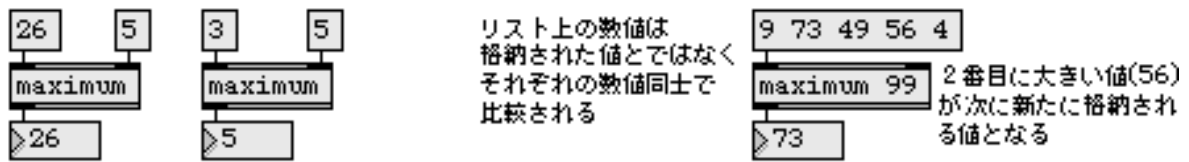
## アーギュメント

- 整数または実数** オプション。第1インレットに受け取った値と比較される初期値を設定する。アーギュメントの値が小数点をもつ場合は、比較に用いられるすべての値は実数として扱われ、出力も実数になる。アーギュメントを設定しない場合は、初期値は0になる。

## 出力

- 整数** **maximum**オブジェクトに格納されている値と、第1インレットに受け取った値を比較し（第1インレットにリストを受け取った場合は、リスト内のすべての数値が互いに比較される）、より大きい値をアウトレットから出力する。
- 実数** アーギュメントの値が小数点をもつ場合に限り、実数として出力される。

## 利用例



2つの値のより大きい方、または数値のリストの中で最も大きい値を出力する

## 参照

**minimum**

数値のリストの中で最も小さい値を出力する

**past**

数値がある値を超えた時に出力する

**Peak**

数値が直前の値より上回る時に出力する

>

2つの数値を比較し、より大きいことを判断する

## 入力

- 整数または実数** 数値は受け取ったすべての数の合計に加えられ、さらにその平均値が出力される。
- bang** 以前出力した数値（格納されている平均値）を出力する。
- リスト** リストの各数値は加算され、その合計はリスト内の項目の数で割られ、求められる平均値が出力される。以前に入力されたすべての数値はメモリから消去される。
- clear** **mean** オブジェクト内の内容を0にリセットする。

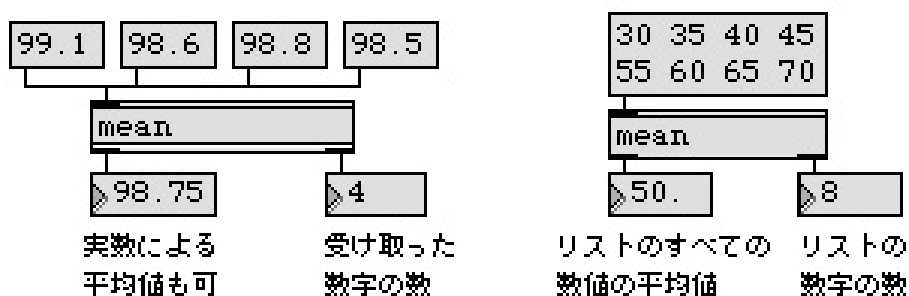
## アーギュメント

なし。

## 出力

- 実数** 第1アウトレット：受け取ったすべての数値の平均値か、あるいはリスト内のすべての数の平均値が出力される。
- 整数** 第2アウトレット：平均値を求めるために用いた数値の総数。

## 利用例



## 参照

- accum** 数値の格納、加算、乗算を行う
- anal** 受け取った2つの数値をペアとして、ヒストグラムを作る
- bag** 数値の集合を格納する
- Histo** 受け取った数値のヒストグラムを生成する
- prob** 重みを付けてランダムな数値の流れを作り出す



**menubar**オブジェクトはコンピュータのOSが提供するメニュー・バーの操作を可能にする。独自のメニュー、そして、通常のFileやEditのメニューに項目を加えることができる。メニューの項目が選択された時、項目の番号がメニューによって異なるアウトレットから出力される。パッチがロックされている状態で、オブジェクトをダブルクリックすれば、テキスト・エディターを開くことができ、そこに書き込まれたスクリプトによって**menubar**オブジェクトを設定する。

## 入力

- 整数** 0ではない数値は、**menubar**オブジェクトのメニューを表示する。0は以前の内容（Maxの通常のメニューか、他の**menubar**オブジェクトのメニュー）にメニュー・バーを戻す。
- checkitem** メニュー番号、項目番号、そして0または1が後に続く**checkitem**は、最後の数値が1である場合は、指定された項目の前にチェック・マークを入れる。0である場合は、チェック・マークを取り除く。
- enableitem** メニュー番号、項目番号、そして0または1が後に続く**enableitem**は、最後の数値が1である場合は、指定された項目を使えるようにする。0である場合は、使えないようにする。
- markitem** Macintoshのみ。メニュー番号、項目番号、そしてASCIIコードの文字番号が後に続く**markitem**は、指定された項目の前に文字を入れる。通常のマーク文字のASCIIコードは、18がチェック・マークで、19がダイヤモンド・マークである。ダッシュ・マーク（209）や砲弾マーク（165）なども使うこともできる。
- (メニュー・バー) **menubar**オブジェクトが（インレットに0ではない数字が送られることによって）使用可能になった状態で、メニューバーから項目が選択された時、**menubar**オブジェクトはメニュー番号と項目番号を受け取り、その項目番号が該当するアウトレットから出力される。

## アーギュメント

**整数** 任意。最初のアーギュメントは、オブジェクトのメニュー・バーに、メニューの数を設定する。このアーギュメントの数値は、少なくとも5（すなわち、1つの追加メニュー）でなければならない。常に表示される4つのメニューは、File、Edit、Windows、Helpである。MacintoshではこれにアップルメニューとMax/MSPメニュー（OS Xのみ）がシステム標準メニューとして追加される。

2番目の任意のアーギュメントは、デフォルトのメニューから特定項目を取り去るための、数値コードである。このコードは、次のようなメニュー項目に対応する数値の合計である。**1=Overdrive**（Optionメニュー）、**2=Resume**（Editメニュー）、**4=Midi Setup**（Fileメニュー）などである。例えば、**Overdrive**と**Midi Setup**のメニュー項目を取り去るためには、2番目のアーギュメントは適切な数値は、5（1+4）となる。

## スクリプト・メッセージ

ロックされているパッチの**menubar**オブジェクトをダブルクリックすれば、テキスト・エディター・ウィンドウが開き、そこに一連のスクリプト・メッセージを書き込むことによって、**menubar**を設定する。スクリプト・ウィンドウを閉じて、修正部分を保存すれば、スクリプト・ファイルの解釈が行われる。エラーがなければ、**menubar**がインレットに0でない数字を受けた時に、独自に用意したメニュー・バーを利用できるようになる。

それぞれのメッセージは#Xから始まり、セミコロン (;) で終わる。最初のスクリプトのメッセージは**apple**で、最後は**end**で終わらなければならない。以下のメッセージの定義の説明の後に、スクリプトの具体例を挙げている。

## 標準のメニューを変更するためのメッセージ

### メッセージ アーギュメント

**apple** Mac OS 9のみ。  
・ Appleメニューの最初の項目のテキスト  
(例えば、About My Program...)

**about** ・ メニューの最初の項目のテキスト  
(例えば、About My Program...)

Mac OS Xでは、About画面はアプリケーションメニュー (Max/MSPメニュー) の最初の項目に位置する。Windowsでは、Helpメニューの最初の項目に位置する。

**file** ・ 出力する項目番号  
・ Fileメニューに追加する項目のテキスト

**file**メッセージは、標準的なFileメニュー (**Midi Setup...**メニューの項目の前) に、項目を挿入する。おのおのの項目は、選択時に出力される番号を持っている。**File**メニューに追加される項目の順序は、項目に割り当てる任意の番号ではなく、スクリプトに現れる順序によって決められる。

**edit** ・ 出力する項目番号  
・ Editメニューに追加する項目のテキスト

**edit**メッセージは、標準的なEditメニューの、**Clear**の後で**Overdrive**と**Resume**の前 (これらは、**menubar**の動作中はEditメニューに移動する) に、項目を追加する。空白の行は、デフォルトの項目と追加された項目とを分ける。それぞれの項目は、選択時に、**menubar**の第3アウトレットから出力される数値に対応する番号を持つ。Editメニューに追加される項目の順序は、項目に割り当てる任意の番号ではなく、スクリプトに現れる順序によって決められる。

**newitem** ・出力する項目番号

0ではない数値が後に続く**newitem**メッセージは、Fileメニューから**New**が選ばれた時に、新しいパッチ・ウィンドウを開くのではなく、**menubar**オブジェクトのFileメニューに対応するアウトレットから、その数値を出力する。**newitem 0**メッセージ（または、**newitem**メッセージがない場合）は、**New**が通常通りに機能するよう設定する。

**open** ・出力する項目番号

0ではない数値が後に続く**open**メッセージは、Fileメニューから**Open...**が選ばれた時に、標準ファイル・ダイアログを開くのではなく、**menubar**オブジェクトのFileメニューに対応するアウトレットから、その数値を出力する。**open 0**メッセージ（または、**open**メッセージがない場合）は、**Open...**が通常通りに機能するよう設定する。

**closeitem** アーギュメントなし

アクティブ・ウィンドウを閉じるために、Fileメニューの**Close**項目を表示するように設定する。

**saveas** ・出力する項目番号

0ではない数値が後に続く**saveas**メッセージは、Fileメニューから**Save As...**が選ばれた時に、標準的なファイル保存動作を行うのではなく、**menubar**オブジェクトのFileメニューに対応するアウトレットから、その数値を出力する。**Save**が選ばれた時は、**Save As...**が選ばれた時よりも1小さい数値がアウトレットから出力される。**saveas 0**メッセージ（または、**saveas**のメッセージがない場合）は、**Save**と**Save As...**が通常通りに機能するよう設定する。

## 新しいメニューと項目を作るためのメッセージ

メッセージ アーギュメント

**munutitle** ・メニュー番号（この数値は5以上でなければならず、**menubar**のアーギュメントで指定されたアウトレットの数を超えてはならない）  
・メニューの名前

**menutitle**メッセージは、メニュー・バーでWindowsメニューの後に新しいメニューを追加する。最初に追加されるメニューの番号は5である。メニュー番号は、メニュー・バーに追加されるメニューの順序と、項目が選択された時に使われるアウトレットの両方を決定する。**menutitle**メッセージは、そのメニュー番号に関連するあらゆる**item**メッセージの前のスクリプトに置かれなければならない。

**item** ・メニュー番号  
・項目番号  
・項目のテキスト  
・特別な記号（任意）

**item**メッセージは、それまでに**menutitle**メッセージで定義した追加のメニューに、メニュー項目を追加する。そのメニューにおいて項目が表れる順序は、項目に割り当てる任意の番号でなく、スクリプトに現れる順序によって決められる。項目番号のアーギュメントは、項目が選択された時に、**menubar**オブジェクトのアウトレットから出力される番号を指定する。項目番号は1から始めて、メニューに表れる順序に従って、項目を列挙するのが良いだろう。

項目のテキストにメタ文字を含めることで、メニュー項目の表示を変えることができる。これらは、Toolboxの**AppendMenu()**関数によって認識される特別な記号である（詳細に関しては、*Inside Macintosh*, Volume I, pp. I-346 - I-349を参照のこと）。認識されるメタ記号の一部を以下に説明する。

**/**の後に続く1文字によって、commandキーとして評価される文字を割り当てる。

**<**の後に続く**B**、**I**、**O**、**S**、**U**によって、フォント・スタイルが決まる（例えば、**O**はアウトライン）。

**!**の後に続く1文字によって、メニュー項目に特定のマーク記号が付けられる。

**(**は、メニュー項目を使用不可にする。

従って、これらの特殊文字は、実際の項目のテキストの一部として用いることができない。例えば、**On/Off**というテキストは「On/Off」ではなく、「**Onff\_O**」として表示される。

## スクリプト定義の完了

|            |             |
|------------|-------------|
| メッセージ      | アーギュメント     |
| <b>end</b> | (アーギュメントなし) |

**end**メッセージはメニューを構築し、また、エラーが起こった場合は、エラーを出力する。

## 出力

**整数** デフォルトの**menubar**オブジェクトは、4つのアウトレットを持っている。**menubar**オブジェクトが（インレットに0ではない数字を受け取ることで）使用可能となった場合に、Appleメニューで最初の項目が選ばれると、第1アウトレットから1を出力する。第2アウトレットは、Fileメニューで追加された項目が選択された時、その項目番号を出力する。第3アウトレットは、Editメニューで追加された項目が選択された時、その項目番号を出力する。第4アウトレットは、Windowsメニューで項目が選択された時、その項目番号を出力する。追加されたメ

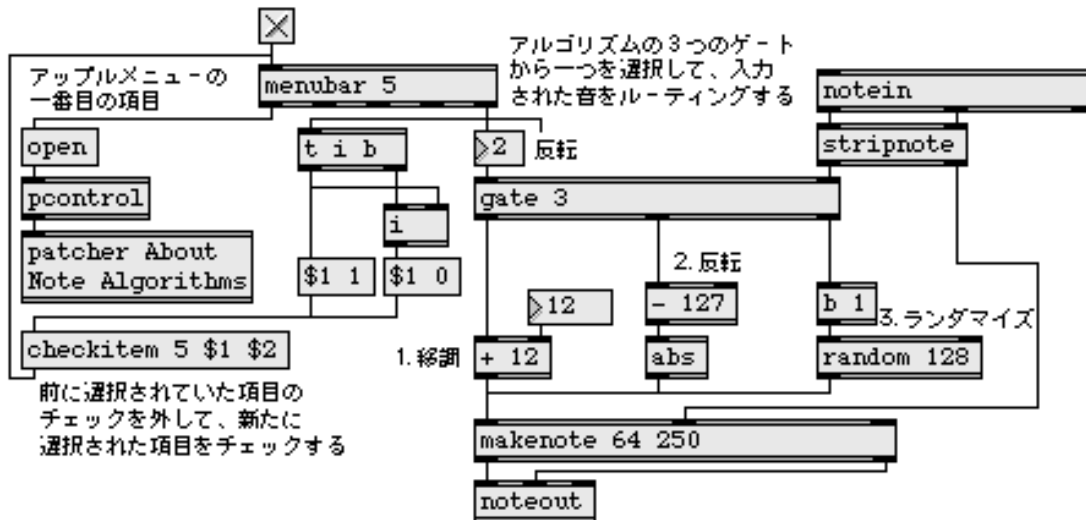
メニューが定義されている場合、5番目のアウトレット以降の追加された右のアウトレットから、選択された項目番号を出力する。

## 利用例

以下は、**menubar**のスキプットの例である。

注記：メニューにおいて他の項目と同じスタイルを保つために、メニュー項目の単語の最初の文字を大文字にすることが推奨される。

上記のスキプットは、以下のパッチの**menubar**オブジェクトで使われている。これは、音を扱う3つのアルゴリズムを切り替えるために、**extra**メニューを用いている。



**menubar** のスキプットの例の活用例

## 参照

**umenu**

コマンドを表示し、出力するポップアップ・メニュー





## 入力

**message**オブジェクト（メッセージを表示し、出力するボックス）は、メッセージ・ボックスと呼ばれることが多い。これは、メッセージ（あるオブジェクトから他のオブジェクトへ実際に送られるデータ）とは異なるものである。

**bang** メッセージ・ボックスの内容を出力する。メッセージ・ボックスをマウス・クリックすることでも、同じ効果がある。

**整数または実数** アーギュメントに**\$1**が含まれている場合は、アーギュメントの**\$1**を数値に置き換えた上で、メッセージ・ボックスの内容を出力する。

**リスト** アーギュメントに**\$**と番号が含まれている場合は、一致する番号の**\$**アーギュメントを、リストのそれぞれの項目に置き換えた上で、メッセージ・ボックスの内容を出力する。

**symbol** **symbol**という単語の後に、アーギュメントとしてのシンボルが続く場合は、**\$1**のアーギュメントをそのシンボルに置き換えた上で、メッセージ・ボックスの内容を出力する。

**color** 数字の0から15までの数値が後に続く**color**は、Objectメニューにある**Color**コマンドと同じように、メッセージ・ボックスの色を、指定した色に設定する。

**set** メッセージが続く**set**は、出力は行わずに、メッセージ・ボックスの内容を新しいメッセージに設定する。**set**メッセージだけであれば、メッセージ・ボックスの内容を消去する。

**append** メッセージが続く**append**は、出力は行わずに、メッセージ・ボックスの内容の最後に（空白文字に続けて）メッセージを付け足す。

**prepend** メッセージが続く**prepend**は、出力は行わずに、メッセージ・ボックスの内容の最初に（空白文字が続く）メッセージを付け足す。

**open** メッセージ・ボックスのインスペクター・ウィンドウを開く。**open**に1が続く場合は、インスペクター・ウィンドウでテキストが変更された時、またはインスペクター・ウィンドウが閉じられた時に、メッセージ・ボックスの内容をアウトレットから出力する。オプションである**open**メッセージの第2アーギュメントは、インスペクター・ウィンドウの上部に表示するプロンプトを指定する。デフォルトのプロンプトは「Set Message Text」である。プロンプトに空白文字を含めたい場合は、シングル・スマート・クォート（option-]とshift-option-]）を利用する。



## インスペクター

メッセージ・ボックスの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意のメッセージ・ボックスを選択することで、メッセージ・ボックスのインスペクターをフローティング・ウィンドウに表示することができる。メッセージ・ボックスを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

*Set Message Text*の部分にテキストを入力することによって、メッセージ・ボックスの内容を設定する。

Revertボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

**任意** メッセージ・ボックスの内容は、パッチ・ウィンドウのロックを外して（エディット・モードにして）書き込むことができる。256項目のメッセージをメッセージ・ボックスに含めることができる。いくつかの記号は特別な意味を持っている。

**\$** ドル記号 (\$) のすぐ後に数字が続くことによって、常に変更できるアーギュメントとなる。このアーギュメントの数値は、インレットに受けたリストで一致する項目に置き換えられる（例えば、**\$2**は、メッセージ・ボックスの内容を送り出す前に、リストでの2番目の項目に置き換えられる）。変更できるアーギュメントの数値は、初期値として0である。

**,** コンマ (,) は、順番に送られる独立したメッセージとして、メッセージを分割する（例えば、**3, 4, 5**は、最初に3、次に4、最後に5が送られる）。

**;** セミコロン (;) は、**receive**オブジェクトにメッセージを送信する。セミコロンに続く最初の項目は、**receive**オブジェクトの名前を指定する。（次のセミコロンの前までの）残りのメッセージが、その**receive**オブジェクトに送信される。これは、メッセージ・ボックスから出力されない。セミコロン後の最初の項目は、変更できるアーギュメントであっても構わない。従って、入力されるメッセージは、その場でメッセージの送り先を設定することができる。



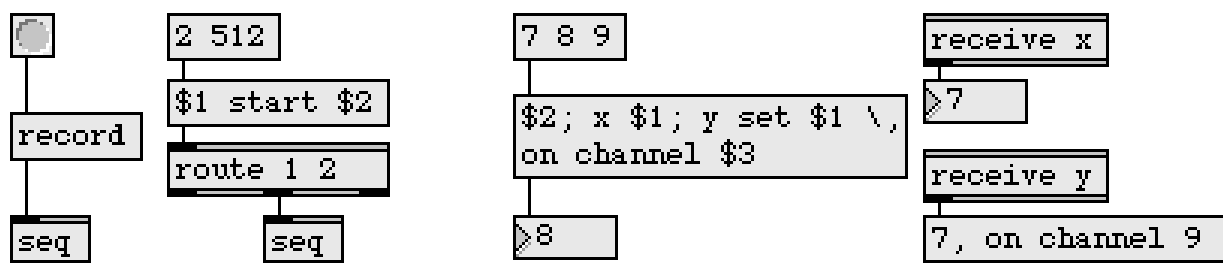


- 1 バックスラッシュ (\) は、特殊文字の特別な役割をを無効にするために使われる。ドル記号、コンマ、セミコロンのすぐ前にバックスラッシュがあれば、その文字は通常の文字として扱われる（例えば、**Note played were C\, E\, and G**）。

## 出力

任意 メッセージ・ボックスの内容は、通常は、アウトレットから出力される。セミコロンがある場合は、残りのメッセージは（または、次のセミコロンまでは）、アウトレットに送られるのではなく、該当する **receive** のオブジェクトに送信される。

## 利用例



単純なメッセージを出力する。また、複雑なメッセージを作り出すことができる

## 参照

**append**  
**prepend**  
**receive**

メッセージの末尾にアーギュメントを付け加える  
メッセージの前に、あるメッセージを付け加える  
パッチ・コードなしでメッセージを受信する

Tutorial 1  
Tutorial 25

Saying “Hello!” (英文PDF参照)  
Managing messages (英文PDF参照)

## 入力

- 整数または実数** 第1インレット：0以外の数値は**metro**オブジェクトをスタートさせる。一定の時間間隔で、**metro**オブジェクトは**bang**をアウトレットから出力する。0は**metro**オブジェクトの動作を止める。
- 第2インレット：この数値はミリ秒で**metro**オブジェクトが**bang**を出力する時間間隔を指定する。第2インレットに新しい数値を受け取っても、次の**bang**が送信されるまで何の効果も生じない。
- bang** 第1インレット：**metro**オブジェクトをスタートさせる。
- stop** 第1インレット：**metro**オブジェクトをストップさせる。
- clock** **setclock**オブジェクトで使われている名前を伴った**clock**メッセージを受け取れば、Max内部のミリ秒クロックではなく、**setclock**オブジェクトにによって**metro**オブジェクトが動作する。**clock**のみのメッセージを受け取った場合は、**metro**オブジェクトはMaxが通常扱うミリ秒・クロックに設定し直す。
- ext** 第1インレット：**ext 1**メッセージで**metro**オブジェクトはOMSタイミング（OMSがインストールされていない時はMIDI Manager）で動作するようになる。これは**metro**オブジェクトと他のOMSを共有したアプリケーションとの同期をとるのに有効である。**ext 0**メッセージは**metro**オブジェクトがMax内部のミリ秒クロックを使用するように設定する。

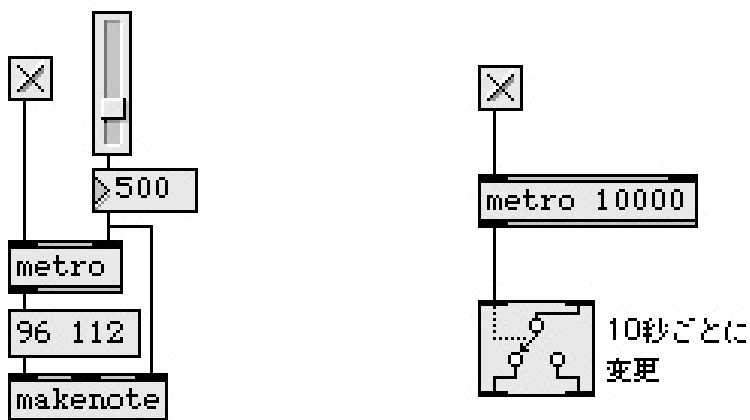
## アーギュメント

- 整数または実数** オプション。最初のアーギュメントは**metro**オブジェクトが**bang**を出力する時間間隔のデフォルト値を設定する。アーギュメントがない場合、時間間隔は5ミリ秒となる。5より小さい数値がアーギュメントとして書き込まれても、5に設定される。2番目のアーギュメントが1の時、**metro**オブジェクトはOMSかMIDI Managerの外部クロック（上記**ext**メッセージを参照）を使用する。2番目のアーギュメントが0であるか、なにも指定されていない時、**metro**オブジェクトはMax内部のミリ秒クロックを使用する。

## 出力

- bang** **metro**オブジェクトが動作している時、**bang**メッセージはただちに出力され、そこから一定の時間間隔で**bang**が出力される。

## 利用例



周期的なメッセージの出力や、処理のトリガーを行う

## 参照

**clocker**  
**counter**  
**setclock**

一定の間隔で経過時間を出力する  
bangメッセージをカウントし、回数を出力する  
タイミング関連のオブジェクトのクロック速度をリモート・コントロールする

**timein**  
**tempo**  
**Uzi**

外部ソースのタイム・コードを出力する  
メトロノームのようなテンポで数値を出力する  
指定した数のbangを出力する

Tutorial 4

Using metro (英文PDF参照)

## 入力

- 整数** **midiflush**オブジェクトは**seq**オブジェクトや**midiin**オブジェクトなどから送られるMIDIデータを受け取る。**midiflush**オブジェクトは受け取ったデータをなにも変えずに出力し、各チャンネルに送られたノート・オン・メッセージに対して、まだノート・オフ・メッセージを出力していないものを監視する。
- bang** **midiflush**オブジェクトが**bang**メッセージを受け取ると、このオブジェクトが作られてから（または最後に**bang**メッセージが送られてから）受け取ったノート・オン・メッセージの中で、まだノート・オフ・メッセージを出力していないものに対して、ノート・オフ・メッセージを出力する。
- clear** ノート・オフを送らずに、**midiflush**オブジェクトが保持しているノート・オンを消去する。

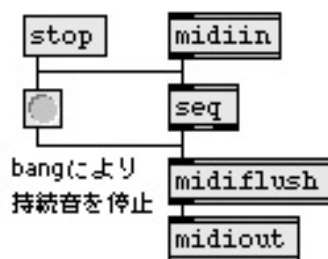
## アーギュメント

なし。

## 出力

- 整数** **midiflush**オブジェクトは入力されたすべての整数をそのまま出力する。そして、受け取ったすべてのノート・オンの中で、まだノート・オフを出力していないものに対して、MIDIノート・オフ・メッセージを送信する。

## 利用例



**bang**を受け取ると、**midiflush** オブジェクトは保持しているノート・オンに対してノート・オフを出力する

## 参照

- flush** 保持しているノートについてノート・オフを作る
- midiin** 生のMIDIデータを受信する
- midiout** 生のMIDIデータを送信する
- seq** MIDIの録音と再生のためのシーケンサー

## 入力

インレットに受け取った数値は、MIDIメッセージのデータとして使われる。データは完全なMIDIメッセージとして（インレットによって決定されたステータス・バイトとともに）フォーマット化され、個々のバイトとしてアウトレットから出力される。

**リスト** 第1インレット：最初の数値をノート・ナンバーの値として、2番目の数値をベロシティの値として、ノート・オンのメッセージを作り出す。

第2インレット：最初の数値をアフタータッチ（プレッシャー）として、2番目の数値をノート・ナンバーとして、ポリフォニック・キーを作り出す。

第3インレット：最初の数値をコントロールの値として、2番目の数値をコントローラー・ナンバーとして、コントロール・メッセージを作り出す。

**整数** 第4インレット：数値はプログラム・チェンジ・メッセージを作り出す。

第5インレット：数値はアフタータッチ（チャンネル・プレッシャー）メッセージを作り出す。

第6インレット：数値はピッチ・ベンド・メッセージを作り出す。

第7インレット：数値はMIDIメッセージのチャンネル・ナンバーとして格納される。実際のステータス・バイトはチャンネルによって異なる。16より大きな数値は、1から16の間の数値として変更される。

**実数** 整数に変換される。

## アーギュメント

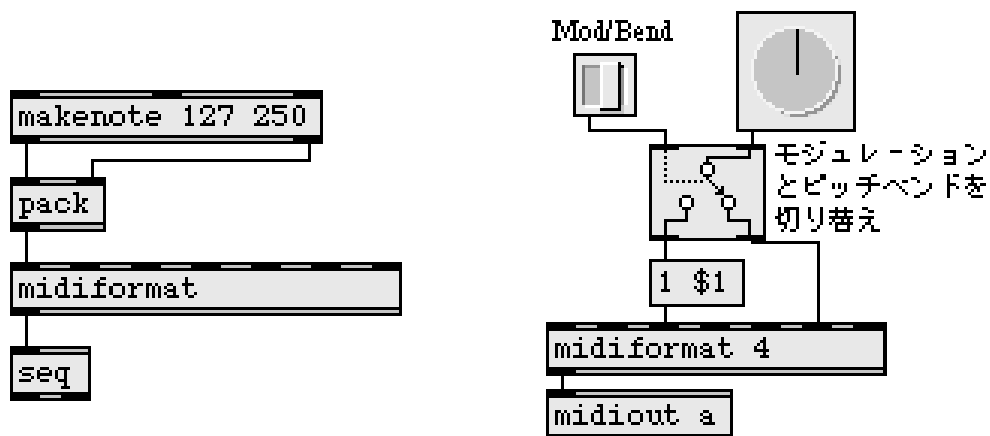
**整数** オプション。MIDIメッセージのチャンネル・ナンバーの初期値を設定する。16より大きな数値は、1から16の間の数値として変更される。アーギュメントがない場合は、チャンネル・ナンバーは初期値で1となる。

**実数** 常に整数に変換される。

## 出力

**整数** MIDIメッセージは、**seq**オブジェクトで記録するため、または**midiout**オブジェクトに送るために、個々のバイトとして出力される。

## 利用例



数値はMIDIメッセージとしてフォーマット化され、個々のバイトとして送り出される

## 参照

**Borax**  
**midiout**  
**midiparse**

ノートオンとノートオフに関する情報を知らせる  
生のMIDIデータを送信する  
生のMIDIデータを解釈する

Tutorial 34

Managing raw MIDI data (英文PDF参照)

## 入力

- (MIDI) **midiiin**はMIDI入力デバイスからの、すべてのMIDIメッセージを受信する。
  
- enable** **enable 0**のメッセージはオブジェクトを使用不能にし、受信するMIDIデータを無視するようになる。**pcontrol**オブジェクトやMIDI Enable/Disableボタンによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを再び受信可能にする。
  
- port** **port**の後に**a**から**z**の文字を続けるか、MIDI入力デバイスの名前を続けることによって、MIDIメッセージを受信するポートを設定する。**port**という単語はオプションで、省略することもできる。
  
- (マウス) **midiiin**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

## アーギュメント

- a - z** オプション。MIDIメッセージを受信するポートを指定する。アーギュメントを指定しない場合、**midiiin**はすべてのポートのすべてのチャンネルから受信する。

## 出力

- 整数** 指定されたポートから受信したMIDIメッセージは、1バイトごとにアウトレットから出力される。

## 利用例



MIDIポートから受信したMIDIメッセージは、**midiiin**オブジェクトから出力される



**参照**

|                  |                                   |
|------------------|-----------------------------------|
| <b>midout</b>    | 生のMIDIデータを送信する                    |
| <b>midiparse</b> | 生のMIDIデータを解釈する                    |
| <b>notein</b>    | MIDIノート・メッセージを出力する                |
| <b>rtin</b>      | MIDIのリアルタイム・メッセージを出力する            |
| <b>sysexin</b>   | MIDIシステム・エクスクルーシブ・メッセージを受信する      |
| <b>xnotein</b>   | ノート・オフ・ベロシティ付きのMIDIノート・メッセージを解釈する |
| <b>xbendin</b>   | MIDIピッチ・ベンド・メッセージ（14ビット）を解釈する     |

Tutorial 34      Managing raw MIDI data（英文PDF参照）

注記：このオブジェクトは、Max 4.1以前のバージョンでは使用できない

## 入力

**整数** 第1インレット：ここに整数が入力されると、**midiinfo**は現在のMIDI出力機器名を含んだメッセージを出力する。これらのメッセージは、**midiinfo**オブジェクトのアウトレットに接続された**umenu**オブジェクトで表示されるポップアップメニューの個々の項目をセットするために使用される。**midiinfo**オブジェクトの第1インレットで受け取られた数値は、**set**メッセージとして送信され、ポップアップメニュー内の該当する項目を選択状態にする。もし数値が負の場合は、**set**メッセージは何も出力されない。

第2インレット：ここに数値が入力されると、**midiinfo**は現在のMIDI入力機器名を含んだメッセージを出力する。これらのメッセージは、**midiinfo**オブジェクトのアウトレットに接続された**umenu**オブジェクトで表示されるポップアップメニューの個々の項目をセットするために使用される。**midiinfo**オブジェクトの第2インレットで受け取られた数値は、**set**メッセージとして送信され、ポップアップメニュー内の該当する項目を選択状態にする。もし数値が負の場合は、**set**メッセージは何も出力されない。

**bang** 第1インレット：整数の場合の同じ動作だが、**umenu**の項目を設定した後に**set**メッセージを送信しない。第2インレットに-1を入力しておくと、**bang**を受け取った時にMIDI入力機器の名前を取得する。

**controllers** 第1インレット：ここに**controllers**メッセージを受け取ると、**midiinfo**は現在のMIDIセットアップ上のすべてのMIDIコントローラー名を含んだメッセージを出力する。これらのメッセージは、**midiinfo**オブジェクトのアウトレットに接続された**umenu**オブジェクトで表示されるポップアップメニューの個々の項目をセットするために使用される。**controllers**の後に数値を入力することにより、ポップアップメニュー生成後に選択される項目を指定することができる。

## アーギュメント

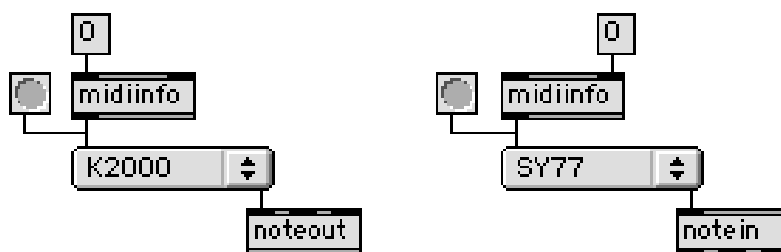
なし。

## 出力

**clear** **midiinfo**は、まず**umenu**オブジェクトから受け取った項目をクリアするために、**clear**メッセージを送信する。

- append** **clear**メッセージを送信するとすぐに、**append**メッセージとしてそれぞれのMIDI入力／出力機器の名前を送信し、接続されている**umenu**オブジェクトに表示する項目をセットする。機器名はMaxのMIDI Setupダイアログ上の順番で表示される。
- set** **midinfo**が受信した整数が0または正の数であった場合、**append**メッセージの後に**set**メッセージが送信され、数値で指定された項目がポップアップメニュー上で選択状態となり表示される。

## 利用例



MIDI出力オブジェクト用に出力機器名をセット    MIDI入力オブジェクト用に入力機器名をセット

## 参照

|               |                          |
|---------------|--------------------------|
| <b>midin</b>  | 生のMIDIデータを受信する           |
| <b>midout</b> | 生のMIDIデータを送信する           |
| <b>umenu</b>  | コマンドを表示し、出力するポップアップ・メニュー |

## 入力

- 整数** 数値は、MIDIメッセージを構成する1バイトとして、指定されたポートから送信される。
- 実数** 整数に変換される。
- リスト** 数値は、MIDIメッセージを構成する個々のバイトとして、指定されたポートから順に送信される。
- enable** **enable 0**メッセージはオブジェクトを使用不可にし、MIDIデータを送信しないように設定する。MIDI Enable/Disableボタンや**pcontrol**オブジェクトによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを送信可能にする。
- port** **port**の後に文字の**a**から**z**の文字を続けるか、MIDI出力デバイスの名前を続けることによって、MIDIメッセージを送信するポートを設定する。**port**という単語はオプションで、省略することもできる。
- (マウス) **midiout**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

## アーギュメント

- a - z** オプション。MIDIデータを送信するポートを指定する。アーギュメントを指定しない場合、**midiout**はポートa、または**MIDI Setup**ダイアログのリストの最初の出力ポートから送信する。

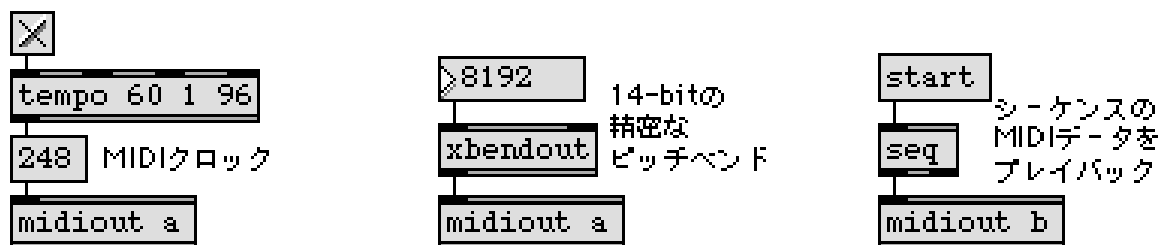
### (MIDI出力デバイス名)

オプション。MIDI出力デバイスの名前は、ポートを指定する最初のアーギュメントとして用いることができる。

## 出力

- (MIDI) アウトレットは存在しない。直接、オブジェクトに指定したMIDI出力ポートから、MIDIメッセージが送信される。

## 利用例



インレットで受信したMIDIバイトは、指定した出力ポートから送信される

## 参照

|                  |                                   |
|------------------|-----------------------------------|
| <b>midformat</b> | MIDIメッセージの形式でデータを用意する             |
| <b>midin</b>     | 生のMIDIデータを受信する                    |
| <b>noteout</b>   | MIDIノート・メッセージを送信する                |
| <b>sxformat</b>  | MIDIシステム・エクスクルーシブ・メッセージを用意する      |
| <b>xbendout</b>  | MIDIピッチ・ベンド・メッセージ（14ビット）を作成する     |
| <b>xnoteout</b>  | ノート・オフ・ベロシティを持つMIDIノート・メッセージを作成する |

Tutorial 34

Managing raw MIDI data（英文PDF参照）

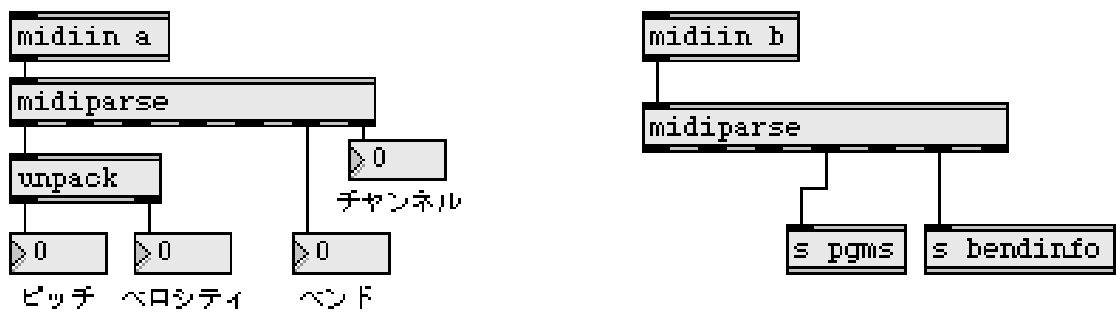
## 入力

- 整数** インレットで受け取る数値は、MIDIメッセージを構成するバイトとみなされ、それは通常、**seq**または**midiiin**オブジェクトから送られる。ステータス・バイトによって、データ・バイトを出力するアウトレットが決定される。
- 実数** 整数に変換される。
- bang** **midiparse**オブジェクトのメモリ中にある、その時点までに入力された部分的なMIDIメッセージを消去する。

## 出力

- リスト** 第1アウトレット：ノート・オン・メッセージ。リストの1番目の数値はノート・ナンバーで、2番目はベロシティの値である。
- 第2アウトレット：ポリフォニック・キー・プレッシャー・メッセージ。リストの1番目の数値はアフター・タッチ（鍵盤を押した時の圧力）の値で、2番目はノート・ナンバーの値（キー・ナンバー）である。
- 第3アウトレット：コントロール・チェンジ・メッセージ。リストの1番目の数値はコントロール・チェンジの値で、2番目はコントロール・ナンバーである。
- 整数** 第4アウトレット：プログラム・チェンジの値。
- 第5アウトレット：アフター・タッチ（チャンネル・プレッシャー）の値。
- 第6アウトレット：ピッチ・ベンドの値。
- 第7アウトレット：MIDIチャンネル・ナンバー。

## 利用例



MIDIデータの意味を翻訳し、異なるタイプごとにデータを振り分ける

## 参照

**Borax**  
**midiformat**  
**midiin**

ノートオンとノートオフに関する情報を知らせる  
MIDIメッセージの形式でデータを用意する  
生のMIDIデータを受信する

Tutorial 34

Managing raw MIDI data (英文PDF参照)



## 入力

- 整数** 第1インレット：入力された数値が、現在**minimum**オブジェクトに格納されている値よりも小さい場合は、入力された値を出力する。そうでない場合は、格納されている値を出力する。
- 第2インレット：この数値は格納され、それより後に第1インレットから入力された値と比較される。
- 実数** アーギュメントが実数でない場合は、整数に変換される。アーギュメントが実数の場合は、すべての数値は実数として扱われる。
- リスト** 第1インレット：リスト内のすべての数値が互いに比較され、そのうち最も小さい値が、アウトレットから出力される。また、リストのうち2番目に小さい値が**minimum**オブジェクトに格納される。**minimum**オブジェクトで扱えるリストの最大要素数は256である。
- bang** 第1インレット：最後に出力した数値を、再びアウトレットから出力する。

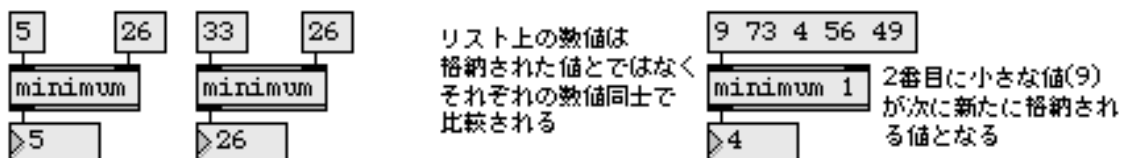
## アーギュメント

- 整数または実数** オプション。第1インレットに受け取った値と比較される初期値を設定する。アーギュメントの値が小数点をもつ場合は、比較に用いられるすべての値は実数として扱われ、出力も実数になる。アーギュメントを設定しない場合は、初期値は0になる。

## 出力

- 整数** **minimum**オブジェクトに格納されている値と第1インレットに受け取った値を比較し（第1インレットにリストを受け取った場合は、リスト内のすべての数値が互いに比較される）、より小さい値をアウトレットから出力する。
- 実数** アーギュメントの値が小数点をもつ場合に限り、実数として出力される。

## 利用例



2つの値のより小さい方、または数値のリストの中で最も小さい値を出力する

## 参照

**maximum**  
**Trough**  
<

数値のリストの中で最も大きい値を出力する  
数値が以前の値より小さければ、それを出力する  
2つの数値を比較し、より小さいことを判断する

## 入力

(キーボード) コンピュータのキーボード上で押さえられたキーが、そのまま **modifiers** オブジェクトに入力される。

**bang** **bang**により現在のモディファイア・キーの状況が出力される。

**interval** **interval**の後ろに数値を続けることにより、モディファイア・キーの状況を監視する間隔をミリ秒単位で指定することができる。0が指定されると、定期的な監視は行われなくなる。

## アーギュメント

**整数** オプション。定期的な監視する間隔をミリ秒単位で指定する。デフォルト値は0（監視を行わない）。

## 出力

**整数** 出力は、コンピュータのキーボード上で何らかのモディファイア・キーが押さえられた時に行われる。モディファイア・キーの状態は、0（押さえられていない）または1（押さえられている）のいずれかで報告される。

第1アウトレット：Shiftキーのオン／オフの状況

第2アウトレット：Caps Lockキーのオン／オフの状況

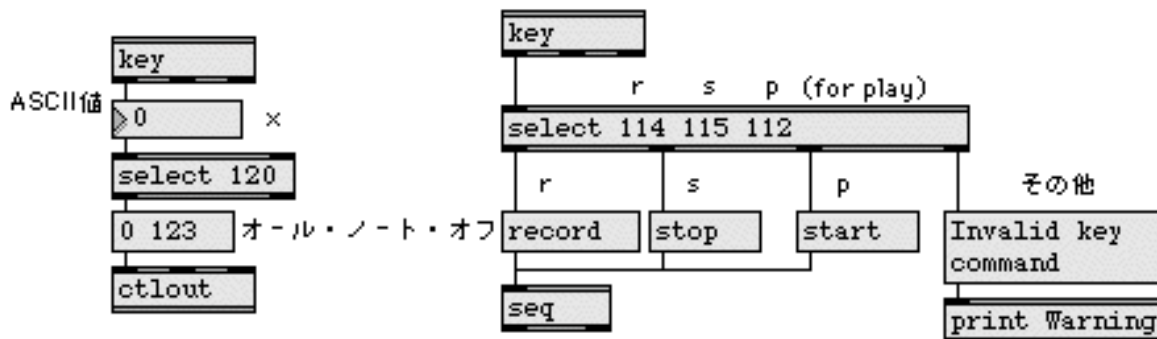
第3アウトレット：Optionキー（Macintosh）あるいはAltキー（Windows）のオン／オフの状況

第4アウトレット：Controlキーのオン／オフの状況

第5アウトレット：Commandキー（Macintosh）あるいはControlキー（Windows）のオン／オフの状況

注記：Macintosh上でのCommandキーはWindows上でのControlキーに相当するため、第4アウトレットと第5アウトレットはWindows上ではどちらもControlキーの状況を報告するが、プラットフォーム間の互換性を保つためには、Windows上では第5アウトレットを使用する必要がある。第4アウトレットは、Macintosh上で作成された（古い）パッチをWindows上でも正しくControlキーの状況を表示するために用意されている。Macintosh上でのControlキーは、通常Windowsでの右クリックに相当する。プラットフォーム間でのキーボードの互換性についての詳細は、Max Tutorials（英文PDF）のfile and key mappingsの項を参照のこと。

## 利用例



モディファイア・キーをタイプすることにより、メッセージをトリガーすることができる

## 参照

- key** コンピューター・キーボードのキー入力を出力する
- keyup** コンピューター・キーボードのキーを離した時の情報を出力する
- numkey** コンピューター・キーボードで入力された数値を解釈する

## 入力

**整数** マウスボタンが押されていない状態の時に、アウトレットから出力される。マウスボタンが押されている間は、数値を無視する。

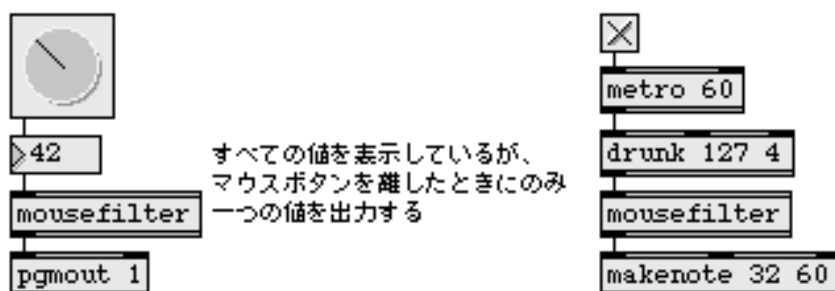
## アーギュメント

なし。

## 出力

**整数** インレットに受け取った数値は、マウスボタンが押されていない状態の時のみ出力される。

## 利用例



マウスボタンが押されている場合には、数値は通過しない

## 参照

**MouseState** マウスの状態や位置を出力する

Tutorial 39 Mouse control (英文PDF参照)

## 入力

- bang** マウスの現在の位置の水平座標と垂直座標を出力する。最後の出力からのマウスの位置の変化量も、同時に出力する。
- poll** マウスが動いた時はいつでも、**bang**を受け取った時と同じように、**MouseState**オブジェクトからマウスの位置と、マウスの位置の変化量を出力する。**poll**メッセージにグラフィック・ウィンドウの名前が続けば、**MouseState**オブジェクトから出力される座標は、グラフィック・ウィンドウを基準とするローカル座標となり、グラフィック・ウィンドウが表示されている間だけ出力する。
- nopoll** **poll**メッセージを取り消し、**MouseState**オブジェクトを、出力する前には**bang**を受け取って出力する通常の状態に戻す。
- zero** **MouseState**オブジェクトがマウスの位置を得る時に原点と見なす位置を再設定する。現在のマウスの位置が、新しい原点とみなされる。
- reset** 原点をデフォルトの設定である画面の左上の隅に戻す。

## アーギュメント

なし。

## 出力

- 整数** **MouseState**オブジェクトが出力を行うためには、そのインレットに**bang**か**poll**メッセージを受け取る必要がある。

第1アウトレット：マウスのボタンが押される度に1が出力される。マウスのボタンが離される度に0が出力される。

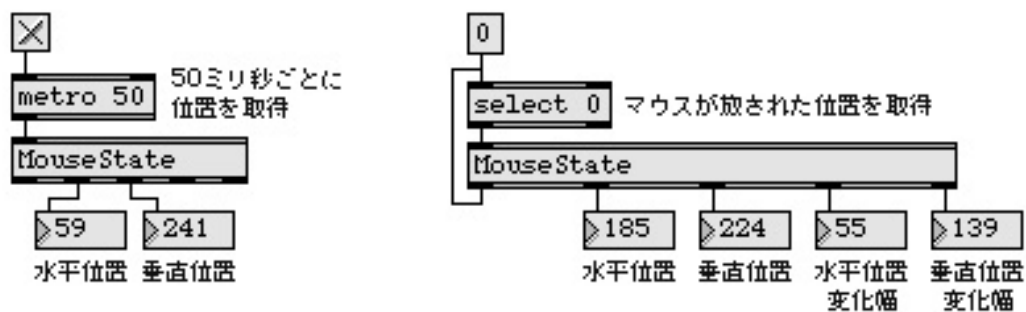
第2アウトレット：マウスが、原点から水平方向に右に何ピクセルの位置にあるかが出力される。

第3アウトレット：マウスが、原点から垂直方向に下に何ピクセルの位置にあるかが出力される。

第4アウトレット：最後にマウスの位置が出力されてからの、マウスの水平方向の位置の変化量が出力される。

第5アウトレット：最後にマウスの位置が出力されてからの、マウスの垂直方向の位置の変化量が出力される。

## 利用例



マウスは、連続した、あるいは不連続の値を出力できる

## 参照

### mousefilter

マウスボタンが押されていない時だけ、数値を通過させる

### Tutorial 39

Mouse control (英文PDF参照)

**movie**オブジェクトと**imovie**オブジェクトはQuickTime機能拡張を必要とする。QuickTimeがインストールされていなければ、このオブジェクトが作られることはない。**movie**オブジェクトはQuickTimeムービーを自分自身のウィンドウで再生する。**imovie**オブジェクトはQuickTimeムービーを、パッチ・ウィンドウ上のボックス内で再生する。

## 入力

以下の**movie**オブジェクトで認識されるすべてのメッセージは、同じように**imovie**オブジェクトでも認識される。

- 整数** ムービーの時間軸上の現在位置を設定する。そのムービーが再生中なら、新しく設定された位置から再生される。**0**は常に最初から再生を始める。終りの時間は1つ1つのムービーによって異なる (**length**メッセージは終りの時間の位置を第1アウトレットから出力する)。
- active** 0でない数値が続く**active**メッセージは、ムービーをアクティブな状態にする (デフォルト)。**0**が続く**active**メッセージは**movie**オブジェクトをアクティブでない状態にする。アクティブでない状態のムービーは再生されず、その再生位置も変わらない。
- autofit** 0でない値が続く**autofit**メッセージは、ムービーを現在表示されているウィンドウにフィットするように伸縮する。
- bang** **resume**メッセージと同じ処理を行う。
- border** **border**には0か1が続き、ムービーの枠線の形状を切り替える。**border 1**メッセージ (デフォルト) は、通常のMacintoshスタイルの枠線をムービー・ウィンドウに用いる。**border 0**メッセージはムービーの再生される矩形だけを表示する。
- clear** アーギュメントのない**dispose**メッセージと同じである。
- dispose** ムービー・ウィンドウが開いていれば、それを閉じ、すべてのムービーを**movie**オブジェクトのメモリから解放する。**dispose**メッセージに、既に読み込まれているムービーの名前が続くなら、その名前で示されたムービーだけを解放する。
- getrate** 現在の再生速度に65535を乗算した値を第2アウトレットから出力する。これに従って、通常の再生速度は**65535**として、半分の再生速度は**32768**として、また2倍の逆再生の速度は**-131072**などとして出力される。そのムービーが再生されていなければその再生速度は**0**と出力され、また、ムービーがまだ読み込まれていなければ何も出力されない。



|                    |                                                                                                                                                          |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>length</b>      | ムービーの時間軸上の最後の位置を出力する。                                                                                                                                    |
| <b>loadintoram</b> | 0でない数値の続く <b>loadintoram</b> メッセージは、ムービーを可能な限り完全にメモリに読み込もうと試みる。デフォルトは0である。                                                                               |
| <b>loop</b>        | 0でない数値の続く <b>loop</b> メッセージは、現在再生されているムービーの繰り返し再生をオンにする。 <b>loop 0</b> メッセージ（デフォルト）は繰り返し再生をオフにする。                                                        |
| <b>loopend</b>     | <b>loopend</b> には1つの数値が続き、ループの終了ポイントを設定する。デフォルト値はそのムービーの終わりと一致する。                                                                                        |
| <b>loopset</b>     | <b>loopset</b> には2つの数値が続き、ループの開始と終了ポイントを設定する。開始のポイントのデフォルト値は0（つまり、フィルムを始め）、終了のポイントのデフォルト値はムービーの終わりの位置である。                                                |
| <b>loopstart</b>   | <b>loopstart</b> には1つの数値が続き、ループの開始のポイントを設定する。デフォルト値は0（つまり、ムービーの始め）である。                                                                                   |
| <b>matrix</b>      | <b>matrix</b> には、9つの実数が続き、現在のムービーをマトリックス（行列）による変型を実行した後に、メモリに読み直す。変型は、QuickTimeによる、1つの座標空間（元の画像）から、もう1つの座標空間（元の画像の拡大縮小、回転、または変換された画像）へのマッピングと同じものが使用される。 |

マトリックスによる変型は9つの行列の要素から成り立っている。

$$\begin{array}{ccc} a & b & u \\ c & d & v \\ t_x & t_y & w \end{array}$$

$u$ と $v$ が0で、 $w$ が1ならば、以下の変型の式に従う。

$$\begin{aligned} x' &= a * x + c * y + t_x; \\ y' &= b * x + d * y + t_y \end{aligned}$$

以下の式は拡大縮小/回転に用いられる。

$$\begin{aligned} a &= xscale * \cos(\theta) \\ b &= yscale * \sin(\theta) \\ c &= xscale * (-\sin(\theta)) \\ d &= yscale * \cos(\theta) \end{aligned}$$

マトリックス変型についての詳しい情報は、以下のAppleのQuickTimeデベロッパードキュメントを参照のこと。

<http://developer.apple.com/techpubs/quicktime/qtdevdocs/INMAC/QT/iqMovieToolbox.c.htm#18006>

- mute** 0でない数値の続く **mute** メッセージは、(サウンドを持つなら) ムービーのサウンドをオフにする。0が続く **mute** メッセージはムービーのサウンドをオンにする (デフォルト)。
- next** **next**には1つの数値が続き、時間軸上の位置をその量だけ先に進める。数値が与えられなかった場合は、**next**メッセージは5だけ時間を先に進める (それらのユニットで意味される実際の時間はムービーごとに異なる)。
- nextmovie** ムービーが再生中であれば、それを止め、現在のムービーの直前に読み込まれたムービーへ切り替える (これらのムービーは読み込まれた順序とは逆の順序で格納されている)。より前のムービーがなければ、**nextmovie**メッセージは最も最近に読み込まれたムービーへと戻る。**nextmovie**メッセージによって現在のムービーが切り替わっても、**movie**オブジェクトのウィンドウのタイトルは自動的に切り替わらないので、注意する必要がある。
- open** ムービー・ウィンドウを最前面に移動させる (**movie**オブジェクトのみで適用され、**imovie**オブジェクトでは適用されない)。
- passive** 0でない数値の続く **passive**メッセージは、パッシブ・モードを設定する。パッシブ・モードでは、ムービーを再生状態にしても、**bang**メッセージを受け取らない限りは、フレームの切り替えは行わない。**passive 0**メッセージ (デフォルト) は**movie**オブジェクトを通常の**start**メッセージに反応するようにする。
- pause** ムービーを停止させる。
- prev** **prev**には1つの数値が続き、時間軸上の位置をその量だけ前へ戻す。数値が与えられなかった場合は、**prev**メッセージは5だけ時間を前へ戻す。
- quality** **quality**には1つの数値が続き、ムービーを描く最小限の間隔をミリ秒単位で設定する。デフォルト値は0 (つまり最小値なし) である。
- rate** **rate**には1つ以上の整数、または実数が続き、ムービーの再生速度を設定する。**rate**メッセージに1つの整数のみが続くなら、この数値はそのまま再生速度となる。また、**rate**メッセージに2つの数値が続くなら、1つ目の数値は分子、2つ目の数値は分母として、元の速度からの比率で指定することになる。**1**は通常速度、**0**はムービーが停止していることを意味する。また、負の値は逆再生である。**rate 1 2**メッセージはムービーを半分の速度で再生する。0でない**rate**メッセージを送ると、ただちにムービーは再生を始める。従って、望む開始位置を指定する

ために、整数を伴う**rate**メッセージを先に送る場合があるかもしれない。

- read** **read**には1つのシンボルが続き、Maxのファイルのサーチ・パス内を、その名前を持つQuickTimeムービー・ファイルを検索する。ファイルが存在していれば、それを開き、そのムービーの最初のフレームをムービー・ウィンドウに表示する。ファイル名が任意の空白文字、また特殊文字を含む場合、名前はシングル・スマート・クオート (option-]とoption-shift-])をキー入力ことで得られる)で囲うか、それぞれの特殊文字をバックスラッシュ (\) に続けなければならない。**read**というメッセージだけの場合は、標準ファイル・ダイアログを開き、選択された任意のムービー・ファイルを開くことになる。**read**メッセージは、少なくともMPEGのようなムービー・ファイル、AIFFやMP3を含むオーディオ・ファイル、また、GIFやJPEGを含む画像ファイルなど、26のQuickTimeで開くことのできる異なるタイプのファイルを開く。
- readany** **readany**メッセージは任意のタイプのファイルを開き、QuickTimeの機能を用いてムービー・ファイルか、またはその他のサポートされているメディア・ファイルとして解釈しようと試みる。
- rect** **rect**には4つの数値が続き、ムービー・ウィンドウ内に表示されるムービーの矩形の大きさを指定する。始めの2つの数値はムービー・ウィンドウ内の矩形の位置を相対座標として指定し、次の2つの数値はその矩形の幅と高さをピクセル単位で指定する。
- resume** ムービーをもっとも最近に設定された再生速度で現在の位置から再生を始める。
- start** ムービーの再生速度を1に設定し、はじめから再生を始める。**start**に読み込まれているムービーを指定する名前が続く場合には、再生を始める前にそのムービーを現在のムービーとして設定する。
- startat** **startat**には1つの数値が続き、ムービーの現在の時間軸上の位置を指定し、そこから再生を始める。
- stop** ムービーを停止させる。
- switch** **switch**には1つのシンボルが続き、その名前のムービーをその再生の状態 (**start**メッセージを参照のこと) を変えずにアクティブにする。
- time** ムービーの現在の時間軸上の位置を出力する。
- title** 現在のムービーの名前をムービー・ウィンドウのタイトルに設定する。これは、**nextmovie**メッセージ (または異なるムービーを指定する**start**メッセージ) との関連で、正確に現在のムービーの名前を**movie**オブ

ジェットのウィンドウに表示させたい場合に必須となる。シンボルの続く**title**メッセージを使うことによって、**movie**オブジェクトのウィンドウに望む任意のタイトルをつけることもできる。

- vol** **vol**には1つの数値が続き、ムービーのサウンドの音量を設定する。1以下の任意の数値は音をミュートする。最大の音量は255である。
- wclose** **movie**オブジェクトのウィンドウを閉じる。
- windowpos** **windowpos**には4つの数値が続き、スクリーン上の**movie**オブジェクトのウィンドウの位置と大きさを指定する。その4つの数値は、それぞれ**movie**オブジェクトのウィンドウの左、上、右、下を絶対座標として指定する。このメッセージは**movie**オブジェクトのみでサポートされ、**imovie**オブジェクトではサポートされていない。

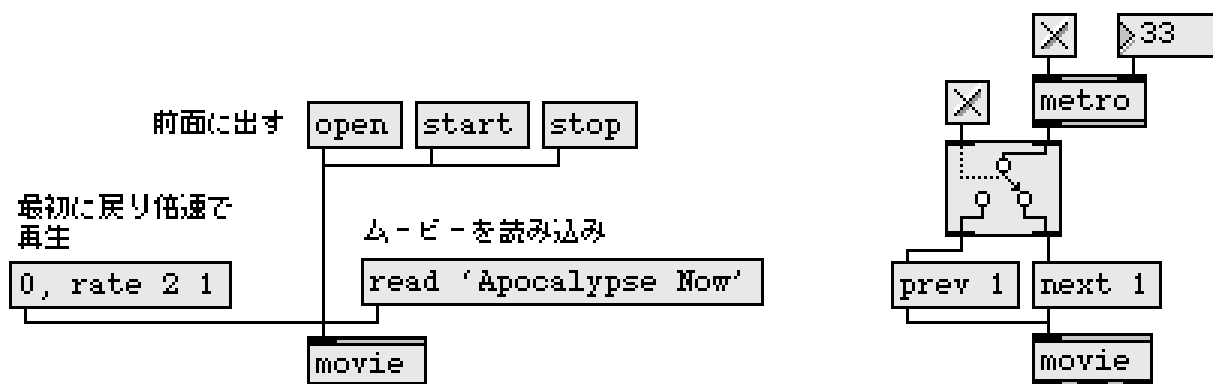
## アーギュメント

- シンボル** オプション。パッチが読み込まれた時に、自動的に**movie**オブジェクトに読み込まれるQuickTimeムービー・ファイルの名前を指定する。**imovie**オブジェクトでは、ロックされていないパッチ内でオブジェクトを選び、Objectメニューから**Get Info...**を選択し、そこでムービー・ファイルを選択することで、同じ効果を得られる。両方のオブジェクトは、そのパッチが保存された時に、読み込まれているムービーの名前を保持し、次回そのパッチが開かれた時に同じムービーを読み込もうと試みる。

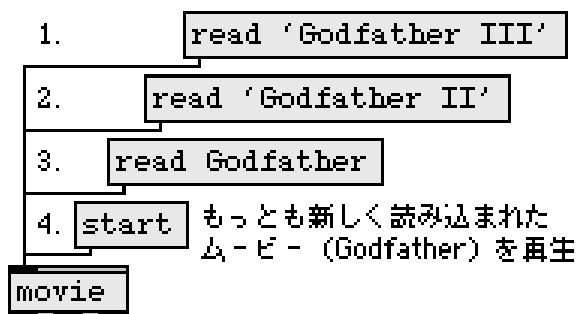
## 出力

- 整数** 第1アウトレット：**time**メッセージを受け取った時に、現在の時間軸上の位置を出力する。また、**length**メッセージを受け取った時に、ムービーの終端の位置を出力する。
- 第2アウトレット：そのムービー上で、マウスでクリックしたり、ドラッグした時に、**movie**オブジェクトのウィンドウかボックスの左上の隅からのマウスの水平方向の相対位置を出力する。
- 第3アウトレット：そのムービー上で、マウスでクリックしたり、ドラッグした時に、**movie**オブジェクトのウィンドウかボックスの左上の隅からのマウスの垂直方向の相対位置を出力する。また、**getrate**メッセージに対して、現在のムービーのレートに65536を乗算した値を第3アウトレットから出力する。

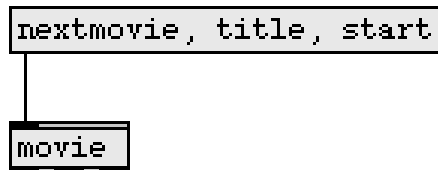
## 利用例



QuickTimeムービーを再生、また様々な方法でそれを進める



GodfatherIIへ飛び、ウィンドウのタイトルを更新して新しいムービーを再生



複数のムービーを保持する (それらは受け取った順番とは逆の順番で格納される)

## 参照

### imovie

パッチ・ウィンドウ内でQuickTimeムービーを再生する

## 入力

- record** 第1インレット：他のインレットに受け取った、すべてのメッセージの記録を開始する。**record**に1つ、または複数のトラック番号が続く場合には、それらのトラックの記録を開始する。
- その他のインレット：そのインレットに対応するトラックの記録を開始する。
- play** 第1インレット：以前に記録されたすべてのメッセージの再生を開始し、対応するアウトレットから記録された時と同じリズム、同じスピードで出力する。**play**に1つ、または複数のトラック番号が続く場合には、それらのトラックの再生を開始する。
- その他のインレット：そのインレットに対応するトラックの再生を開始する。
- stop** 第1インレット：**mtr**オブジェクトが再生、記録をしている時には、それを停止させる。**stop**に1つ、または複数のトラック番号が続く場合には、それらのトラックを停止させる。
- その他のインレット：そのインレットに対応するトラックを停止させる。
- next** 第1インレット：それぞれのトラックで、記録されたシーケンスの次のメッセージだけを出力させる。**next**メッセージを受け取った時には、トラック番号、またメッセージそれぞれのデルタ・タイム（次のイベントが起こるまでの時間間隔）を第1アウトレットからリストとして出力する。**next**に1つ、または複数のトラック番号が続く場合には、それらのトラックに記録されている次のメッセージを出力する。
- その他のインレット：そのインレットに対応したトラック上に記録されている次のメッセージを出力する。
- rewind** 第1インレット：**mtr**オブジェクトを記録されたシーケンスの先頭に再設定する。このコマンドは**next**メッセージでメッセージを出力し終えた時に、シーケンスの先頭に戻るのに使われる。再生中や記録中にシーケンスの先頭に戻ることは、**play**メッセージ、または**record**メッセージを繰り返すことになる。**mtr**オブジェクトが再生や記録している時には、**stop**メッセージを**rewind**メッセージに先行させるべきである。**rewind**に1つ、または複数のトラック番号が続く場合には、それらのトラックを先頭に戻す。
- その他のインレット：そのインレットに対応したトラックの先頭にポインターを戻す。



- mute** 第1インレット：再生を続けながら（シーケンスを前に進めながら）、**mtr**オブジェクトの出力を停止させる。**mute**に1つ、または複数のトラック番号が続く場合には、それらのトラックをミュートする。
- その他のインレット：そのインレットに対応したトラックをミュートする。
- delay** 第1インレット：**delay**にはミリ秒単位での数値が続き、それぞれのトラックのはじめのデルタ・タイムの値をその数値に設定する。**play**メッセージを受け取った後に、その時間に従って、すべてのトラックの再生を始める。
- その他のインレット：そのインレットに対応したトラックの初期のデルタ・タイムを設定する。
- first** 第1インレット：**first**にはミリ秒単位の数値が続き、**mtr**オブジェクトを**play**メッセージを受け取った後に、その時間の量だけ再生前に待機させる。**delay**メッセージと違うのは、**first**メッセージはそのトラックのはじめのイベントのデルタ・タイムの値を変化させないことであり、これはただ最初から再生させる前に（はじめのデルタ・タイムに加えて）一定時間待機させるだけである。
- write** 第1インレット：標準ファイル保存ダイアログを呼び出し、**mtr**オブジェクトの内容を独立したファイルとして保存する。**mtr**オブジェクトの内容を保存する唯一の方法は**write**メッセージを使うことであり、このオブジェクトの内容はパッチ・ファイルに埋め込みされない。
- その他のインレット：そのインレットに対応したトラックのみを含んだファイルを書き出す。
- read** 第1インレット：標準ファイル・ダイアログを呼び出し、以前に保存されたファイルを**mtr**に読み込むことができる。
- その他のインレット：そのインレットに対応したトラックのみを含むファイルを開く。
- 整数** 第1インレット以外の任意のインレット：そのトラックが記録状態であれば、そのトラックのインレットに受け取った数値は、デルタ・タイム（以前のイベントから経過したミリ秒単位の数値）とともに、**mtr**オブジェクトに格納される。
- リスト** 第1インレット以外の任意のインレット：そのトラックが記録状態であれば、そのトラックのインレットに受け取ったリストは、デルタ・タイムとともに、**mtr**オブジェクトに格納される。
- 任意のシンボル** 第1インレット以外の任意のインレット：そのトラックが記録状態であれば、そのトラックのインレットに受け取ったシンボルは、デル



タ・タイムとともに、**mtr**オブジェクトに格納される。

また、**mtr**オブジェクトは**midin**オブジェクトから受け取ったMIDIメッセージのバイトを個々に記録できるが、それぞれのバイトは独立したデルタ・タイムで格納され、**seq**オブジェクトのようにMIDIメッセージの形式とはならない。完全なMIDIメッセージを記録し、後で編集しようと思うなら、**seq**オブジェクトの方が適している。一方では、**mtr**オブジェクトは、ほとんどのMaxオブジェクト（**notein**オブジェクトや**pgmin**オブジェクトのような特定のMIDIオブジェクト、**number box**オブジェクト、**slider**オブジェクト、また**dial**オブジェクトのようなユーザー・インターフェース・オブジェクト、またその他のオブジェクト）からの数値、リスト、そしてシンボルには、十分に活用することができる。

再生のために**mtr**オブジェクトに読み込ませるファイルは、適切な形式になっていなければいけない。さらに、1つの**mtr**オブジェクトの複数のトラックのシーケンスは、特定の形式に従ったテキスト・ファイルとして入力することができる。異なったトラックの内容は順番にファイル内に列挙され、また、それぞれのトラックの形式は以下になる。それぞれの列はセミコロン (;) で終わることに注意が必要である。

1行目: **track** <トラック番号>; (続くデータが格納されるトラック)  
2行目以降: <デルタ・タイム> <メッセージ>;  
最終行: **end**; (トラックのデータの終わり)

**clear** 第1インレット: **mtr**オブジェクトの内容を消去する。1つ、また複数のトラック番号が続く**clear**メッセージは、それらのトラックを消去する。

その他のインレット: そのインレットに対応したトラックを消去する。

**unmute** 第1インレット: これまでに受け取った**mute**メッセージを取り消す。1つ、また複数のトラック番号が続く**unmute**メッセージは、それらのトラックのミュートを取り消す。

その他のインレット: そのインレットに対応するトラックのミュートを取り消す。

## アーギュメント

**整数** オプション。**mtr**オブジェクトのトラック数を指定する。トラック数は、第1インレットや第1アウトレットに加わる形で、インレットやアウトレットの数を決定する。上限は32トラックである。アーギュメントが設定されなければ、1つのトラックのみとなる。

## 出力

**任意** すべてのアウトレット：第1インレットに**play**メッセージを受け取った時に、それぞれのトラックに格納されたメッセージは、そのトラックのアウトレットから、記録された時と同じリズム、また同じスピードで出力される。個々のトラックのインレットに受け取った**play**メッセージは、その特定のトラックを再生する。

第1インレットに**next**メッセージを受け取った時は、それぞれのトラックの次のメッセージが、それらに対応するアウトレットから出力される。個々のトラックのインレットに受け取った**next**メッセージは、そのトラックの次のメッセージを出力する。

**リスト** 第1アウトレット：**next**メッセージに反応し、値が出力される時は常に、その出力される値のトラック番号とデルタ・タイムが、2つの項目を持つリストとして第1アウトレットから出力される。

## 利用例



MIDIデータ、またはその他のイベントを記録する

## 参照

- hslider** 画面上のスライダーを動かすことによって数値を出力する
- multislider** 複数のスライダーとスクロール表示
- seq** MIDIの録音と再生のためのシーケンサー
- timeline** Maxメッセージを時間軸で扱うスコア
- rslider** 数値の範囲を変更し、それを表示する
- uslider** 画面上でスライダーを動かすことによって数値を出力する

- Tutorial 12 Sliders and dials (英文PDF参照)
- Tutorial 36 Multi-track sequencing (英文PDF参照)



## 入力

- 整数** すべてのスライダーの値と位置を、受け取った数値に設定し、その値を反映したリストを出力する。**multiSlider**オブジェクトのデータ型が実数に設定されている場合には、入力される値は実数に変換される。
- 実数** すべてのスライダーの値と位置を、受け取った数値に設定し、その値を反映したリストを出力する。**multiSlider**オブジェクトのデータ型が整数に設定されている場合には、入力される値は四捨五入され整数に変換される。
- リスト** リストの1つ目の値を1つ目のスライダーに設定するように、左から右へとリストの値に対応させて、それぞれのスライダーを設定する。**multiSlider**オブジェクトが、リスト項目数以上の数のスライダーを持つ場合には、スライダーの数はリストの項目数へと変更される。その場合、**multiSlider**オブジェクト自体の外側の大きさは変更されず、個々のスライダーの幅、また高さが変更される。
- bang** 現在のスライダーの値をリストとして出力する。
- border** **border**には1つの整数が続き、**multiSlider**オブジェクトにその外側の枠線の描画方法を指定する。この機能は、複数の**multiSlider**オブジェクトを隣り合わせて置く時に有用である。**border**メッセージのアーギュメントは、以下に示す通りである。
- |                 |           |
|-----------------|-----------|
| <b>border 0</b> | 枠線を描画しない  |
| <b>border 1</b> | 左の枠線を描画する |
| <b>border 2</b> | 右の枠線を描画する |
| <b>border 4</b> | 上の枠線を描画する |
| <b>border 8</b> | 下の枠線を描画する |
- 枠線の任意の組み合わせは、これらの値を加算することによって描画される。例えば、**border 15**メッセージはすべての枠線を描画する。
- brgb** **brgb**には0から255の範囲の3つの数値が続き、**multiSlider**オブジェクトの背景色のRGB値を設定する。デフォルト値は白 (**brgb 255 255 255**)である。
- cont\_data** スクロール表示しない状態でのcontinuous outputモードのオン・オフを切り替える。このモードがオンになっている場合は、その**multiSlider**オブジェクトは、ドラッグ操作中にマウスが動く度に、現在のスライダーの値のリストを出力する。また、このモードがオフになっている場合は、その**multiSlider**オブジェクトは、マウスのボタンが押された時、またそれが離された時にだけリストを出力する。



- displayonly** display onlyモードのオン・オフを切り替える。display onlyモードがオンになっている場合は、**multiSlider**オブジェクトは表示を変えるようなユーザーの操作を禁止する。デフォルトはオフ (0) である。
- echo** echoモードのオン・オフを切り替える。echoモードがオンの場合は、その**multiSlider**オブジェクトは、インレットに任意のリストを受け取った時に出力をするようになる。デフォルトはオフ (0) である。
- fetch** **fetch**には1つの数値が続き、その番号のスライダーの値を第3アウトレットから出力する。
- frgb** **frgb**には0から255の範囲の3つの数値が続き、**multiSlider**オブジェクトのスライダーのRGB値を設定する。デフォルト値は黒 (**frgb 0 0 0**) である。
- interpolate** interpolationモードのオン・オフを切り替える。interpolationモードがオンの場合は、その**multiSlider**オブジェクトはスライダーが動かされた時に、それぞれのスライダーの補完された値を出力する。
- limit** **limit**には2つの実数が続き、**multiSlider**オブジェクトの上限と下限の値を設定する。デフォルト値は-1.0と1.0である。上下の限界値は、**limit\_lo**メッセージと**limit\_hi**メッセージを使うことでも、個々に設定できる。
- limit\_hi** **limit\_hi**には実数が続き、**multiSlider**オブジェクトの上限値を設定する。デフォルト値は1.0である。
- limit\_lo** **limit\_lo**には実数が続き、**multiSlider**オブジェクトの下限値を設定する。デフォルト値は-1.0である。
- max** すべてのスライダーを最大値に設定する。
- maximum** **max**メッセージと同じである。
- min** すべてのスライダーを最小値に設定する。
- minimum** **min**メッセージと同じである。
- (マウス) **multiSlider**オブジェクトがマウスに反応するかどうかは、設定されている表示形式 (以下のアーギュメントの項を参照のこと) によって決定される。**multiSlider**オブジェクトは、その表示形式がスクロールしないモード (Thin LineかBar) の時か、逆方向へのスクロールのモード (Rev pointかRev Line) の時には、マウスのクリックに反応する。正方向へスクロール表示するように設定された**multiSlider**オブジェク



ト (Point ScrollかLine Scroll) をクリックすることは、何の効果も生じない。

表示形式がスクロールしないように設定されている場合に、**multiSlider**オブジェクト内の任意のスライダーをクリックすれば、直ちにクリックされた位置へスライダーを変化させ、その時のすべてのスライダーの値を出力する。1つの**multiSlider**オブジェクト上を横断しながらドラッグすれば、同じように他のスライダーを設定することになる。continuous outputモードが有効になっている場合には、現在の値のリストは、ドラッグ中にマウスを動かす度に出力される。また、continuous outputモードがオフになっている場合には、そのリストはマウスのボタンが押された時や離された時のみ出力される。

表示形式が逆方向へのスクロールになっている時、出力は同じように行われるが、表示が連続的に更新され、過去の**multiSlider**オブジェクトの状態を表示する。この表示の連続的な更新は、必ずしも**multiSlider**オブジェクトによる連続的な出力を意味しない。それは**multiSlider**オブジェクトのインスペクターで、Continuous Outputオプションのチェックによって決まる。

- peak\_hold** peak holdモードのオン・オフを切り替える。peak holdモードがオンの場合は、それぞれのスライダーのピークの値が細い線によって表示される。
- peak\_reset** 現在のピークの値を現在のスライダーの値へと再設定する。
- quantiles** 第1インレット : **quantiles**には実数のリストが続き、リストの要素を**multiSlider**オブジェクトのすべての値の合計と乗算する。この結果は、次に215 (32768) で除算する。そして、**multiSlider**オブジェクトはそのアドレスまでのすべての値の合計が、リストの要素それぞれについての結果より大きいか等しい時に、そのアドレスを出力する。
- select** スライダーの値を選択的に設定する。例えば**select 1 30 2 4 5 50**というメッセージは、最初のスライダーを30に設定し、2つ目を4に、そして5つ目のスライダーを50に設定する (最も上のスライダー、また最も左のスライダーの番号は、常に1である)。
- set** **set**にはスライダーの番号と値が続き、出力を行わずに、その番号のスライダーの値を設定する。
- setborder** 1つの整数が続く**setborder**メッセージは、**border**メッセージとほとんど同じであるが、このメッセージは**multiSlider**オブジェクトにその外側の枠線を描画方法を指定することに加え、**multiSlider**オブジェクトのインスペクター (Objectメニューの**Get Info**...を選択して開く) 内の





枠線描画のモードを設定する。**setborder**メッセージのアーギュメントは、以下に示す通りである。

|                    |           |
|--------------------|-----------|
| <b>setborder 0</b> | 枠線を描画しない  |
| <b>setborder 1</b> | 左の枠線を描画する |
| <b>setborder 2</b> | 右の枠線を描画する |
| <b>setborder 4</b> | 上の枠線を描画する |
| <b>setborder 8</b> | 下の枠線を描画する |

**border**メッセージと同じように、枠線の任意の組み合わせは、それらの値を加算することで描画できる。例えば、**setborder 15**メッセージはすべての枠線を描画する。デフォルトではすべての枠線が描画されている (**setborder 15**)。

**setcontdata** 0か1が続く**setcontdata**メッセージは、**cont\_data**メッセージとほとんど同じであるが、表示形式を設定することに加え、**multiSlider**オブジェクトのインスペクター (Objectメニューの**Get Info**...を選択して開く) 内のcontinuous outputモードのオン・オフを切り替える。デフォルトはオフである (**setcontdata 0**)。

**setminmax** **setminmax**には2つの実数が続き、**multiSlider**オブジェクトの最小値、また最大値を設定する。デフォルト値は-1.0と1.0である。

**setpeakhold** **setpeakhold**メッセージは、**peak\_hold**メッセージとほとんど同じであるが、これは表示形式を設定することに加え、**multiSlider**オブジェクトのインスペクター (Objectメニューの**Get Info**...を選択して開く) 内のpeak holdモードのオン・またはオフを切り替える。デフォルトはオフである (**setpeakhold 0**)。

**setstyle** **setstyle**には0から5まで整数が続き、**multiSlider**オブジェクトの表示形式を設定する。デフォルトはThin Line (**setstyle 0**) である。表示形式の値は、以下に示す通りである。

|                   |                                      |
|-------------------|--------------------------------------|
| <b>setstyle 0</b> | Thin Line (細い線)                      |
| <b>setstyle 1</b> | Bar (棒状)                             |
| <b>setstyle 2</b> | Point Scroll (点によるスクロール)             |
| <b>setstyle 3</b> | Line Scroll (線によるスクロール)              |
| <b>setstyle 4</b> | Reverse Point Scroll (点による逆方向のスクロール) |
| <b>setstyle 5</b> | Reverse Line Scroll (線によるの逆方向のスクロール) |

その表示形式がThin LineかBarに設定されている場合は、それぞれのスライダーはその現在の値を細い線で表示する。その他の (スクロールする) 表示形式のうち1つが選択されている場合は、それぞれのスライダーは現在または最も最近の過去の値を連続的にスクロールする



表示になる（表示される過去の値の数は、**multiSlider**オブジェクトの表示領域のピクセル数によって決定される）。

注記：スクロール表示に設定された**multiSlider**オブジェクトは、値を受け取った時間間隔で更新できないことがある。これは、結果として、若干のデータのポイントが表示されないことになる。

**setype** 0か1が続く**setype**メッセージは、**multiSlider**オブジェクトのインスペクター（Objectメニューの**Get Info...**を選択して開く）で、そのオブジェクトを整数（0）か実数（1）か、どちらを扱うかを設定し現在選択されているオブジェクトの振る舞いを設定する。デフォルト値は実数である（**setype 0**）。

**sum** 現在のすべてのスライダーの値の合計を実数として出力する。

## インスペクター

**multiSlider**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**multiSlider**オブジェクトを選択することで、**multiSlider**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**multiSlider**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**multiSlider**オブジェクトのインスペクターでは、以下の属性を設定できる。

- *Slider Range Minimum*および*Maximum*の値。Min.のデフォルト値は-1。Max.のデフォルト値は1である。
- *Number of Sliders*。デフォルト値は1である。また、スライダーが整数か、実数かも選択できる。デフォルトは実数である。
- *Slider Style*では、Thin line、Bar、Point Scroll、Line Scroll、Reverse Point Scroll、Reverse Line Scrollから選択する。表示形式がThin Line（デフォルト）かBarに設定されている時は、それぞれのスライダーは、その現在の値を細い線で表示する。また、その他の（スクロールする）表示が選択されていれば、それぞれのスライダーは、現在または最も最近の過去の値を連続的にスクロールする表示になる（表示される過去の値の数は、**multiSlider**オブジェクトの表示領域のピクセル数によって決定される）。また、Continuous Data OutputモードとPeak Hold表示モードも選択できる（デフォルトではどちらのモードもオフである）。





- *Orientation*は、データを水平に表示するか、垂直（デフォルト）に表示するかを選択する。
- *Draw Borders*チェック・ボックスでは、**multiSlider**オブジェクトの四方の枠線を描画を指定する。
- *Color*オプションは、**swatch**オブジェクトのカラー・ピッカーか、RGB値を用いることで、**multiSlider**オブジェクトの色と背景の色を設定する。スライダーのデフォルトの色は0 0 0、またデフォルトの背景の色は255 255 255である。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

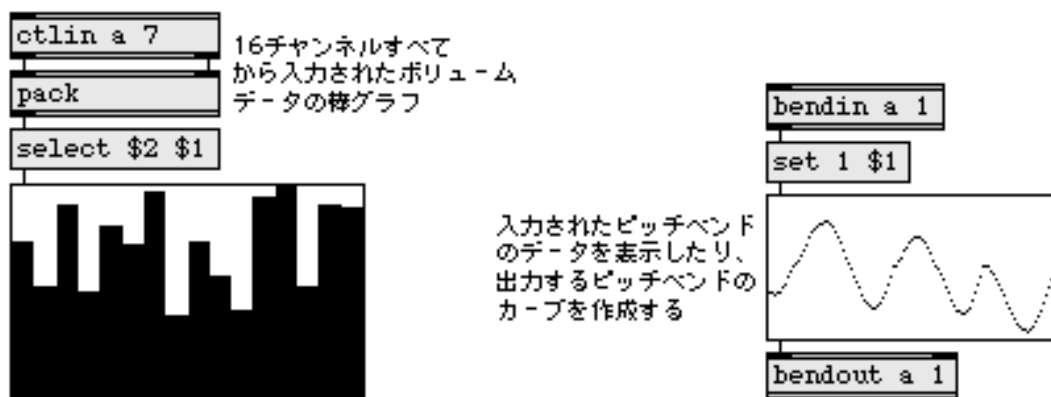
## アーギュメント

なし。

## 出力

- リスト **multiSlider**オブジェクトがリスト、整数、または実数をインレットに受け取った時に、現在の値のリストを出力する。スライダーがマウスによって変更された時にも、リストが出力される。

## 利用例



multiSlider オブジェクトの描画形式



## 参照

|                   |                            |
|-------------------|----------------------------|
| <b>dial</b>       | ダイアル操作によって数値を出力する          |
| <b>hslider</b>    | 画面上のスライダーを動かすことによって数値を出力する |
| <b>kslider</b>    | 画面上のキーボードから数値を出力する         |
| <b>matrixctrl</b> | マトリックス状のスイッチ・コントロール        |
| <b>pictctrl</b>   | ピクチャーを用いたコントロール            |
| <b>rslider</b>    | 数値の範囲を変更し、それを表示する          |
| <b>slider</b>     | 画面上でスライダーを動かすことによって数値を出力する |
| <b>uslider</b>    | 画面上でスライダーを動かすことによって数値を出力する |

Tutorial 12      Slider and dials (英文PDF参照)

## 入力

**任意のメッセージ** メッセージは、それらが同じ論理的イベントの一部であるかどうか判定するためにテストされる。論理的イベントとは、以下のうちの1つである。マウス・クリック、マウス・ドラッグの進行中の監視、スケジューラーによって発生するイベント（例えば**metro**から出力される**bang**）、MIDIイベント、キーボード・イベント。**next**は、現在のメッセージが先に受け取ったメッセージと同じイベントであるかどうか判定する。例えば、2回**bang**をクリックするならば、2つの**bang**は同じ論理的イベントではない。しかし、**bang, bang**というメッセージ・ボックスを用いた場合や、連続して2回**bang**を出力する**Uzi**オブジェクトを使用した場合は、これらの**bang**は同じ論理的イベントとなる。

## アーギュメント

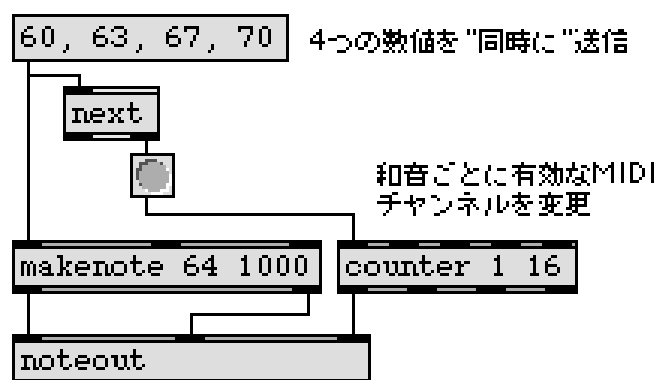
なし。

## 出力

**bang** 第1アウトレット：現在のメッセージが、先に受け取ったメッセージと同じ論理的イベントでない場合、**bang**を出力する。

第2アウトレット：現在のメッセージが、先に受け取ったメッセージと同じ論理的イベントの場合、**bang**を出力する。

## 利用例



**next**は、異なるMaxメッセージが同じ論理的イベントとして起こっているかを検出する

## 参照

**Uzi** 指定した数の**bang**を出力する  
**defer** メッセージの優先を解除する  
**delay** **bang**を遅らせて出力する

## 入力

- (MIDI) MIDI入力デバイスから送信されたMIDIノート・オン・メッセージまたはノート・オフ・メッセージを受信する。
- enable** **enable 0**のメッセージはオブジェクトを使用不能にし、受信するMIDIデータを無視するようになる。**pcontrol**オブジェクトやMIDIEnable/Disableボタンによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを再び受信可能にする。
- port** **port**の後に**a**から**z**の文字を続けるか、MIDI入力デバイスの名前を続けることによって、ノート・メッセージを受信するポートを設定する。**port**という単語はオプションで、省略することもできる。
- (マウス) **notein**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

## アーギュメント

- a - z** オプション。ノート・メッセージを受信するポートを指定する。アーギュメントを指定しない場合、**notein**はすべてのポートのすべてのチャンネルから受信する。

### (MIDI入力デバイス名)

オプション。入力ポートを指定するために、第1アーギュメントにMIDI入力デバイスの名前を使うことができる。

- a - zと整数** 空白文字で区切られた文字と数値の組み合わせは、ノート・メッセージを受信するポートとMIDIチャンネルを指定する。チャンネル・ナンバーが16より大きい場合は、1から16の範囲に収まるように変換される。

- 整数** 文字と数値の組み合わせの代わりに、数値だけを指定しても良い。チャンネル・ナンバーのアーギュメントの正確な意味は、*MIDI Setup*ダイアログで指定したポートごとのチャンネル・オフセットによって決まる。

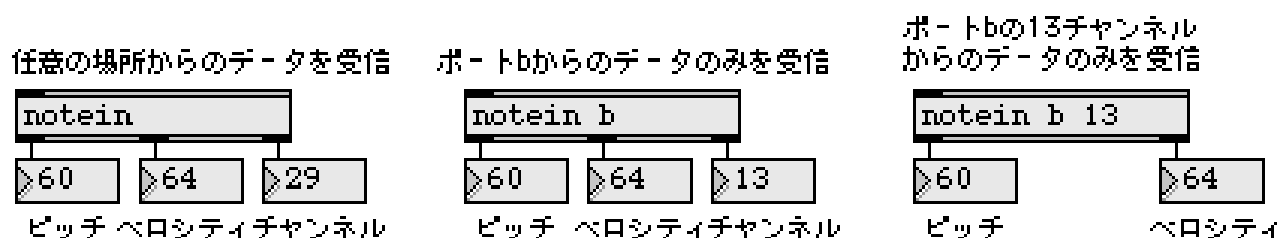
## 出力

- 整数** 第1アウトレット：受信したノート・メッセージのノート・ナンバーの値。
- 第2アウトレット：0はノート・オフ・メッセージを表し、0以外は受

受信したノート・オン・メッセージのベロシティ値を表す。リリース・ベロシティを受信するためには**xnotein**を使用する。

チャンネル・ナンバーをアーギュメントに指定した場合は、**notein**は2つのアウトレットのみを持つ。チャンネル・ナンバーを指定しない場合は、**notein**は一番右に第3アウトレットを持つ。これは受信したノート・メッセージのチャンネル・ナンバーを出力する。

## 利用例



ノート・オン・メッセージは、どこでも受信することができ、ポートのみを指定するか、ポートとチャンネルを指定することができる

## 参照

|                |                                   |
|----------------|-----------------------------------|
| <b>ctlin</b>   | MIDIコントロール・チェンジを受信する              |
| <b>midin</b>   | 生のMIDIデータを受信する                    |
| <b>noteout</b> | MIDIノート・メッセージを送信する                |
| <b>rtin</b>    | MIDIのリアルタイム・メッセージを出力する            |
| <b>xbendin</b> | MIDIピッチ・ベンド・メッセージ (14ビット) を解釈する   |
| <b>xnotein</b> | ノート・オフ・ベロシティ付きのMIDIノート・メッセージを解釈する |

Tutorial 12

Sending and receiving MIDI notes (英文PDF参照)

## 入力

**整数** 第1インレット：数値は、MIDIノート・メッセージのノート・ナンバーの値として扱われ、指定されたチャンネルとポートから送信される。数値は0から127の間に制限される。

第2インレット：数値はノート・メッセージのベロシティとして格納され、第1インレットで受け取られるノート・ナンバーとともに使用する。数値は0から127の間に制限される。ベロシティが0であれば、ノート・オフ・メッセージとして、ベロシティが1から127までであれば、ノート・オン・メッセージとして扱われる。

第3インレット：ノート・オン・メッセージを送信するチャンネル・ナンバーとして格納される。

**実数** 整数に変換される。

**リスト** リストの1番目の数値はノート・ナンバーとして、2番目はベロシティとして、3番目はチャンネル・ナンバーとして扱われ、MIDIノート・メッセージを送信する。

**enable** **enable 0**メッセージはオブジェクトを使用不可にし、MIDIデータを送信しないように設定する。MIDI Enable/Disableボタンや**pcontrol**オブジェクトによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを送信可能にする。

**port** 第1インレット：**port**の後に文字の**a**から**z**の文字を続けるか、MIDI出力デバイスの名前を続けることによって、ノート・メッセージを送信するポートを設定する。**port**という単語はオプションで、省略することもできる。

(マウス) **noteout**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

## アーギュメント

**a - z** オプション。ノート・メッセージを送信するポートを指定する。文字のアーギュメントを指定している時に、チャンネル・ナンバーが16より大きければ、1から16の範囲に収まるように変換される。アーギュメントが存在しない場合は、**noteout**は初期状態として、ポートaからMIDIチャンネル1として送信する。

**a-zと整数** 空白で区切られた文字と数値の組み合わせによって、送信するノート・メッセージのポートとチャンネル・ナンバーを指定する。チャンネル・ナンバーが16より大きければ、1から16の範囲に収まるように変換される。

### (MIDI出力デバイス名)

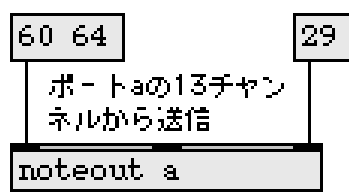
オプション。MIDI出力デバイスの名前は、ポートを指定する最初のアーギュメントとして用いることができる。

**整数** 文字と数値の組み合わせの代わりに、数値だけを指定しても良い。チャンネル・ナンバーのアーギュメントの正確な意味は、*MIDI Setup*ダイアログで指定したポートごとのチャンネル・オフセットによって決まる。

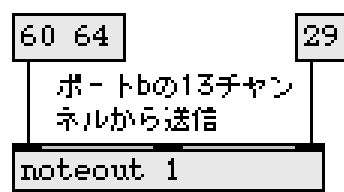
## 出力

(MIDI) アウトレットは存在しない。オブジェクトに指定したMIDI出力ポートから、MIDIノート・メッセージが送信される。

## 利用例



文字のアーギュメントは、送信ポートを限定する



または、数値によってポートとチャンネルを指定する

## 参照

|                 |                                   |
|-----------------|-----------------------------------|
| <b>ctlout</b>   | MIDIコントロール・チェンジを送信する              |
| <b>midout</b>   | 生のMIDIデータを送信する                    |
| <b>notein</b>   | MIDIノート・メッセージを出力する                |
| <b>xbendout</b> | MIDIピッチ・ベンド・メッセージ (14ビット) を作成する   |
| <b>xnoteout</b> | ノート・オフ・ベロシティを持つMIDIノート・メッセージを作成する |

Tutorial 12 Sending and receiving MIDI notes (英文PDF参照)





## 入力

**整数、または実数** インレットに受け取った数値は格納され、**number box**オブジェクト上に表示され、また、アウトレットから出力される。実数は、整数の**number box**オブジェクトでは整数に変換される。逆の場合は、その逆となる。

アクティブになっているパッチ・ウィンドウがロックされている時は、オブジェクト上でマウスをクリックし、コンピューターのキーボードから数値を入力することで、数値を**number box**オブジェクトに入力できる。returnキー(Macintosh)またはenterキー(Macintosh、Windows)を押すことや、**number box**オブジェクト外をクリックすることで、数値をアウトレットから出力する。

マウスを用いて（パッチ・ウィンドウがロックされている時に）**number box**オブジェクト上で上下にドラッグすることで、表示されている値を上下に動かし、新しい値を連続的に出力する。実数の**number box**オブジェクトは、小数点の左側をドラッグすれば、値を1ずつ変化させる。小数点の右側をドラッグすれば、その数値の端数を0.01ずつ変化させる。

- bang** 現在表示されている数値をアウトレットから出力する。
- brgb** **brgb**には、0から255までの範囲の3つの数値が続き、**number box**オブジェクトの背景色のRGB値を設定する。デフォルトは灰色（**brgb 211 211 211**）である。
- color** **color**には0から15のうち、1つの数値が続き、**number box**オブジェクトの背景色を、Objectメニューの**Color**サブ・メニュー内で利用可能な標準的なオブジェクトの色に設定する。
- flags** **flags**には1つの数値が続き、**number box**オブジェクトの外見の特徴、また振る舞いを設定する。その特徴（以下のアーギュメントの項の下に記述する）は、希望の特徴を指し示す以下の数値を加えることで設定される。4-太字、16-16進表示、32-三角形を表示しない、64-マウスを上げた時のみ出力、128-マウスによって変更不可、256-MIDI C3表示、1024-Roland 8進表示、2048-2進表示、4096-MIDI C4表示、8192-透明表示モード（他のオブジェクト上での表示、編集に便利である）。従って、例えば**flags 180**（4+16+32+128=180）メッセージは、**number box**オブジェクトを、16進表示、太字、三角形は表示せず、マウスでの変更不可というように設定することになる。
- frgb** **frgb**には0から255までの範囲の3つの数値が続き、**number box**オブジェクトに表示される数値の色を、RGB値で設定する。デフォルトは黒（**frgb 0 0 0**）である。



- max** **max**に1つの数値が続く場合は、**number box**オブジェクトで表示や出力できる最大値を設定する。数値を伴わない**max**メッセージであれば、最大値をなしに設定する（以前に設定された最大値を削除する）。
- min** **min**に1つの数値が続く場合は、**number box**オブジェクトで表示や出力できる最小値を設定する。数値を伴わない**min**メッセージであれば、最小値をなしに設定する（以前に設定された最小値を削除する）。
- rgb2** **rgb2**には0から255までの範囲の3つの数値が続き、**number box**オブジェクトが更新され、ハイライト表示されている時に表示される数値の色をRGB値で設定する。デフォルト値は白 (**rgb2 255 255 255**) である。
- rgb3** **rgb3**には0から255までの範囲の3つの数値が続き、**number box**オブジェクトが更新され、ハイライト表示されている時の背景色をRGB値で設定する。デフォルト値は黒 (**rgb3 0 0 0**) である。
- set** **set**には1つの数値が続き、その数字が格納され、表示されるが、出力を行わない。

(キーボード入力) パッチ・ウィンドウ内で**number box**オブジェクトがハイライト状態（三角形が隠されることで示される）であれば、キーボードの数値入力は、その**number box**オブジェクトへその値を変更するために送られる。マウスをクリックすることや、returnキー(Macintosh)、またはenterキー(Macintosh、Windows)を押すことで、入力されている数値を格納する。

(Fontメニュー) **number box**オブジェクトのフォントやサイズは、Fontメニューから異なったフォントか、サイズを選択することで変更できる。

## インスペクター

**number box**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**number box**オブジェクトを選択することで、**number box**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**number box**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**number box**オブジェクトのインスペクターでは、以下の属性を設定することができる。

*Range Min.*と*Max.*ボックスに値を入力することで、格納、表示、入力、通過させる数値の範囲を設定することができる。*No Min.*や*No Max.*チェック・ボックスがチェックされていれば（デフォルト状態）、



**number box**オブジェクトはそれらの最小、また最大の値をNone. (なし) に設定する。それらのボックスをチェックしなければ、最小値や最大値は0に設定される。

インスペクターのオプションの部分では、**number box**オブジェクトの表示属性を設定できる。**number box**オブジェクトのインスペクターで利用可能なオプションは、*Bold* (太字の書体で表示する)、*Draw Triange* (矢印のようなポインターを表示し、独特の外観を与える)、*Output Only on Mouse-Up* (数値を連続的にではなく、マウスのボタンを離れた時のみ出力する)、*Can't Change* (マウスやコンピューターのキーボードによる変更を禁止する)、*Transparent* (ボックスは表示せずに、**number box**オブジェクト内の数値のみ表示する。従って、**number box**オブジェクトは**comment**オブジェクトに良く似た外観になる) である。

*Display Style*のポップ・アップ・メニューでは、数値の表示の方法を選択できる。*Decimall*は数値を表示するデフォルトの方法である。*Hex*は数値を16進で表示し、MIDI関係のアプリケーションに有用である。*Roland Octal*では、数値を、いくつかのハードウェア・デバイスで使われている、それぞれの桁で1から8までの範囲の形式で表示する。11は0で、88は63を表す。*Binary*は、数値を1と0で表示する。*MIDI Note Names*では60をC3と表示するように、MIDIピッチの値に従って数値を表示する。*Note Names C4*は60をC4と表す以外は*MIDI Note Names*と同じである。すべての表示モードにおいて、それらが表示される形式と同じ形式で、数値を入力しなければいけない。

*Color*オプションでは、**swatch**オブジェクトのカラー・ピッカーか、RGB値を用いることで、数値のボックス、またその背景の通常、またハイライト状態で用いられる色を指定できる。*Number*は表示される数値の色 (デフォルトは0 0 0) を設定し、*Background*では**number box**オブジェクトそれ自体の色 (デフォルトは221 221 221) を、*Highlighted Number*は**number box**オブジェクトが選択されている時か、その値が更新された時の数値の表示される色 (デフォルトは222 222 222)、また*Highlighted Background*は、ハイライト表示になっている時や、更新された時の**number box**オブジェクトの色 (デフォルトは0 0 0) をそれぞれ設定する。

**Revert**ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。



## アーギュメント

なし。

## 出力

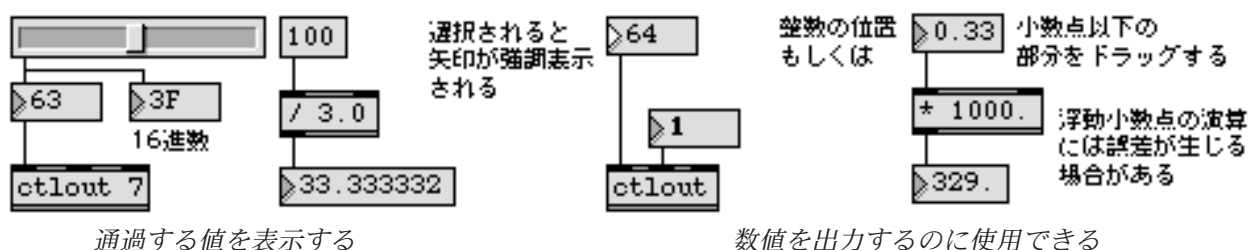
### 整数または実数

その数値を**number box**オブジェクト内に表示し、アウトレットから出力する。インレットに受け取った数値やコンピューター・キーボードによって入力された数値はその**number box**に設定された限度をこえることができるが、格納、表示、出力される値は、自動的に指定された範囲に制限される。

**number box**オブジェクトは、格納している数値の大きさに従って自動的にサイズを変更しない。受け取った数値が、**number box**オブジェクトに表示するには長過ぎるならば、整数の**number box**オブジェクトの場合は、省略記号 (...) が続く省略された形で表示され、実数の**number box**オブジェクトの場合は、プラス記号 (+) として表示される。

数値は、省略した形で表示されていても、通常通りに**number box**オブジェクトに格納され、出力される。

## 利用例



## 参照

**float**  
**int**

実数を格納する  
整数を格納する

Tutorial 3  
Tutorial 10

About numbers (英文PDF参照)  
Number boxes (英文PDF参照)

## 入力

**整数** 受け取る数値は**key**オブジェクトまたは**keyup**オブジェクトから入力されたASCII値である。コンピューター・キーボードから数字が入力されると、**numkey**オブジェクトはASCII値を識別して押された数値に解釈する。

**numkey**オブジェクトによって識別されるキーは0~9の数字、delete (backspace)、ピリオド、return、そしてenterキーである。**numkey**オブジェクト内に格納された数字は1つの数値に結合される。

**bang** **numkey**オブジェクト内に格納されている数値を第1アウトレットから出力し、格納される数値を0にリセットする。

**clear** 格納されている数値をリセットする。

## アーギュメント

オプション。アーギュメントに実数を与えると、**numkey**オブジェクトはピリオドと小数点以下の数値を理解し、整数の代わりに実数を出力する（しかし、アーギュメントがない場合、**numkey**オブジェクトに設定される初期値は常に0である）。

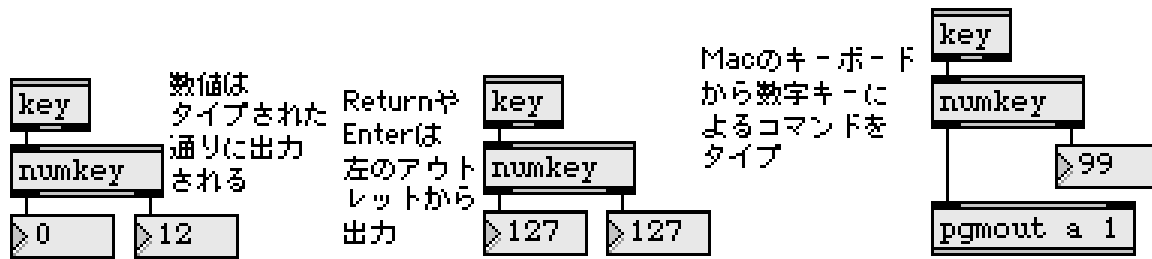
## 出力

**整数** コンピューター・キーボードから数字が入力され、(**key**オブジェクトまたは**keyup**オブジェクトから) ASCII値がインレットに入力されると、数字は1つの数値に結合されて**numkey**オブジェクトに格納される。格納された数値は第2アウトレットから順次出力されてゆく。deleteキー (Macintosh) あるいはBackspaceキー (Windows) は最後に入力された数字を消去してから、格納されている数値を第2アウトレットから出力する。ピリオド・キーは小数点として扱われる。そして (アーギュメントが実数でない時) 次の数字が新たに始められるまで、**numkey**オブジェクトは次の数字を格納しない。returnまたはenterキーは、格納している数値を第1インレットから出力し、新たに数値を入力できるように格納している数値を0にリセットする。

**実数** アーギュメントに実数を与えられている時、**numkey**オブジェクトは小数点と小数点以下の数値を理解して整数の代わりに実数を出力する。



## 利用例



タイプされたすべての数値を認識する

## 参照

**key**

コンピューター・キーボードのキー入力を出力する

**keyup**

コンピューター・キーボードのキー入力を出力する

**number box**

数値を表示し、出力する

Tutorial 20

Using the computer keyboard (英文PDF参照)

## 入力

- list** 第1インレット：最初の数値をx値、2番目の数値をy値としたx,yを1組のリストとして、**offer**オブジェクト内に格納される。最初の数値は整数でなければならない。2番目の数値は実数でも良いが、整数に変換される。
- 整数** 第1インレット：この数値はx,y組のうちのx値にあてられる。y値がすでに第2インレットに入力されているならば、この2つの数は一緒に**offer**オブジェクト内に格納される。さもなければ、**offer**オブジェクトは入力された数と一致する格納されているx値を探し出し、それに対応するy値を出力した後、その組を消去する。**offer**オブジェクト内に一致するx値がない時、**offer**オブジェクトは何もしない。
- 第2インレット：この数値は**offer**オブジェクト内に格納されるy値にあてられる。その後、第1インレットにx値（整数）が入力されると、この2つの数値はx,yの組として**offer**オブジェクト内に格納される。
- 実数** 第2インレット：整数に変換される。
- clear** 第1インレット：**offer**オブジェクト内に格納されたすべてのものを消去する。

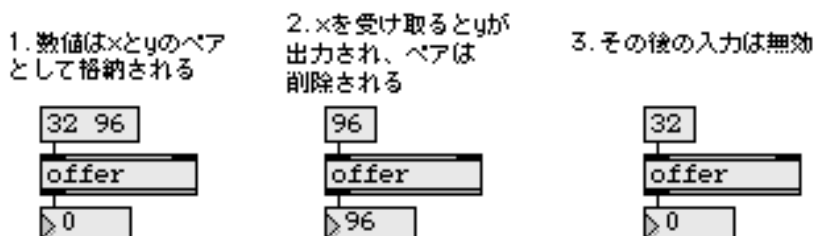
## アーギュメント

なし。

## 出力

- 整数** **offer**オブジェクト内に格納されているx,yの組のx値に一致する数値が第1インレットに入力されると、そのx値に対応するy値が出力され、この組は消去される。

## 利用例



組になった数値が格納され、一回だけ呼び出される

## 参照

**coll**  
**funbuff**  
**table**

様々な種類のメッセージを格納し、編集する  
xとyの数値のペアを対にして格納する  
数値の配列をグラフィカルに編集し、格納する



注記：このオブジェクトは、Mac OS 9上でのみ動作する。

## 入力

**シンボル** 第1インレット：現在のOMSセットアップに登録してあるMIDI出力デバイスの名前を受け取る。そのMIDIデバイスが持つすべてのコントローラー（コントロール・チェンジ・メッセージ）の名前を、**umenu** オブジェクトが受け取る形式で出力する。

第2インレット：現在のOMSセットアップに登録してあるMIDI入力デバイスの名前を受け取る。そのMIDIデバイスが持つすべてのコントローラー名を出力する。

**generic** 第1インレットまたは第2インレット：MIDI規格として標準化されているコントローラー名を出力する。

**整数** 第3インレット：コントローラー名を探すためのMIDIチャンネルを指定する。どのMIDIデバイスであるかは、その後に第1インレットまたは第2インレットに入力することになる。

第4インレット：MIDIデバイス名を受け取った時にコントローラー名を出力する、特定のパッチ・ナンバーを指定する。パッチ・ナンバーとして-1を指定すれば、パッチに依存しないコントローラーの名前を探すことになる（これは、多くのシンセサイザーでは、マスター設定の中に見い出される）。

**シンボル+整数** 第1インレット：出力デバイス名の後に1つまたは2つの整数が続くメッセージの場合、1番目の整数はMIDIチャンネルを表し、2番目はパッチ・ナンバー（または、上記で記述した-1）を表す。出力デバイス名の後に3つの整数が続く場合は、第2アウトレットから、ひとつのコントローラー名を出力する。3つの整数は、1番目がMIDIチャンネル、2番目はパッチ・ナンバー（または-1）、3番目は名前を得たいコントローラー・ナンバーを表す。

第2インレット：第1インレットと同様だが、入力デバイスの名前と整数を入力する。

## アーギュメント

**整数** オプション。第1アーギュメントでは、コントローラー名を探すためのMIDIチャンネル・ナンバーを指定する。どのMIDIデバイスであるかは、第1インレットまたは第2インレットに入力する。第2アーギュメントは、探し出すパッチ・ナンバーを指定する。デバイス名を受け取ると、コントローラー名が出力される。パッチ・ナンバーのアーギュメントとして-1を指定すると、パッチに依存しないコントローラーの名前を探すことになる。アーギュメントがない場合は、

**omscontrollers**はMIDIチャンネル1のパッチに依存しないコントローラーの名前を探す。

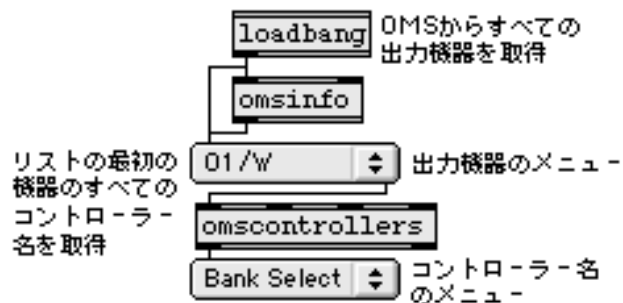
## 出力

**clear** 第1アウトレット：第1インレットまたは第2インレットにデバイス名を受け取れば、**omscontrollers**は最初に、すでに存在している項目を**umenu**から消去するために、**clear**メッセージを出力する。

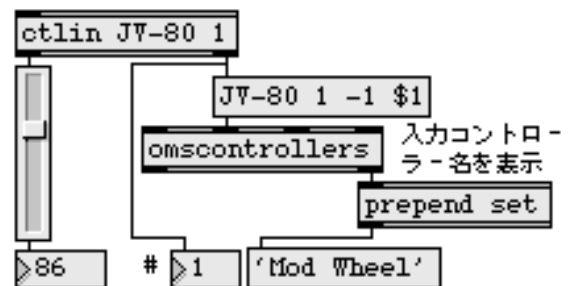
**append** 第1アウトレット：**clear**メッセージを出力した後に、接続された**umenu**オブジェクトに項目をセットするために、**omscontrollers**は指定されたデバイス（とMIDIチャンネル）が持つコントローラー名に、**append**メッセージをつけて出力する。注記：ほとんどの場合、128個のコントローラー名が出力されるので、それらを受け取る**umenu**は、多くの項目を格納できるように設定しなければならない（すなわち、**umenu**オブジェクトのインスペクター・ウィンドウで、Maximum Itemsの数値を128以上にすることがある）。

**シンボル** 第2アウトレット：デバイス名、MIDIチャンネル、パッチ・ナンバー、コントロール・ナンバーを特定するメッセージを受け取るごとに、1つのコントローラー名を出力する。

## 利用例



デバイスが持つすべてのコントローラー名をメニューに格納する...



または、1つのコントローラー名を出力する

## 参照

- omsinfo** OMSデバイス名をポップアップ・メニューに設定する
- omsnotes** OMSデバイスのノート名を出力する
- omspatches** OMSデバイスのパッチ名を出力する

注記：このオブジェクトは、Mac OS 9上でのみ動作する。

## 入力

**整数** 第1インレット：現在のOMS出力デバイスの名前を含む一連のメッセージを出力する。**omsinfo**オブジェクトのアウトレットから接続している、ポップアップ・メニューである**umenu**オブジェクトに、個々のデバイス名を設定するために、これらのメッセージが用いられる。**omsinfo**オブジェクトの第1インレットに受け取る整数は、表示されたメニュー項目をセットするために、**set**メッセージとともに出力される。

第2インレット：現在のOMS入力デバイスの名前を含む一連のメッセージを出力する。これらのメッセージは、**omsinfo**オブジェクトのアウトレットから接続している、ポップアップ・メニューである**umenu**オブジェクトに、個々のデバイス名を設定するために用いられる。**omsinfo**オブジェクトの第1インレットに受け取る整数は、表示されたメニュー項目をセットするために、**set**メッセージとともに出力される。これは、整数が0未満でない場合であり、それ以外の場合には**set**メッセージは出力されない。

**bang** 第1インレット：整数と同様だが、**umenu**の項目にセットする**set**メッセージは出力されない。入力デバイス名を出力するためには、第2インレットに**bang**を送るが、これは第2インレットに-1を送ることと同じである。

**controllers** 第1インレット：現在OMSセットアップのすべてのMIDIコントローラー（MIDI送信デバイス）の名前を含む一連のメッセージを出力する。これらのメッセージは、**omsinfo**オブジェクトのアウトレットから接続している、ポップアップ・メニューである**umenu**オブジェクトに、個々のコントローラー名を設定するために用いられる。**controllers**に続く整数によって、メニュー項目が作られた後、**umenu**で選択表示される項目番号を設定する。

## アーギュメント

なし。

## 出力

**clear** 最初に、**omscontrollers**は、すでに存在している項目を**umenu**から消去するために、**clear**メッセージを出力する。

**append** **clear**メッセージの出力した後に、接続された**umenu**オブジェクトに項目をセットするために、**omsinfo**は、各OMS入力デバイス名または

出力デバイス名に**append**メッセージをつけて出力する。デバイス名はMaxの**MIDI Setup**ダイアログで設定した順序で出力されることになる。

**set omsinfo**に入力されるメッセージが0以上の整数であれば、**umenu**で選択表示される項目番号を指定するために、**append**メッセージの後に**set**メッセージを出力する。

## 利用例



MIDI出力のオブジェクトのために、出力デバイス名を得る

そしてMIDI入力のオブジェクトにセットする

## 参照

**omscontrollers**

OMSデバイスのコントローラー名を出力する

**omsinfo**

OMSデバイス名をポップアップ・メニューに設定する

**omsnotes**

OMSデバイスのノート名を出力する

**omspatches**

OMSデバイスのパッチ名を出力する

注記：このオブジェクトは、Mac OS 9上でのみ動作する。

## 入力

**シンボル** 第1インレット：現在のOMSセットアップに登録してあるMIDI出力デバイスの名前を受け取る。そのMIDIデバイスが持つ、特定のMIDIチャンネルにおけるノート名を、**umenu**オブジェクトが受け取る形式で出力する。

第2インレット：現在のOMSセットアップに登録してあるMIDI入力デバイスの名前を受け取る。そのMIDIデバイスが持つ、特定のMIDIチャンネルにおけるノート名を、**umenu**オブジェクトに出力する。

**整数** 第3インレット：第1または第2インレットでMIDIデバイス名を受け取った時に、どのMIDIチャンネルノート名を出力するか、を指定する。

第4インレット：MIDIデバイス名を受け取った時に、どのパッチ・ナンバーのノート名を出力するか、指定する。パッチ・ナンバーとして-1を指定すれば、パッチに依存しないノート名を出力する。

**シンボル+整数** 第1インレット：出力デバイス名の後に1つまたは2つの整数が続くメッセージの場合、1番目の整数はMIDIチャンネルを表し、2番目はパッチ・ナンバー（または、上記で記述した-1）を表す。出力デバイス名の後に3つの整数が続く場合は、全体の中から1つのノート名だけを第2アウトレットから出力する。1番目の整数はMIDIチャンネル、2番目はパッチ・ナンバー（または-1）、3番目の整数がノート名を特定するために使われる。

第2インレット：第1インレットと同様だが、入力デバイスの名前と整数を入力する。

## アーギュメント

**整数** オプション。第1アーギュメントでは、ノート名を探すためのMIDIチャンネル・ナンバーを指定する。どのMIDIデバイスであるかは、第1インレットまたは第2インレットに入力する。第2アーギュメントはパッチ・ナンバーを特定する。デバイス名を受け取ると、ノート名が出力される。パッチ・ナンバーとして-1を指定すると、パッチに依存しないノート名を検出する。

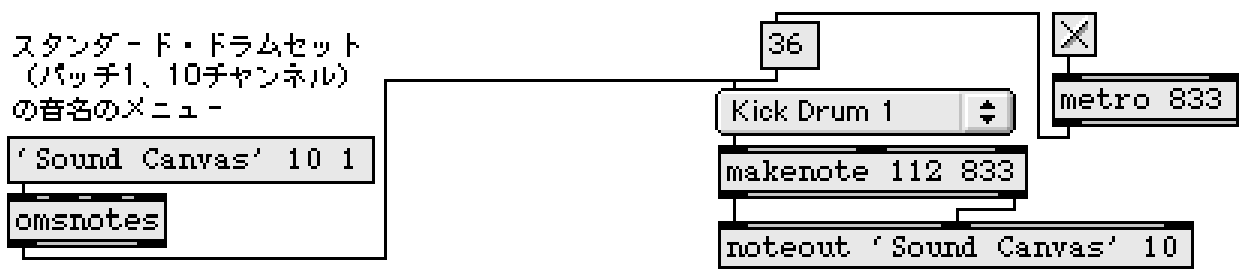
## 出力

**clear** 第1アウトレット：第1インレットまたは第2インレットにデバイス名を受け取れば、**omsnotes**は最初に、すでに存在している項目を**umenu**から消去するために、**clear**メッセージを出力する。

**append** 第1アウトレット：**clear**メッセージを出力した後に、接続された**umenu**オブジェクトに項目をセットするために、**omsnotes**は指定されたデバイスとMIDIチャンネルが持つノート名に、**append**メッセージをつけて出力する。注記：ほとんどの場合、128個のノート名が出力されるので、それらを受け取る**umenu**は、多くの項目を格納できるように設定しなければならない（すなわち、**umenu**オブジェクトのインスペクター・ウィンドウで、Maximum Itemsの数値を128以上にすることがある）。ただし、名前が与えられていないノートもあるので、メニュー項目が空の場合もある。

**シンボル** 第2アウトレット：デバイス名、MIDIチャンネル、パッチ・ナンバー、ノート・ナンバーを特定するメッセージを受け取るごとに、ノート名を出力する。

## 利用例



デバイスが持つノート名をメニューに格納し、メニューからノートを選択する

## 参照

- omscontrollers** OMSデバイスのコントローラー名を出力する
- omsinfo** OMSデバイス名をポップアップ・メニューに設定する
- omspatches** OMSデバイスのパッチ名を出力する



注記：このオブジェクトは、Mac OS 9上でのみ動作する。

## 入力

**シンボル** 第1インレット：現在のOMSセットアップに登録してあるMIDI出力デバイスの名前を受け取る。そのMIDIデバイスが持つすべてのパッチ名を、**umenu**オブジェクトが受け取る形式で出力する。

第2インレット：現在のOMSセットアップに登録してあるMIDI入力デバイスの名前を受け取る。そのMIDIデバイスが持つすべてのパッチ名を出力する。

**整数** 第3インレット：パッチ名を探すためのMIDIチャンネルを指定する。どのMIDIデバイスであるかは、その後に第1インレットまたは第2インレットに入力することになる。

**シンボル+整数** 第1インレット：出力デバイス名の後に整数が続くメッセージの場合、整数はMIDIチャンネルを表す。出力デバイス名の後に2つの整数が続くメッセージでは、全体の中から1つのパッチ名だけを第2アウトレットから出力する。1番目の整数はMIDIチャンネル、2番目の整数がパッチ・ナンバーを表す。

第2インレット：第1インレットと同様だが、入力デバイスの名前と整数を入力する。

## アーギュメント

**整数** オプション。パッチ名を探すためのMIDIチャンネル・ナンバーを指定する。どのMIDIデバイスであるかは、第1インレットまたは第2インレットに入力する。

## 出力

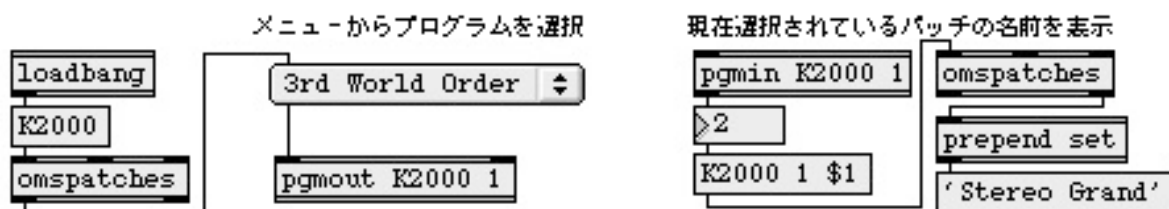
**clear** 第1アウトレット：第1インレットまたは第2インレットにデバイス名を受け取れば、**omspatches**は最初に、すでに存在している項目を**umenu**から消去するために、**clear**メッセージを出力する。

**append** 第1アウトレット：**clear**メッセージを出力した後に、接続された**umenu**オブジェクトに項目をセットするために、**omspatches**は指定されたデバイス（とMIDIチャンネル）が持つパッチ名に、**append**メッセージをつけて出力する。受信するシンセの音色バンクは多くの項目を含むために、それらを受け取る**umenu**は、多くの項目を格納できるように設定しなければならない（すなわち、**umenu**オブジェクトのインスペクター・ウィンドウで、Maximum Itemsの数値を128以上にすることがある）。



シンボル 第2アウトレット：デバイス名、MIDIチャンネル、パッチ・ナンバーを特定のメッセージを受け取ることに、パッチ名を出力する。

## 利用例



デバイスが持つすべてのパッチ名をメニューに格納する...

またはパッチ名を出力する

## 参照

**omscontrollers**

OMSデバイスのコントローラー名を出力する

**omsinfo**

OMSデバイス名をポップアップ・メニューに設定する

**omsnotes**

OMSデバイスのノート名を出力する

## 入力

**bang** 第1インレット：最後に**bang**を出力してから現在までに、第2インレットに**bang**を受け取っている場合のみ**bang**を出力する。

第2インレット：次に第1インレットに**bang**を受け取った時に、それを通過させるように**onebang**をリセットする。

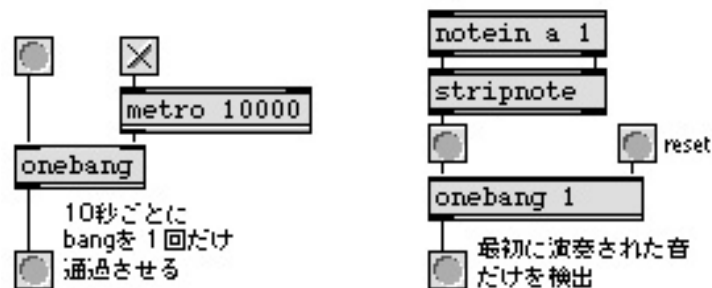
## アーギュメント

**整数** オプション。0でない値を設定した場合、第1インレットに最初に**bang**を受け取った時に、**onebang**はそれを通過させる。

## 出力

**bang** 最後に**bang**を出力してから現在までに、第2インレットに**bang**を受け取っている場合のみ**bang**を出力する。

## 利用例



沢山の **bang**を受け取っても、1つの**bang**だけを通過させる

## 参照

**gate**  
**Ggate**  
**speedlim**

入力を特定のアウトレットから出力する  
2つのうち1つのアウトレットから数値を出力する  
メッセージが通過するスピードを制限する

**onecopy**オブジェクトは、Extrasメニューに入れたいパッチの中にこのオブジェクトを作り、そのパッチをextrasフォルダに入れて使用する。Extrasメニューからパッチの名前を選択すると、そのパッチが既に読み込まれているならば、重複して開かれることはなく最前面に表示される。そのパッチが現在開かれていなければ、読み込まれる。**onecopy**オブジェクトは、Extrasメニューと協調して、1度に開かれるパッチが1つだけになるようにする。しかし、**onecopy**オブジェクトを含むパッチでも、Fileメニューの**Open...**から開いた場合は、重複して開かれる。

## 入力

なし。

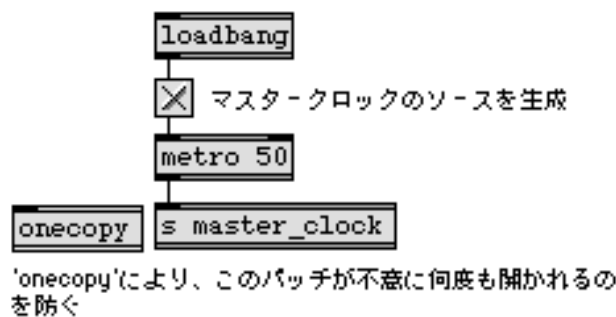
## アーギュメント

なし。

## 出力

なし。

## 利用例



Extrasメニューを使って同じパッチが重複して開かれるのを防ぐ。

## 参照

**thispatcher**  
**pcontrol**

パッチにメッセージを送る  
パッチ内のサブ・ウィンドウを開いたり、閉じたりする

## 入力

- bang** ファイルを選ぶための標準ファイル・ダイアログを開く。
- set** ファイル・タイプを特定する4文字からなるシンボル（例えば、**TEXT**、**maxb**）を伴った**set**メッセージは、**opendialog**オブジェクトが標準ファイル・ダイアログを開く時に、指定されたファイル・タイプを探すように設定する。
- sound** **opendialog**オブジェクトが、オーディオ・ファイル（AIFF、Sound Designer II、NeXT/Sun、そしてWAVなどの一般的なデータ・ファイル・タイプ）を表示するように設定する。
- types** 4文字までのタイプを表す記号を1つ以上伴った**types**メッセージは、**opendialog**オブジェクトによって読まれるファイルの種類を確定する。Macintoshが扱うファイルのタイプを表す記号の例を挙げると、**TEXT**はテキスト・ファイル、**maxb**はMaxのバイナリー形式によるパッチ・ファイル、そして**AIFF**はAIFF形式のオーディオ・ファイルなどがある。アーギュメントを含まない**types**メッセージは、すべてのファイル・タイプを対象とするよう設定する。これはデフォルトの設定である。
- 任意のシンボル 1つ以上のシンボルは、1つ以上のタイプを表す記号として解釈され、**opendialog**オブジェクトによって表示されるファイルの種類を指定する。

## アーギュメント

- fold** オプション。**opendialog**オブジェクトがファイルではなくフォルダを選ぶように設定する。
- sound** オプション。**opendialog**オブジェクトがオーディオ・ファイル（AIFF、Sound Designer II、NeXT/Sun、そしてWAVなどの一般的なデータ・ファイル・タイプ）を表示するように設定する。QuickTime Appendixに、開くことができるすべてのファイル・タイプがリスト・アップされている。
- 任意のシンボル オプション。1つ以上のシンボルは、1つ以上のタイプを表す記号として解釈され、**opendialog**オブジェクトによって表示されるファイルの種類を指定する。

## 出力

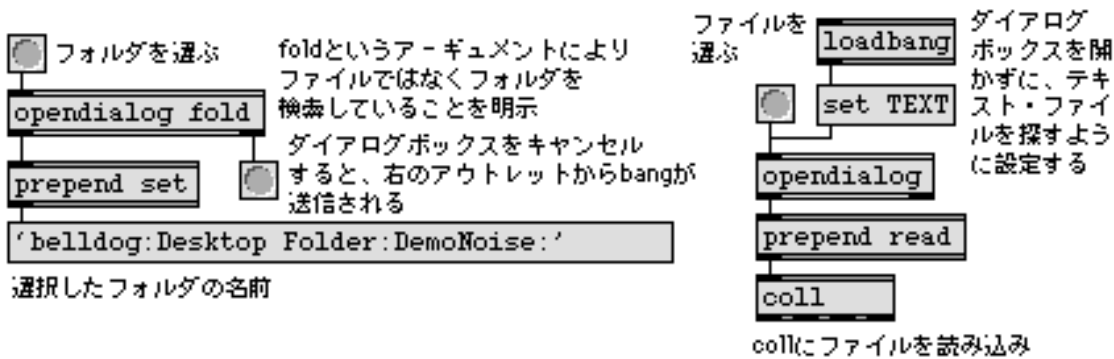
**シンボル** 第1アウトレット：ユーザーが選んだファイルの完全なパス名がシンボルとして出力される。絶対パスはMax 4.1以前のバージョン(コロンを使用)を除き、スラッシュで区切られて表示される。具体的には下記のようなになる。

**“C:/Max Folder/extras/mystuff/mypatch.pat”**

パスの形式や種類を変更したい場合は、**conformpath**オブジェクトを使用する。

**bang** ユーザーがダイアログ・ボックスをキャンセルした場合、第2アウトレットから**bang**が出力される。

## 利用例



フォルダや特定の種類のファイルを表示する

## 参照

|                   |                           |
|-------------------|---------------------------|
| <b>dialog</b>     | テキスト入力のためのダイアログ・ボックスを開く   |
| <b>dropfile</b>   | ファイルをドラッグ&ドロップする領域を設定する   |
| <b>date</b>       | 現在の日付と時刻を出力する             |
| <b>filedate</b>   | ファイルの最終修正日を出力する           |
| <b>filein</b>     | ファイルからバイナリー・データを読み出す      |
| <b>filepath</b>   | 現在のMaxサーチ・パスの情報を出力する      |
| <b>folder</b>     | 指定したフォルダー内のファイル・リストを表示する  |
| <b>opendialog</b> | ファイルやフォルダを選択するためのダイアログを開く |
| <b>strippath</b>  | フル・パス名からファイル名を取得する        |



## 入力

(パッチ) パッチ内の各**outlet**オブジェクトは、そのパッチが他のパッチの中に（オブジェクトかサブ・パッチとして）使われる時に、オブジェクト・ボックスの下のアウトレットとして現れる。サブパッチ内の**outlet**オブジェクトにメッセージが入力されると、このオブジェクト・ボックスの対応するアウトレットから出力されることになる。

## インスペクター

インスペクターを使用して**outlet**オブジェクトとして表示するアシスタンス・メッセージを指定し、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**outlet**オブジェクトを選択することで、**outlet**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**outlet**オブジェクトを選択して、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

*Describe Outlet* エリアにテキストを入力することで、アシスタンス・メッセージの内容を指定できる。

*Revert* ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

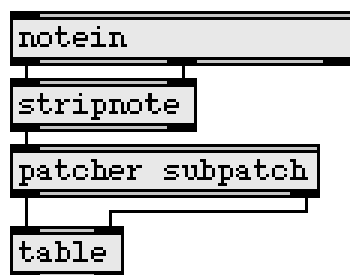
なし。

## 出力

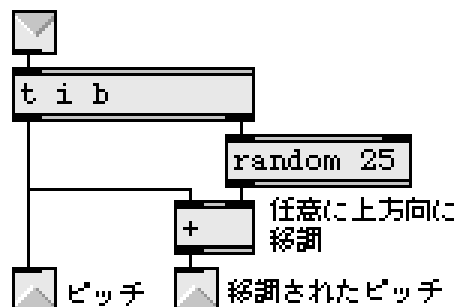
任意のメッセージ サブ・パッチ内の**outlet**オブジェクトへ入力されるすべてのメッセージは、パッチ・コードを通してサブパッチのアウトレットから出力される。



## 利用例



サブパッチの内容:



サブパッチ・オブジェクトのアウトレットは、サブパッチ内の **outlet** オブジェクトに対応している

## 参照

|                 |                                |
|-----------------|--------------------------------|
| <b>bpatcher</b> | ボックス内に埋め込まれた可視型のサブ・パッチ         |
| <b>forward</b>  | 様々なオブジェクトにパッチ・コードなしでメッセージを送信する |
| <b>int</b>      | 整数を格納する                        |
| <b>patcher</b>  | パッチの中でサブ・パッチを作る                |
| <b>receive</b>  | パッチ・コードなしでメッセージを受信する           |
| <b>send</b>     | パッチ・コードなしでメッセージを送信する           |

Tutorial 26

The **patcher** object (英文PDF参照)



## 入力

**bang** 第1インレット：座標、ドローイング・モード、色の設定に従って、楕円形を描く。

**整数** 第1インレット：楕円形の左端の座標を設定し、楕円形を描く。

第2インレット：楕円形の上端の座標を設定する。

第3インレット：楕円形の右端の座標を設定する。

第4インレット：楕円形の下端の座標を設定する。

第5インレット：楕円形のドローイング・モードを設定する。以下はドローイング・モードの定数である。オペレーティング・システムによっては、すべてのモードが利用できる訳ではない。

|         |   |             |    |
|---------|---|-------------|----|
| Copy    | 0 | blend       | 32 |
| Or      | 1 | addPin      | 33 |
| Xor     | 2 | addOver     | 34 |
| Bic     | 3 | subPin      | 35 |
| NotCopy | 4 | transparent | 36 |
| NotOr   | 5 | adMax       | 37 |
| NotXor  | 6 | subOver     | 38 |
| NotBic  | 7 | adMin       | 39 |

第6インレット：グラフィック・ウィンドウの現在のパレットに従って、楕円形を描くパレット番号（色）を設定する。モニターが白黒の場合は、この設定は効果を及ぼさない。

**frgb** 第1インレット：**frgb**は0から255の間の3つの数字が後に続き、その後に描かれる楕円形の色を、RGB値で設定する。

**priority** 第1インレット：**priority**の後に0より大きな数字が続き、**oval**オブジェクトが描くスプライトの、グラフィック・ウィンドウにおける優先順位を設定する。低い優先順位のオブジェクトは、高い優先順位の後ろに描かれることになる。

## アーギュメント

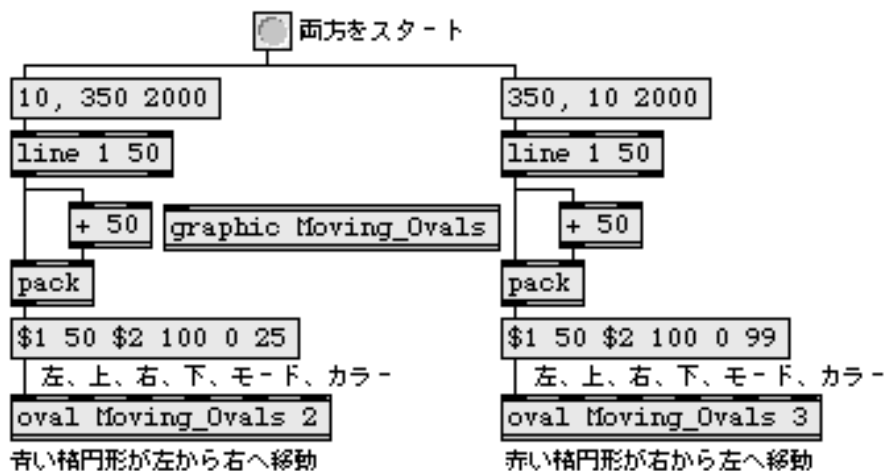
**任意のシンボル** 必須。**oval**の最初のアーギュメントは、楕円形が描かれるグラフィック・ウィンドウの名前でなければならない。**oval**のオブジェクトが作られる時に、そのウィンドウが存在する必要はないが、名前が一致するウィンドウが存在していないか、表示されていないか、楕円形は描かれない。

**整数** オプション。楕円形のスプライトの優先順位の初期値を設定する。優先順位が指定されていない場合、デフォルト値は3である。

## 出力

(表示) **oval**オブジェクトに関連付けられたグラフィック・ウィンドウが表示されている時に、**bang**メッセージか数字を第1インレットに受け取れば、ウィンドウに楕円形が描かれる。この時、以前に描かれた楕円形は消去される。

## 利用例



右側の **oval** オブジェクトはスプライトの優先順位が高いので、両方がスクリーン上で重なり合う時は、左側のもオブジェクトより前に表示される

## 参照

**frame** グラフィック・ウィンドウに矩形の枠を描く  
**graphic** スプライトによるグラフィックを描くためのウィンドウ  
**lcd** パッチ・ウィンドウ内に画像を描画する  
**rect** グラフィック・ウィンドウに塗り潰した矩形を描く  
**ring** グラフィック・ウィンドウに楕円形の枠を描く

## 入力

**整数** この数値は、受け取ったインレットに対応する**pack**オブジェクト内のリスト内の位置に、1つの項目として格納される。第1インレットに受け取った数値は、リストの1つ目の項目として格納され、アウトレットからリスト全体を出力する。その数値を受け取ったインレットが、実数やシンボルのアーギュメントによって初期化されている場合は、入力された数値はそれぞれ、実数と空白のシンボルに変換される。

**実数** この数値は、受け取ったインレットに対応する**pack**オブジェクト内のリスト内の位置に、1つの項目として格納される。第1インレットに受け取った数値は、リストの1つ目の項目として格納され、アウトレットからリスト全体を出力する。その数値を受け取ったインレットが、整数やシンボルのアーギュメントによって初期化されている場合は、入力された数値はそれぞれ、整数と空白のシンボルに変換される。アーギュメントが指定されていなければ、実数は整数に変換される。

**bang** 第1インレット：現在**pack**オブジェクトに格納されている内容を出力する。

**任意のシンボル** そのシンボルを受け取ったインレットが、シンボルのアーギュメントによって初期化されている場合には、そのシンボルは**pack**オブジェクトの対応する位置に格納される。それ以外の場合は、シンボルは格納される前に0に変換される。第1インレットに受け取ったシンボルは、**pack**オブジェクトの内容を出力する。

**リスト** 任意の複数の項目を持つメッセージは、それが数値で始まっている場合でも、**pack**オブジェクトによってリストとして扱われる。入力されたリストの1つ目の項目は、**pack**オブジェクトのリストを受け取ったインレットに対応する位置に格納され、続くそれぞれの項目は、続くインレットに受け取られるように（利用できるインレットの数によって制限される）格納される。第1インレットに受け取ったリストは、格納されたリスト全体をアウトレットから出力させる。

**set** **set**には任意のメッセージが続き、いかなる出力も引き起こさずに、メッセージを**pack**オブジェクトに受け取らせることを可能にする。**set**メッセージは、どのインレットでも受け取ることができるが、第1インレット（出力を引き起こすインレット）の場合でのみ意味がある。その他のインレットでは、**set**メッセージは無視し、残りのメッセージを通常通りに扱う。

**nth** **nth**には1つのインレットの番号（第1インレットより1から始まる）が続き、**pack**オブジェクト内のその位置に格納された項目の値を出力する。

**send** 第1インレット：**send**には**receive**オブジェクトの名前が続き、**pack**オブジェクトのアウトレットではなく、その名前をもった**receive**オブジェクトすべてに、現在格納されているリストを送信する。

## アーギュメント

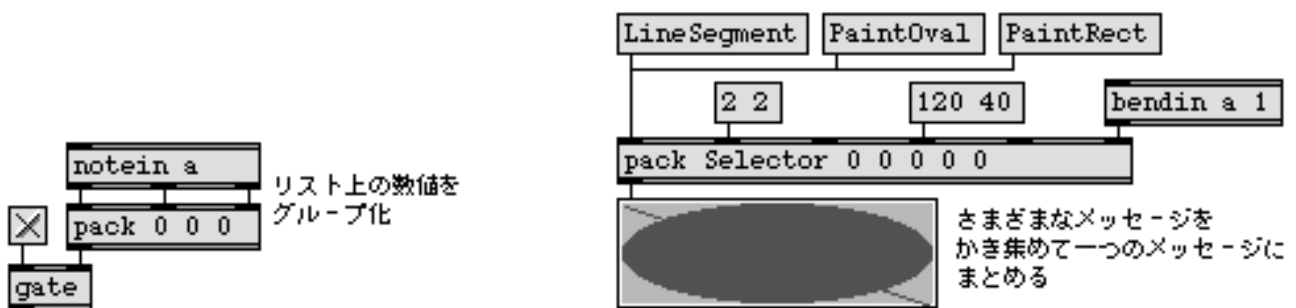
**整数、実数、シンボル** オプション。インレットの数はアーギュメントの数によって決定される。それぞれのアーギュメントは、**pack**オブジェクトに格納されるリストの項目の初期の型と値を設定する。数値のアーギュメントが小数点を含む場合は、その項目は実数として格納されることになる。シンボルのアーギュメントが設定されていれば、その項目はシンボルとして格納される。また、アーギュメントが設定されていない場合には、2つのインレットが生成され、初期値を0（整数）として設定する。注記：オブジェクト・ボックスにリストを入力すると、自動的にそれは**pack**オブジェクトとして認識され、オブジェクト・ボックスに**pack**と入力するのを省略できる。また、入力したアーギュメントのリスト（最低2つの項目をもち、数値ではじまる）を初期値として設定する。

## 出力

**リスト** リストの長さは、アーギュメントの数によって設定される。第1インレットに入力を受け取った場合には、格納されたリストをアウトレットから出力する。

**整数、実数、シンボル** **nth**メッセージを受け取った時に、その指定された項目の値を出力する。

## 利用例



数値やシンボルは、**pack**オブジェクト内に必要に応じて混在する

## 参照

|               |                           |
|---------------|---------------------------|
| <b>bondo</b>  | 一群のメッセージを同期する             |
| <b>buddy</b>  | 入力されたデータを同期させ、それらを同時に出力する |
| <b>match</b>  | 数値の並びを探し、リストとして出力する       |
| <b>swap</b>   | 2つの数値の一連の順番を逆にする          |
| <b>thresh</b> | 短時間に受け取った数値をリストとしてまとめる    |
| <b>unpack</b> | リストを個々のメッセージに分割する         |
| <b>zl</b>     | 多目的なリスト処理                 |

Tutorial 30      Number groups (英文PDF参照)



## 概要

**panel**オブジェクトはユーザー・インターフェースとして利用できる矩形の背景パネルを作成できる。また、丸みを付けた矩形の作成や、**ubutton**オブジェクトと共に利用し、ボタンとしても使用できる。

## 入力

- border** **border**には1つの数値が続き、枠線の幅をピクセル単位で設定する。デフォルト値は1である。
- brgb** **brgb**には0から255までの範囲の3つの数値が続き、**panel**オブジェクトの色をRGB値で設定する。デフォルトは灰色 (**brgb 192 192 192**) である。
- frgb** **frgb**には0から255までの範囲の3つの数値が続き、**panel**オブジェクトの枠線の色をRGB値で設定する。デフォルトは黒 (**frgb 0 0 0**) である。
- rounded** **rounded**には1つの数値が続き、**panel**オブジェクトの角の丸みの大きさをピクセル単位で設定する。デフォルト値は0 (丸めない) である。
- shadow** **shadow**には正の、また負の数値が続き、**panel**オブジェクトの影付け効果の大きさをピクセル単位で設定する。正の数値は浮き上がって見えるような効果を、負の数値は凹んで見えるような効果を作り出す。デフォルト値は0 (影付けをしない) である。
- size** **size**には2つの数値が続き、**panel**オブジェクトの幅と高さをピクセル単位で指定する。デフォルトのパネルの大きさは、幅が69、高さが57である。

## インスペクター

**panel**オブジェクトの振舞はインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**panel**オブジェクトを選択することで、**panel**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**panel**オブジェクトを選び、オブジェクト・メニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

*Width*と*Height*のナンバー・ボックスは、パネルの大きさを設定するのに使用される。デフォルトのパネルの大きさは幅が69、高さが57である。*Border Size*は、パネルの枠線の幅をピクセル単位で指定する。デフォルト値は1である。*Shadow Size*のナンバー・ボックスに値を入



力すれば、パネルの影の幅を設定する。デフォルト値は0（影付けをしない）である。*Rounded Corners*のボックスにピクセル単位で数値を入力すれば、パネルの角の丸みを指定できる。

*Color*オプションは、**swatch**オブジェクトのカラー・ピッカーか、RGB値を用いることで、枠線の色と背景の色を設定する。*Frame*は、**panel**オブジェクトの枠線の色を設定し（デフォルト値は0 0 0）、*Background*はパネルの色を設定する（デフォルト値は192 192 192）。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

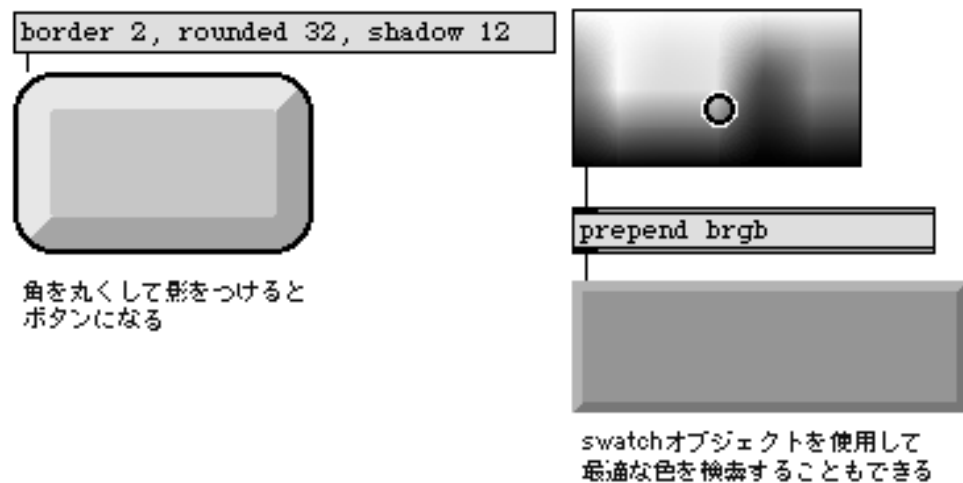
## アーギュメント

なし。

## 出力

なし。

## 使用例



## 参照

**fpic**  
**lcd**  
**pics**  
**pict**  
**ubutton**

画像ファイルより画像を表示する  
 パッチ・ウィンドウ内に画像を描画する  
 グラフィック・ウィンドウでのアニメーション  
 グラフィック・ウィンドウでピクチャーを描く  
 透明なボタン、**bang**を出力する



## 入力

- リスト** リストの数値はアーギュメントと比較される。リストのすべての数値が、対応するアーギュメントより大きいか同じである場合は、アウトレットから**bang**が出力される。**bang**が再び出力されるためには、**past**に**clear**メッセージを送る必要がある。あるいは、同じ数値か、アーギュメントより小さい数値に戻るようしてから、それを上回る数値の別のリストを送らなければならない。
- 整数** アーギュメントが1つだけの場合で、入力がアーギュメントより大きい値か同じ値であり、以前の入力がそれより大きくない値か同じ値である時に、**past**はアウトレットから**bang**を出力する。
- 実数** 整数に変換される。
- clear** 以前に入力から受け取った数値を消去し、再び**bang**メッセージを出力する状態にする。
- set** **set**の後に1つかそれ以上の数値が続き、**past**オブジェクトのインレットに受け取った数値と同じ値であるか、それより大きい数値でなければならない数値を設定する。

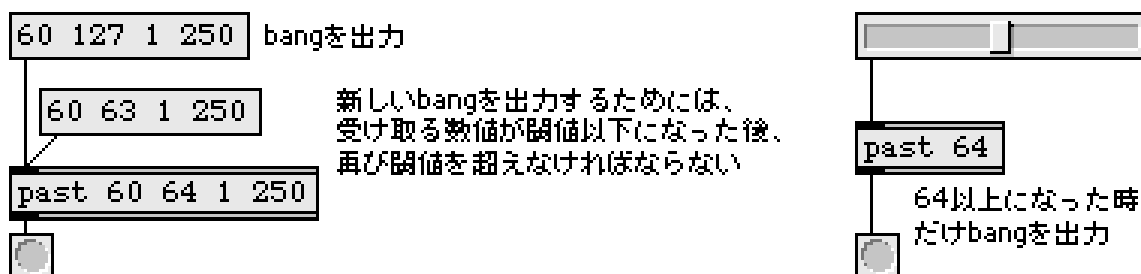
## アーギュメント

- リスト** インレットに受け取る数値と同じか、それより大きい数値でなければならない数値のリストを設定する。
- 整数** インレットに受け取る数値と同じか、それより大きい数値でなければならない単一の数値を設定する。

## アウトプット

- bang** すべてのアーギュメントが、インレットに受け取った数値と同じか、それより大きい数値である場合、**past**は**bang**を出力する。そうでない場合は、**past**は何もしない。数値が閾値を超える時のみ、**bang**が出力される。一度、閾値を超えた後に、新しい**bang**が出力されるためには、受け取る数値が閾値以下になった後、再び閾値を超えなければならない。

## 利用例



入力される数値が、増加方向での数値で閾値を超えた時のみ、**bang** が出力される

## 参照

**cd**  
**timeline**  
**maximum**  
**Peak**  
>

オーディオCDをコントロールする  
Maxメッセージを時間軸で扱うスコア  
数値のリストの中で最も大きい値を出力する  
数値が直前の値より上回る時に出力する  
2つの数値を比較し、より大きいことを判断する

## 入力

任意 **patcher** オブジェクトのインレットの数は、そのサブ・パッチ・ウィンドウ内に含まれている **inlet** オブジェクトの数によって決定される。

## アーギュメント

任意のシンボル オプション。サブ・パッチはアーギュメントによって名前をつけることができ、その名前はサブ・パッチ・ウィンドウのタイトル・バーに表示される。サブ・パッチ・ウィンドウのタイトル・バーにある名前は、同じファイルの一部であることを示すために、括弧を付けて表示される。アーギュメントを入力しなければ、サブ・パッチ・ウィンドウは [sub patch] という名前が付けられる。異なる **patcher** オブジェクトは同じ名前であっても、異なったサブ・パッチとなり、同じ内容を共有することはない。

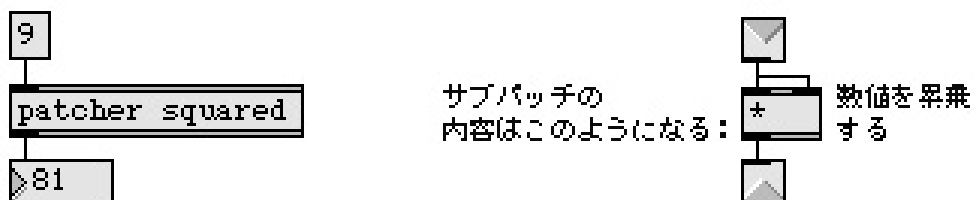
## 出力

任意 **patcher** オブジェクトのアウトレットの数は、そのサブ・パッチ・ウィンドウ内に含まれている **outlet** オブジェクトの数によって決定される。サブ・パッチ内に含まれる **send** オブジェクトや **value** オブジェクトによっても、サブ・パッチからの出力を行うことができる。実際の **patcher** オブジェクトから出力されるメッセージは、サブ・パッチの内容によって異なる。

**patcher** オブジェクトが最初に作られた時には、そのサブ・パッチ・ウィンドウを編集するために自動的に開く。それ以後に、**patcher** オブジェクト（および、任意のサブ・パッチ・オブジェクト）の内容を表示したり、編集するには、パッチ・ウィンドウがロックされている状態でそのオブジェクト上でダブル・クリックする。

**patcher** オブジェクトによるサブ・パッチ内のすべてのオブジェクトは、その **patcher** オブジェクトを含むパッチの一部として保存される。

## 利用例



パッチは、他のパッチの一部として含まれ、保存される

## 参照

|                    |                           |
|--------------------|---------------------------|
| <b>bpatcher</b>    | ボックス内に埋め込まれた可視型のサブ・パッチ    |
| <b>inlet</b>       | パッチの外からメッセージを受け取る         |
| <b>outlet</b>      | パッチの外へメッセージを出力する          |
| <b>pcontrol</b>    | パッチ内のサブ・ウィンドウを開いたり、閉じたりする |
| <b>thispatcher</b> | パッチにメッセージを送る              |

Tutorial 26      The patcher object (英文PDF参照)

## 入力

- open** **pcontrol** オブジェクトのアウトレットに接続された、任意のサブ・パッチやオブジェクトのパッチ・ウィンドウを開く。
- close** **pcontrol** オブジェクトのアウトレットに接続された、任意のサブ・パッチや**patcher**オブジェクトのパッチ・ウィンドウを閉じる。
- enable** 0以上の値が続く**enable**メッセージは、**pcontrol**オブジェクトのアウトレットに接続された、任意のサブ・パッチや**patcher**オブジェクト内のMIDI関係のオブジェクトの動作を有効にする。**enable 0**メッセージは、それらのサブ・パッチ内のMIDI関係のオブジェクトの動作を無効にする。
- load** **load**にはパッチ・ファイルの名前が続き、その名前のファイルがMaxのサーチ・パス内で見つければ、そのファイルを開く。ファイル名は、オプションとして、9つまでの数値か、またシンボルを続けることができ、それらは、開かれるパッチが持つ#アーギュメント（#1から#9）に置き換えられることになる。
- shroud** **shroud**にはパッチ・ファイルの名前が続き、その名前のファイルを、そのウィンドウを表示せずに開く。（表示されずに開かれるパッチは、場合によっては、何か予想しない結果を生じさせることがあるので、このメッセージは注意して使うこと）
- help** **help**には1つのシンボルが続き、MaxのHelpディレクトリーから、指定されたシンボル名に.helpが続くファイルを開く。

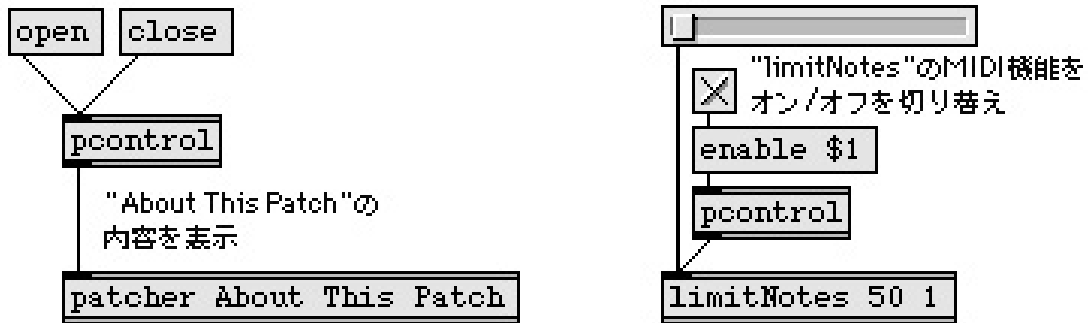
## アーギュメント

なし。

## 出力

**pcontrol**オブジェクトのアウトレットに接続された、任意のサブ・パッチや**patcher**オブジェクトのパッチ・ウィンドウを開いたり、閉じたりすることができる。**pcontrol**オブジェクトのインレットに適当なメッセージを受け取った時に、MIDIを有効/無効にすることもできる。

## 利用例



サブ・パッチ・ウィンドウの表示/非表示を切り替え、またそれらのMIDIオブジェクトを有効/無効にする

## 参照

|                    |                        |
|--------------------|------------------------|
| <b>bpatcher</b>    | ボックス内に埋め込まれた可視型のサブ・パッチ |
| <b>inlet</b>       | パッチの外からメッセージを受け取る      |
| <b>patcher</b>     | パッチの中でサブ・パッチを作る        |
| <b>thispatcher</b> | パッチにメッセージを送る           |

Tutorial 40      Automatic actions (英文PDF参照)

## 入力

**整数** 第1インレット：入力が現在**Peak**に格納されている数値よりも大きい時、その数値を新しいピーク値として格納し、出力する。

第2インレット：数値は新しいピーク値として**Peak**に格納し、出力する。

**実数** 整数に変換する。

**リスト** 第1インレット：2番目の数値は新しいピーク値として格納され、出力される。その後、1番目の数値が第1インレットに受け取られる。

**bang** 第1インレット：第1アウトレットから現在のピーク値を送り出す。

## アーギュメント

なし。**Peak**に格納される初期値は0である。

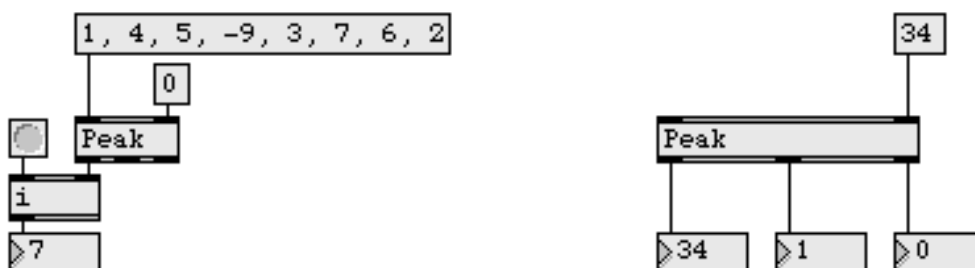
## 出力

**整数** 第1アウトレット：新しいピーク値が出力される（第2インレットに受け取った数値は、常に新しいピーク値となる）。

第2アウトレット：受け取った数値が新しいピーク値である場合は、1が出力される。第1インレットに受け取った数値が新しいピーク値でない場合は、0が出力される。

第3アウトレット：受け取った数値が新しいピーク値である場合は、0が出力される。第1インレットに受け取った数値が新しいピーク値でない場合は、1が出力される。

## 利用例



連続する数値からピーク値を見つける

第2インレットに受け取る数値は、常に新しいピーク値を設定する



## 参照

**maximum**

数値のリストの中で最も大きい値を出力する

**past**

数値がある値を超えた時に出力する

**Trough**

数値が直前の値より下回る時に出力する

**>**

2つの数値を比較し、より大きいことを判断する

## 入力

- (MIDI) **pgmin**は、MIDI入力デバイスから受信したMIDIプログラム・チェンジ・メッセージを、入力として受け取る。
- enable** **enable 0**メッセージはオブジェクトを使用不可にし、その後受信されるMIDIデータを無視する。MIDI Enable/Disableボタンや**pcontrol**オブジェクトによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを再び動作可能にする。
- port** **port**の後に**a**から**z**の文字を続けるか、MIDI入力デバイスの名前を続けることによって、プログラム・チェンジ・メッセージを受け取るポートを設定する。**port**という単語はオプションで、省略することもできる。
- 整数** 数値は、受信されるMIDIプログラム・チェンジの値のように扱われる。第2アウトレットが存在する場合、MIDIチャンネル・ナンバーとして0が出力される。プログラム・ナンバーに1を足した数値が、第1アウトレットから出力され、その範囲は1から128に限定されない。

## アーギュメント

- a-z** オプション。プログラム・チェンジ・メッセージを受け取るポートを指定する。アーギュメントがない場合は、**pgmin**は、すべてのポートのすべてのチャンネルから受け取る。
- (MIDI入力デバイス名) オプション。MIDI入力デバイスの名前は、ポートを指定する最初のアーギュメントとして用いることができる。
- a-zと整数** (空白文字によって分けられる) 文字と数値の組み合わせは、プログラム・チェンジ・メッセージを受け取るポートとMIDIチャンネルを指定する。チャンネル・ナンバーが16より大きい場合は、1から16の範囲に収まるように変換される。
- 整数** 文字と数値の組み合わせの代わりに、数値だけを指定しても良い。チャンネル・ナンバーのアーギュメントの正確な意味は、MIDI Setupダイアログで指定したポートごとのチャンネル・オフセットによって決まる。

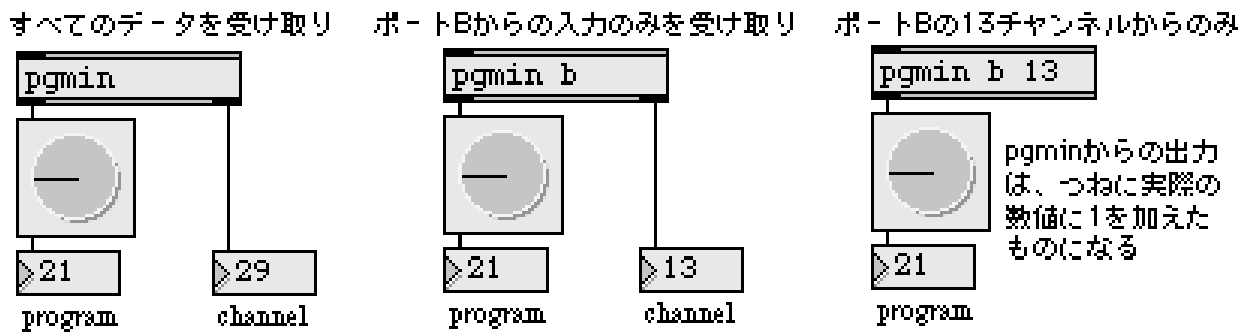
## 出力

**整数** 特定のチャンネル・ナンバーがアーギュメントに含まれている場合、アウトレットは1つのみとなる。その出力は、指定したチャンネルとポートから受け取ったプログラム・ナンバーである。

注記：**pgmin**オブジェクトは、受信したプログラム・ナンバーより1大きな数値を出力する。従って、受信したプログラム・チェンジの値が32であれば、**pgmin**のアウトレットからは33が出力される。

アーギュメントにチャンネル・ナンバーを指定していない場合は、**pgmin**は2番目のアウトレットを持つ。第2アウトレットからは、受信したプログラム・チェンジ・メッセージのチャンネル・ナンバーを出力する。

## 利用例



指定したポートや、指定したポートとチャンネルから、どこでもプログラム・チェンジを受け取ることができる

## 参照

**midin**      生のMIDIデータを受信する  
**pgmout**      MIDIプログラム・チェンジを送信する

Tutorial 16      More MIDI ins and outs (英文PDF参照)

## 入力

**整数** 第1インレット：数値は1引かれ、指定されたチャンネルとポートにプログラム・チェンジの値として送信される。数値は1から128の範囲に限られており、プログラム・チェンジとしては0から127までの値として送信される。

第2インレット：数値は、プログラム・チェンジ・ナンバーとして送られるチャンネル・ナンバーとして格納される。

**実数** 整数に変換される。

**リスト** 第1インレット：最初の数値はプログラム・ナンバーに1が足された数値として、2番目の番号はチャンネル・ナンバーとして、MIDIプログラム・チェンジ・メッセージが指定されたチャンネルとポートから送信される。

**enable** **enable 0**メッセージはオブジェクトを使用不可にし、MIDIデータを送信しないように設定する。MIDI Enable/Disableボタンや**pcontrol**オブジェクトによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを動作可能にする。

**port** **port**の後に文字の**a**から**z**の文字を続けるか、MIDI入力デバイスの名前を続けることによって、プログラム・チェンジ・メッセージを送信するポートを設定する。**port**という単語はオプションで、省略することもできる。

## アーギュメント

**a-z** オプション。プログラム・チェンジのメッセージを送信するポートを指定する。文字のアーギュメントを指定している時に、チャンネル・ナンバーが16より大きければ、1から16の範囲に収まるように変換される。アーギュメントが存在しない場合は、**pgmout**は初期状態として、ポートaからMIDIチャンネル1として送信する。

**a-zと整数** (空白文字によって分けられる) 文字と数値の組み合わせは、プログラム・チェンジ・メッセージを送信するポートとMIDIチャンネルを指定する。チャンネル・ナンバーが16より大きければ、1から16の範囲に収まるように変換される。

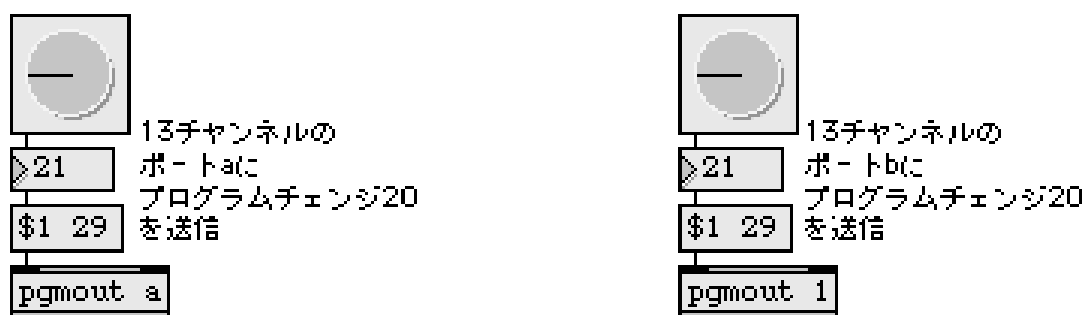
(MIDI出力デバイス名) オプション。MIDI出力デバイスの名前は、ポートを指定する最初のアーギュメントとして用いることができる。

**整数** 文字と数値の組み合わせの代わりに、数値だけを指定しても良い。チャンネル・ナンバーのアーギュメントの正確な意味は、MIDI Setupダイアログで指定したポートごとのチャンネル・オフセットによって決まる。

## 出力

**(MIDI)** アウトレットは存在しない。オブジェクトに指定したMIDI出力ポートから、直接MIDIプログラム・チェンジ・メッセージが送信される。

## 利用例



文字のアーギュメントであれば、1つのポートのみから送信する  
そうでなければ、数値はポートとチャンネルの両方を指定する

## 参照

**midout**  
**pgmin**

生のMIDIデータを送信する  
MIDIプログラム・チェンジを受信する

Tutorial 16

More MIDI ins and outs (英文PDF参照)

## 入力

第1インレット：オブジェクトに関連付けられたグラフィック・ウィンドウが表示されている状態であれば、指定したフレーム番号の画像を描く。**pics**オブジェクトでは、最初のフレーム番号は0である。あるフレーム番号の画像が描かれれば、次のフレームへポインターが移動する。従って、その後に**bang**のメッセージを続けて送れば、次のフレームが描かれることになる。

第2インレット：グラフィック・ウィンドウの左端からの相対位置で、画像の左端をピクセル単位で設定する（次にフレームが描かれる時から有効となる）。

第3インレット：グラフィック・ウィンドウの上端からの相対位置で、画像の上端をピクセル単位で設定する（次にフレームが描かれる時から有効となる）。

**bang** 次のフレームを描く。最後のフレームが描かれた後は、最初のフレームへ戻り、循環する。

**priority** **priority**の後に0より大きな数値が続き、オブジェクトが描くスプライトの優先順位を設定する。スプライトの優先順位については、マニュアルの**Tutorial and Topics**の**Graphics**のセクションを参照のこと。

## アーギュメント

**pics2** オプション。オブジェクトが**pics2**として作られた場合は、最初のフレームは、後に続くフレームによって消されることのない、背景フレームとして扱われる。PICSファイルのフォーマットでは、最初のフレームが何をするか定かではないので、**pics**で適切な結果が得られない場合は、オブジェクトを**pics2**として作成すべきである。

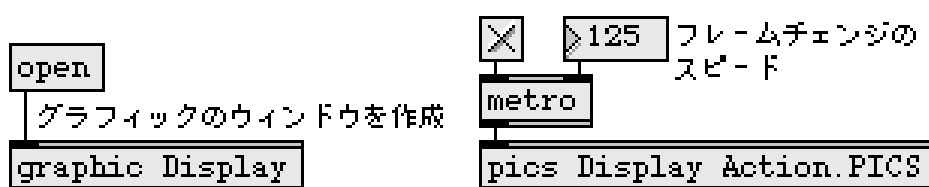
**シンボル** 必須。**pics**への最初のアーギュメントは、アニメーションが描かれるウィンドウの**graphic**オブジェクトの名前を指定する。2番目のアーギュメントは、オブジェクトが作成される時に読み込まれる、PICSフォーマットのファイルの名前を指定する。PICSファイルのサイズは通常とても大きく、ファイルを読み込むのに時間がかかるので注意すること。

**整数** オプション。ウィンドウの名前とファイルの名前の後に、0より大きな数値が続けば、スプライトの優先順位の初期値として設定される。優先順位のデフォルト値は3である。

## 出力

(表示) **pics**オブジェクトに関連付けられているグラフィック・ウィンドウが表示されている時に、**bang**メッセージ、または数値をインレットに受け取れば、PICSファイルのフレームがウィンドウに描かれる。以前に描かれたフレームがある場合は、それを消去する（ただし、スクリーンに常に最初のフレームを残すために、オブジェクトが**pics2**として作成されている場合は、これに該当しない）。

## 利用例



**metro** オブジェクトは、常に一定の速度でPICSファイルを循環させるのに使われる

## 参照

**graphic  
pict**

スプライトによるグラフィックを描くためのウィンドウ  
グラフィック・ウィンドウでピクチャーを描く



## 入力

- bang** 関連付けられたグラフィック・ウィンドウが表示されている場合は、**pict**オブジェクトが格納しているピクチャーを描く。
- clear** グラフィック・ウィンドウに描かれているピクチャーを消去する。
- 整数** 第1インレット：関連付けられたウィンドウが表示されている場合は、0でない数値によって、ピクチャーを描く。0はピクチャーを消去する。
- 第2インレット：グラフィック・ウィンドウの左端からの相対位置で、ピクチャーの左端をピクセル単位で設定する（次にピクチャーが描かれる時から有効となる）。
- 第3インレット：グラフィック・ウィンドウの上端からの相対位置で、ピクチャーの上端をピクセル単位で設定する（次にピクチャーが描かれる時から有効となる）。
- priority** **priority**の後に0より大きな数値が続き、オブジェクトが描くスプライトの優先順位を設定する。スプライトの優先順位については、マニュアルの**Tutorial and Topics**の**Graphics**のセクションを参照のこと。

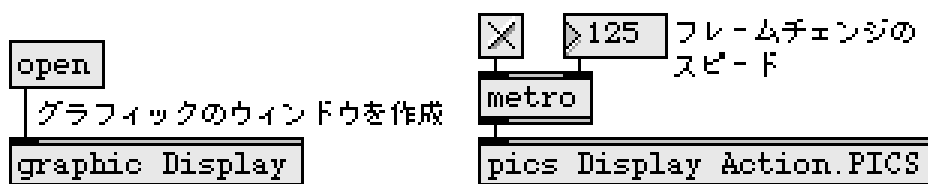
## アーギュメント

- シンボル** 必須。**pict**の最初のアーギュメントは、ピクチャーが描かれるウィンドウの**graphic**オブジェクトの名前を指定する。2番目のアーギュメントは、オブジェクトが作成される時に読み込まれる、グラフィック・ファイルの名前を指定する。
- 整数** オプション。ウィンドウの名前とファイルの名前の後に、0より大きな数値が続けば、スプライトの優先順位の初期値として設定される。優先順位のデフォルト値は0である。この場合、ピクチャーは他のすべてのオブジェクトより下に描かれる。優先順位の数値に続く、2つのアーギュメントは、グラフィック・ウィンドウの秒が領域の左上隅からの相対位置として、ピクチャーの左と上のオフセットの値を、ピクセルで指定する。

## 出力

- (表示) **pict**オブジェクトに関連付けられたグラフィック・ウィンドウが表示されている時に、**bang**メッセージ、または0でない整数がインレットに受け取れば、格納しているピクチャーをウィンドウに描く。

## 利用例



ピクチャーを表示し、グラフィック・ウィンドウ内を移動させることもできる

## 参照

|                |                           |
|----------------|---------------------------|
| <b>frame</b>   | グラフィック・ウィンドウに矩形の枠を描く      |
| <b>graphic</b> | スプライトによるグラフィックを描くためのウィンドウ |
| <b>lcd</b>     | パッチ・ウィンドウ内に画像を描画する        |
| <b>oval</b>    | グラフィック・ウィンドウに塗り潰した楕円形を描く  |
| <b>pics</b>    | グラフィック・ウィンドウでのアニメーション     |
| <b>rect</b>    | グラフィック・ウィンドウに塗り潰した矩形を描く   |
| <b>ring</b>    | グラフィック・ウィンドウに楕円形の枠を描く     |

Tutorial 42

Graphics (英文PDF参照)



**pictctrl**はボタン、スイッチ、ノブなどを作るためのユーザー・インターフェース・オブジェクトである。PICTファイルや、QuickTimeがインストールされていれば、QuickTimeでサポートされている他のファイル形式の画像ファイルを扱うことができる。**pictctrl**オブジェクトはピクチャー・ファイルの画像を用いて表示を作り出すので、望み通りの外見のコントロールを作ることができる。

## 入力

- 整数** ボタンやノブなどのコントロールの値を設定し、アウトレットから現在の値を出力する。ボタン・モードやトグル・モードでは、値は0か1になる。ダイアル・モードでの値の範囲は、**pictctrl**オブジェクトのRange属性によって決定される。
- set** **set**には1つの数値が続き、値の出力をせずに、ボタンやノブなどの値を設定する。
- bang** 現在の**pictctrl**の値を出力する。
- clickincrement** 0でない値が続く**clickincrement**メッセージは、このオブジェクトがクリックされる度に1ずつ値を加算して出力するように設定する。クリックした後のいかなるマウスの動きも無視される。最大値に達した時には、次のクリックで値を0に戻す。他のすべてのマウスの動きの検出は無効になる。**clickincrement 0**メッセージは、この加算を無効にする。
- clickesimage** 0でない値が続く**clickesimage**メッセージは、ユーザーがオブジェクト上でクリックして、マウス・ポインターをドラッグする時に、ダイヤルに異なる外観を与えるために、ピクチャー・ファイル内の1つおきのイメージ・フレームの集合を使うように**pictctrl**オブジェクトを設定する。**clickesimage 0**メッセージはこの機能を無効にする。
- picture** **picture**には、ファイル名を指定するシンボルが続き、**pictctrl**オブジェクトがボタンやダイアルのコントロールとして使う画像ファイルを指定する。ファイル名として用いられるシンボルは、Maxのサーチ・パスに含まれているか、または絶対パスで指定する必要がある。(例えば “MyDisk:/Documents/UI Pictures/Cool-Knob.pct”)
- active** **active**には0か1が続き、**pictctrl**オブジェクトのマウスによるコントロールを切り替える。デフォルト値は1 (有効) である。**pictctrl**オブジェクトがアクティブでない状態の画像を表示していて、inactive images属性が設定されていれば、この**active**メッセージは、コントロールの外見も変える。



- inactiveimage** 0でない値が続く **inactiveimage** メッセージはアクティブでない状態の画像の列を、指定したファイルが持っていることを **pictctrl** オブジェクトに指示する。デフォルト値は0（アクティブでない状態はない）である。
- imagemask** 0でない値が続く **imagemask** メッセージは、指定した画像がイメージ・マスクを持っていることを **pictctrl** オブジェクトに指示する。デフォルト値は0（イメージ・マスクなし）である。
- soundfx** **soundfx** には、0か1が続き、Mac OS 8.5以降に実装されているアピアランス・マネージャーのサウンド効果の使用を切り替える。 **soundfx 0** メッセージはサウンドをオフにする。
- tracking** **tracking** には0か1が続き、リアルタイムの位置検出を切り替える。リアルタイムの位置検出がオンになっていれば、 **pictctrl** オブジェクトは、マウスのボタンを押したままで、 **pictctrl** オブジェクトの矩形領域に入ったり、出たりした時に状態を変える。 **tracking 0** メッセージはその機能をオフにする。
- range** **range** には、1つの数値が続き、 **pictctrl** オブジェクトがダイアル・モードの時に、その範囲を設定する。デフォルト値は128である。
- offset** **offset** には、1つの数値が続き、オフセット値を設定する。 **pictctrl** オブジェクトがダイアル・モードの時には、ここで設定されたオフセット値は、アウトレットから出力される前にオブジェクトの値に加算される。デフォルトのオフセット値は0である。
- multiplier** **multiplier** には1つの数値が続き、乗数を設定する。 **pictctrl** オブジェクトがダイアル・モードの時に、オブジェクトの値は出力される前に、この値によって乗算される。乗算は **offset** で設定された値が加算される前に行われる。
- frame** **frame** には1つの数値が続き、ピクチャー・ファイルに含まれる画像の数（列）を指定する。フレームの数はコントロールの範囲と同じである必要はなく、 **pictctrl** オブジェクトは、与えられた値に従って最も近い画像を用いる。
- trackhorizontal** 0でない値が続く **trackhorizontal** メッセージは、オブジェクトをクリックし、水平方向にマウスをドラッグさせた時に反応するように設定する。マウスを右に動かすとオブジェクトの値を増加させ、左に動かすことで値を減少させる。このモードを有効にすれば、Circular Tracking と Click to Increment モードは無効になる（ **clickincreament** メッセージ、および **trackcirculer** メッセージを参照のこと）。



- trackvertical** 0でない値が続く**trackvertical**メッセージは、オブジェクトをクリックし、垂直方向にマウスをドラッグさせた時に反応するように設定する。マウスを上を動かすとオブジェクトの値を増加させ、下を動かすことで値を減少させる。このモードを有効にすれば、Circular TrackingとClick to Incrementモードは無効になる（**clickincrement**メッセージ、および**trackcirculer**メッセージを参照のこと）。
- trackcircular** 0でない値が続く**trackcircular**メッセージは、オブジェクトを、クリックしコントロールの中心から相対的に円形の弧上でマウスをドラッグさせた時に反応するように設定する（Circular Trackingモード）。マウスを時計回りに動かすとオブジェクトの値を増加させ、逆時計回りに動かすことで値を減少させる。このモードを有効にすれば、他のトラッキング・モードは無効になる。**trackcircular 0**メッセージはこのトラッキングを無効にする。
- ratio** **ratio**には1つの値が続き、何ピクセル、マウスを動かせば、ダイアルの値が変わるか指定する。**pictctrl**オブジェクトがcircularトラッキングを使用していれば、**ratio**メッセージはオブジェクトの中心から、相対的に何度カーソルを動かせば値が変わるかを指定する。

## インスペクター

**pictctrl**オブジェクトの振舞はインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**pictctrl**オブジェクトを選択することで、**pictctrl**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**pictctrl**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**pictctrl**オブジェクトの属性は、このオブジェクトの3つのモードである**Button**モード、**Toggle**モード、**Dial**モードのいずれかに関連付けられている。**pictctrl**オブジェクトのインスペクターは以下の属性をセットできる。

**Button**モードは、グラフィカル・ユーザー・インターフェース、例えばダイアログ・ボックスなどに見られる「OK」ボタンや「Cancel」ボタンなどの、単純なボタンの振舞いを模倣する。このモードでは、**pictctrl**オブジェクトは、このオブジェクト上でユーザーがクリックした時には1を、ユーザーがこのオブジェクト上からマウスを外に動かした時か、マウスのボタンを離れた時に0を出力する。ボタン・モードは、LEDオブジェクトや状態を表示するオブジェクトなどに似た、表示用オブジェクトの作成に役立つ。





*Toggle*モードは、クリックごとにオブジェクトの状態を0から1（また1から0に）切り替える以外は、*Button*モードと良く似ている。*Toggle*モードはチェック・ボックスの振舞いを模倣する。

*Live Tracking*をチェックすることは、**pictctrl**オブジェクトを*Button*モードにしている場合のみ可能である。このチェック・ボックスがチェックされている時は、マウスのボタンを押しながらオブジェクトの矩形領域の中へ入ったり、出たりするのに応じて、**pictctrl**オブジェクトはその状態を切り替える。

*Dial*モードは、ノブや2つ以上の異なる値を持つ他のコントロールの様に振舞うコントロールを作成することができる（*Dial*モードを用いることでスライダーを作ることができる。しかし、**pictslider**の方がこの用途には適している）。*Dial*モードでは、Maxの**hslider**オブジェクト、**uslider**オブジェクト、また**dial**オブジェクトのように範囲、オフセット、乗数を設定できる。このオブジェクトをクリックし、ドラッグした時に、その値が変化する。**pictctrl**オブジェクトは、水平、垂直それぞれ、または両方のカーソルの動き、あるいは、円弧上でのカーソルの動きを追うことができる。ドラッグした後に続く動きは無視する。*dial*モードを使用する場合、使用するピクチャー・ファイル（下記を参照のこと）に含まれるイメージのフレームの数を指定しなくてはならない。画像の数は、値の範囲と一致する必要はない。例えばノブは範囲を128まで設定できるが、画像については30しか割り当てることができない。その範囲にそれ以上の数のフレームを割り当ててコントロールを作っても、何の出力もせずその外観のみを変えることになるので、意味がない。

**pictctrl**オブジェクトが*Dial*モードである場合、インレットに受け取った数値を自動的に0から指定した値より1小さい数値までに制限する *Range*、出力される前にすべての数値を指定された値で乗算する *Multiplicier*、そして乗算後に加算される *Offset*のそれぞれの値を指定できる。それぞれのデフォルト値は、*Range*は128、*Multiplicier*は1、そして *Offset*は0である。

*Image Frame*のボックスでは、ピクチャー・ファイル内に含まれる別個の画像の数（列）を指定する。フレームの数は、コントロールの範囲と一致する必要はなく、**pictctrl**オブジェクトは、与えられた値に従って最も近い画像を用いる。

*Horizontal Tracking*か*Vertical Tracking*がチェックされている場合は、それぞれ対応する方向にマウスをドラッグさせた時に反応するように**pictctrl**オブジェクトは設定される。マウスを右、または上にドラッグするとオブジェクトの値を増加させ、左、または下にドラッグすることで値を減少させる。このどちらかの属性を有効にすることは、*Circular Tracking*と*Click to Increment*モード（以下を参照）を無効にすることになる。



*Circular Tracking*がチェックされている場合は、オブジェクトをクリックし、コントロールの中心から、相対的に円形の弧上でマウスをドラッグさせた時に反応するように、**pictctrl**オブジェクトは設定される。マウスを時計回りに動かすとオブジェクトの値を増加させ、逆時計回りに動かすことで値を減少させる。*Circular Tracking*を有効にすることは、他のトラッキング・モードを無効にすることになる。

*Click to Increment*がチェックされている場合は、このオブジェクトがクリックされる度に、コントロールの値を1ずつ増加させるように設定する。クリックした後のいかなるマウスの動きも無視される。最大値に達した時は、次のクリックで値を0に戻す。*Click to Increment*を有効にすることは、他のトラッキング・モードを無効にすることになる。

*Clicked Image*がチェックされている場合は、ユーザーがオブジェクト上でクリックして、マウス・ポインターをドラッグする時、ダイアルに異なる外観を与えるために、ピクチャー・ファイル内の1つおきのイメージ・フレームの集合を使うように**pictctrl**オブジェクトを設定する。

*Tracking Ratio*属性は、何ピクセル、マウスを動かせば、ダイアルの値が変わるかを指定する。**pictctrl**オブジェクトがcircularトラッキングを使用していれば、**ratio**メッセージはオブジェクトの中心から、相対的に何度カーソルを動かせば値が変わるかを指定する。

*Has Inactive Image*と*Image Masks*のチェック・ボックスは、アクティブでない状態の画像の列、およびイメージ・マスクを指定したファイルが持っていることを指定する。

*Audible Feedback*がチェックされている場合は、クリックされた時に、Mac OS アピアランス・マネージャーのサウンド効果を鳴らすように設定する。

*Picture File*オプションでは、Openボタンを押すことによって、**pictctrl**オブジェクトのノブに使用するピクチャー・ファイルを選ぶことができる。現在、使用されているファイル名はボタンの左にあるテキスト・ボックスに表示される。このボックスにファイル名を直接入力することや、Finderからファイルのアイコンをこのボックスへドラッグしてこることでファイルを選択できる。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。





## アーギュメント

なし。

## 画像ファイルの形式

パッチ・ウィンドウ内に新しい **pictctrl** オブジェクトを作った時は、画像ファイルは設定されていない。インスペクターのOpenボタンを使うことで、コントローラーに画像ファイルを選択できる。PICTファイルか、QuickTimeがインストールされていれば、QuickTimeでサポートされているファイル形式のファイルを使用することができる。ピクチャー・ファイルのレイアウトは、**pictctrl** オブジェクトで使用されるモードによって異なる。3つのモードすべてにおいて、ピクチャー・ファイルは、同じ幅と高さを持つイメージのグリッドであることが必要である。

*Button*モードは単純なレイアウトを持つ。

|                                  |                               |
|----------------------------------|-------------------------------|
| Not Clicked<br>value = 0         | Clicked<br>value = 1          |
| inactive<br>value = 0            | inactive<br>value = 1         |
| Not-Clicked<br>Mask<br>value = 0 | Clicked<br>Mask<br>value = 1  |
| inactive<br>Mask<br>value = 0    | inactive<br>Mask<br>value = 1 |

1つ目の画像の列は必須である。その2つの画像は、ボタンのアイドル時、またクリック時の状態（対応する値は0、また1）の時に使われる。次の画像の列は、コントロールがアクティブでない状態の時に使われる。次に続く列は、それぞれ、1つ目の列やアクティブでない状態の時の画像の列のマスクとして使用される。



Toggleモードも似たレイアウトを持っている。

|                                  |                               |
|----------------------------------|-------------------------------|
| Not Clicked<br>value = 0         | Clicked<br>value = 0          |
| Not Clicked<br>value = 1         | Clicked<br>value = 1          |
| Inactive<br>value = 0            | Inactive<br>value = 1         |
| Not-clicked<br>Mask<br>value = 0 | Clicked<br>Mask<br>value = 0  |
| Not-clicked<br>Mask<br>value = 1 | Clicked<br>Mask<br>value = 1  |
| Inactive<br>Mask<br>value = 0    | Inactive<br>Mask<br>value = 1 |

このモードでは、先頭の2つの列が必須である。1つ目の画像の列はコントロールの値が0の時に使用され、次の画像の列はその値が1の時に使用される。3つ目の画像の列はオプションであり、コントロールがアクティブでない状態の時に使われる。（注記：アクティブでない状態の“クリック時”の画像は必要ではない。アクティブでない状態では、コントロールはマウスのクリックを無視する。）次の列は、それぞれの画像のマスクとして使用される。



Dialモードのレイアウトは、いくつかのイメージ・フレームを持っているかによって異なる。また、それらはインスペクターのImage Framesのパラメーターと一致している必要がある。

|                                |                                |     |                                  |
|--------------------------------|--------------------------------|-----|----------------------------------|
| Not Clicked<br>image 0         | Not Clicked<br>image 1         | ... | Not Clicked<br>image n-1         |
| Clicked<br>image 0             | Clicked<br>image 1             | ... | Clicked<br>image n-1             |
| Inactive<br>image 0            | Inactive<br>image 1            | ... | Inactive<br>image n-1            |
| Not-clicked<br>Mask<br>image 0 | Not-clicked<br>Mask<br>image 1 | ... | Not-clicked<br>Mask<br>image n-1 |
| Clicked<br>Mask<br>image 0     | Clicked<br>Mask<br>image 1     | ... | Clicked<br>Mask<br>image n-1     |
| Inactive<br>Mask<br>image 0    | Inactive<br>Mask<br>image 1    | ... | Inactive<br>Mask<br>image n-1    |

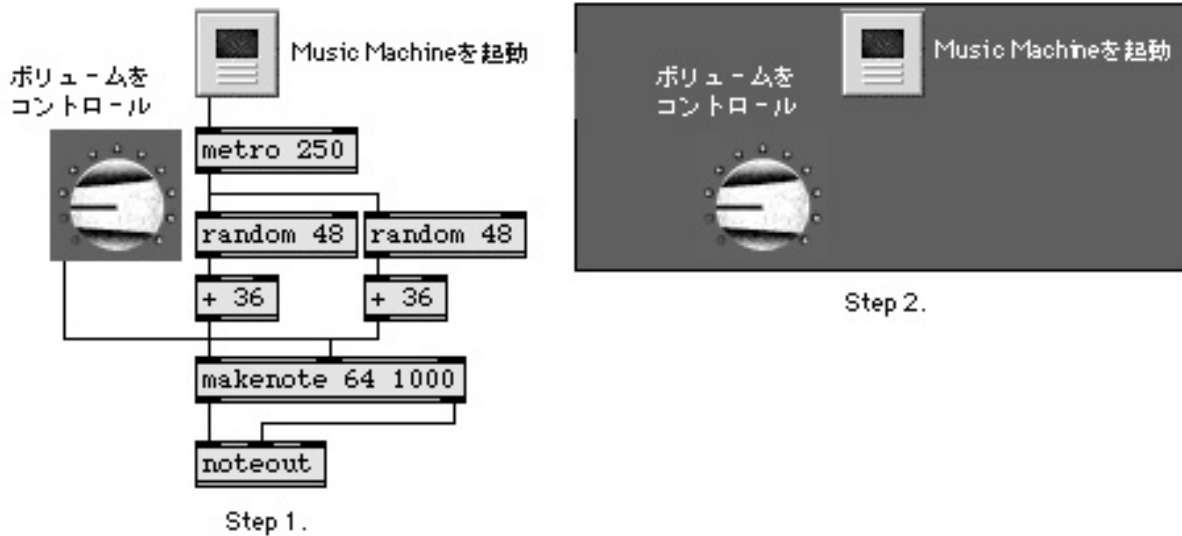
1つ目の画像の列はコントローラの視覚的に異なったそれぞれの状態を表す画像として、必須である。ダイヤルは、必要とされる視覚的に異なる状態と同じ数だけの画像が必要とされる。ダイヤルは画像によって表示される数よりも、大きい値の入出力が可能であることに注意する（例えば、8つの画像しかその範囲を表現するのに使用しなかったとしても、ダイヤルは128の範囲を持つことができる）。次の画像の列はオプションであり、ユーザーが値を変えるためにそのオブジェクト上でクリックし、ドラッグした時に使用される。その次の画像の列もまたオプションである。（注記：アクティブでない状態の“クリック時”の画像は必要ではない。アクティブでない状態では、コントロールはマウスのクリックを無視する。）続く列は、それぞれの画像のマスクとして使用される。

## 出力

**整数** 現在のpictctrlオブジェクトの値を出力する。ToggleやButtonモードでは、0か1になる。Dialモードでは、その値はインスペクターで設定された範囲、オフセット値、乗数によって決められる。



## 利用例



より面白いユーザー・インターフェースを作るためにカスタマイズされたコントロールを作る

## 参照

|                   |                            |
|-------------------|----------------------------|
| <b>dial</b>       | ダイアル操作によって数値を出力する          |
| <b>hslider</b>    | 画面上のスライダーを動かすことによって数値を出力する |
| <b>kslider</b>    | 画面上のキーボードから数値を出力する         |
| <b>matrixctrl</b> | マトリックス状のスイッチ・コントロール        |
| <b>pictslider</b> | ピクチャーを用いたスライダー・コントロール      |
| <b>rslider</b>    | 数値の範囲を変更し、それを表示する          |
| <b>slider</b>     | 画面上でスライダーを動かすことによって数値を出力する |
| <b>ubutton</b>    | 透明なボタン、 <b>bang</b> を出力する  |
| <b>uslider</b>    | 画面上でスライダーを動かすことによって数値を出力する |

Tutorial 12

Sliders and dials (英文PDF参照)



**pictslider**オブジェクトは、外部ファイルのピクチャーを使用して、その外観をカスタマイズできるスライダー・コントロールである。このオブジェクトは2つの画像を使用する。1つはノブであり（マウスを用いて動かす部分、実際のスライダーで指を使って動かす部分に相当する）、もう1つはノブが動く背景となる部分となる。**pictslider**オブジェクトは、他のピクチャーを使用しようと望まなかった時のためのデフォルトのピクチャーを持っているが、本来意図されている使用法は、カスタマイズされた外観のコントロールをつくることである。

**pictslider**オブジェクトは、水平、垂直方向のスライダー、つまり2次元のコントロール（仮想トラック・パッドか、ジョイスティック）を作るためにも用いることができる。

## 入力

**bang** 第1インレット：現在の**pictslider**オブジェクトの値をアウトレットから出力する。水平方向の値は第1アウトレットから、垂直方向の値は第2アウトレットから出力される。

**整数** 第1インレット：**pictslider**オブジェクトの水平方向の値を設定する。また、値は第1アウトレットから、垂直方向の値も第2アウトレットから出力される。

第2インレット：**pictslider**オブジェクトの垂直方向の値を設定する。また、値は第2アウトレットから、**pictslider**オブジェクトの水平方向の値も第1アウトレットから出力される。

**実数** 整数に変換される。

**リスト** 第1インレット：第1インレットに送られた2つの数値によるリストは、1つ目の値は水平方向の値を、2つ目の値は垂直方向の値を設定する。それらの2つの値は第1、第2アウトレットから出力される。

**active** 第1インレット：**active**には0か1が続き、**pictslider**オブジェクトをマウスで制御できるかどうかを切り替える。デフォルト値は1（有効）になっている。**pictctrl**オブジェクトの画像ファイルにアクティブでない状態の画像のセットが用意されていて、Inactive Images属性がセットされていれば、**active**メッセージはコントロールの外観もまた、変化させる。

**bkgndpicture** **bkgndpicture**にはファイル名を指定するシンボルが続き、**pictslider**オブジェクトがコントロールの背景として使用する画像ファイルを指定する。ファイル名として使用するシンボルはMaxのサーチ・パス内のファイル名か、ファイルのフル・パス名（例えば“MyDisk:/Documents/UIPictures/CoolBkgnd.pct”）でなくてはならない。



|                           |                                                                                                                                                                                |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>bkgndsize</b>          | 第1インレット：0でない値が続く <b>bkgndsize</b> メッセージは、背景の画像の大きさにオブジェクトの大きさを合わせるように設定する。このメッセージを受け取った後には、オブジェクトの大きさは変更できなくなる。 <b>bkgndsize 0</b> メッセージは左下の隅をドラッグする通常の方法でのコントロールのサイズ変更が可能になる。 |
| <b>bottommargin</b>       | 第1インレット： <b>bottommargin</b> には0または0より大きい1つの整数が続き、 <b>pictslider</b> オブジェクトの下の余白をピクセル単位で指定する。余白はノブの動く範囲を制限する。余白が0に設定されれば、ノブはスライダーの下まですべて動くことができる。                              |
| <b>bottomvalue</b>        | 第1インレット： <b>bottomvalue</b> には1つの整数が続き、可能な限り下までノブが動かされた時に <b>pictslider</b> オブジェクトから出力される値を設定する。 <b>bottomvalue 100</b> メッセージは、ノブが最も下まで動かされた時にコントロールから100を第1アウトレットから出力させる。     |
| <b>clickedimage</b>       | 第1インレット：0でない値が続く <b>clickedimage</b> メッセージは、 <b>pictslider</b> オブジェクトで使用されている画像ファイルが、コントロールがクリックされた時に表示される画像を含んでいることを指定する。                                                     |
| <b>horizontaltracking</b> | 第1インレット： <b>horizontaltracking</b> には実数が続き、マウスの動きに対する、 <b>pictslider</b> オブジェクトのノブの動きの割合を設定する。デフォルト値は1.0。1より大きな値はドラッグした時にノブをより早く動かす。1より小さい値はよりゆっくりと動かす。                        |
| <b>imagemask</b>          | 第1インレット：0でない値が続く <b>imagemask</b> メッセージは <b>pictslider</b> オブジェクトで使われる画像ファイルが、マスク画像を含んでいることを指定する。                                                                              |
| <b>inactiveimage</b>      | 第1インレット：0でない値が続く <b>inactiveimage</b> メッセージは <b>pictslider</b> オブジェクトで使われる画像ファイルが、オブジェクトがアクティブでない状態の画像を含んでいることを指定する。                                                           |
| <b>invisiblebkgnd</b>     | 第1インレット：0でない値が続く <b>invisiblebkgnd</b> メッセージは、 <b>pictslider</b> オブジェクトに背景画像を描画しないことを指定する。ノブはその下のオブジェクトなどの上に浮いているように見える。                                                       |
| <b>jump</b>               | 第1インレット：0でない値が続く <b>jump</b> メッセージは、ノブの外側でオブジェクトがクリックされた時にも、ノブをカーソルの位置に <b>pictslider</b> オブジェクトに動かさせる。 <b>jump 0</b> メッセージはこの振る舞いを無効にする。ノブを動かすためには、そのノブそのものをクリックしなくてはならない。    |
| <b>knobpicture</b>        | <b>knobpicture</b> にはファイル名を指定するシンボルが続き、 <b>pictslider</b> オブジェクトがコントロールのノブとして使用する画像ファイルを指定する。                                                                                  |





ファイル名として使用するシンボルはMaxのサーチ・パス内のファイル名か、ファイルのフル・パス名（例えば“**MyDisk:/Documents/UI Pictures/Cool-Knob.pct**”）でなくてはならない。

- leftmargin** 第1インレット：**leftmargin**には0または0より大きい1つの整数が続き、**pictslider**オブジェクトの左の余白をピクセル単位で指定する。余白はノブの動く範囲を制限する。余白が0に設定されれば、ノブはスライダーの左まですべて動くことができる。
- leftvalue** **leftvalue**には1つの整数が続き、可能な限り左までノブが動かされた時に**pictslider**オブジェクトから出力される値を設定する。**leftvalue 100**メッセージは、ノブが最も左まで動かされた時にコントロールから100を第1アウトレットから出力させる。
- movehorizontal** 第1インレット：0でない値が続く**movehorizontal**メッセージは、マウスが水平方向に動かされた時にノブを動かすことを可能にする。**movehorizontal 0**メッセージは、マウスが水平方向に動かされた時にノブが動くことを妨げる。
- movevertical** 第1インレット：0でない値が続く**movevertical**メッセージは、マウスが垂直方向に動かされた時にノブを動かすことを可能にする。**movevertical 0**メッセージは、マウスが垂直方向に動かされた時にノブが動くことを妨げる。
- rightmargin** 第1インレット：**rightmargin**には0または0より大きい1つの整数が続き、**pictslider**オブジェクトの右の余白をピクセル単位で指定する。余白はノブの動く範囲を制限する。余白が0に設定されれば、ノブはスライダーの右まですべて動くことができる。
- rightvalue** 第1インレット：**rightvalue**には1つの整数が続き、可能な限り右までノブが動かされた時に**pictslider**オブジェクトから出力される値を設定する。**rightvalue 100**メッセージは、ノブが最も右まで動かされた時にコントロールから100を第1アウトレットから出力させる。
- scaleknob** 第1インレット：0でない値が続く**scaleknob**メッセージは、**pictslider**オブジェクトを、このオブジェクトの大きさが変更された時に、それに従って、ノブの大きさも拡大伸縮させるように設定する。**scaleknob 0**メッセージは、ノブを常に元の大きさのまま描画させる。
- set** 第1インレット：**set**には数値が続き、**pictslider**オブジェクトの水平方向の値をセットするが、第1アウトレットからの値の出力は行わない。2つの数値が続く**set**メッセージは、**pictslider**オブジェクトの水平方向の値をリストの1つ目の数値に、垂直方向の値を2つ目の数値に設定するが、アウトレットからの値の出力は行わない。





第2インレット：**set**には数値が続き、**pictslider**オブジェクトの垂直方向の値をセットするが、第2アウトレットからの値の出力は行わない。

**soundfx** 第1インレット：**soundfx**には0か1が続き、Mac OS 8.5以降に実装されているアピアランス・マネージャーのサウンド効果の使用を切り替える。**soundfx 0**メッセージはサウンドをオフにする。

**topmargin** 第1インレット：**topmargin**には0または0より大きい1つの整数が続き、**pictslider**オブジェクトの上の余白をピクセル単位で指定する。余白はノブの動く範囲を制限する。余白が0に設定されれば、ノブはスライダーの上まですべて動くことができる。

**topvalue** 第1インレット：**topvalue**には1つの整数が続き、可能な限り上までノブが動かされた時に**pictslider**オブジェクトから出力される値を設定する。**topvalue 100**メッセージは、ノブが最も上まで動かされた時にコントロールから**100**を第1アウトレットから出力させる。

**track** 第1インレット：**track**には実数が続き、マウスの動きに対する**pictslider**オブジェクトのノブの水平方向の動きの割合を設定する。

第2インレット：**track**には実数が続き、マウスの動きに対する**pictslider**オブジェクトのノブの垂直方向の動きの割合を設定する。

**verticaltracking** 第1インレット：**verticaltracking**には実数が続き、マウスの動きに対する、**pictslider**オブジェクトのノブの動きの割合を設定する。デフォルト値は1.0。1より大きな値はドラッグした時にノブをより早く動かす。1より小さい値はよりゆっくりと動かす。

## インスペクター

**pictslider**オブジェクトの振舞はインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**pictslider**オブジェクトを選択することで、**pictslider**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**pictslider**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでもインスペクターを表示することができる。

**pictslider**オブジェクトのインスペクターでは以下の属性を設定できる。

*Margin*のナンバー・ボックスは、**pictslider**オブジェクトのそれぞれに対応する余白をピクセル単位で設定する。余白はノブの動く範囲を制限する。余白が0に設定されれば、ノブはスライダーのそれぞれ対応する端まですべて動くことができる。例えば、左の余白を5に設定した場合、ノブはスライダーの左の端から5ピクセルの近さまでしか動くことができない。



*Value*のナンバー・ボックスは、それぞれ対応する方向に可能な限りノブが動かされた時に**pictslider**オブジェクトから出力される値を設定する。例えば、右手のナンバー・ボックスに**100**を設定すると、ノブが最も右まで動かされた時にコントロールから**100**を第1アウトレットから出力させる。(第1アウトレットはノブの水平方向の動きの値を出力するため、この値は第1アウトレットから出力される。) 中間の位置にあるノブの値は、左と右、また上と下の値をそれぞれ補完し算出される。それぞれの値のペアのうち、どちらでも大きく設定できる。例えば、上の値を**-100**に下の値を**50**に設定した場合、ノブを下から上に動かした時には垂直方向の値は50から-100まで減らされる。

*Move Horizontal*か、*Move Vertical*チェック・ボックスがチェックされている場合、ノブはマウスを用いて、クリックしドラッグすることで対応した方向に動かすことができる。従来のような横、または縦のみに動くスライダーを作りたい場合は、適切なチェックボックスのみチェックし、他はチェックしないままにする。

*Audible Feedback*オプションをチェックした場合、**pictslider**オブジェクトはクリックされた時に、Mac OS 8.5以降に実装されているアピラランス・マネージャーのサウンド効果を鳴らす。

*Knob Jump to Click Location*オプションを選択した場合は、**pictslider**オブジェクトの矩形範囲内のどこかをクリックした場合、その場所へノブをジャンプさせる。チェックされていない場合は、ノブを動かすためには、ノブそれ自体をクリックしてドラッグしなければならない。

*Has Inactive Images*チェック・ボックスは**pictslider**オブジェクトで使われる画像ファイルが、オブジェクトがアクティブでない状態の画像を含んでいることを指定する。そのコントロールに使用されるピクチャ・ファイルがその画像を含んでいなければ、そのボックスをチェックしないままにする。

*Tracking Ratio*の値はマウスの動きに対する、ノブの反応を決定する。デフォルト値は1.0。1より大きな値はドラッグした時にノブをより早く動かす。1より小さい値はよりゆっくりと動かす。

インスペクター内の以下の4つの属性のリストは、スライダーのノブの外観を変える。Openボタンをクリックすることで、スライダーのノブに使用される画像ファイルを選択することができる。現在使用されているファイル名は、ボタンの左のテキスト・ボックスに表示される。このボックスに直接、ファイル名を入力するか、Finderからファイルのアイコンをこのボックスにドラッグしてこることもファイルを選択することができる。



*Scale Knob When Control Size Changes* オプションをチェックすることで、**pictslider** オブジェクトをサイズ変更した時に、ノブの画像をオブジェクトの矩形の範囲、また背景の画像と相対的に、拡大縮小する。チェックしなければ、ノブの画像は常に元の大きさと描画される。伸縮した画像はむらがあり、不均等に見えるようになるので、大抵の場合はノブが、ノブに指定した画像の大きさと描画されて欲しいと思うだろう。このノブの伸縮の属性は、ノブの画像ファイルを書き直す必要なく **pictslider** オブジェクトのサイズとレイアウトを実験することに役立つ。

*Clicked Image* オプションをチェックすることは、**pictslider** オブジェクトで使用されている画像ファイルが、コントロールがクリック、またドラッグされた時にノブの異なる外観を与えるための画像のフレームを含んでいることを指定する。

ノブが描画される時に、ノブとして使用される画像ファイルにマスクをかけたいと望む場合は、*Image Mask* オプションを選択する。マスクは矩形ではない形のノブを作る時に利用できる。ノブとして使用される画像が、クリックされた時、またはアクティブでない状態の時の画像のセットを持っているのなら、マスクはそれぞれに用意しなければならない。

インスペクター内の以下の3つの属性のリストは、スライダーの背景の外観を変える。Open ボタンをクリックすることで、スライダーの背景に使用される画像ファイルを選択することができる。現在使用されているファイル名は、ボタンの左のテキスト・ボックスに表示される。このボックスに直接、ファイル名を入力するか、Finder からファイルのアイコンをこのボックスにドラッグしてこることもファイルを選択することができる。

*Size Control to Background Image* がチェックされている場合は、**pictslider** オブジェクトのサイズを、背景として選択された画像の大きさに合うように調整する。この属性が有効な場合、このオブジェクトの右下をクリックし、ドラッグするという通常の方法によるオブジェクトのサイズ変更はできなくなる。このサイズは固定になる。チェックされていない場合は、この画像はスライダーの大きさに合わせ、拡大縮小される。伸縮した画像はむらがあり、不均等に見えるようになるので、大抵の場合はスライダーが、スライダーに指定した画像の大きさと描画されて欲しいと思うだろう。この伸縮の属性をチェックしないことは、スライダーの画像ファイルを書き直す必要なく **pictslider** オブジェクトのサイズとレイアウトを実験することに役立つ。

*Invisible Background* ボックスをチェックすることは、**pictslider** オブジェクトに背景画像を描画しないことを指定する。ノブはその下のオブジェクトなどの上に“浮いて”いるように見える。



Revertボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

なし。

## 画像ファイルの形式

**pictslider**オブジェクトは2つの画像ファイルを使用する。1つは背景のために、もう1つはマウスによって背景上を動かされるノブのためのものである。

背景の画像ファイルはPICTファイルか、QuickTimeがインストールされていれば、QuickTimeでサポートされているファイル形式のファイルを使用することができる。背景の画像ファイルは以下のレイアウトになっていなければならない。

|                 |                   |
|-----------------|-------------------|
| Normal<br>Image | Inactive<br>Image |
|-----------------|-------------------|

1つの画像のみが必須である。1つの画像しか提供されなかった場合、それはすべての状態の背景の描画に使用される。追加される画像は1つ目の画像の右に位置させる。コントロールがアクティブでない状態の時の画像を加えることができる。アクティブでない状態の時の画像はコントロールが **active 0**メッセージを受け取った後に使用される。

ノブのファイルは以下のレイアウトのPICT形式のファイルでなくてはならない。

|                      |                  |                   |
|----------------------|------------------|-------------------|
| Not-Clicked<br>Image | Clicked<br>Image | Inactive<br>Image |
| Not-Clicked<br>Mask  | Clicked<br>Mask  | Inactive<br>Mask  |

画像は1つ以上の画像のグリッドで構成される。すべての画像は同じ幅と高さをもつ。





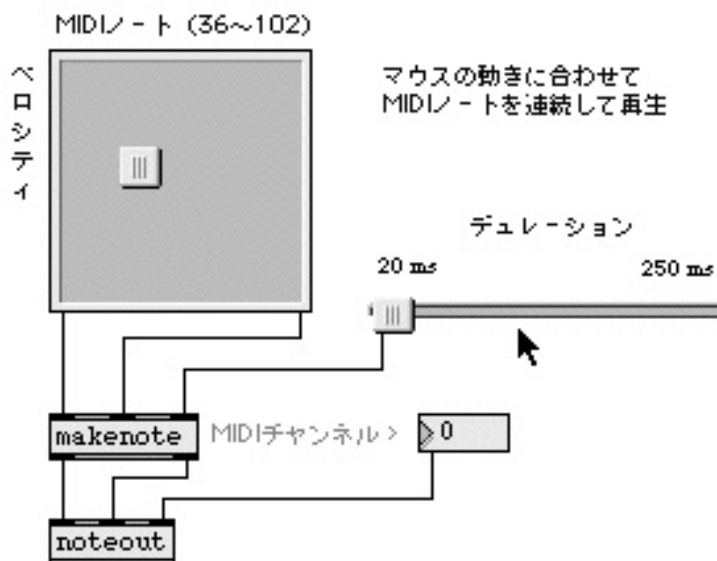
1つの画像のみが必須である。1つの画像しか提供されなかった場合、それはすべての状態のノブの描画に使用される。追加される画像は1つ目の画像の右に位置させる。“クリック時”、またコントロールがアクティブでない状態の画像を両方加えることができる。“クリック時”の画像は、ユーザーがコントロールのノブをドラッグした時に表示される。アクティブでない状態の時の画像は、コントロールが**active 0**メッセージを受け取った後に使用される。

マスク画像は、矩形ではない輪郭のノブを作る時に使用される。それらのマスクは、ファイル内のそれぞれに対応する画像に直接適用される。それぞれのノブの画像にマスクをかけたいと思うなら、それらすべてにマスクを用意する必要がある。それぞれの画像は対応するマスク画像の列を持っている。マスク画像内の黒いピクセルは対応する画像を描画する範囲を作り、白いピクセルは見えない範囲を作る。

## 出力

**整数** マウスを用いてクリック、またドラッグすることによって、スライダーのノブを動かすことか、オブジェクトのどちらかのインレットに値を送ることは、第1アウトレットから水平方向の値を、第2アウトレットから垂直方向の値を出力させる。入力される値は、インスペクター内で設定される上下、左右の値によって決定された範囲内に制限される。

## 利用例



pictslider オブジェクトは1次元、2次元両方の要素のユーザー・インターフェースを作れる



## 参照

|                 |                            |
|-----------------|----------------------------|
| <b>dial</b>     | ダイアル操作によって数値を出力する          |
| <b>hslider</b>  | 画面上のスライダーを動かすことによって数値を出力する |
| <b>kslider</b>  | 画面上のキーボードから数値を出力する         |
| <b>pictctrl</b> | ピクチャーを用いたコントロール            |
| <b>rslider</b>  | 数値の範囲を変更し、それを表示する          |
| <b>slider</b>   | 画面上でスライダーを動かすことによって数値を出力する |
| <b>ubutton</b>  | 透明なボタン、 <b>bang</b> を出力する  |
| <b>uslider</b>  | 画面上でスライダーを動かすことによって数値を出力する |

Tutorial 14

Slider and dials (英文PDF参照)

## 入力

- 整数** 第1インレット：第1アウトレットから出力する前に、指定したミリ秒分遅らせて、数値を出力する。第3インレットがあれば、第2インレットに受け取る数値も遅らせて、第2アウトレットから出力する。
- 整数または実数** 右インレット：他のインレットに受け取った数値を遅らせて出力する時間を、ミリ秒単位で設定する。
- bang** 第1インレット：（既に遅らせている数値があれば、それらに加えて）指定したミリ秒分遅らせて、現在**pipe**が保持している数値を再度出力する。
- 実数** 第1インレットと中央のインレット：実数のアーギュメントを指定していない限り、整数に変換される。
- リスト** 第1インレット：複数の数値を等しく遅らせるように、**pipe**のインレットに受け取る。右インレットに対応する数値があれば、それ以外の数値を遅らせる時間を設定する。
- clear** 第1インレット：その時点で**pipe**が遅らせている数値を消去する。
- clock** 既存の**setclock**オブジェクトの名前が続く**clock**メッセージは、Maxの内部ミリ秒クロックではなく、その**setclock**オブジェクトに従って、**pipe**が動作するように設定する。**clock**メッセージのみの場合は、Maxの通常のミリ秒クロックに従うように設定する。
- flush** 第1インレット：その時点で**pipe**が遅らせているすべての数値を、すぐさま出力し、**pipe**のメモリーを消去する。これらの数値は、インレットに対応するアウトレットから、受け取った順序とは逆の順序で出力する。

## アーギュメント

- 整数** オプション。最後のアーギュメントは、ミリ秒単位でディレイ時間の初期値を設定する。アーギュメントを指定しなければ、ディレイ時間は0である。2つのアーギュメントがある場合は、最初のアーギュメントは**pipe**に格納される数値の初期値を設定し、2番目のアーギュメントはディレイ時間を設定する。2つ以上のアーギュメントがある場合は、**pipe**はインレットとアウトレットを追加し、第1インレットに数値を受け取った時に、追加の数値も遅らせて出力する。
- 実数** 最後のアーギュメントは整数に変換される。他の実数のアーギュメントは、対応するアウトレットから実数を出力するよう設定する。



## 出力

**整数** **pipe**オブジェクトの第1インレットに数値を受けば、指定した時間だけ遅らせて、第1アウトレットから出力する。中央にインレットがあれば、第1インレットに数値を受け取った時点から、それらのインレットに受け取った数値も遅らせて、対応するアウトレットから出力する。**delay**オブジェクトとは異なり、**pipe**は同時に複数の数値を遅らせることができる。右インレットに新しいディレイ時間を受け取った時は、既に遅らせている数値には影響はしない。

## 利用例



**pipe**は1つ、または複数の数値を遅らせることができる

## 参照

**delay** **bang**を遅らせて出力する

Tutorial 22 Delay lines (英文PDF参照)



## 入力

**bang** **playbar**オブジェクトの第1アウトレットが、**movie**または**imovie**オブジェクトに繋がっている場合に、**bang**はそれらのオブジェクトを結び付け、**playbar**がQuickTimeムービーを操作できるようにする。**playbar**と**movie**とを結び付ければ、**playbar**を操作することで、再生位置や再生状態を変化させるメッセージが、**movie**オブジェクトに送られる。(playbarとmovieが繋がったパッチ・ファイルを読み込めば、それらは自動的に結び付けられる。従って、playbarにbangを送ることは、パッチを作っている時だけ必要である。)

## アーギュメント

なし

## 出力

(内部) 第1アウトレット：**playbar**と**movie**オブジェクトが結び付けられれば、**playbar**はQuickTimeムービーをコントロールする。**playbar**は、1つの**movie**オブジェクトだけに繋ぐようになっている。この接続は、パッチ・コードを使って行わなければならない。従って、**send**オブジェクトと**receive**オブジェクトを使って接続することはできない。

**整数** 第2アウトレット：**playbar**によって処理されたコマンドは、数値として第2アウトレットから出力される。コマンド番号の一覧とその意味は、QuickTime Standard Movie Play Controllerというドキュメントに書かれている。これらのコマンドを適切に解釈することによって、ムービーの操作だけでなく、他の目的のために**playbar**を用いることができる。しかし、長さが0ではないQuickTimeムービーが**playbar**に結び付けられない限り、再生範囲が設定されないので、コントローラーのツマミを動かすことはできない。

## 利用例



movieやimovieとともにplaybarを用いる



**参照**

**movie**  
**imovie**

QuickTimeムービーをウィンドウ内で再生する  
パッチ・ウィンドウ内でQuickTimeムービーを再生する

## 入力

**実数** 第1インレット：極座標における振幅（音量）部分。これは直交座標（実数部／虚数部）に変換される。

第2インレット：極座標における位相部分。これは直交座標（実数部／虚数部）に変換される。

## アーギュメント

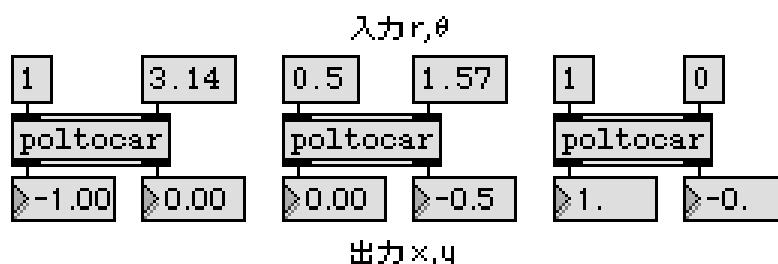
なし

## 出力

**実数** 第1アウトレット：直交座標（周波数領域）における実数部。

第2アウトレット：直交座標（周波数領域）における虚数部。

## 利用例



極座標を直交座標に変換する

## 参照

- cos** コサイン関数
- cartopol** 直交座標を極座標に変換する
- lcd** パッチ・ウィンドウ内に画像を描画する
- sin** サイン関数

## 入力

- リスト** 第1インレット：ノート・ナンバーとベロシティの組み合わせとして、1番目の数値はノート・ナンバー、2番目の数値はベロシティ値として扱われる。ベロシティが0でなければ、そのノート・オンのノート・ナンバーにボイス番号を割り当てて出力する。ベロシティが0である場合、ノート・オフとして、すでに保持されているそのノート・ナンバーのボイス番号を出力し、それに割り当てられたボイス番号を解除する。
- 整数** 第1インレット：ノート・ナンバーとベロシティの組み合わせのノート・ナンバーとして出力される。
- 第2インレット：数値はベロシティとして格納され、第1インレットで受け取った整数と組み合わせて使用される。
- 実数** 整数に変換される。
- stop** 第1インレット：現在保持しているノートの情報をノート・オフとして出力し、それに割り当てられたボイス番号を解除する。

## アーギュメント

- 整数** オプション。第1アーギュメントは**poly**で割り当てることができるボイス番号の数を設定する（従って、一度に保持できるノート数を制限することができる）。アーギュメントがなければ、**poly**は16個のノートを保持することになる。第2アーギュメントがなければ、またはそれが0である場合、**poly**が第1アーギュメントで決めた割り当てられるボイス番号数以上は保持することができない、というオーバーフロー・メッセージを出力する第4アウトレットを持つことになる。第2アーギュメントが0でない場合は、自動的にボイス番号を更新する。つまり、割り当てられるボイス番号数より多くのノートを受け取った時、受け取ってから一番長く保持しているノートをノート・オフとして出力し、そのボイス番号を解除し、新しく受け取ったノートにそのボイス番号を割り当てる。
- 実数** 整数に変換される。

## 出力

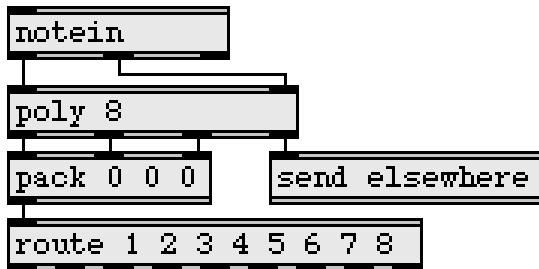
- 整数** 第1アウトレット：受け取ったノート・オンまたはノート・オフに割り当てたボイス番号。

第2アウトレット：受け取ったノート・オンまたはノート・オフのノート・ナンバー。

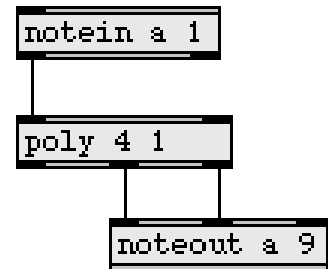
第3アウトレット：受け取ったノート・オンまたはノート・オフのベロシティ。

リスト 第4アウトレット：1番目の数値はノート・ナンバー、2番目はベロシティを表し、それは**poly**が保持することが出来ないことを示すオーバーフロー・メッセージとして出力する。第2アークギュメントに0以外の整数を指定した場合は、オーバーフローのメッセージを出力する第4アウトレットを持たず、その場合は随時ボイス番号を更新することになる。

## 利用例



異なる場所にボイスを振り分ける



同時発音数を制限する

## 参照

**Borax**  
**flush**  
**makenote**

ノートオンとノートオフに関する情報を知らせる  
保持しているノートについてノート・オフを作る  
ノート・オンに続けてノート・オフ・メッセージを生成する

## 入力

- (MIDI) MIDI入力デバイスから受信したMIDIポリフォニック・キー・プレッシャー・メッセージを受け取る。
- enable** **enable 0**のメッセージはオブジェクトを使用不能にし、受信するMIDIデータを無視するようになる。**pcontrol**オブジェクトやMIDIEnable/Disableボタンによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを再び受信可能にする。
- port** **port**の後に**a**から**z**の文字を続けるか、MIDI入力デバイスの名前を続けることによって、ポリフォニック・キー・プレッシャー・メッセージを受信するポートを設定する。**port**という単語はオプションで、省略することもできる。
- (マウス) **polyin**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

## アーギュメント

- a-z** オプション。ポリフォニック・キー・プレッシャーを受信するポートを指定する。アーギュメントを指定しない場合、**polyin**はすべてのポートのすべてのチャンネルから受信する。
- (MIDI入力デバイス名) オプション。入力ポートを指定するために、第1アーギュメントにMIDI入力デバイスの名前を使うことができる。
- a-zと整数** 空白文字で区切られた文字と数値の組み合わせは、ポリフォニック・キー・プレッシャーを受信するポートとMIDIチャンネルを指定する。チャンネル・ナンバーが16より大きい場合は、1から16の範囲に収まるように変換される。
- 整数** 文字と数値の組み合わせの代わりに、数値だけを指定しても良い。チャンネル・ナンバーのアーギュメントの正確な意味は、**MIDI Setup**ダイアログで指定したポートごとのチャンネル・オフセットによって決まる。

## 出力

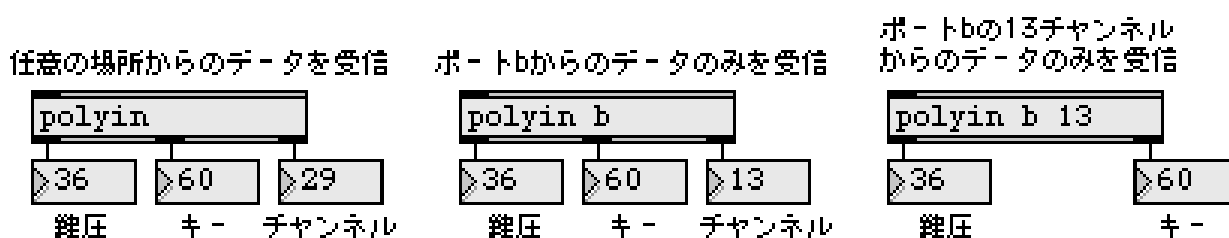
- 整数** 第1アウトレット：受信したポリフォニック・キー・プレッシャーの値。



第2アウトレット：受信したポリフォニック・キー・プレッシャーのノート・ナンバー。

チャンネル・ナンバーをアーギュメントに指定した場合は、**polyin**は2つのアウトレットのみを持つ。チャンネル・ナンバーを指定しない場合は、**polyin**は一番右に第3アウトレットを持つ。これは受信したポリフォニック・キー・プレッシャーのチャンネル・ナンバーを出力する。

## 利用例



ポリフォニック・キー・プレッシャーは、どこでも受信することができ、ポートのみを指定するか、ポートとチャンネルを指定することができる

## 参照

**midin**  
**polyout**

生のMIDIデータを受信する  
MIDIポリフォニック・キー・プレッシャーを送信する

Tutorial 16

More MIDI ins and outs (英文PDF参照)

## 入力

**整数** 第1インレット：数値はMIDIポリフォニック・キー・プレッシャーの値として扱われ、指定されたチャンネルとポートから送信される。数値は0から127の間に制限される。

第2インレット：数値はノート・ナンバーとして格納され、第1インレットで受け取るポリフォニック・キー・プレッシャーの値とともに使用される。数値は0から127の間に制限される。

第3インレット：数値はポリフォニック・キー・プレッシャーを送信するチャンネル・ナンバーとして格納される。

**実数** 整数に変換される。

**リスト** リストの1番目の数値はポリフォニック・キー・プレッシャーの値として、2番目はノート・ナンバーとして、3番目はチャンネル・ナンバーとして扱われ、MIDIポリフォニック・キー・プレッシャーを送信する。

**enable** **enable 0**メッセージはオブジェクトを使用不可にし、MIDIデータを送信しないように設定する。MIDI Enable/Disableボタンや**pcontrol**オブジェクトによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを送信可能にする。

**port** 第1インレット：**port**の後に文字の**a**から**z**の文字を続けるか、MIDI出力デバイスの名前を続けることによって、ポリフォニック・キー・プレッシャーを送信するポートを設定する。**port**という単語はオプションで、省略することもできる。

(マウス) **polyout**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

## アーギュメント

**a-z** オプション。ポリフォニック・キー・プレッシャーを送信するポートを指定する。文字のアーギュメントを指定している時に、チャンネル・ナンバーが16より大きければ、1から16の範囲に収まるように変換される。アーギュメントが存在しない場合は、**polyout**は初期状態として、ポートaからMIDIチャンネル1として送信する。

**a-zと整数** 空白で区切られた文字と数値の組み合わせによって、送信するポリフォニック・キー・プレッシャーのポートとMIDIチャンネルを指定する。チャンネル・ナンバーが16より大きければ、1から16の範囲に収まるように変換される。

(MIDI出力デバイス名) オプション。MIDI出力デバイスの名前は、ポートを指定する最初のアーギュメントとして用いることができる。

**整数** 文字と数値の組み合わせの代わりに、数値だけを指定しても良い。チャンネル・ナンバーのアーギュメントの正確な意味は、**MIDI Setup**ダイアログで指定したポートごとのチャンネル・オフセットによって決まる。

## 出力

(MIDI) アウトレットは存在しない。オブジェクトに指定したMIDI出力ポートから、MIDIポリフォニック・キー・プレッシャー・メッセージが送信される。

## 利用例



文字のアーギュメントは、送信ポートを限定する  
または、数値によってポートとチャンネルを指定する

## 参照

**midout**  
**polyin**

生のMIDIデータを送信する  
MIDIポリフォニック・キー・プレッシャーを受信する

Tutorial 16

More MIDI ins and outs (英文PDF参照)

底の値を指数乗する。指数は第2インレットで、底の値は第1インレットで設定する。

## 入力

**実数または整数** 第1インレット：底の値を設定する。

第2インレット：指数を設定する。

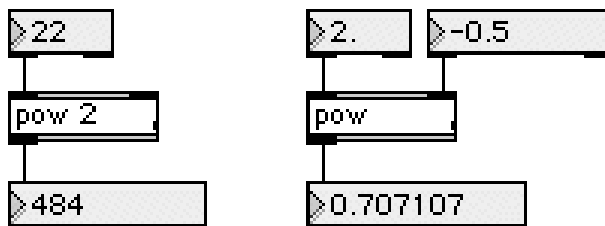
## アーギュメント

**実数または整数** オプション。指数を設定する。デフォルト値は0である。

## 出力

**実数** 底の値（第1インレットで指定）を指数乗（第2インレットで設定）した値。

## 利用例



pow を用いて、2乗（あるいは、それ以外の累乗）の値を求める

## 参照

- expr** 数式の値を求める
- >>** 数値のビットを右へシフトする
- <<** 数値のビットを左へシフトする

# prepend

メッセージの前に、  
あるメッセージを付け加える

## 入力

**set** **set**メッセージに続くメッセージを出力せずに、**prepend**に格納する。

### set以外の任意のメッセージ

**prepend**に格納されたメッセージを、インレットに受け取ったメッセージの前に付け加える。繋がられた一連のメッセージが、アウトレットから出力される。

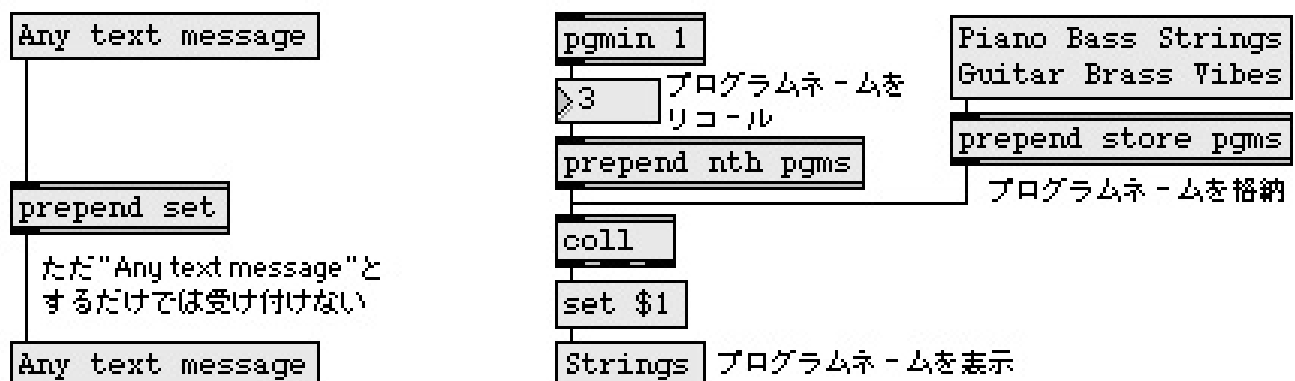
## アーギュメント

任意のメッセージ 必須。入力されたメッセージの前に付け加えるメッセージを設定する。第1アーギュメントはシンボルでなければならない。

## 出力

任意のメッセージ **prepend**に格納されているメッセージを、インレットに受け取ったメッセージの前に付け加え、それをアウトレットから出力する。出力されるメッセージの長さは最大256項目以内でなければならない。

## 利用例



**prepend**を用いて、シンボルを意味のあるメッセージにすることができる

## 参照

**append**      メッセージの末尾にアーギュメントを付け加える  
**message**    任意のメッセージを送る  
**route**      選択的に入力を特定のアウトレットから出力する

Tutorial 25      Managing messages (英文PDF参照)



## 入力

- 整数** プリセット番号を指定し、そのプリセットに格納された設定を、**preset**オブジェクトに繋がれたオブジェクト、または、繋がれていないすべてのオブジェクトに出力する。プリセットの設定は、マウスでプリセットをクリックすることでも出力することができる。
- 実数** 整数に変換される。
- bang** 整数またはマウス・クリックによって、直前に読み出したプリセットの設定を出力する。
- clear** 直前に出力したプリセットの内容を消去される。**clear**に数値が続けば、その数値で指定したプリセット番号の内容が消去される。
- clearall** すべてのプリセット内容を消去する。
- list** **bang**と同じ。
- name** **word**に続くシンボルによって、プリセットのID名を設定する。ID名によって、**preset**は他と重複しないIDを持ち、そのIDに対応するファイルが、他の**preset**では読み込めないようにする。
- read** **read**または**read**に続く数値によって、読み込むプリセット・データのファイルを選択する標準ファイル・ダイアログが表示される。プリセット・ネーム・コードが与えられた場合、コードで指定したファイルだけを表示する。**read**に数値が続く場合は、そのプリセット番号のプリセットにファイルのデータを読み込む。数値が0または-1の場合、ファイルのデータは、そのファイルが持つプリセット番号の最初から、順にプリセット番号に読み出される。**read**に続くシンボルまたは数値とシンボルでは、標準ファイル・ダイアログは表示されない。その代わりに、このシンボルはプリセットに読み込むファイル名を指定する。シンボルと数値の場合での数値は、読み込むプリセット番号を指定する。
- store** **store**に続く数値によって、数値によりプリセット番号を指定し、同じウィンドウにあるすべてのユーザー・インターフェイス・オブジェクトの現在の設定を、そのプリセット番号に格納する。オブジェクトがパッチ・コードで**preset**オブジェクトの第1アウトレットに繋がっている場合、繋がれたオブジェクトだけの設定を格納する。

プリセット番号は左から右、次いで上から下の順に設定される。プリセットに設定が格納されると、そこに何かのデータが含まれていることを示すドットが現れる。Shiftキーを押しながらプリセットボタンをクリックすることによっても、他のオブジェクトの設定をそのプリセットに格納することができる。



**write** **write**または**write**に続く数値によって、プリセット・データをファイルとして保存するための標準ファイル保存ダイアログが開く。プリセットにプリセット・ネーム・コードを与えない場合、そのファイルのファイル形式のコードが与えられる。**write**に続く数値は、保存するデータのプリセット番号を指定する。0または-1の数値はすべてのプリセットのデータを保存する。**word**に続くシンボルまたは数値とシンボルでは、標準ファイル保存ダイアログは表示されない。代わりに、このシンボルは保存するファイル名を指定する。保存ファイルは現在のデフォルトのディレクトリーに置かれる。シンボルと数値の場合での数値は、保存するデータのプリセット番号を指定する。

## インスペクター

**preset**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**preset**オブジェクトを選択することで、**preset**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**preset**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

インスペクターでは、他の**preset**オブジェクトを区別するために、ID名を**preset**オブジェクトに指定する。4文字の名称を入力すれば、それはこのオブジェクトによって保存されるプリセット・ファイルに設定されるMacintoshのファイル・タイプを指定したことになる。ID名を持つ**preset**オブジェクトに**read**メッセージを送れば、このID名（ファイル・タイプ）に一致するタイプのファイルだけが、標準ファイル・ダイアログに現れる。このことによって、他の**preset**オブジェクトに設定されたプリセット・ファイルを開かないようにし、特定のプリセット・ファイルのためのファイル・タイプを設定することができる。*Save Presets with Patcher*のチェック・ボックスをクリックすれば、**preset**の内容をパッチ・ファイルとともに保存することができる。

**Revert**ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

なし。





## 出力

**整数または実数** 第1アウトレット：プリセットが、マウス・クリックまたはインレットへの数値によって読み出された時、プリセットに格納されている設定を、**preset**オブジェクトのアウトレットから繋がっているオブジェクトに出力する。**preset**に何もオブジェクトが繋がっていない場合は、ウィンドウ内のすべてのユーザー・インターフェイス・オブジェクトに出力する。

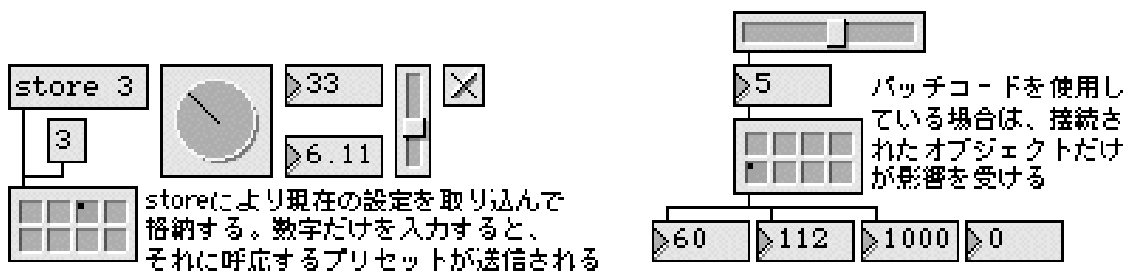
**整数** 第2アウトレット：読み出されたプリセット番号を出力する。

(インターナル) **preset**の第3アウトレットに繋がっている任意のユーザー・インターフェイス・オブジェクトは、**preset**の設定の読み出しには影響されない。(このことは特に、**preset**の設定の効果を与えたいオブジェクトが多く、設定の影響を与えたくないオブジェクトが少ない場合に役に立つ。)

プリセットに格納できるデータを持つオブジェクトは、**dial**、**Ggate**、**Gswitch**、**hslider**、**led**、**number box** (整数と実数の両方)、**slider**、**toggle**、そして**uslider**である。**table**の内容も**preset**によって格納し、読み出すことができる。しかし、**table**は**preset**オブジェクトのアウトレットから繋がっていないなければならない。**preset**オブジェクトのアウトレットには、同じアーギュメント名をもつ**receive**オブジェクトに繋がったオブジェクトと交信するために、**send**オブジェクトを繋ぐことができる。

**preset**オブジェクトの大きさを変えることによって、表示されるプリセットの数を調節することができる。1つの**preset**オブジェクトにおけるプリセット数の最大は2048個である。

## 利用例



多くの設定を格納し、後からそれらの設定を読み出す

## 参照

**grab** 他のオブジェクトの出力を取得する

Tutorial 37 Data structures (英文PDF参照)

## 入力

任意 メッセージは**print**オブジェクトによって解釈されることはない。それらは、Maxウィンドウに単にそのまま一語一語出力されるだけである。

(マウス) 任意の**print**オブジェクトをダブル・クリックすることで、Maxウィンドウを開くか、前面に持ってくるができる。

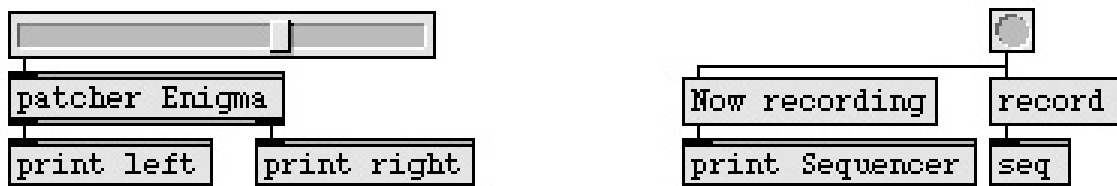
## アーギュメント

任意 オプション。アーギュメントは**print**オブジェクトを識別できるようにする。Maxウィンドウに出力されるそれぞれのメッセージは、**print**オブジェクトの名前とコロンの(:)に続くことになる。その名前は、スペース、または特殊文字を含むことはできず、1つの数値、または1つの単語である必要がある。アーギュメントがなければ、その**print**オブジェクトの名前は**print**になる。**print**オブジェクトにアーギュメントを指定することは、2つ以上の**print**オブジェクトを識別するために役立つ。

## 出力

任意 アウトレットはない。インレットに受け取ったメッセージはMaxウィンドウに出力される。

## 利用例



出力を表示する。またイベントが起こったことを通知する

## 参照

Tutorial 1      Saying “Hello!” (英文PDF参照)

## 入力

- リスト** 数値は、ある数値に続いてある数値が出現する確率マトリックスに登録される。(これは1次マルコフ連鎖として知られている)。リストは3つの整数から成る。それらは現在の値、次に来る値、現在の値に続いて次の値が来る確率である。リストの最初の2つの整数は、出力される数値の並びを指定する。これは、1番目の整数の後に2番目の整数が出力される並び方である。3番目の整数は、数値の並びが起こる相対的な確率を設定する。1番目の整数が出力されると、その数値に続いて出現する可能性のある数値の相対的確率に基づいて、次に出力される値が決定される。
- bang** 直前の出力値と、それに続いて出現する可能性のある数値の相対的確率に基づいて、重みを付けてランダムに数値を選んで出力する。
- 整数** 現在の値を設定する (出力はしない)。**bang**メッセージによって出力される値は、設定された数値に続いて出現する確率マトリックスに基づいて決定される。
- reset** 整数が続く**reset**メッセージによって、次に続く可能性を持たない数値によって行き詰まった時に、指定した数値の回数だけ以前の状態へ戻る。
- dump** 格納されている確率マトリックス (マルコフ連鎖) の完全なリストを、Maxウィンドウに表示する。
- embed** 0以外の数値が続く**embed**メッセージによって、**prob**の内容が、それを含むパッチの一部として保存されるようになる。**embed 0**メッセージは、パッチを閉じた時に、に**prob**に含まれる内容を消去するよう設定する。
- clear** **prob**の内容を消去する。

## アーギュメント

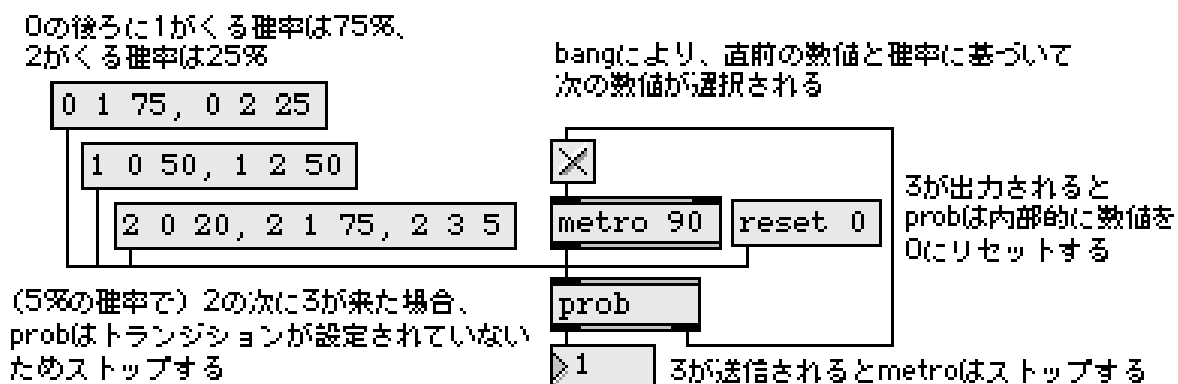
なし。

## 出力

- 整数** 第1アウトレット：インレットに**bang**を受け取った時に、**prob**は数値を出力する。直前に出力した数値と、その数値に続いて出現する数値の相対的確率に基づいて、出力される数値が決定される。
- bang** 第2アウトレット：現在の数値 (最後に出力された数値) に対して、次に出現する可能性のある数値が確率マトリックスに存在しないなら

ば、インレットに**bang**を受け取った時に、第2アウトレットから**bang**が出力される（第1アウトレットからは何も出力されない）。

## 利用例



出力値の確率は、直線の出力値によって決まる

## 参照

**anal**  
**Histo**  
**mean**

受け取った2つの数値をペアとして、ヒストグラムを作る  
受け取った数値のヒストグラムを生成する  
連続した数値の平均値を出力する

## 概要

**pv**オブジェクトは、**value**オブジェクトとほぼ同じように動作するが、2つの例外がある。1つ目は、同じ名前を持つ**pv**オブジェクトは、同じパッチか、その中のサブ・パッチの1つの中にある場合のみ、同じ値を共有する。2つ目は、**pv**オブジェクトはメッセージ・ボックスによって送られるリモート・メッセージ（セミコロンの後にシンボルが続くもの）を受けることができない。つまり、**pv**はprivate value、すなわち、その値はオブジェクト間で共有されるが、1つのパッチ内のみ限定されることを意味している。

## 入力

**任意のメッセージ** そのメッセージはオブジェクト内に格納され、そのオブジェクトを含むパッチか、そのサブ・パッチ内（または、サブ・パッチ内にそのオブジェクトがある場合は、その親となるパッチ内）で、同じ名前を持つ**pv**オブジェクトと共有される。同じように、その他の**pv**オブジェクトが受け取ったメッセージは、その格納されたメッセージを変更する。

**bang** 格納されたメッセージを出力する。

## アーギュメント

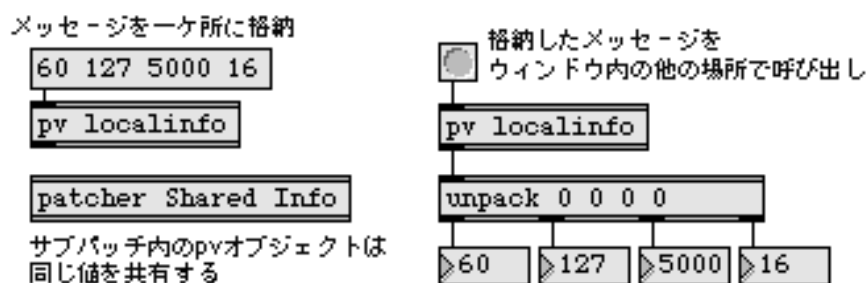
**任意のシンボル** 必須。1つ目のアーギュメントは識別するための名前を指定する。パッチ内で同じ名前を持つすべての**pv**オブジェクトは、同じ値を共有する。

**任意のメッセージ** オプション。1つ目のアーギュメントの後に入力された任意のメッセージは、**pv**オブジェクトに格納される初期値として格納される。1つのパッチ・ファイルの中にある2つ以上の同じ名前を持つ**pv**オブジェクトが別の値で初期化された時は、最後に初期化されたものが値を決定することに注意しよう。**pv**オブジェクトを初期化する命令が正確に決定できなくなるため、一番良い方法は、関連する**pv**オブジェクトのうち、1つのみで初期化を行うことである。

## 出力

**任意のメッセージ** そのインレットに**bang**が受け取られた時に、格納されたメッセージが出力される。

## 利用例



## 参照

**float**  
**int**  
**receive**  
**send**  
**value**

実数を格納する  
整数を格納する  
パッチ・コードなしでメッセージを受信する  
パッチ・コードなしでメッセージを送信する  
他のオブジェクトと格納されている値を共有する

**pvar**は、パッチの他の部分の「処理」パートに接続されたパッチの中の一部に、ユーザー・インターフェースを構築することを可能にする。**send**オブジェクトと**receive**オブジェクトとは異なり、**pvar**はグローバルには作用しない。つまり、**pvar**オブジェクトとそれに関連付けられたオブジェクトは同じパッチの中になければならない。

## 入力

**任意のメッセージ** メッセージは現在**pvar**に関連付けられている名前を付けられたMax/MSPオブジェクトに送信される。

**set** 任意のシンボルを伴う**set**は、**pvar**に関連付けられた名前を付けられたオブジェクトにメッセージを送信する。名前を付けられたオブジェクトは、**pvar**オブジェクトに対するアーギュメント、または**setname**メッセージを用いることによって設定することができる。

**setname** 1つのシンボルを伴う**setname**は、**pvar**が接続するMax/MSPオブジェクトの名前を指定する。名前を付けられたオブジェクトは、**pvar**オブジェクトと同じパッチの中になければならない。

## アーギュメント

**シンボル** オプション。1番目のアーギュメントは、**pvar**が接続するオブジェクトの名前を指定する。名前が与えられていない場合には、**setname**メッセージを後で入力することができる。

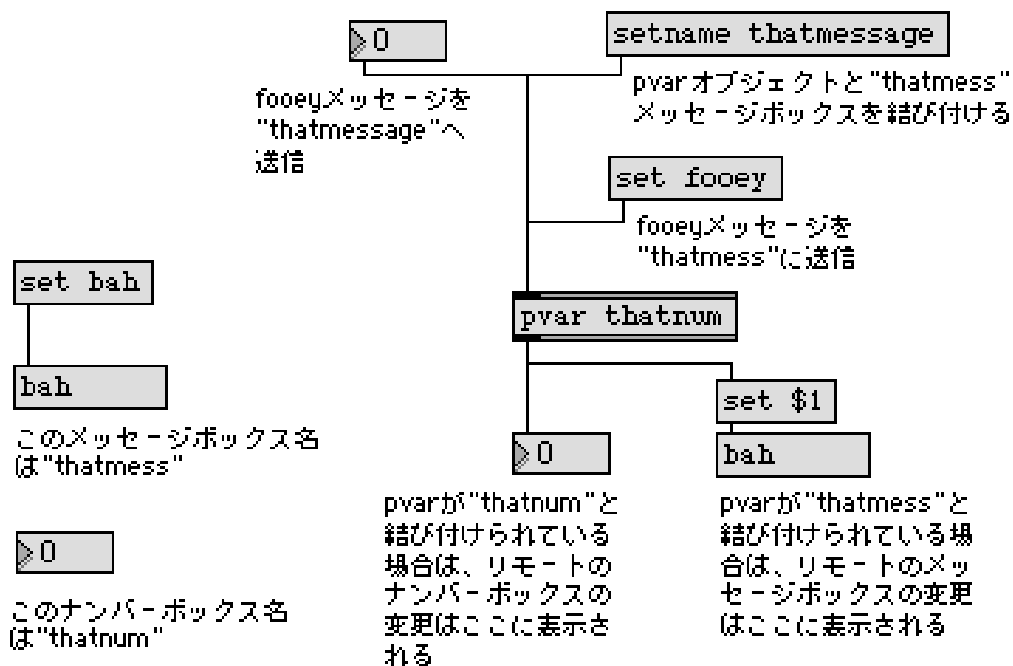
**整数** オプション。2番目のアーギュメントは、**pvar**オブジェクトのアウトレットの数を指定する。**pvar**は、アーギュメントとして指定した数よりも多くなならない限りは、関連付けられたオブジェクトが持つのと同じだけの数のアウトレットを接続することができる。アウトレットのデフォルト数は1である。

## 出力

**任意のメッセージ** **pvar**に接続された名前を付けられたオブジェクトに対して行われた任意の変更は、オブジェクトのアウトレットから出力される。



**利用例**



**参照**

**receive**  
**send**  
**thispatcher**  
**value**

パッチ・コードなしでメッセージを受信する  
パッチ・コードなしでメッセージを送信する  
パッチにメッセージを送る  
他のオブジェクトと格納されている値を共有する



**radiogroup**オブジェクトには、ラジオ・ボタンとチェック・ボックスの2つの操作モードがある。ラジオ・ボタン・モードでは、**radiogroup**オブジェクトは、1つのグループ内でユーザーの宣言する数のボタンを提供する。それらは一度に1つだけ選択することができるようになっている。チェック・ボックス・モードでは、**radiogroup**オブジェクト内のインジケータはオン・オフを表示するチェック・ボックスの集まりとして機能する。チェック・ボックス・モードはまた、チェック・ボックスを、整数を表す2進数のビット・パターンを表示のために使うことにも対応している（以下の**flagmode**メッセージを参照のこと）。

注記：ボタン、またはボックスの右に位置するコメント・ボックスの下の部分を引き伸ばすことで、**radiogroup**オブジェクトは水平方向にリサイズできる。同じように、ボタンの右側のテキストをクリックすることでも、ボタンの選択やボックスの状態を設定することができる。

## 入力

(マウス) ラジオ・ボタン・モードでは、ラジオ・ボタンをクリックすることで、そのラジオ・ボタンを選択し、そのボタンに対応する番号を出力する（番号は0から始まる）。

チェック・ボックス・モードでは、チェック・ボックスをクリックすることで、その状態を変え（1から0へ、または0から1へ）、それらのボックスのオン・オフの状態に対応した0と1からなるリストを出力する。全体のボタンやボックスのグループがアクティブでない（灰色になっている）状態では、クリックには反応しない。個々の要素が無効になっている場合は（灰色になっている）、それらはクリックには反応しないが、そのグループ内の有効な要素は通常通りクリックに反応する。チェック・ボックス・モードのバリエーションであるフラグ・モードでは、チェック・ボックスを2進数表現でのビットの位置に対応するものとして扱う（すなわち、最初のチェック・ボックスは1の桁に対応し、2つ目は2の桁、3つ目は4の桁というように対応する）。チェック・ボックスをクリックすれば、そのチェック・ボックスのチェックを変えて、そのビット・パターンに対応する整数の値を出力する。

**bang** ラジオ・ボタン・モード：**bang**は、現在選択されているラジオ・ボタンの番号を出力する。

チェック・ボックス・モード：**bang**はそれらのチェック・ボックスのオン・オフの状態を表した0と1からなるリストを出力する。

フラグ・モード：**bang**は現在のチェック・ボックスのビット・パターンに対応した整数を出力する（すなわち、1つ目、2つ目、3つ目のチェック・ボックスがチェックされていれば、**bang**は7という値を出力する）。



- 整数** ラジオ・ボタン・モード：整数はラジオ・ボタンの選択を行い、その入力された値を出力する。番号付けは0から始まり、負の数値はどのボタンも選択されないことを示す。
- フラグ・モード：**radiogroup**オブジェクトのインレットに受け取った整数は、ボタン、またはチェック・ボックスをその整数の値のビットパターンを反映するように設定する（すなわち、19の値は2進数の**10011**に対応し、1つ目、2つ目、また5つ目のボックスを選択させることになる）。また、**radiogroup**オブジェクトのアウトレットから、その整数を出力する。
- 実数** ラジオ・ボタン・モードとチェック・ボックス・モード：整数に変換される。
- リスト** チェック・ボックス・モード：チェック・ボックス・モードを指定し、さらに**flagmode 1**メッセージを用いてフラグ・モードを設定している場合には、0と1から成るリストによって、それらのチェック・ボックスの状態を設定し、そのリストを出力する。
- disableitem** ラジオ・ボタン・モードとチェック・ボックス・モード：指定された番号の項目を無効にする（それらは灰色に描画され、クリックにも反応しなくなるが、**set**メッセージ、整数、実数には反応する）。
- enableitem** ラジオ・ボタン・モードとチェック・ボックス・モード：**enableitem**には1つの数値、または数値のリストが続き、その指定された番号の項目が既に**disableitem**メッセージによって無効になっている場合に、その項目を有効にする。
- flagmode** チェック・ボックス・モード：0でない値の続く**flagmode**メッセージは、その**radiogroup**オブジェクトをフラグ・モードに設定する。このモードでは、それぞれのチェック・ボックスは1つの整数の値のビットに対応する（すなわち、1つ目のラジオ・ボタンかチェック・ボックスは1の桁のビットに、2つ目のボタンかチェック・ボックスは2の桁のビットに、また3つ目のボタンかチェック・ボックスは4の桁のビットにというように対応する）。**flagmode 0**メッセージはこのモードを無効にする（デフォルト）。
- itemtype** ラジオ・ボタン・モードとチェック・ボックス・モード：**itemtype**には0か1が続き、**radiogroup**オブジェクトのモードを設定する。**itemtype 0**メッセージはラジオ・ボタン・モードに、**itemtype 1**メッセージはチェック・ボックス・モードに設定する。
- inactive** ラジオ・ボタン・モードとチェック・ボックス・モード：**inactive**には0か1が続き、ラジオ・ボタンまたはチェック・ボックスのグループ全体を有効な状態、または有効でない状態へと切り替える。**inactive 0**メ



メッセージ（デフォルト）は、それらのボックスが有効な状態であることを意味し、マウスのクリックにも反応する。**inactive 1**メッセージはラジオ・ボタン、チェック・ボックスの表示を灰色にし、マウスのクリックにも反応しないようにする（しかし、**set**メッセージ、整数、リストを用いることで、それらの状態を設定できる）。

**offset** ラジオ・ボタン・モードとチェック・ボックス・モード：**offset**には1つの数値が続き、ボタンまたはボックスの上部からのオフセット位置をピクセル単位で設定する。最小のオフセット位置は14ピクセルで、デフォルトのオフセット位置は16ピクセルである。

**set** ラジオ・ボタン・モード：**set**には1つの整数が続き、出力を行わずに、現在のラジオ・ボタンの選択を設定する。

チェック・ボックス・モード：**set**には0と1からなるリストが続き、出力を行わずに、チェック・ボックスの状態を設定する。

チェック・ボックス・モードで、フラグ・モードに設定されている場合には、出力を行わずに、最初の32個のチェック・ボックスの状態を、**set**に続く数値のビット・パターンに対応した状態に設定する（詳細は**flagmode**メッセージの部分参照のこと）。

**size** ラジオ・ボタン・モードとチェック・ボックス・モード：**size**には1つの数値が続き、ボタンやボックスの数を変更する。デフォルト値は2で、最大数は64である。

注記：**radiogroup**オブジェクトにチェック・ボックス・モードを使用し、フラグ・モードを有効にしていれば、チェック・ボックスの数は32にしか設定できない。

## インスペクター

**radiogroup**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**radiogroup**オブジェクトを選択することで、**radiogroup**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**radiogroup**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**radiogroup**オブジェクトのインスペクターでは、ボタンの数（デフォルト値は2）や、ボタンのオフセット位置（デフォルト値は16ピクセル）を指定できる。*Button Type*オプションでは、ラジオ・ボタン（デ



フォルト) とチェック・ボックスとを選択できる。チェック・ボックス・オプションを選択していれば、*Flag Mode*オプションも、選択できるようになる。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

なし。

## 出力

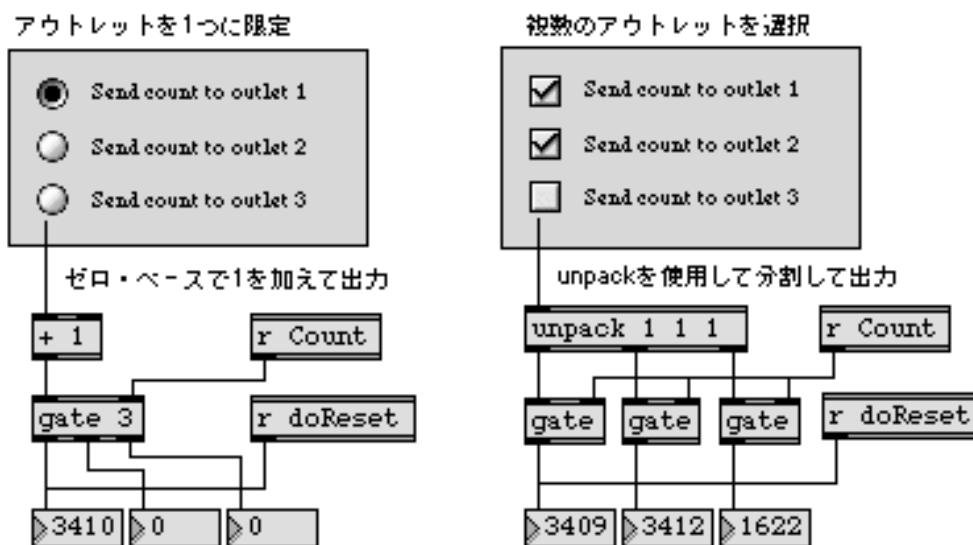
**整数** ラジオ・ボタン・モード：ラジオ・ボタンをクリックすることで、選択されたラジオ・ボタンに対応する整数を出力する。番号は0から始まる。

フラグ・モード：チェック・ボックスをクリックすることで、チェックされているボックスによって表されるビット・パターンに対応する整数を出力する（すなわち、1つ目、2つ目、3つ目のチェック・ボックスがチェックされていれば、**bang**は7という値を出力する）。

**リスト** チェック・ボックス・モード：**bang**を受け取れば、チェック・ボックスのオン・オフの状態を示す0と1から成るリストを出力する。



## 利用例



ラジオ・ボタンは1つの選択のみが可能で、チェック・ボックス・モードでの複数の選択は、複数の **gate** オブジェクトを制御できる

## 参照

**button**  
**matrixctrl**  
**pictctrl**  
**toggle**  
**ubutton**

任意のメッセージによって動作し、**bang**を送信する  
マトリックス状のスイッチ・コントロール  
ピクチャーを用いたコントロール  
オンとオフ（1と0）を切り替える  
透明なボタン、**bang**を出力する

## 入力

- bang** 第1インレット：0から、上限値より1小さい数値の間で、ランダムに発生させた数値を出力する。
- 整数** 第2インレット：この値は、出力される乱数の上限値として格納される。出力は常に0から、この上限値より1小さい数値の間になる。
- seed** 第1インレット：**seed**には、1つの数値が続き、ランダム・ジェネレーターにシード（種子）となる値を与え、特定の（再現可能な）疑似乱数の連続を発生させる。0の数値は、シード値として、システムが起動してからの経過時間（予測できない値）を用いる。この予測できないシードは、デフォルトとして**random**オブジェクトが生成された時刻が使用される。

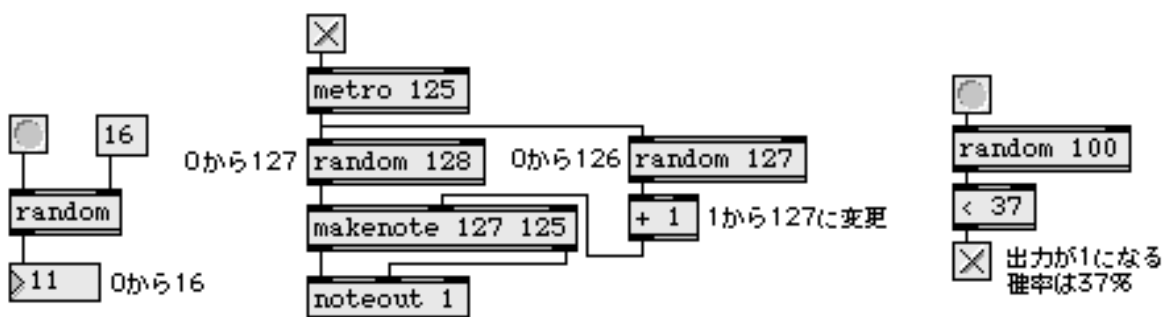
## アーギュメント

- 整数** オプション。出力される乱数の上限値の初期値を設定する。出力は常に0から、上限値より1小さい数値の間になる。アーギュメントを指定しなかった場合、上限値の初期値は1に設定され、**random**オブジェクトが**bang**を受け取る度に0を出力する。
- 整数** オプション。2つ目のアーギュメントはランダム・ジェネレーターのシードとなる値を設定する。アーギュメントが指定されなかった場合、時刻がシードの初期値として使用される。

## 出力

- 整数** **bang**を第1インレットに受け取った時に、**random**オブジェクトは0から、上限値より1小さい数値の範囲の値を生成する。

## 利用例



ランダムなイベントを生成する。また、可能性に基づく判定を行う



## 参照

**decide**  
**drunk**  
**urn**

オンあるいはオフ（1あるいは0）をランダムに選択する  
ある範囲を移動するようなランダムな数値を出力する  
重複しないランダムな数値のジェネレーター

Tutorial 22

Delay lines（英文PDF参照）

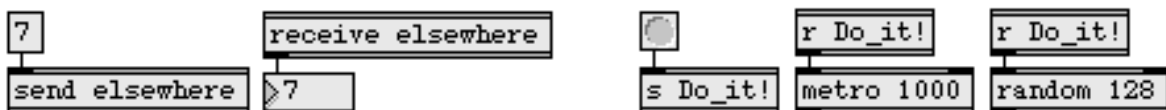
## 入力

- 任意のメッセージ 同じ名前の**send**オブジェクトまたは**forward**オブジェクトから受け取ったものが**receive**オブジェクトに入力される。開いている他のパッチ内に、同じ名前の**receive**オブジェクトが存在する場合も同様である。
- (マウス) **receive**オブジェクトをダブルクリックすると、同じ名前の**send**オブジェクトを含むパッチ・ウィンドウが開く。くり返し**receive**オブジェクトをダブルクリックすることにより、さらに他のウィンドウを開くことができる。
- set** アーギュメントを指定しない場合、**receive**は1つのインレットを持つ。setに続くシンボルで、最初からアーギュメントとして名前を設定すると同じように、**receive**の名前を指定し、変更することができる。
- 任意のシンボル オプション。**receive**に名前を付ける。アーギュメントを指定しない場合、**receive**は1つのインレットを持ち、何かを受け取るよりも前に、**set**メッセージを用いて**receive**オブジェクトの名前を指定しなければならない。

## 出力

- 任意のメッセージ 同じ名前の**send**オブジェクトまたは**forward**オブジェクトのインレットに受け取ったメッセージ、または、**int**、**float**、**grab**オブジェクトやメッセージボックスから送られたものが、アウトレットから出力される。開いている他のパッチの中に、同じ名前の**receive**オブジェクトが存在する場合も同様である。

## 利用例



同じ名前を持ったすべての**send**オブジェクトと**receive**オブジェクトは繋がっている

## 参照

- float** 実数を格納する
- forward** 様々なオブジェクトにパッチ・コードなしでメッセージを送信する
- int** 整数を格納する
- message** 任意のメッセージを送る
- route** 選択的に入力を特定のアウトレットから出力する
- send** パッチ・コードなしでメッセージを送信する
- value** 他のオブジェクトと格納されている値を共有する

## 入力

- bang** 第1インレット：現在設定しているスクリーン座標、表示モード、表示色によって矩形を表示する。
- 整数** 第1インレット：グラフィック・ウィンドウの左上隅からの相対位置として、矩形の左座標を設定し、形状を描く。
- 第2インレット：矩形を描く上座標を設定する。
- 第3インレット：矩形を描く右座標を設定する。
- 第4インレット：矩形を描く下座標を設定する。
- 第5インレット：矩形を描く表示モードを設定する。（**oval**オブジェクトの表示モードの定数リストを参照のこと）
- 第6インレット：グラフィック・ウィンドウの現在のパレットに従って、矩形の表示色のインデックス番号を設定する。モニターが白黒モードである時は、0以外のインデックス番号では黒であり、0は白である。
- frgb** 第1インレット：**frgb**の後に続く0から255までの3つの数値によって、次に描かれる矩形の表示色のRGB値を設定する。
- priority** 第1インレット：**priority**の後に続く0以上の数値によって、グラフィック・ウィンドウ内の**rect**オブジェクトを重ね合わせる優先順位を設定する。優先順位の低いオブジェクトは、優先順位の高いオブジェクトの後ろに描かれる。

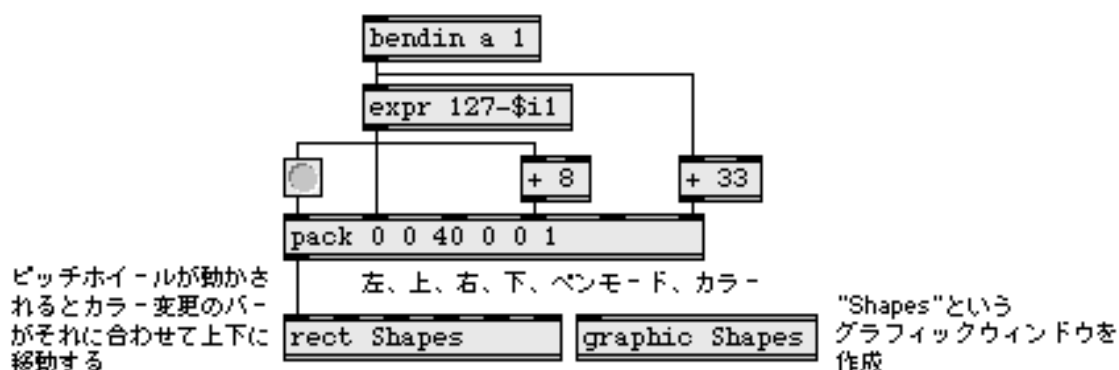
## アーギュメント

- 任意のシンボル** 必須。**rect**の最初のアーギュメントは、矩形が描かれるグラフィック・ウィンドウの名前でなければならない。**rect**オブジェクトを編集している時に、グラフィック・ウィンドウが存在する必要はないが、名前が一致するウィンドウが開かれるまでは、矩形は描かれない。
- 整数** **rect**を重ねる優先順位の初期値を設定する。優先順位が指定されていない場合の初期値は3である。

## 出力

(ビジュアル) **rect**オブジェクトに関連付けられたグラフィック・ウィンドウが表示されている場合、第1インレットに**bang**メッセージか数値を受け取れば、ウィンドウに矩形が描かれる。そして、先に描かれている矩形があれば、それは消去される。

## 利用例



MIDIデータや数値の変更によって、矩形を移動することができる

## 参照

**frame**  
**graphic**  
**lcd**  
**oval**  
**ring**

グラフィック・ウィンドウに矩形の枠を描く  
スプライトによるグラフィックを描くためのウィンドウ  
パッチ・ウィンドウ内に画像を描画する  
グラフィック・ウィンドウに塗り潰した楕円形を描く  
グラフィック・ウィンドウに楕円形の枠を描く

## 入力

シンボル シンボルとしてのフォルダやファイルの絶対パス名。絶対パスの表記は以下のようなになる。  
“MyDisk:/Max Folder/extras/filename”

## アーギュメント

なし。

## 出力

シンボル Maxアプリケーションのフォルダからのファイルやフォルダの相対位置のパス名をシンボルとして出力する。入力されたパス名がMaxアプリケーションのフォルダを含んでいれば、パス名は“./”で始まり、パスのフォルダ名が続く形に変換される。それ以外の場合は、入力されたシンボルがそのまま出力される。

## 利用例



## 参照

**absolutepath**

ファイル名を絶対パスに変換する

**opendialog**

ファイルやフォルダを選択するためのダイアログを開く

**strippath**

フル・パス名からファイル名を取得する

## 入力

- bang** 第1インレット：現在設定しているスクリーン座標、表示モード、表示色によって楕円形の枠（輪）を表示する。
- 整数** 第1インレット：グラフィック・ウィンドウの左上隅からの相対位置として、輪の左座標を設定し、形状を描く。
- 第2インレット：輪を描く上座標を設定する。
- 第3インレット：輪を描く右座標を設定する。
- 第4インレット：輪を描く下座標を設定する。
- 第5インレット：輪を描く表示モードを設定する。（**oval**オブジェクトの表示モードの定数リストを参照のこと）
- 第6インレット：グラフィック・ウィンドウの現在のパレットに従って、輪の表示色のインデックス番号を設定する。モニターが白黒モードである時は、0以外のインデックス番号では黒であり、0は白である。
- frgb** 第1インレット：**frgb**の後に続く0から255までの3つの数値によって、次に描かれる輪の表示色のRGB値を設定する。
- priority** 第1インレット：**priority**の後に続く0以上の数値によって、グラフィック・ウィンドウ内の**ring**オブジェクトを重ね合わせる優先順位を設定する。優先順位の低いオブジェクトは、優先順位の高いオブジェクトの後ろに描かれる。

## アーギュメント

- 任意のシンボル** 必須。**ring**の最初のアーギュメントは、輪が描かれるグラフィック・ウィンドウの名前でなければならない。**ring**オブジェクトを編集している時にグラフィック・ウィンドウがある必要はないが、名前が一致するウィンドウが開かれるまでは輪は描かれない。
- 整数** **ring**の重ねる優先順位の初期値を設定する。優先順位が指定されていない場合の初期値は3である。

## 出力

(ビジュアル) **ring**オブジェクトと関連付けられたグラフィック・ウィンドウが表示している場合、第1インレットに**bang**メッセージか数値を受け取れば、ウィンドウに形状が描かれる。そして、先に描かれている輪があれば、それは消去される。

## 利用例

**oval**または**ring**の利用例を参照のこと。**ring**は**oval**や**rect**、**frame**の例に直接置き換えることが可能。

## 参照

**frame**  
**graphic**  
**lcd**  
**oval**  
**rect**

グラフィック・ウィンドウに矩形の枠を描く  
スプライトによるグラフィックを描くためのウィンドウ  
パッチ・ウィンドウ内に画像を描画する  
グラフィック・ウィンドウに塗り潰した楕円形を描く  
グラフィック・ウィンドウに塗り潰した矩形を描く



## 入力

**任意** メッセージの1つ目の項目が**route**オブジェクトのアーギュメントのうちどれかと同じだった場合、その残りのメッセージをそのアーギュメントに対応するアウトレットから出力する。1つ目の項目がアーギュメントのどれとも一致しなかった場合、そのメッセージ全体は一番右に位置するアウトレットから出力される。

## アーギュメント

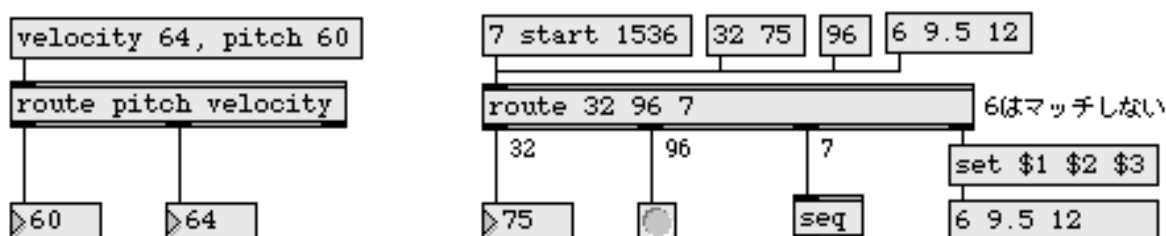
**任意** オプション。アーギュメントは、整数、実数、シンボルを混在させて指定できる。アーギュメントの数はアウトレットの数を決定し、それに一番右に位置するアウトレットが加えられる。それぞれのアーギュメントは、アウトレットに対して名前や数値を割り当てる。アーギュメントが設定されていない場合は、0の数値が割り当てられたアウトレットが1つ作られる。

## 出力

**任意** インレットに受け取った任意のメッセージの1つ目の項目が、アーギュメントと比較される。アーギュメントのうちの1つと一致すれば、その残りのメッセージが指定されたアウトレットから出力される。それ以外の場合は、メッセージ全体は一番右に位置するアウトレットから出力される。

**bang** メッセージの1つ目の項目がアーギュメントのうちの1つと一致し、そのメッセージが他に項目を持たない場合、**bang**が指定されたアウトレットから出力される。

## 利用例



アーギュメントは名前か数値をアウトレットに割り当て、入力を適切なアウトレットへ出力する

## 参照

|                |                                |
|----------------|--------------------------------|
| <b>Bucket</b>  | 順番にアウトレットからアウトレットへ数値を渡す        |
| <b>forward</b> | 様々なオブジェクトにパッチ・コードなしでメッセージを送信する |
| <b>gate</b>    | 入力を特定のアウトレットから出力する             |
| <b>pack</b>    | 数値やシンボルをひとつのリストに結合させる          |
| <b>receive</b> | パッチ・コードなしでメッセージを受信する           |
| <b>route</b>   | 選択的に入力を特定のアウトレットから出力する         |
| <b>select</b>  | 特定の入力を選択し、それ以外は通過させる           |
| <b>send</b>    | パッチ・コードなしでメッセージを送信する           |
| <b>sprintf</b> | 単語や数値のメッセージをフォーマットする           |
| <b>zl</b>      | 多目的なリスト処理                      |

Tutorial 17

Gate and Switches (英文PDF参照)



## 入力

- 整数** 第1インレット：数値は、色のついた範囲で表示される**rslider**の範囲幅の最小値を設定する。そして、その範囲の最小値と最大値が出力される。**rslider**の許容範囲を超える数値は、**rslider**の範囲内に収まるよう変更される。
- 第2インレット：数値は、色のついた範囲で表示される**rslider**の範囲幅の最大値を設定する。**rslider**の設定範囲を超える数値は、**rslider**の範囲内に収まるよう変更される。
- 最小値と最大値は、**rslider**の範囲をマウスでドラッグすることでも、設定や出力ができる。
- リスト** 第1インレット：2つの数値によって、最小値と最大値を設定し、出力する。
- bang** 第1インレット：現在設定されている範囲の最小値と最大値を出力する。
- color** **color**に続く0から15のメッセージによって、または、Objectメニューの**Color**コマンドのオブジェクト色によって、**rslider**が範囲を表示する色を設定できる。
- 実数** 整数に変換される。
- mult** 第1インレット：**mult**に続く数値によって、乗数を指定する。アウトレットから出力する前に、**rslider**オブジェクトの値は、この乗数が掛け合わされる。乗数のデフォルト値は1である。
- set** **set**に続く2つの数値のメッセージによって、数値を出力せずに、範囲を指定する最小値と最大値を設定する。
- size** 第1インレット：**size**に続く整数によって、**rslider**のサイズ（総範囲）を設定する。**rslider**は0から、サイズ値より1小さい値の間の数値を取る。1より小さい**size**メッセージは、自動的に2にセットされる。デフォルトでの**rslider**のサイズ値は128である。

## インスペクター

**rslider**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**rslider**オブジェクトを選択することで、**rslider**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**rslider**オブジェクトを選び、Objectメニューから**Get Info...**を選択



することでも、インスペクターを表示することができる。  
インスペクターでは、Maximumに最大値を入力する。0から指定した最大値より1小さい値までの範囲で、インレットに受け取られる数値は、自動的に制限される。デフォルトの範囲値は128である。また、**rslider**のインスペクターではMultiplierに乗数を指定できる。**rslider**オブジェクトの値は、乗数を掛けた後、アウトレットから出力される。乗数のデフォルト値は1である。

Revertボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

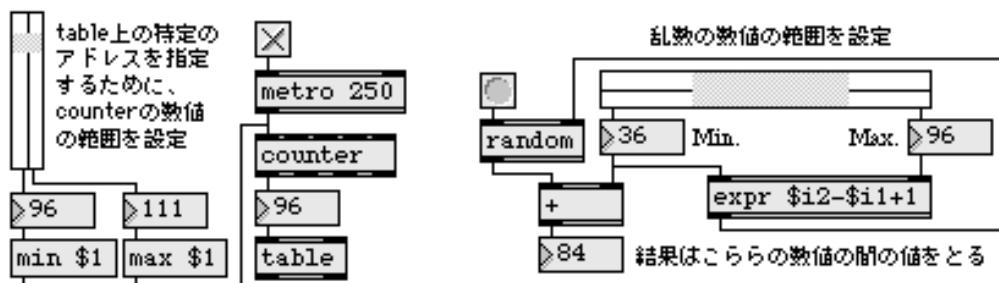
## アーギュメント

なし。

## 出力

**整数** 設定された範囲の最大値が第2アウトレットから、最小値が第1アウトレットから出力される。これらは、第1インレットに新しい最小値（またはbang）を受け取った時や、**rslider**をマウスでドラッグした時に、出力される。

## 利用例



他のオブジェクトの範囲を決めるために、最小値と最大値を出力する

## 参照

- hslider** 画面上のスライダーを動かすことによって数値を出力する
- pictctrl** ピクチャーを用いたコントロール
- pictslider** ピクチャーを用いたスライダー・コントロール
- slider** 画面上でスライダーを動かすことによって数値を出力する
- split** 数値を範囲によって振り分ける
- uslider** 画面上でスライダーを動かすことによって数値を出力する

## 入力

- (MIDI) **rtin**はMIDI入力デバイスよりMIDIリアルタイム・メッセージを受信する。
- enable** **enable 0**のメッセージはオブジェクトを使用不能にし、受信するMIDIデータが無視するようになる。**pcontrol**オブジェクトやMIDIEnable/Disableボタンによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを再び受信可能にする。
- port** **port**の後にaからzの文字を続けるか、MIDI入力デバイスの名前を続けることによって、MIDIリアルタイム・メッセージを受信するポートを設定する。**port**という単語はオプションで、省略することもできる。
- (マウス) **rtin**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

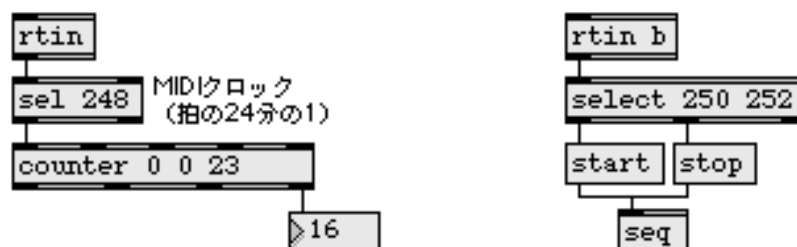
## アーギュメント

- a - z** オプション。MIDIリアルタイム・メッセージ入力を受信するポートを指定する。アーギュメントを指定しない場合、**rtin**はポートaまたは**MIDI Setup**ダイアログのリストの最初の入力ポートから受信する。

## 出力

- 整数** 指定したポートで受信したMIDIリアルタイム・メッセージ（MIDIクロック、スタート、ストップやコンティニュー）をアウトレットから出力する。

## 利用例



Maxと外部のMIDIイベントと同期させるために、MIDIリアルタイム・メッセージを利用する

**参照****clocker**

一定の間隔で経過時間を出力する

**metro**一定の時間間隔で**bang**メッセージを出力する**midin**

生のMIDIデータを受信する

**timein**

外部ソースのタイム・コードを出力する

**seq**

MIDIの録音と再生のためのシーケンサー

Tutorial 16

More MIDI ins and outs (英文PDF参照)

## 入力

- bang** ユーザーがファイル名を入力し、保存する場所を選ぶための標準ファイル保存ダイアログを開く。場所とファイル名は、シンボルとして出力される。
- set** ファイル・タイプを指定する4文字のシンボル（例えば、**TEXT**、**MAXB**）が続く**set**メッセージによって、ダイアログを開くことなく、指定したファイル・タイプとしてファイルを保存するよう設定する。指定したファイル・タイプは、Saveボタンが押された時に、第2アウトレットから出力される。
- 任意** 1つ、またはそれ以上の4文字のファイル・タイプを指定する。これは、ダイアログにポップアップ・メニューとして表示される。Macintoshのファイル・タイプの例としては、テキスト・ファイルの**TEXT**、Maxバイナリー形式でのパッチ・ファイルの**maxb**、AIFFフォーマットのオーディオ・ファイルの**AIFF**がある。シンボルの**fold**は、ユーザーに保存する場所だけを選択するように設定する。

## アーギュメント

- 任意** オプション。ユーザーが選択できる1つまたはそれ以上のファイル・タイプを設定する。**fold**というシンボルは、ユーザーに保存する場所だけを選択するように設定する。

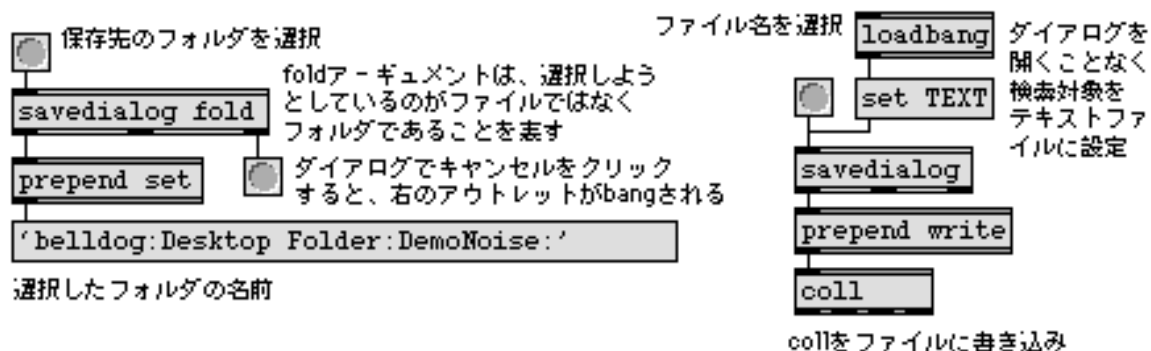
## 出力

- シンボル** 第1アウトレット：ファイルの完全なパス名がシンボルとして出力される。絶対パスはMax 4.1以前のバージョン(コロンを使用)を除き、スラッシュで区切られて表示される。具体的には下記のようなになる。
- “**C:/Max Folder/extras/mystuff/mypatch.pat**”
- パスの形式や種類を変更したい場合は、**conformpath**オブジェクトを使用する。
- シンボル** 第2アウトレット：現在指定しているファイル・タイプの4文字をシンボルとして出力する。



**bang** 第3アウトレット：ユーザーがダイアログでCancelを選んだ時に、**bang**を出力する。

## 利用例



ファイルを保存するために、ファイル形式を指定する。または、保存するフォルダーを選択する

## 参照

- dialog** テキスト入力のためのダイアログ・ボックスを開く
- filedate** ファイルの最終修正日を出力する
- filein** ファイルからバイナリー・データを読み出す
- filepath** 現在のMaxサーチ・パスの情報を出力する
- opendialog** ファイルやフォルダを選択するためのダイアログを開く

注記：このオブジェクトは、Max 4.1以前のバージョンでは使用できない

## 入力

- 整数** 実数に変換される。
- 実数** 第1インレット：入力値は、アーギュメントあるいは他のインレットから受け取られた数値により指定されたマッピングに基づいて拡大縮小される。
- 第2インレット：入力値の下限を設定する。
- 第3インレット：入力値の上限を設定する。
- 第4インレット：出力値の下限を設定する。
- 第5インレット：出力値の上限を設定する。
- 第6インレット：指数関数的に拡大縮小する際の基準値を設定する。最小値は1.0となり、値は正比例に変化する。適正值は1.06となる。
- bang** 第1インレット：直前の入力値に対して拡大縮小を実行する。第1インレットに値が入力されてから拡大縮小範囲が変更された場合は、新しい値を使用して拡大縮小を行う。

## アーギュメント

- 整数または実数** オプション。1つ目のアーギュメントは入力値の最小値を、2つ目のアーギュメントは入力値の最大値を表す。同様に3つ目と4つ目のアーギュメントはそれぞれ出力値の最小値と最大値を表す。5つ目のアーギュメントはオプションで、拡大縮小の比例カーブを設定する。1から値が大きくなるほど、指数関数のカーブは急峻になる。1以下の数値が入力されると、値は正比例で変化する。この第5アーギュメントには、1より大きな実数を指定するのが適切である。これら5つの値は、オブジェクトの5つのインレットに入力される値により変更される。整数の値を出力するには、5つ目のアーギュメントを指定せずに、最初の4つのアーギュメントすべてに整数を指定する必要がある。

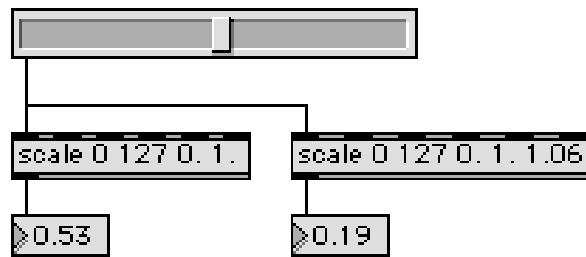
## 出力

- 整数** アーギュメントが4つだけ指定され、それらがすべて整数である場合にのみ整数の値が出力される。それ以外の場合は、実数で出力される。
- 実数** **scale** オブジェクトが第1インレットに値を受け取ると、その値は指定された出力範囲内で呼応するポジションになるように拡大縮小される。

# scale

入力値を指定した数値の範囲にマッピングし、指定した出力範囲に拡大縮小および再マッピングして出力する

## 利用例



整数のスライダー値を、実用的な実数に変換した利用例

## 参照

### zmap

入力値を指定した数値の範囲にマッピングし、指定した出力範囲に再マッピングして出力する

### expr

数式の値を求める

## 入力

**bang** メインのスクリーンの大きさ、また全体の複数のモニターのスクリーンの矩形のサイズを、アウトレットから出力する。

## アーギュメント

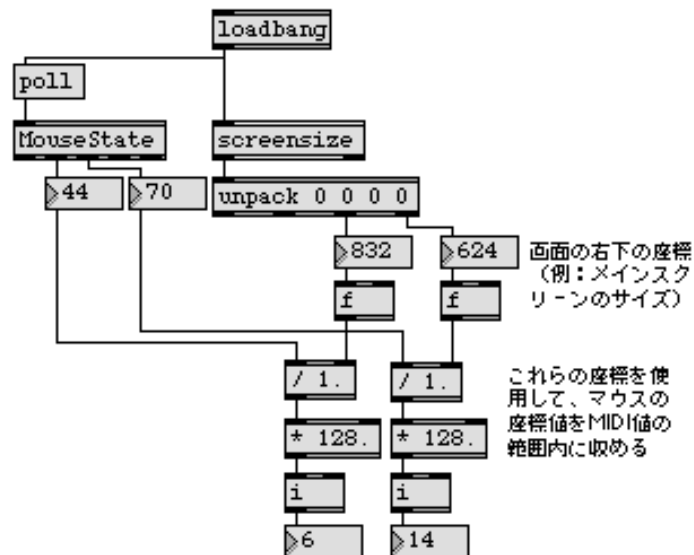
なし。

## 出力

**リスト** 第1アウトレット：メインのスクリーンのサイズの座標（左、上、右、下と続く）を出力する。

第2アウトレット：すべてのモニターを合わせたサイズの座標が出力される。1つしかモニターがない場合は、出力は第1アウトレットのものと同じになる。

## 利用例



## 参照

**gestalt**  
**menubar**  
**thispatcher**

現在のシステムについて調べる  
独自のメニュー・バーを作る  
パッチにメッセージを送る

## 入力

**任意のメッセージ** 第1インレット：入力がアーギュメントのうち1つと一致した場合には、そのアーギュメントに対応するアウトレットから**bang**を出力する。それ以外の場合は、入力が一番右に位置するアウトレットから出力される。

注記：**select**オブジェクトは、それらの値が同じでも、整数を実数と一致するとは見なさず、また逆の場合でも同じである。例えば、**4.0**は**4**のアーギュメントと一致するとは見なされず、同じように**4**も**4.0**とは一致しない。

**整数** 第2インレット：アーギュメントの値を置き換える。第2インレットは、アーギュメントとして整数が1つだけ指定されている場合のみ存在する。

**bang** 第1インレット：**bang**というシンボルに変換され、他のシンボルと同じように扱われる。

## アーギュメント

**任意** オプション。アーギュメントは、整数、実数、またシンボルを混在させて指定できる。アーギュメントの数はアウトレットの数を決定し、それに一番右に位置するアウトレットが加えられる。アーギュメントが設定されていなければ、0の数値が割り当てられたアウトレットが1つ作られる。

**整数** アーギュメントとして整数が1つだけ指定されている場合（または、アーギュメントが指定されていない場合）、2つ目のインレットが右につくられる。そのインレットに受け取った数値は、アーギュメントとして格納される。1つ以上のアーギュメントがある場合、またはその1つだけのアーギュメントが整数でなかった場合には、その第2インレットは作られない。

## 出力

**bang** 第1インレットに受け取った数値かシンボルが、アーギュメントのうち1つと一致した場合は、そのアーギュメントに対応するアウトレットから、**bang**が出力される。

**任意** 第1インレットに受け取った数値かシンボルが、アーギュメントのうち1つとも一致しなかった場合は、入力が一番右に位置するアウトレットから出力される。

## 利用例



アーギュメントは名前か数値をアウトレットに割り当て、入力がそれらと一致した時に**bang** を出力する

## 参照

|              |                        |
|--------------|------------------------|
| <b>if</b>    | if/then/else形式の条件文     |
| <b>match</b> | 数値の並びを探し、リストとして出力する    |
| <b>route</b> | 選択的に入力を特定のアウトレットから出力する |
| <b>==</b>    | 2つの数値を比較し、等しければ1を出力する  |

Tutorial 17      Gate and switches (英文PDF参照)

## 入力

任意のメッセージ インレットに受け取ったメッセージは、同じ名前の**receive**オブジェクトのアウトレットから出力される。開いている他のパッチの中に、同じ名前の**receive**オブジェクトが存在する場合も同様である。

(マウス) **send**オブジェクトをダブルクリックすると、同じ名前の**receive**オブジェクトを含むすべてのウィンドウが開く。

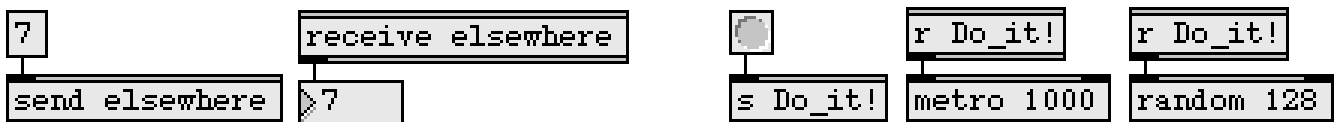
## アーギュメント

任意のシンボル 必須。**send**オブジェクトに名前を付ける。

## 出力

任意のメッセージ アウトレットは持たない。**send**オブジェクトのインレットに受け取ったメッセージは、同じ名前の**receive**オブジェクトのアウトレットから出力される。開いている他のパッチ内に、同じ名前の**receive**オブジェクトが存在する場合も同様である。

## 利用例



同じ名前を持ったすべての**send**オブジェクトと**receive**オブジェクトは繋がっている

## 参照

**forward message** 様々なオブジェクトにパッチ・コードなしでメッセージを送信する  
任意のメッセージを送る  
**pv** パッチとそのサブ・パッチ内に限定して、変数を共有する  
**receive value** パッチ・コードなしでメッセージを受信する  
他のオブジェクトと格納されている値を共有する

Tutorial 24 send and receive (英文PDF参照)



## 入力

- bang** **seq**に格納しているシーケンスの再生を開始する。
- start** アーギュメントがない**start**メッセージは、**bang**と同じ働きをする。**start**に数値が続く場合は、その数字によって決定されるテンポで、格納しているシーケンスを再生する。**start 1024**メッセージは通常のテンポを指定する。数値が512なら、本来のテンポの半分の速度で再生し、2048であれば2倍の速度で再生する。
- start -1**メッセージは、シーケンスの再生を開始するが、Maxのミリ秒のクロックに従うのではなく、**tick**メッセージを受け取って**seq**のクロックを進めるようになる。以下の**tick**メッセージを参照のこと。
- record** インレットに受け取るMIDIメッセージの録音を開始する。
- stop** 録音や再生を行っていれば、シーケンサーを停止する。再生から録音へ切り替える場合や、あるいは、その逆の場合でも、**stop**メッセージを送る必要はなく、直接切り替えることができる。
- append** すでに存在しているシーケンスを消去せずに、格納しているシーケンスの最後から録音を開始する。
- 整数** **seq**が録音中であれば、受け取った数値はMIDIメッセージのバイトとして解釈される (**midiformat**や**midiiin**から数値を受け取るのが一般的である)。**seq**はMIDIチャンネル・メッセージやシステム・エクスクルーシブ・メッセージを録音するが、**start**、**stop**、MIDIクロックなどのMIDIリアルタイム・メッセージに直接反応することはない。
- 実数** 整数に変換される。
- tick** **seq**が**start -1**メッセージを受け取った後、クロックを進めるために**tick**メッセージを待つ。録音された本来のテンポでシーケンスを再生するためには、**seq**は1秒に48回の割合で**tick**メッセージを受け取る必要がある。これは、120MMのテンポで4分音符につき24回の**tick**メッセージを受け取ることと同じである (標準のMIDIクロック・メッセージと同じ)。**tick**メッセージを使ってシーケンサーを進めることで、再生のテンポを変化させたり、他のタイミング・ソース (例えば、受信したMIDIクロック・メッセージ) と**seq**を同期させることができる。
- delay** **delay**に続く数字によって、録音されたシーケンスにおける最初のイベントのオンセットの時刻をミリ秒で設定する。これによって、シーケンスのすべてのイベントが移動し、最初のイベントは指定したオンセットの時刻に始まる。

- hook** **hook**に続く数値を、格納しているシーケンスのすべてのイベントの時刻に掛ける。例えば、数値が2.0であれば、すべてのイベントの時刻は2倍になり、シーケンスは半分の速度で再生される。シーケンスが再生中でも、この掛け算を行うことができる。
- write** 標準ファイル保存ダイアログを開き、録音したシーケンスを独立したファイルとして保存する。テキスト・エディターでシーケンスを編集する場合は、ダイアログの*Save As Text*をチェックしておく。
- read** アーギュメントがなければ、**read**は標準ファイル・ダイアログを開き、以前に録音されたシーケンスを**seq**に読み込み、現在のシーケンスと置き換える。アーギュメントとしてシンボルを指定すれば、ファイル名がシンボルに一致するファイルを探して、**seq**オブジェクトに読み込む。

注記：**seq**オブジェクトは、単一トラック（フォーマット0）のスタンダードMIDIファイルとして読み書きを行う。**seq**はテキスト・ファイルとしても読み書きをすることができ、各行ごとにミリ秒単位での開始時刻（シーケンスの最初からの経過時間）と、空白文字で区切ったMIDIメッセージのバイト列がある。例えば、

```
0 144 60 122
1000 144 60 0
1500 192 31
1500 144 60 112
2500 144 60 0
```

は、チャンネル1の中央のCを一秒間鳴らし、500ミリ秒後にプログラム・ナンバーを31に変えて、中央のCを1秒間鳴らす。

- print** 録音されたシーケンスの最初の16個のイベントを、Maxウィンドウに表示する。
- dump** 標準ファイル・ダイアログが開くので、保存されたシーケンスかスタンダードMIDIファイルを選択する。選択されたファイルをテキストとして開き、新しいUntitledという名前でテキスト・ウィンドウに表示する。テキストを編集したり、保存することもできる。

## アーギュメント

任意のシンボル オプション。パッチが読まれた時、自動的に**seq**に読み込まれるファイルの名前を指定する。

## 出力

**整数** 第1アウトレット：**bang**や**start**をインレットに受け取った時、**seq**に格納されているシーケンスを、個々のMIDIバイトとしてアウトレットから出力する。この出力は、**midiparse**や**midiout**に送るのが一般的である。

**bang** 第2アウトレット：現在のシーケンスの再生を終えたことを示す（シーケンスの最後のイベントが再生される直前に、**bang**を出力する）。

## 利用例



ライブ・パフォーマンスの録音や再生を行う。あるいは、以前に録音したシーケンスを再生する

## 参照

**coll** 様々な種類のメッセージを格納し、編集する  
**follow** ライブなものと録音されたものを比較する  
**mtr** マルチ・トラック・シーケンサー

Tutorial 35 seq and follow (英文PDF参照)

**serial**オブジェクトは、標準のシリアル・ドライバーがサポートしているポートとデバイスのみ機能する。USB-シリアル・アダプターが繋がれていない場合は、USBポートやUSBデバイスに対しては機能しない。

## 入力

- 整数** **serial**オブジェクトがアクセスしているシリアル・ポートから、数値を送信する。0から255の範囲を超える数値は送信しない。
- リスト** リストの数値を左から右の順序で、シリアル・ポートから送信する。0から255の範囲を超える数値は送信しない。
- bang** 最後に**bang**のメッセージを受け取った以降に、シリアル・ポートに受信した文字を、受信した順序で、**serial**オブジェクトのアウトレットから整数値として出力する。
- bufsize** **bufsize**に続く数値によって、**serial**オブジェクトが使用するシリアル・ポートの受信用バッファ・サイズを設定する。**bufsize 0**メッセージは、ハードウェアによって決まるシリアル・ポートのバッファ・サイズの初期値に戻す。
- baud** **baud**に続く適切な速度の数値によって、**serial**オブジェクトが使用するシリアル・ポートのボーレートを設定する。適切な速度のリストは、以下のアーギュメントを参照のこと。

## アーギュメント

- シンボル a-z** 必須。オブジェクトが使用するシリアル・ポートを指定する。**a**はコンピュータの最初の論理的シリアル・ポートを指定し、**b**から**z**は追加されたポートを指定する。**serial**オブジェクトが作られる時に、そのポートがすでに使われている場合は、エラー・メッセージが表示され、オブジェクトは機能しない。
- 整数** オプション。シリアル・ポートのボーレートを指定する。デフォルト値は4800baudである。この他に指定できるボーレートは、**300**、**600**、**1200**、**1800**、**2400**、**3600**、**7200**、**9600**、**19200**、**38400**、**57600**である。
- 整数** オプション。ボーレートの次のアーギュメントは、シリアル・ポートのデータ・ビット数を指定する。デフォルト値は8であり、この他に指定できる数値は**5**、**6**、**7**である。
- 整数** オプション。次のアーギュメントは、シリアル・ポートのストップ・ビット数を指定する。デフォルト値は1である。この他に指定できる数値は**1.5**と**2**である。

**シンボル** オプション。次のアーギュメントはシリアル・ポートのパリティーを設定する。デフォルト値はno（パリティーなし）である。この他に指定できる値は、**even**（偶数）と**odd**（奇数）である。

## 出力

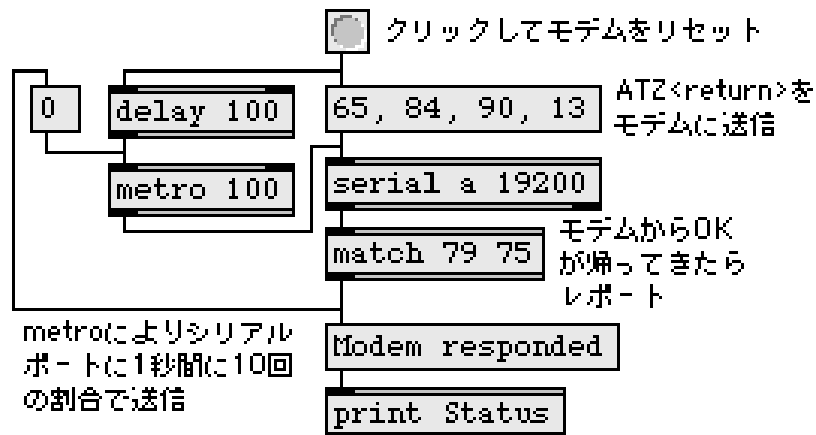
### （シリアル・ポート出力）

インレットに数値やリストを受け取った場合は、**serial**オブジェクトは指定されたシリアル・ポートから、現時点でのボーレートでデータを送信する。

**整数** 第1アウトレット：**serial**オブジェクトが**bang**メッセージを受け取り、シリアル・ポートが受信した文字があれば、受信した順序で、受信した文字を数値として出力する。

第2アウトレット：エラー・メッセージを出力する。

## 利用例



ボタンがクリックされた時、このパッチはモデムをリセットし、  
応答を受け取るためポーリングを開始する。応答があれば、ポーリングを停止する

## 参照

**match**  
**spell**  
**vdp**

数値の並びを探し、リストとして出力する

入力をASCIIコードに変換する

シリアル・ポートを通じて、ビデオディスク・プレーヤーをコントロールする

## 入力

**bang** 第1インレット：**setclock**オブジェクト自身の現在のクロックに従って、現在の時間の値を出力する。**clocker**、**line**、**metro**、**pipe**、**tempo**、**timeline**のようなタイミング関連のオブジェクトは、Maxの標準のミリ秒クロックではなく、クロック・ソースとして**setclock**オブジェクトを使うことができる。

**整数または実数** 第1インレット：数値の意味は、**setclock**オブジェクトの第2アーギュメントによって設定される操作モードに依存する。モードが**pass[ive]**（デフォルトのモード）の場合は、タイミング関連のオブジェクトが、時間の初期値と比較するために用いる絶対的なクロック・タイムを、その数値によって設定する。また、**add[itive]**モードの場合には、その数値は**setclock**オブジェクトの現在のクロック・タイムに加えられる。**interp[olate]**モードの場合は、**setclock**オブジェクトは、以前に受け取った数値からの経過時間によって決定される時間間隔において、その数値によってクロック・タイムを増加させて設定する（しかし、負の値ならクロック・タイムは減少することになる）。また、**ext[ernal]**モードや**mul[tiplicative]**モードの場合は、その数値は単純に無視される。**mul[tiplicative]**モードであれば、その数値はタイミング関連のオブジェクト群の乗数として用いられる。例えば、0.5という数値は、タイミング関連のオブジェクト群の増加レート（速度）を半分にする。

第2インレット：**setclock**オブジェクトが、クロック情報をタイミング関連のオブジェクト群に送信する時間間隔を、ミリ秒単位で設定する。デフォルトは5ミリ秒である。

**set** **setclock**オブジェクトが、**pass[ive]**モードや**add[itive]**モードの場合は、**set**メッセージに続く数値によって、クロック・タイムを設定する。**setclock**オブジェクトが他のモードの場合は、**set**メッセージは無視される。

**reset** **setclock**オブジェクトが**interp[olate]**モードの場合には、**reset**メッセージに続く数値によって、クロック・タイムを設定し、その後、そのオブジェクトが行った最後の補完を繰り返して行う。

## アーギュメント

任意のシンボル 必須。1つ目のアーギュメントは、**clocker**、**line**、**metro**、**pipe**、**tempo**、**timeline**オブジェクトのようなタイミング関連のオブジェクトによって参照される、**setclock**オブジェクトの名前を指定する。タイミング関連の（以前に**setclock**オブジェクトの名前が続く**clock**メッセージを受け取っている）オブジェクトは、Max内部の標準のミリ秒クロックではなく、その**setclock**オブジェクトをタイミング・ソースとして使



用する。**setclock**オブジェクトは、そのオブジェクトを参照しているタイミング関連のオブジェクトと、同じパッチ内に存在している必要はない。1つ以上の**setclock**オブジェクトが同じ名前が存在することもあり、それらは同じクロック・タイムの情報を共有する。(注記：同じ名前のアーギュメントを持つ異なる**setclock**オブジェクトは、異なるモードをアーギュメントとして設定できるが、実際にはその名前ですべて最初に読み込まれた**setclock**オブジェクトのモードによる操作となる。従って、同じ名前が異なるモードが設定された**setclock**オブジェクトはそれらがMaxに読み込まれた順序がわからなければ、予想できない振る舞いをするかもしれない。)

第2アーギュメント (オプション) は、**setclock**オブジェクトに設定されるクロックの操作モードを表す。第2アーギュメントとして設定可能なモードは、以下の通りである。

- pass** *passive*モードを指定する。このモードでは、**setclock**オブジェクトの現在のクロック・タイムは、第1インレットに受け取った数値によって設定される。タイミング関連のオブジェクトは、標準的に進行するミリ秒クロックであるかのように、**setclock**のクロック・タイムに従うようになる。2つ目のアーギュメントが指定されていない場合は、この**pass**モードがデフォルトとなる。
- add** *additive*モードを指定する。第1インレットに受け取った数値は、現在のクロック・タイムに加えられ、新しいクロック・タイムを決定する。
- mul** *multiplicative*モードを指定する。第1インレットに受け取った数値は、すべてのタイミング関連のオブジェクトが、それらのタイミングの設定を変更する乗数として用いられる。例えば2.0という乗数は、**setclock**オブジェクトをクロック・ソースとして用いているタイミング関連のオブジェクトのタイミングの値を2倍にする (速度を半分にする) ことになる。**mul**モードの振る舞いを理解するもう1つの (おそらく、より本来の) 方法は、入力された実数を、**setclock**オブジェクトが、それ自身のクロック・タイムの進み具合を除算するための除数として考えることである。このように、**setclock**オブジェクトが2.0という数値を受け取った時には、**setclock**のクロック速度を2.0によって除算し、そのクロックを参照しているオブジェクトを2倍遅くさせる。
- interp** *interpolate*モードを指定する。第1インレットに受け取った数値は、以前に数値が受け取った時から経過した時間によって決定される時間間隔として、**setclock**オブジェクトの現在時間に段階的に加えられる。その時間の間に、**setclock**オブジェクトはクロックをリニアに補完する。
- 実数** 2つ目のアーギュメントが**mul**の場合、オプションとしての3つ目のアーギュメントは、すべてのタイミング関連のオブジェクトのタイミン



グの乗数を設定する。3つ目のアーギュメントが指定されていない場合は、デフォルトとして乗数は1.0に設定される。

**ext** *external*クロック・モードを指定する。OMSがインストールされていれば、**setclock**オブジェクトはOMSによるタイミングの時間情報と同期することになる。**ext[ernal]**モードの**setclock**オブジェクトからタイミングを取得しているタイミング関連のオブジェクトは、この方法でOMSタイミングから、タイミングをミリ秒単位で取得する。時間を進めるためには、OMSタイミングは、サポートされているアプリケーションによって開始する必要がある。Maxでは、**timeout**オブジェクトに**start**メッセージを送ることで開始することができる。

2つ目のアーギュメントにオプションとして加えることのできるモードは、以下の通りである。

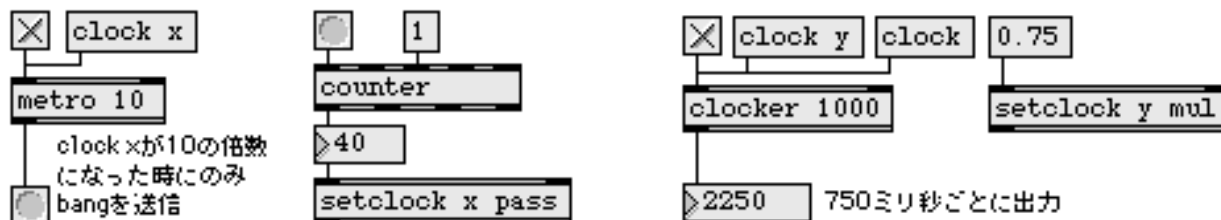
**beats** OMSタイミングから取得するタイム・ユニットが、ミリ秒ではなくティックス（四分音符）であること以外は、**ext**モードと同じである。任意のbeatsモードの**setclock**オブジェクトから時間を取得しているタイミング関連のオブジェクトは、基準となるユニットに従って、ティックスをミリ秒へ置き換える。ユニットのレートは、OMSタイミングのテンポ（Maxでは**timeout**オブジェクトに**tempo**メッセージを送ることによって設定できる）と、Maxが**setclock**オブジェクトにビートを出力するために使用しているティックスの精度（Maxでは**timeout**オブジェクトに**granularity**メッセージを送ることで設定できる）に従って、決定される。時間を進めるためには、**ext**モードと同じように、OMSタイミングを開始させなくてはならない。

**smpte** OMSのタイミングから取得するタイム・ユニットが、ミリ秒ではなくOMSタイミングの現在のSMPTEフレーム・レートのビット（1フレームの1000分の80）であること以外は、**ext**と同じである。*smpte*モードの**setclock**オブジェクトから時間を取得しているタイミング関連のオブジェクトは、基準となるユニットに従って、ビットをミリ秒へと置き換える。フレーム・レートは、**timeout**オブジェクトに**format**メッセージを送ることで変更できる。時間を進めるためには、**ext**モードと同じように、OMSタイミングを開始させなくてはならない。

## 出力

**整数** 第1インレットに**bang**を受け取った時に、**setclock**オブジェクトは設定されている現在の時間をアウトレットから出力する。

## 利用例



**setclock** オブジェクトは **metro** オブジェクト  
のクロックになる

**setclock** オブジェクトは **clocker** オブジェクト  
の時間を修正する

## 参照

**clocker**  
**metro**  
**timeline**  
**timer**

一定の間隔で経過時間を出力する  
一定の時間間隔でbangメッセージを出力する  
Maxメッセージを時間軸で扱うスコア  
2つのイベント間の経過時間を出力する

## 入力

実数または整数 ラディアン単位でのサイン関数への入力。

## アーギュメント

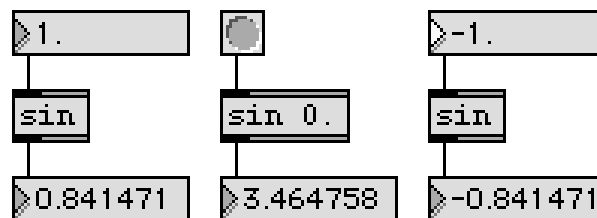
なし。

## 出力

整数または実数 ラディアン単位で入力された値のサイン値。

## 利用例

・実数による入力



・入力値に対するサイン関数

## 参照

|              |                    |
|--------------|--------------------|
| <b>acos</b>  | アーク・コサイン関数         |
| <b>asin</b>  | アーク・サイン関数          |
| <b>atan</b>  | アーク・タンジェント関数       |
| <b>atan2</b> | アーク・タンジェント関数 (2変数) |
| <b>cos</b>   | コサイン関数             |
| <b>cosh</b>  | 双曲線コサイン関数          |
| <b>sinh</b>  | 双曲線サイン関数           |
| <b>tanh</b>  | 双曲線タンジェント関数        |

## 入力

**実数または整数** 双曲線サイン関数への入力。

**bang** 第1インレット：現在格納されている数値の双曲線サイン値を計算する。アーギュメントがない場合は、**sinh**への初期値は0である。

## アーギュメント

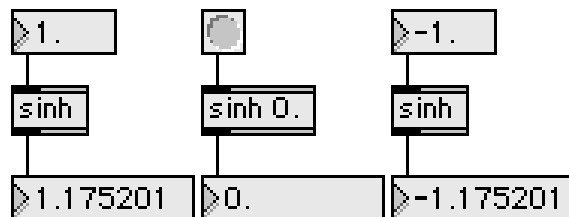
**実数または整数** オプション。双曲線サイン関数への初期値を設定する。

## 出力

**整数または実数** 入力された値の双曲線サイン値。

## 利用例

・実数による入力



・入力値に対する双曲線サイン関数

## 参照

|              |                    |
|--------------|--------------------|
| <b>acos</b>  | アーク・コサイン関数         |
| <b>asin</b>  | アーク・サイン関数          |
| <b>atan</b>  | アーク・タンジェント関数       |
| <b>atan2</b> | アーク・タンジェント関数 (2変数) |
| <b>cos</b>   | コサイン関数             |
| <b>cosh</b>  | 双曲線コサイン関数          |
| <b>sin</b>   | サイン関数              |
| <b>tanh</b>  | 双曲線タンジェント関数        |



## 入力

**整数** インレットに受け取った数値は、**slider**オブジェクトがグラフィカルに表示し、アウトレットから出力する。オプションとして、出力する前に、その数値に乗数を掛け算し、オフセットを足し算することもできる。

(マウス) **slider**をマウスで直接ドラッグすることによって、数値を出力する。

**実数** 整数に変換される。

**bang** **slider**に現在格納されている数値を出力する。

**min** **min**に続く数値によって、出力する前に、**slider**オブジェクトの値に足し算する数値（オフセット）を設定する。デフォルト値は0である。

**mult** **mult**に続く数値によって、掛け算する乗数を設定する。**slider**オブジェクトの値は、最初に掛け算が行われた後、オフセットの数値を足し算し、出力される。デフォルト値は1である。

**set** **set**に続く数値によって、出力することなく、**slider**が表示する数値を設定する。

**size** **size**に続く数値によって、**slider**オブジェクトの範囲を設定する。デフォルト値は128である。

## インスペクター

**slider**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**slider**オブジェクトを選択することで、**slider**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**slider**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**slider**のインスペクターでは、*Slider Range*を設定することができる。インレットに受け取った数値は、0から範囲値（Slider Rangeの数値）より1小さい数値の自動的に変更される。デフォルトの範囲は128である。掛け算をした後、数値に足される*Offset*の数値を設定することができる。デフォルトのオフセットは0である。*Multiplier*では掛け算の乗数を設定することができる。**slider**オブジェクトの数値は、アウトレットから出力する前に、乗数を掛けられる。掛け算はオフセットの足し算の前に行なわれる。デフォルトの乗数は1である。



## アーギュメント

**slider**の範囲は、(パッチ・ウィンドウがロックされていない状態で)オブジェクトを選んで、Objectメニューより**Get Info...**を選ぶことにより設定できる。**slider**は、その新しい範囲に合わせて自動的に大きさを変える。

インスペクターではMultiplierとOffsetを設定することができる。Multiplierは、**Slider**の数値を出力する前に掛け算する。Offsetは、掛け算の後に数値を足す。新しく作られた**slider**は、範囲は128、乗数は1、オフセットは0となっている。

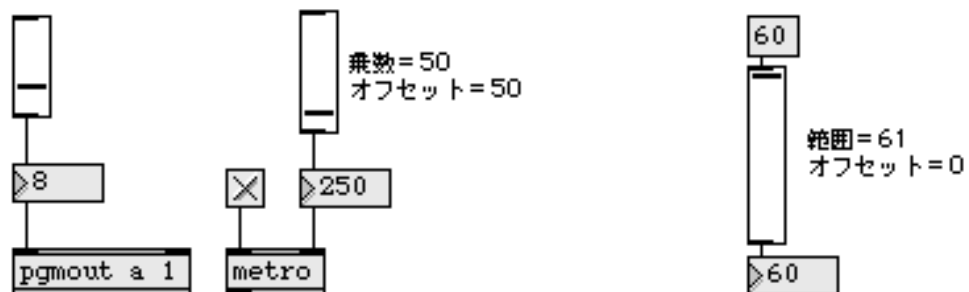
Revertボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## 出力

**整数** インレットに数値を受け取ったり、マウスで**slider**をドラッグすることによって、**Slider**の数値に乗数を掛け算し、オフセットを足し算し、得られた結果をアウトレットから出力する。

ドラッグによって出力する数値は**slider**の範囲に限られるが、インレットに受け取る数値は、アウトレットから出力される前には、制限されない。

## 利用例



**Slider**をドラッグすることで数値を出力する。または、通過する数値を表示する



## 参照

|                   |                            |
|-------------------|----------------------------|
| <b>dial</b>       | ダイアル操作によって数値を出力する          |
| <b>hslider</b>    | 画面上のスライダーを動かすことによって数値を出力する |
| <b>kslider</b>    | 画面上のキーボードから数値を出力する         |
| <b>pictctrl</b>   | ピクチャーを用いたコントロール            |
| <b>pictslider</b> | ピクチャーを用いたスライダー・コントロール      |
| <b>rslider</b>    | 数値の範囲を変更し、それを表示する          |
| <b>uslider</b>    | 画面上でスライダーを動かすことによって数値を出力する |

|             |                             |
|-------------|-----------------------------|
| Tutorial 9  | Using the slider (英文PDF参照)  |
| Tutorial 14 | Sliders and dials (英文PDF参照) |



## 入力

- 任意** 第1インレット：特定の最小限の時間が以前の出力から経過した時に、メッセージをアウトレットから出力する。それ以外の場合は、そのメッセージはその時間が経過するまで保持されている（または、新たに入力されたメッセージによって、上書きされる）。
- 整数** 第2インレット：その数値は、次の出力までのミリ秒単位の最小限の時間として格納される。
- clock** 第1インレット：**clock**には、既に存在している**setclock**オブジェクトの名前が続き、Max内部のミリ秒クロックではなく、その**setclock**オブジェクトによって**speedlim**オブジェクトがコントロールされるように設定する。**clock**だけのメッセージなら、Max内部のミリ秒クロックによって、**speedlim**オブジェクトがコントロールされることになる。

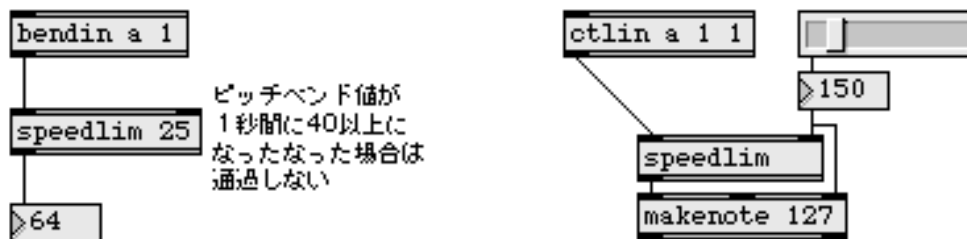
## アーギュメント

- 整数** オプション。出力の間隔の最小限の時間の初期値を、ミリ秒単位で設定する。アーギュメントが設定されてなければ、最小限の時間は0ミリ秒に設定される。

## 出力

- 任意** 特定の最小限の時間が以前の出力から経過した時に、第1インレットに受け取ったメッセージを出力する。それ以外の場合は、**speedlim**オブジェクトはその時間が経過するまで待機し、そして、以前の出力をした時以降に受け取った最後のメッセージを出力する。

## 利用例



数値の大量の流れを間引く、また、連続的な流れを分離したステップに変えるのに用いられる

## 参照

- delay** **bang**を遅らせて出力する
- mousefilter** マウスボタンが押されていない時だけ、数値を通過させる
- thresh** 短時間に受け取った数値をリストとしてまとめる
- timer** 2つのイベント間の経過時間を出力する

## 入力

- 任意のシンボル** シンボル内のそれぞれの文字や数字などのASCII値を1度に1文字ずつ出力する。
- 整数** その数値の各桁の数字のASCII値を、1度に1つずつ出力する。
- リスト** リスト内のそれぞれの整数を上記のようにASCII値へと変換し、空白文字（32）をリストの項目の間に出力する。リストの項目内の任意の実数やシンボルは無視される。

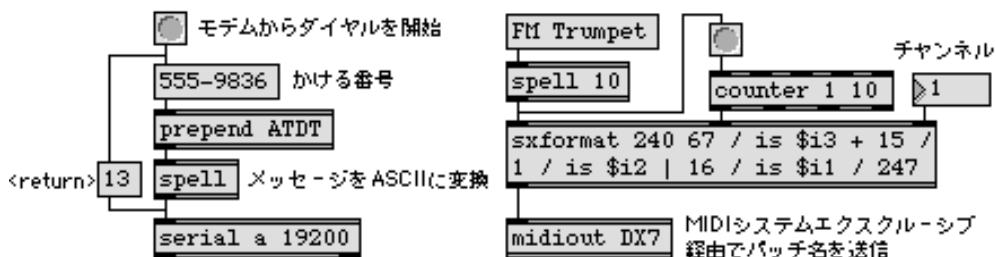
## アーギュメント

- 整数** オプション。1つ目のアーギュメントは出力の最小値を設定する。その最小の長さに満たない任意の入力は、その最小の要求を満たすために特定の文字（デフォルト値では空白文字、ASCII値では32にあたる）によって、十分な長さになるように埋められる。オプションとして、2つ目のアーギュメントによって、その埋めるための文字を32以外に指定することができる。その埋めるための文字に0を指定したい場合は、任意の負の値を**spell**オブジェクトの第2アーギュメントとして用いれば良い。

## 出力

- 整数** 入力値のASCII表現を、1度に1文字ずつ出力する。

## 利用例



**spell** オブジェクトを用いて、モデム・コマンドの文字列、またシンセサイザー・パッチ名を、人間に理解できる形からコンピューターが理解できる形に変換でき、ASCII表現でシリアル・ポートより出力することができる

## 参照

- key** コンピューター・キーボードのキー入力を出力する
- keyup** コンピューター・キーボードのキー入力を出力する
- message** 任意のメッセージを送る
- sprintf** 単語や数値のメッセージをフォーマットする

## 入力

**整数または実数** 第1インレット：その数値が指定された範囲内にある場合には、それを第1アウトレットから出力する。それ以外の場合には、第2アウトレットから出力する。

第2インレット：その数値は**split**オブジェクトによって検索される数値の範囲の最小値として格納される。その値が整数なら**split**オブジェクトはすべての実数の値を整数へと変換する。

第3インレット：その数値は**split**オブジェクトによって検索される数値の範囲の最大値として格納される。

**リスト** 第1インレット：2つ目の数値を範囲の最小値として格納し、3つ目の値を範囲の最大値として格納する。1つ目の値はその範囲と比較され、2つのアウトレットのどちらかから出力される。

## アーギュメント

**整数または実数** オプション。1つ目のアーギュメントは第1アウトレットから出力される最小値を設定する。その1つ目のアーギュメントが整数である場合、**split**オブジェクトはすべての実数の値を整数へと変換する。2つ目のアーギュメントは第1アウトレットから出力される最大値を設定する。**split**オブジェクトの1つ目のアーギュメントが整数である場合、その出力は整数になる。実数の場合には出力は実数となる。これは入力の型に関係なく行われる。

## 出力

**整数** 第1インレットに受け取った数値が、指定された最小値以上で指定された最大値以下なら、その数値を第1アウトレットから出力する。それ以外の場合では、その数値を第2アウトレットから出力する。

## 利用例



異なる用途のために、特定の範囲の数値を分配するために用いられる

## 参照

### **route select**

**<=**

**>=**

選択的に入力を特定のアウトレットから出力する

特定の入力を選択し、それ以外は通過させる

2つの数値を比較し、より小さいか等しいことを判断する

2つの数値を比較し、より大きいか等しいことを判断する

Tutorial 20

Using the computer keyboard (英文PDF参照)

## 入力

**リスト** リストの1つ目の数値はアウトレットの番号を指定し、そのアウトレットからリストの2つ目の数値を出力する。リストにさらに数値があれば、それらは、リストの1つ目の数値によって指定されたアウトレットの右に続くアウトレットから、それぞれ出力される。リストは整数だけしか含むことができない。実数（また、シンボル）は無視される。

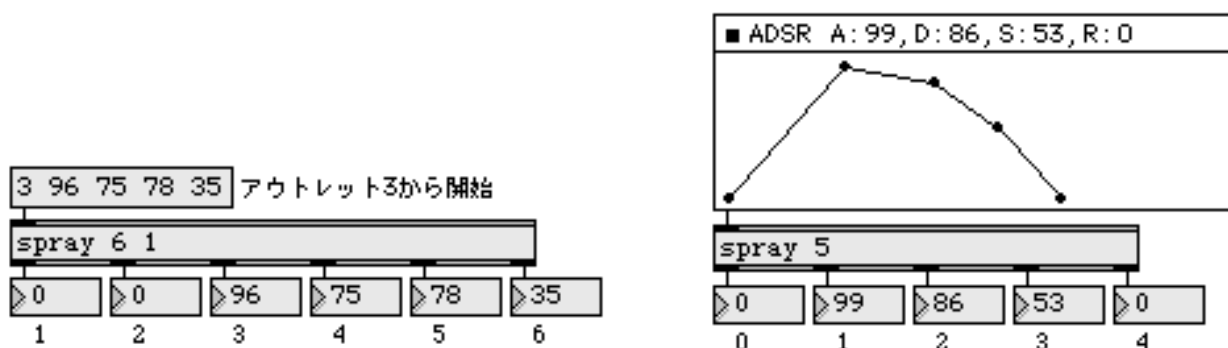
## アーギュメント

**整数** オプション。1つ目のアーギュメントはアウトレットの数を指定する。アーギュメントが指定されていない場合には、このオブジェクトは2つのアウトレットを持つことになる。2つ目のアーギュメントはアウトレットの番号のオフセット値を設定する。2つ目のアーギュメントが指定されていない場合には、アウトレットは0から番号をつけられる。

## 出力

**整数** 整数のリストが**spray**オブジェクトに受け取った時に、リストの1つ目の数値はアウトレットを指定するために用いられ、2つ目の数値はそのアウトレットから出力される。リストにオプションとして追加される数値は、右に続くアウトレットから出力される。**env**オブジェクト、また**envi**オブジェクトのアウトレットを**spray**オブジェクトのインレットに接続することで、エンベロープの値を異なるアウトレットから出力することができる。

## 利用例



リストを分割し、その項目を特定のアウトレットから出力するために用いる

## 参照

|               |                                   |
|---------------|-----------------------------------|
| <b>cycle</b>  | 独立したアウトレットからデータの流を出力する            |
| <b>env</b>    | スクリプト定義できるエンベロープ・エディター            |
| <b>envi</b>   | パッチ・ウィンドウに置く、スクリプトによって定義できるエンベロープ |
| <b>funnel</b> | 受け取ったインレットの番号を付け加えて出力する           |
| <b>gate</b>   | 入力を特定のアウトレットから出力する                |
| <b>route</b>  | 選択的に入力を特定のアウトレットから出力する            |
| <b>unpack</b> | リストを個々のメッセージに分割する                 |

## 入力

- 整数** **%ld**か、**%c**のアーギュメントに対応する任意のインレットに受け取ることができる。この数値は、そのアーギュメントの位置に格納される。**%c**アーギュメントは、整数に対応するASCII文字に変換する。
- 実数** **%f**アーギュメントに対応する任意のインレットに受け取ることができる。この数値は、そのアーギュメントの位置に格納される。
- シンボル** **%s**アーギュメントに対応する任意のインレットに受け取ることができる。このシンボルは、そのアーギュメントの位置に格納される。
- リスト** 第1インレット：リストのそれぞれの項目は、インレットの数まで、それぞれのインレットに受け取られたかのように扱われる。
- bang** 第1インレット：現在格納している値を用いて、メッセージをフォーマットする。

上記のメッセージのいずれかを第1インレットに受け取れば、メッセージをフォーマットし、出力する。置き換えができるアーギュメント(**%ld**か**%f**)に値を受け取っていない場合は、そのアーギュメントは0となる。**%s**か**%c**アーギュメントに値を受け取っていない場合は、そのアーギュメントは空白となる。

## アーギュメント

- symout** オプション。1つ目のアーギュメントが**symout**というシンボルの場合は、**sprintf**オブジェクトは1つのシンボルとして生成した文字列を出力する。それ以外の場合は、シンボル、数値のリストを出力する。**symout**自体は、**sprintf**オブジェクトの出力に含まれない。

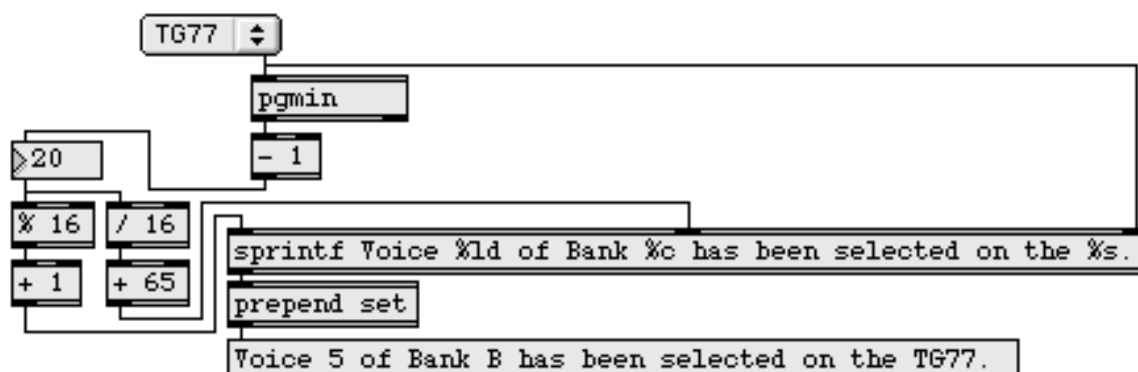
必須。アーギュメントはC言語に準拠したフォーマットで、出力するメッセージを構成する。アーギュメントは、単語や数字と、入力されるシンボル (**%s**)、整数 (**%ld**)、実数 (**%f**)、ASCII文字としてフォーマットされる整数 (**%c**) などの代入を示すアーギュメントによって構成される。インレットの数は、代入を示すアーギュメントの数によって決まり、インレットはアーギュメントの並びに対応する。

## 出力

- 任意** アーギュメントによって指定されたメッセージを、代入できるアーギュメントを置き換えてフォーマットし、出力する。



## 利用例



代入できるアーギュメントは、インレットに受け取った値によって置き換えられる

## 参照

**fromsymbol**  
**key**  
**keyup**  
**message**  
**spell**  
**tosymbol**

シンボルを個々の数値かメッセージに変換する  
コンピューター・キーボードのキー入力を出力する  
コンピューター・キーボードのキー入力を出力する  
任意のメッセージを送る  
入力をASCIIコードに変換する  
メッセージ、数値、リストを1つのシンボルに変換する

## 入力

**整数または実数** **sqrt**は、入力された値の平方根を出力する。入力が負の場合、解は実在しないので、**NaN** (Not a Number、数値ではない、の略) が出力される。

**bang** 現在格納されている数値の平方根値を出力する。

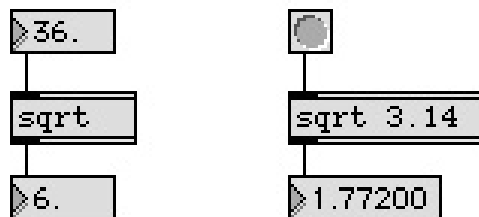
## アーギュメント

**整数または実数** オプション。アーギュメントの値の平方根が出力される。

## 出力

**実数** 入力の平方根。

## 利用例



## 参照

**expr** 数式の値を求める

注記：スタンドアロンのアプリケーションの作成は、2003年9月現在Windows版ではサポートされていない。**standalone**オブジェクトとそのインスペクター自体は、Windows版のパッケージにも含まれているが、スタンドアロン・アプリケーションの作成機能が追加されるまでWindows上では有効ではない。

**standalone**オブジェクトは、EditメニューのBuild Application/Collective...を選択してMax/MSPのパッチからスタンドアロン・アプリケーションを作成する際のオプションを指定することができる。アプリケーションが複数のサブパッチから生成される場合、**standalone**オブジェクトは最上位のパッチに1つだけ配置する必要がある。

## 入力

スタンドアロン・アプリケーションのすべてのパラメーターは、**standalone**オブジェクトのインスペクターにより設定される。

## インスペクター

**standalone**オブジェクトの動作は、インスペクターに表示され編集することができる。WindowsメニューからShow Floating Inspectorを選択しておくこと、**standalone**オブジェクトを選択した際に**standalone**インスペクターがフローティング・ウィンドウとして表示される。あるいは、オブジェクトを選択しObjectメニューからGet Info...を選択することによってもインスペクターを表示させることができる。**standalone**インスペクターでは、下記の各項目を設定することができる。

*Application Creator Code*は、ファイルタイプを表す半角英数4文字からなるコードで、使用しているコンピュータのファイルシステム上で独自のファイルタイプとしてふるまうための情報を、作成したアプリケーションに与える。

File Optionの欄では、主に初期設定情報の取り扱いについて指定することができる。

*Use Own Property List (plist) Resource*をチェックすると、作成したアプリケーションに独自の“plist”リソースを持たせて、任意のアイコンを持たせることができるようにする（Mac OS 9上のBNDLリソースに相当するものである）。

もしパッチ内のMax/MSPのオブジェクトが使用するサポートファイルがコレクティブに含まれない場合は、*Search for Files Not in the Application's Collective*をチェックするとよい。

*Utilize Search Path in Preferences File*オプションをチェックすると、デフォルトのサーチパスに代わりPreferencesファイルで指定されたサー

チパスを使用することができる。

作成するアプリケーションの初期設定を、デフォルトのMax Preferencesファイルの代わりに独自のものを使用したい場合は、*Preference File Name*のボックス内に任意のファイル名を指定することができる。

Optionsの欄では、スタンドアロン・アプリケーションの動作や外観を設定することができる。

*Status Window Visible at Startup*をチェックすると、アプリケーションの起動時にStatusウィンドウ（Maxウィンドウに相当するもの）を表示する。

*Prevent Loadbang Defeating with Cmd-Shift*をチェックすると、最上位のパッチがロードされた時にすべてのオブジェクトに対して**loadbang**メッセージが送られるのをユーザーが中止させてしまうのを防ぐことができる。

*Overdrive Enabled*および*All Windows Active Enabled*をチェックすると、Max/MSPのOptionメニューにあるOverdriveおよびAll Windows Activeに相当する動作が有効になる。

*User Can't Close top-level Patcher Windows*をチェックすると、スタンドアロン・アプリケーションのユーザーが最上位のパッチ・ウィンドウを閉じることをできなくすることができる。

## アーギュメント

なし。

## 出力

なし。

## 参照

Collectives      Grouping files to create a single application（英文PDF参照）

## 入力

**リスト** 第1インレット：2番目の数値はベロシティとして格納し、1番目の数値はMIDIノート・オン・メッセージのノート・ナンバーとして扱う。2番目の数値が0でない場合は、第2アウトレットから出力し、1番目の数値も第1アウトレットから出力する。2番目の数値が0であれば、何も出力しない。

**整数** 第1インレット：数値をノート・ナンバーの値として扱う。**stripnote**が格納しているベロシティの値が0でなければ、そのベロシティを第2アウトレットから出力し、ノート・ナンバーを第1アウトレットから出力する。

第2インレット：数値をベロシティとして格納し、第1インレットで受け取るノート・ナンバーとともに扱われることになる。

**実数** 整数に変換される。

## アーギュメント

なし。

## 出力

**整数** 第1アウトレット：ベロシティが0でなければ、第1インレットで受け取ったノート・ナンバーを出力する。

第2アウトレット：ベロシティが0でなければ、ノート・オンのベロシティの値を出力する。

## 利用例



ノート・オフ・メッセージによって繰り返されるノート・ナンバーとベロシティ0を取り除く

## 参照

**makenote**  
**sustain**

ノート・オンに続けてノート・オフ・メッセージを生成する  
ノート・オフ・メッセージを保持して、コマンドによりそれを出力する

## 入力

**シンボル** シンボルとしてのフォルダやファイルのフル・パス名。絶対パスはMax 4.1以前のバージョン(コロンを使用)を除き、スラッシュで区切られて表示される。具体的には下記のようなになる。

“C:/Max Folder/extras/mystuff/mypatch.pat”

パスの形式や種類を変更したい場合は、**conformpath**オブジェクトを使用する。

## アーギュメント

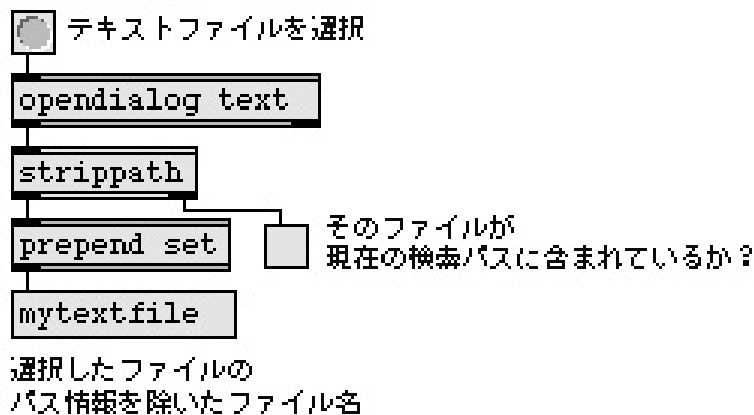
なし。

## 出力

**シンボル** 第1アウトレット：その前のすべてのパス情報を除いたファイル名。

**整数** 第2アウトレット：ファイルが現在のMaxのサーチ・パス内に見つかれば1を、それ以外の場合は0を出力する。

## 利用例



**strippath** オブジェクトはファイルのパス名からパスの情報を取り除き、ファイル名を残す

## 参照

**absolute****path**

ファイル名を絶対パスに変換する

**dropfile**

ファイルをドラッグ&ドロップする領域を設定する

**opendialog**

ファイルやフォルダを選択するためのダイアログを開く

**relative****path**

絶対パスを相対パスへ変換する

**savedialog**

保存するファイルの名前を尋ねるダイアログを開く

## 入力

- 任意** 第1インレット：入力はそのまま出力されるが、受け取ったメッセージが指定したシンボルか数値と一致する要素を含む場合は、その要素を置換用のシンボルで置き換えて、メッセージを出力する。
- 任意** 第2インレット：**substitute**オブジェクトは、第2インレットに2つの数値または2つのシンボルのメッセージを受け取る。1つ目の数値かシンボルは、入力されるメッセージの中で置き換える要素を指定する。2つ目の数値かシンボルは、一致したシンボルを置き換える数値かシンボルを指定する。
- set** 第2インレット：**set**という単語が無視される以外は、任意のメッセージと同じように処理される。

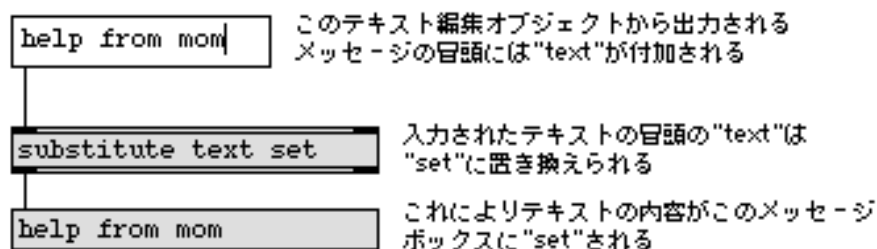
## アーギュメント

- 任意** オプション。1つ目の数値かシンボルは、入力されるメッセージの中で置き換える要素を指定する。アーギュメントが設定されていなければ、その**substitute**オブジェクトは何もしない。
- 任意** オプション。2つ目の数値かシンボルは、一致したシンボルを置き換える数値かシンボルを指定する。

## 出力

- 任意** 第1アウトレット：指定した数値かシンボルと一致する要素を、置換用の数値かシンボルに置換えた上で、入力されたメッセージを出力する。
- bang** 第2アウトレット：入力されたメッセージを出力する時に、置き換えがなかった場合に、**bang**を出力する。

## 利用例



**substitute**オブジェクトは、あるオブジェクトから出力されるメッセージを、他のオブジェクトが必要とするメッセージに変換することができる



# substitute

メッセージ内のシンボルを  
他のシンボルに置き換える

---

## 参照

|                |                        |
|----------------|------------------------|
| <b>route</b>   | 選択的に入力を実定のアウトレットから出力する |
| <b>sprintf</b> | 単語や数値のメッセージをフォーマットする   |
| <b>zl</b>      | 多目的なリスト処理              |

# suspend

アプリケーションが一時停止されたか、再開された時に出力する

## 入力

なし。

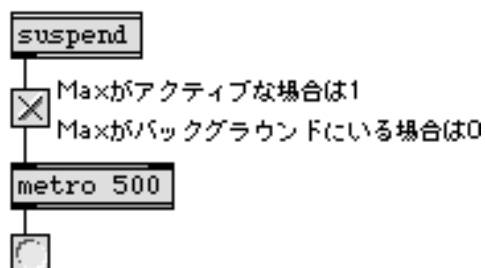
## アーギュメント

なし。

## 出力

**整数** 第1アウトレット：アプリケーションが一時停止された時（バック・グラウンドになった時）に、**0**を出力する。アプリケーションが再開された時（最前面に戻った時）には、**1**を出力する。

## 利用例



**suspend** オブジェクトは、Maxが最前面であるか否かによって、処理を開始または停止させる

## 参照

### active gestalt

パッチ・ウィンドウがアクティブであれば1を、そうでなければ0を出力  
現在のシステムについて調べる

## 入力

第1インレット：2番目の数値をベロシティとして格納し、1番目の数値をMIDIノート・オン・メッセージのノート・ナンバーとして扱う。それらがノート・オン（ベロシティが0でない）場合、ベロシティを第2アウトレットから出力し、ノート・ナンバーを第1アウトレットから出力する。ノート・オフ（ベロシティが0のノート・オン）は、そのまま通過するか、**sustain**が保持する。

**整数** 第1インレット：数値はノート・ナンバーとベロシティの組み合わせにおけるノート・ナンバーの数値である。**sustain**が現在保持しているベロシティの数値が0でない場合は、すぐにその組み合わせを出力する。ベロシティが0である場合は、**sustain**が有効であるか無効であるかによって、ノート・オフを出力するか、保持するか、いずれかの動作になる。

第2インレット：数値をベロシティとして格納し、第1インレットで受け取るノート・ナンバーと組み合わせられることになる。

第3インレット：数値が0でない場合は、**sustain**が有効になり、すべてのノート・オフを保持する。数値が0である場合は、**sustain**は無効になり、すべてのノート・オフをすぐに出力する。

**実数** 整数に変換される。

## アーギュメント

なし。

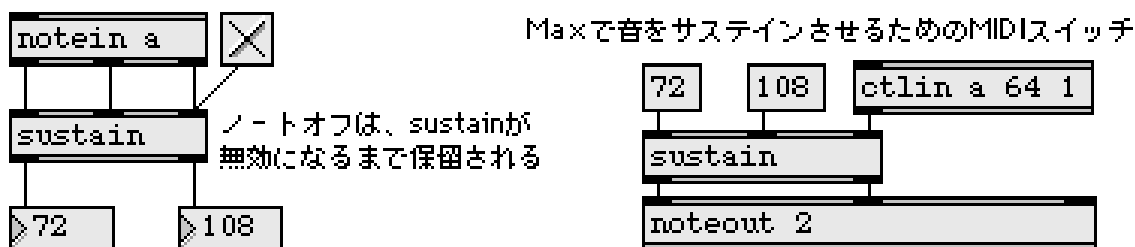
## 出力

**整数** 第1アウトレット：ノート・ナンバーとベロシティの組み合わせにおけるノート・ナンバーの値。

第2アウトレット：ノート・ナンバーとベロシティの組み合わせにおけるベロシティの値。

ノート・オンは、常にすぐに出力する。**sustain**が有効であれば、**sustain**が無効になるまで、ノート・オフを保持する。**sustain**が無効であれば、ノート・オフをすぐに出力する。

## 利用例



ピアノのサステイン・ペダルのように、**sustain**はすべての保持しているノートを一度に離す

## 参照

**flush**  
**makenote**  
**stripnote**

保持しているノートについてノート・オフを作る  
ノート・オンに続けてノート・オフ・メッセージを生成する  
ノート・オフ・メッセージを取り除き、ノート・オン・メッセージのみを出す

## 入力

- 整数** 第1インレット：この数値は第2アウトレットから出力され、第2インレットに受け取った数値が第1アウトレットから出力される。
- 第2インレット：第1インレットに数値を受け取った時に、第1アウトレットから出力するために、この数値は格納される。
- 実数** アーギュメントとして実数が指定されていない場合は、この数値は整数に変換される。実数が指定されている場合は、第2インレットに受け取った数値は、実数として格納される。
- リスト** 第1インレット：これらの数値は、**swap**オブジェクトに格納される。1つ目の値は第2アウトレットから出力され、2つ目の数値は第1アウトレットから出力される。
- bang** 第1インレット：**swap**オブジェクトに現在格納されている数値を入れ替え、出力する。

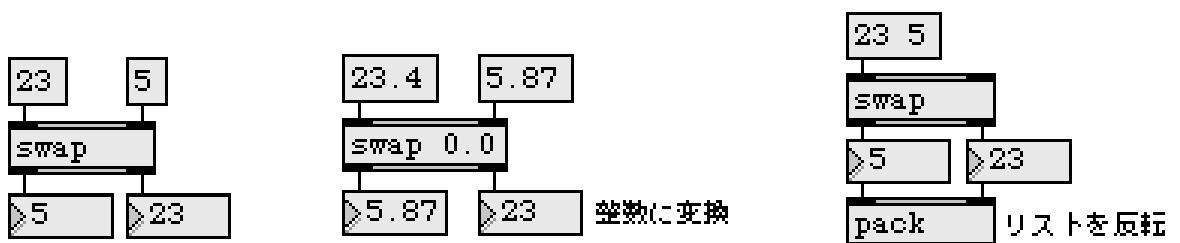
## アーギュメント

- 整数または実数** オプション。第1アウトレットから出力する数値の初期値を設定する。実数のアーギュメントは第1アウトレットから実数を出力する（第2アウトレットから出力する数値は、常に整数である）。アーギュメントがなければ、初期値は0となる。

## 出力

- 整数** 第1インレットに数値を受け取った時に、それぞれのインレットに受け取った数値を、逆のアウトレットから出力する。
- 実数** 実数のアーギュメントがあれば、第1アウトレットから出力する数値は実数となる。

## 利用例



受け取った順番と逆に、数値を出力する

## 参照

**buddy**

入力されたデータを同期させ、それらを同時に出力する

**fswap**

左右のインレットに受け取った2つの実数を交換して出力する

**pack**

数値やシンボルをひとつのリストに結合させる

**unpack**

リストを個々のメッセージに分割する

Tutorial 30

Number groups (英文PDF参照)



**swatch**オブジェクトの2次元のカラー・スペースは、横軸に色相 (hue)、縦軸に明度 (lightness) を表す。3番目の色の次元である彩度 (saturation) は、**saturation**メッセージによって設定する。

## 入力

**整数** 第1インレット：0から255までの数値は、赤色の成分を設定し、出力を行う。

第2インレット：0から255までの数値は、緑色の成分を設定し、出力を行う。

第3インレット：0から255までの数値は、青色の成分を設定し、出力を行う。

注記：ほとんどの**Max**オブジェクトとは異なり、3つのインレットのいずれに入力しても、**swatch**は現時点の色彩の位置を計算し直し、出力を行う。

**実数** 整数に変換される。

**(マウス)** **swatch**をクリックまたはドラッグすれば、2次元のカラー・スペースにおいて指定された位置 (x, y) と彩度の値に従って、RGBカラーを計算し、出力を行う。

**bang** 2次元のカラー・スペースにおいて指定された位置 (x, y) と彩度の値に従って、現在のRGBカラーを出力する。

**リスト** 0から255までの3つの数値のリストによって、3つのRGBカラーの成分 (赤、緑、青) を設定し、表示を更新して、出力を行う。

**set** **set**に続く0から255までの3つの数値のリストによって、3つのRGBカラーの成分 (赤、緑、青) を設定し、表示を更新するが、出力は行わない。

**saturation** **saturation**に続く0から255までの数値によって、表示している2次元のカラー・スペース (色相と明度) に対する彩度を設定する。この時、指定されている位置 (x, y) によって新しいRGBカラーを計算し、出力を行う

**(プリセット)** **preset**オブジェクトを用いれば、**swatch**オブジェクトのRGBカラーを保存し、再現することができる。





## アーギュメント

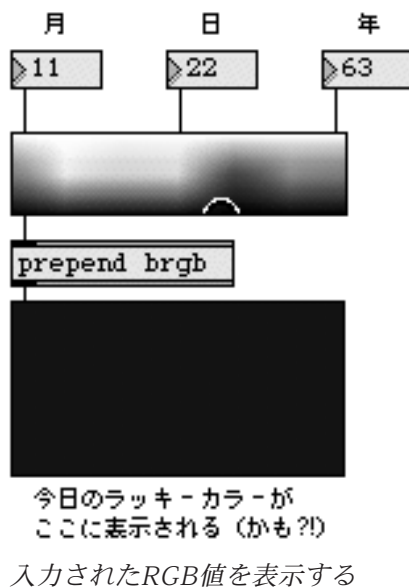
なし

## 出力

**リスト** 第1アウトレット：3つのRGB（赤、緑、青）カラーの数値から成るリストを出力する。

**整数** 第2アウトレット：現在の彩度の値を出力する。これはRGBのリストの受け取った時に計算されるか、あるいは、**saturation**メッセージを受け取った時に出力される。

## 利用例



## 参照

### colorpicker panel

モーダル・ダイアログによって色を選択する  
色のついた背景領域

## 入力

- 整数** 第1インレット：使用するインレットを指定する値。指定されたインレットに入力されたメッセージが、アウトレットから出力される。指定以外のすべてのインレットは閉じられ、メッセージを通過させない。0を指定した場合は、すべてのインレットが閉じられる。
- 任意のメッセージ** 任意のインレット：指定されたインレットに受け取ったメッセージは、アウトレットから出力される。閉じられたインレットに受け取ったメッセージは無視される。
- 実数** 第1インレット：整数に変換される。
- bang** 第1インレット：開いているインレットの番号を出力する。すべてのインレットが閉じられている場合は、**0**を出力する。

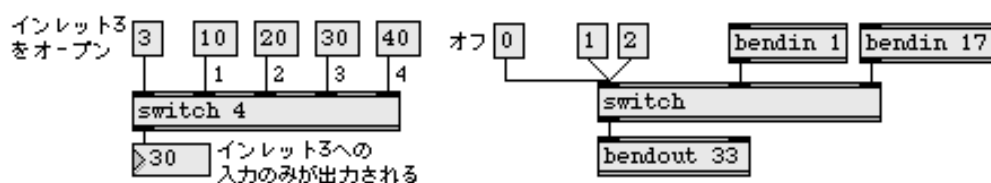
## アーギュメント

- 整数** オプション。インレットの数（100以下）を指定する。指定したインレット数に、第1インレットが付け加えられる。アーギュメントを指定しない場合は、第1インレットの他に2つのインレットが作られる。

## 出力

- 任意のメッセージ** 第1インレットの値が0以下の場合、絶対値として使用するインレットを指定する（-1の場合はインレット1を使用、-2の場合はインレット2を使用する、など）。絶対値が存在しているインレットの数よりも大きい値の場合は、最も右のインレットに受け取った値を出力する。

## 利用例



1度に1つのインレットからメッセージを受け取る、または、すべてのインレットからの入力を無視する

## 参照

- forward** 様々なオブジェクトにパッチ・コードなしでメッセージを送信する
- funnel** 受け取ったインレットの番号を付け加えて出力する
- gate** 入力を特定のアウトレットから出力する
- Ggate** 2つのうち1つのアウトレットから数値を出力する
- Gswitch** 2つのうち1つのインレットから数値を受け取る
- receive** パッチ・コードなしでメッセージを受信する
- send** パッチ・コードなしでメッセージを送信する

## 入力

**整数** 第1インレット：オブジェクト・ボックスでの**\$i1**アークギュメントを数値で置き換え、アークギュメントのリスト全体を処理し、一つずつアウトレットから出力する。

他のインレット：オブジェクト・ボックスの対応する**\$i1**アークギュメントを置き換える数値として用いられるが、第1インレットに数値を受け取るまで、数値を保持する。

**bang** 第1インレット：最も最近に受け取った数値を用いて、フォーマットしたメッセージのバイト列を出力する。

## アークギュメント

**リスト** 必須。このアークギュメントは、MIDIシステム・エクスクルーシブ・メッセージの個々のバイトの数値を表す数値のリストである。1番目の数値はシステム・エクスクルーシブのステータス・バイトである**240**（または**0xF0**）であり、最後の数値はエンド・バイトである**247**（または**0xF7**）でなければならない。これらの間には、任意の数のデータ・バイトを入れることができる。

データ・バイトのアークギュメントは、**expr**や**if**オブジェクトでの表記と同じような数式を用いることができる。この式は、アウトレットから出力する前に評価される。数式は**\$i**のすぐ後にインレット番号を続ける形式（例えば、**\$i2**）として、置き換え可能であるアークギュメントを含めても構わない。置き換え可能なアークギュメントは、指定したインレットに受け取った数値によって置き換えられる。数値を置き換える数式は、**is**という単語を先に付け、数式の両側にスラッシュ (/) で他のアークギュメントと区切る必要がある（利用例を参照のこと）。

数式を評価した結果の値が0より小さい場合は、その数式による数値を出力しない。これによって、長さが変化するシステム・エクスクルーシブ・メッセージを送信することができる。

## 出力

**整数** 第1インレットに数値を受け取った時、アークギュメントのすべての数式を評価し、アークギュメントのリストの数値を連続的に出力する。これらは、MIDIシステム・エクスクルーシブ・メッセージのバイト列として、**midout**オブジェクトを使って送信することができる。

## 利用例



**sxformat** は、完全なMIDIシステム・エクスクルーシブ・メッセージを、**midiout**へ1バイトずつ送ることができる

## 参照

- expr** 数式の値を求める
- midiout** 生のMIDIデータを送信する
- sysexin** MIDIシステム・エクスクルーシブ・メッセージを受信する

Tutorial 34 Managing raw MIDI data (英文PDF参照)

## 入力

- (MIDI) **sysexin** オブジェクトは、MIDI入力デバイスからMIDIシステム・エクスクルーシブ・メッセージを受信する。
- enable** **enable 0**のメッセージはオブジェクトを使用不能にし、受信するMIDIデータを無視するようになる。**pcontrol** オブジェクトやMIDIEnable/Disableボタンによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**メッセージは、オブジェクトを再び受信可能にする。
- port** **port**の後に**a**から**z**の文字を続けるか、MIDI入力デバイスの名前を続けることによって、システム・エクスクルーシブ・メッセージを受信するポートを設定する。**port**という単語はオプションで、省略することもできる。
- (マウス) **sysexin** オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

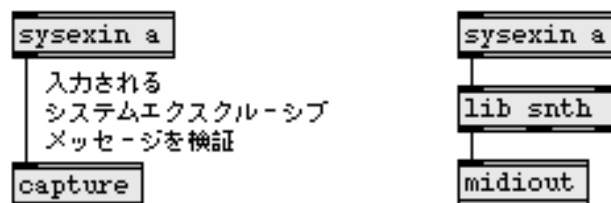
## アーギュメント

- a~z** オプション。MIDIシステム・エクスクルーシブ・メッセージを受信するポートを指定する。アーギュメントを指定しない場合は、ポートa (または**MIDI Setup**・ダイアログのリストの最上位にあるポート) からMIDIメッセージを受信する。

## 出力

- 整数** 受信したMIDIシステム・エクスクルーシブ・メッセージを、バイトごとに出力する。

## 利用例



MIDIシステム・エクスクルーシブ・メッセージは **lib** オブジェクトによって解釈することができる。

## 参照

- midin** 生のMIDIデータを受信する
- sxformat** MIDIシステム・エクスクルーシブ・メッセージを用意する

## 入力

- リスト** 第1インレット：リストの2番目の数値は**table**オブジェクトに格納されるが、その場所は1番目の数値によって指定されるアドレス（インデックス）となる。
- 整数** 第1インレット：**table**オブジェクト内のアドレスを指定する。アドレスに格納されている数値は第1アウトレットから出力される。しかし、第2インレットに数値が入力される場合は、**table**オブジェクトは指定されているアドレスへ数値を格納するが、出力はしない。
- 第2インレット：**table**オブジェクト内に格納する数値を指定する。第1インレットに入力された数値の次のアドレスに数値は格納される。
- 実数** 整数に変換される。
- bang** 第1インレット：**quantile**メッセージと同じ働きをする。その場合、0から32,768までのランダムな数値のアーギュメントを伴うものとして扱われる。
- cancel** 第1インレット：第2インレットに入力された数値を忘れる。それによって第1インレットに次のアドレスを示す数値が入力されると、そのアドレスに入力された数値ではなく、もともと格納されていた数値を出力する。
- clear** 第1インレット：**table**オブジェクト内のすべての数値をすべて0に設定する。
- const** 第1インレット：数値を伴った**const**メッセージは、**table**オブジェクト内のすべてのアドレスへ、その数値を格納する。
- dump** 第1インレット：**table**オブジェクト内に格納されているアドレス0からのすべての数値を、瞬時に第1アウトレットから出力する。
- flags** 第1インレット：**table**オブジェクトのインスペクター（下記参照）に見られる保存オプションを切り替える。**flags**メッセージは2つのアーギュメントを伴う。1つ目のアーギュメントは*Save with patcher*オプションに関連し、2つ目のアーギュメントは*Don't Save*オプションに関連する。アーギュメントが0以外の時オプションはチェックされ、アーギュメントが0ならばオプションはチェックされない。例えば、**flags 1 1**というメッセージならば**table**オブジェクトの内容はパッチの一部として保存され、Maxは**table**内の内容に変化が加えられても保存するかどうかを確認しない。










- fquantile** 第1インレット：0から1の間の数を伴った**fquantile**メッセージは、**table**オブジェクト内に格納されているすべての数値の合計とその数値とを掛け算する。そして、**table**オブジェクトは格納されているすべての数値を順に加算し、先ほど求めた結果より大きいかまたは等しくなった時点でそのアドレスを出力する。
- getbits** 第1インレット：**table**オブジェクト内に格納された1つまたはそれ以上の数値のビットを得て、第1アウトレットから出力する。**getbits**は3つのアーギュメントを持つ。最初のアーギュメントはアドレスを示し、2番目のアーギュメントはそのアドレスに格納されている数値の最初のビット位置（ビット位置は最も小さいビットから大きなビットまでを0から31の数で示す）を示し、3番目のアーギュメントは最初のビット位置からどのくらいビットを出力するかを示す。これによって特定されたビットはアウトレットから10進法の整数として出力される。
- 例えば、**getbits 61 4 3**というメッセージは**table**オブジェクトのアドレスで61を探し、ビット位置を4から始め（右から5番目のビット）、そしてその位置から3ビット分に対応する10進法の数値を出力する。このアドレスが61の場所に87という数値が格納されているとする。87はバイナリー形式で1010111である。ビット位置4から始めて3ビットをとると101であり、このバイナリー形式は10進法で5にあたるので、アウトレットからは**5**が出力される。
- goto** 第1インレット：数値を伴う**goto**メッセージはポインターを数値に対応するアドレスに設定する。ポインターは通常**table**オブジェクトの最初のアドレスに設定されている。
- inv** 第1インレット：数値を伴う**inv**メッセージは、その数値より大きいかまたは同じ数値を探し、最初のアドレスを第1アウトレットから出力する。
- length** 第1インレット：第1アウトレットから**table**オブジェクトの長さ（サイズ）を出力する。
- load** 第1インレット：**table**オブジェクトを**load**モードにする。**load**モードの時は、第1インレットに入力されるすべての数値を**table**オブジェクト内に格納する。これはアドレス0から始まって**table**オブジェクトが満たされるまで（あるいは**normal**メッセージが送られ、**load**モードから抜け出すまで）続けられる。**table**オブジェクトの容量より多くの数値が入力されるとすると、超過した数値は無視される。
- max** **table**オブジェクト内に格納されている数値の最大値を、第1アウトレットから出力する。



- min** **table**オブジェクト内に格納されている数値の最小値を、第1アウトレットから出力する。
- next** 第1インレット：**goto**メッセージで第1アウトレットから出力されるポインタのアドレスと、そこに格納されている数値を出力し、ポインタを次のアドレスに設定する。ポインタが**table**オブジェクトの最後のアドレスにあれば、ポインタは最初のアドレスに移動する。
- normal** 第1インレット：**load**メッセージを無効にする。つまり、**table**オブジェクトは**load**モードを抜けて通常の状態に戻る。
- open** 第1インレット：**table**オブジェクトのグラフィック・エディター・ウィンドウを開き、それを最前面に表示する。**table**オブジェクトをダブル・クリックしても同じ効果が得られる。
- prev** 第1インレット：**next**メッセージと同じ出力をするが、ポインタは進むのではなく減少してゆく。ポインタが**table**オブジェクトの最初のアドレスにあれば、ポインタは最後のアドレスに移動する。
- quantile** 第1インレット：数値を伴った**quantile**メッセージは、**table**オブジェクト内に格納されたすべての数値の合計とその数値を掛ける。その結果を $2^{15}$  (32,768) で割る。そして、**table**オブジェクトは格納されているすべての数値を順に加算し、先ほど求めた結果より大きいかまたは等しくなった時点でそのアドレスを出力する。
- read** 第1インレット：名前を伴った**read**メッセージはテキストかMaxバイナリー形式のファイルを開き、データを読み込む。アーギュメントがない場合、標準ファイル・ダイアログを開いて、どのファイルを開くかを選ぶことになる。そのファイルが有効なデータを持っていた場合、**table**オブジェクト内のすべての情報はそのデータに置き換えられる。
- refer** 第1インレット：他の**table**オブジェクトの名前を伴った**refer**メッセージは、その名前の**table**オブジェクトからデータを読み込むように設定する。
- send** **receive**オブジェクトの名前とアドレスの数値を伴った**send**メッセージは、そのアドレスに格納されている数値を、名前の一致するすべての**receive**オブジェクトに対して、**table**オブジェクトのアウトレットを使わずに送信する。
- set** 第1インレット：数値のリストを伴った**set**メッセージは、指定されたアドレスへ数値を格納する。**set**メッセージの後にくる最初の数値はアドレスを指定する。その次の数値は指定されたアドレスへ格納する数値であり、その後の数値は順次、次のアドレスへ格納される。

- setbits** 第1インレット：**table**オブジェクトに格納されている数値の1つあるいはそれ以上のビットを変化させる。**setbits**は4つのアーギュメントを持つ。最初のアーギュメントはアドレスを示し、2番目のアーギュメントはそのアドレスに格納されている数値の最初のビット位置（ビット位置は最も小さいビットから大きなビットまでを0から31の数で示す）を示し、3番目のアーギュメントは変更するビットを、指定されたビット位置から何ビットにするかを示し、そして4番目のアーギュメントはそれらのビットを変更する数値（10または16進法で表される）を示す。
- 例えば、**setbits 47 5 3 6**というメッセージは**table**オブジェクトのアドレスで47を探し、ビット位置を5から始め（右から6番目のビット）、そしてその位置から3ビット分にあたる範囲を110（10進法の整数で6に相当するバイナリー表記）に置き換える。アドレスが47の場所に87という数値が格納されているとする。87はバイナリー形式で1010111であり、ビット位置5から始めて3ビット分を110に置き換えると、数値は1110111に変わる。このバイナリー形式は10進法の整数で119にあたる。こうしてアドレス47に新しく格納される数値は119となる。
- size** 第1インレット：数値を伴った**size**メッセージは、**table**オブジェクトをその数値のサイズに設定する。
- sum** 第1インレット：第1アウトレットから**table**オブジェクト内に格納されたすべての数値の総和を出力する。
- wclose** 第1インレット：**table**オブジェクトに関連付けられているグラフィック・エディター・ウィンドウを閉じる。
- write** 第1インレット：**table**オブジェクトのデータを保存するために、標準ファイル保存ダイアログを開く。ファイルはテキスト形式かMaxバイナリー形式で保存できる。
- (マウス) **table**オブジェクト内に格納される数値は、マウスを使ってグラフィカルに入力や編集ができる。パッチ内に**table**オブジェクトが作られると、**table**オブジェクトのグラフィック・エディター・ウィンドウが開かれ、マウスをその中でドラッグすることで数値を入力できる。このグラフィック・エディター・ウィンドウは編集ツールのパレットを持っている。
- パッチ・ウィンドウがロックされている時、**table**オブジェクトをマウスでダブル・クリックすることで、グラフィック・エディター・ウィンドウを開くことができる。

|                                                                                   |                       |
|-----------------------------------------------------------------------------------|-----------------------|
|  | 正確に作画するための十字カーソルを表示   |
|  | カット、コピー、消去、ペーストの範囲を選択 |
|  | 画面をスクロールさせて他の場所を表示させる |
|  | 画面上でマウスを使用して作画        |
|  | 画面上で始点と終点をクリックしてして作画  |
|  | 画面を横方向に拡大縮小           |
|  | 画面を縦方向に拡大縮小           |

**table**データは新しいTableウィンドウを開き、FileメニューからSaveを選ぶことで、独立したファイルを作ることができる。また、新しいTextファイルを開き、**table**という言葉から始まる独立したファイルからも**table**データを作ることができる。この**table**という言葉の後には、空白文字で区切られた数値が続くが、これは**table**オブジェクト内に格納される数値を示す。

独立したファイルとして保存された**table**データは、Fileメニューの**Open as Text...**を選ぶことでテキスト・ファイルとして表示や編集ができる。このテキスト形式のファイル内にある数値に、**capture**オブジェクトのエディター・ウィンドウや、または別のワープロ・ソフトで書かれたものをペーストすることができる。また、**capture**オブジェクトからのテキストは、**table**オブジェクトのグラフィック・エディター・ウィンドウへ直接ペーストすることができる。

## インスペクター

**table**オブジェクトの振舞はインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**table**オブジェクトを選択することで、**panel**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**table**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

*Table Size*は**table**オブジェクトに格納される数値の数を決定する。**table**オブジェクトが新たに作られた時、128個の数値が割り当てられている。これは0から127で示される。

*Table Range*はエディター・ウィンドウに表示されるy軸の範囲を決定する。**table**オブジェクトが新たに作られた時、128が割り当てられていて、これは0から127で示される。

*Save Table with Patcher*がチェックされると、**table**オブジェクト内の数値はパッチの一部として保存される。そうでなければ、**table**オブジェクトの数値は独立したファイルに保存されることになる。

*Don't Save*がチェックされると、**table**オブジェクトを持つパッチが閉じられる時に、Maxは**table**の変更を保存するかどうか、確認しないようになる。

*Use Note Name Legend*がチェックされると、y軸の数値を数値の代わりにMIDIノート・ネームとして表示する。

*Signed Values*がチェックされると、**table**オブジェクトは正の数と同様に負の数も表示する。この時、0から上下方向へ*Range*の値が割り当てられ、表示される数値の範囲が決定される。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

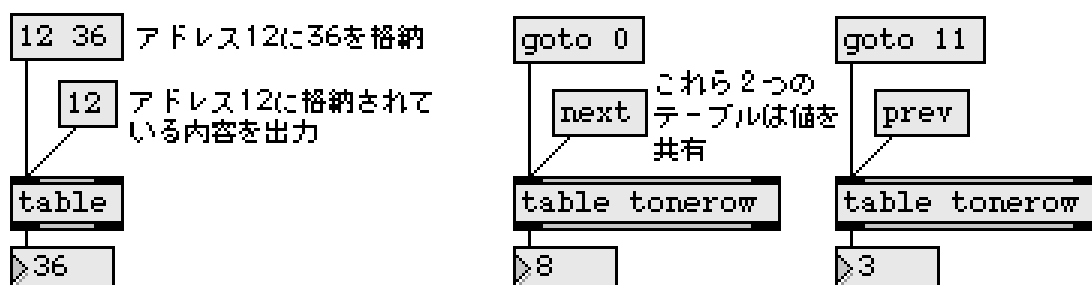
任意のシンボル オプション。**table**オブジェクトに名前を付ける。Maxは同じ名前の独立したファイルを探す。2つまたはそれ以上の**table**オブジェクトが同じ名前を共有していれば、それらはまた同じ数値を共有する。Maxがファイルを見つけられなかった時も同様である。

## 出力

**整数** **table**オブジェクトのすべての数値が第1アウトレットから出力される。

**bang** グラフィック・エディター・ウィンドウで**table**オブジェクトの内容が変更されると、**bang**が第2アウトレットから出力される。

## 利用例



任意のサイズと範囲の配列を格納し、呼び出し、修正ができる

## 参照

|                    |                           |
|--------------------|---------------------------|
| <b>capture</b>     | 数値を格納し、表示・編集する            |
| <b>coll</b>        | 様々な種類のメッセージを格納し、編集する      |
| <b>funbuff</b>     | $x$ と $y$ の数値のペアを対にして格納する |
| <b>Histo</b>       | 受け取った数値のヒストグラムを生成する       |
| <b>multiSlider</b> | 複数のスライダーとスクロール表示          |
| <b>Text</b>        | 数値をテキスト・ファイルとしてフォーマットする   |

Tutorial 32

The table object (英文PDF参照)

## 入力

**実数または整数** タンジェント関数への入力。

**bang** 第1インレット：現在格納されている数値のタンジェント値を計算する。アーギュメントがない場合は、**tan**への初期値は0である。

## アーギュメント

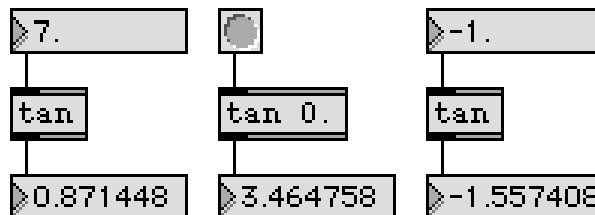
**実数または整数** オプション。タンジェント関数への初期値を設定する。

## 出力

**整数または実数** 入力された値のタンジェント値。

## 利用例

・実数による入力



・入力値に対するタンジェント関数

## 参照

|              |                    |
|--------------|--------------------|
| <b>acos</b>  | アーク・コサイン関数         |
| <b>asin</b>  | アーク・サイン関数          |
| <b>atan</b>  | アーク・タンジェント関数       |
| <b>atan2</b> | アーク・タンジェント関数 (2変数) |
| <b>cos</b>   | コサイン関数             |
| <b>cosh</b>  | 双曲線コサイン関数          |
| <b>sin</b>   | サイン関数              |
| <b>sinh</b>  | 双曲線サイン関数           |
| <b>tanh</b>  | 双曲線タンジェント関数        |

## 入力

**実数または整数** 双曲線タンジェント関数への入力。

**bang** 第1インレット：現在格納されている数値の双曲線タンジェント値を計算する。アーギュメントがない場合は、**tanh**への初期値は0である。

## アーギュメント

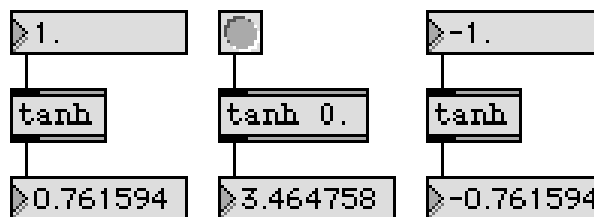
**実数または整数** オプション。双曲線タンジェント関数への初期値を設定する。

## 出力

**整数または実数** 入力された値の双曲線タンジェント値。

## 利用例

・実数による入力



・入力値に対する双曲線タンジェント関数

## 参照

|              |                    |
|--------------|--------------------|
| <b>acos</b>  | アーク・コサイン関数         |
| <b>asin</b>  | アーク・サイン関数          |
| <b>atan</b>  | アーク・タンジェント関数       |
| <b>atan2</b> | アーク・タンジェント関数 (2変数) |
| <b>cos</b>   | コサイン関数             |
| <b>cosh</b>  | 双曲線コサイン関数          |
| <b>sin</b>   | サイン関数              |
| <b>sinh</b>  | 双曲線サイン関数           |



## 入力

**bang** 第1インレット：**tempo**オブジェクトを始動させる。**tempo**オブジェクトが動作中であれば、再始動させる。

**stop** 第2インレット：**tempo**オブジェクトを止める

**整数** 第1インレット：数値が0でない場合は、**bang**と同じ動作をする。0の場合は**stop**と同じ動作をする。

**整数または実数** 第2インレット：数値は、1分ごとの拍数（BPM、1分ごとの4分音符の数）のテンポとして格納される。テンポは1分間に5から300までに制限される。

第3インレット：数値は1拍の時間を決める乗数である。乗数によって、テンポは遅くなる。例えば、乗数が2であれば、**tempo**オブジェクトの出力の早さは1/2になる。

第4インレット：数値は全音符の分割数を示し、**tempo**オブジェクトが出力するリズム値になる。例えば、この数値が8であれば、指定されたテンポに対して、8分音符の間隔で出力が行われる。アウトレットからは、0から全音符の分割数より1小さい数までの数値が、繰り返して出力される。この全音符の分割数は1から96まででなければならない。

**tempo** 第1インレット：**tempo**メッセージに続く実数によって、現在のテンポを設定する。

**clock** 既存の**setclock**オブジェクトの名前が続く**clock**メッセージは、Maxの内部クロックでなく、指定された**setclock**によって、**tempo**オブジェクトを制御するように設定する。**clock**メッセージだけであれば、Maxの内部クロックによって、**tempo**オブジェクトを制御するように設定する。

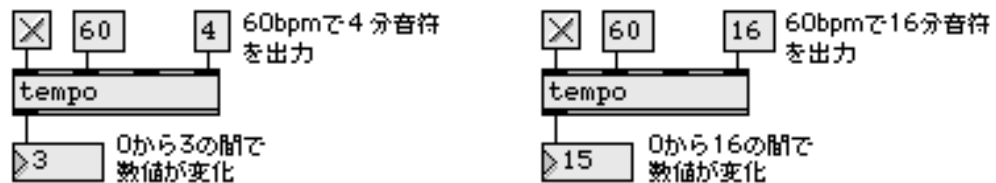
## アーギュメント

**整数または実数** オプション。第1アーギュメントでは、5から300までの範囲で、テンポ（BPM）の初期値を設定する。アーギュメントを指定しない場合は、初期値は120BPMである。第2アーギュメントには、1拍の時間を決める乗数を指定する。初期値は1である。第3アーギュメントでは、全音符の分割数の初期値を指定する。数値は全音符（1）から64分音符の3連符（96）までの範囲である。アーギュメントを指定しない場合は、初期値は16である。

## 出力

**整数** **tempo** オブジェクトが始動すれば、0から全音符の分割数より1小さい数までの数値が繰り返し出力される。出力される速度は、テンポの値（1分ごとの4分音符の数）と全音符の分割数によって決まる。

## 利用例



テンポ (60) は1分ごとの4分音符数を指定し、分割数は出力される拍数を指定する

## 参照

**clocker**

一定の間隔で経過時間を出力する

**metro**

一定の時間間隔で**bang**メッセージを出力する

**setclock**

タイミング関連のオブジェクトのクロック速度をリモート・コントロールする

**timein**

外部ソースのタイム・コードを出力する

Tutorial 31

Using timers (英文PDF参照)

## 入力

- clear** **Text**オブジェクトの内容を消去する。
- cr** 新しい行にするために、**Text**オブジェクトの内容の最後に改行を入れる。**Text**オブジェクトの最後が空白文字であれば、空白文字を取り除いて、改行する。
- line** **line**メッセージに続く数値によって、その行番号にあたる**Text**オブジェクトの内容を、**set**というメッセージを最初に付けて、出力する（最大256文字）。行番号は1から始まる。1未満の数値は1に変換され、存在しない行番号が指定された場合は、何も出力されない。
- open** オブジェクトの内容を編集するためにテキスト・ウィンドウを開く。パッチがロックされた状態で、オブジェクトをダブルクリックしても、同じ動作になる。テキスト・ウィンドウが開いている状態では、そのテキストを変えるメッセージを入力しても、無視される。**capture**オブジェクトとは違い、テキスト・ウィンドウの内容を変えれば、**Text**オブジェクトの内容も変更される。
- read** ファイル名を表すシンボルが続く**read**メッセージは、そのテキスト・ファイルの内容を、最大32,000文字まで**Text**オブジェクトに読み込む。ファイル名もパス名も指定しない場合は、**read**メッセージによって標準ファイル・ダイアログが開き、読み込むテキスト・ファイルを指定できる。
- settitle** **settitle**に続く任意の単語によって、テキスト・ウィンドウの名前を設定する。1単語以上をデフォルトのタイトルとして指定したい場合は、シングル・スマート・クォート（‘はoption-[、’はshift-option-[として入力する）によって単語を囲み、空白文字の前にバック・スラッシュ (\) を挿入する。
- symbol** **symbol**メッセージに続く任意の単語によって、その単語を**Text**オブジェクトの内容の最後に格納する。これは特定のメッセージを単語として**Text**オブジェクトに格納するために役立つ。例えば、**symbol clear**と入力すると、**Text**オブジェクトの内容は消去されず、**clear**という単語が格納されることになる。
- tab** **tab**メッセージは**Text**オブジェクトの内容の最後にタブ文字を入れる。最後の文字が空白文字であれば、空白文字を取り除いて、タブ文字を入れる。
- wclose** テキスト・ウィンドウを閉じる。
- write** ファイル名を表すシンボルが続く**write**メッセージは、**Text**オブジェクトの内容をテキスト・ファイルに保存する。フル・パス名を指定しな

ければ、テキスト・ファイルは現在のデフォルト・ディレクトリに保存される。ファイル名もパス名も指定しない場合は、標準ファイル保存ダイアログが開き、**Text**オブジェクトの内容をテキスト・ファイルとして保存することができる。

## その他任意のメッセージ

メッセージは**Text**オブジェクトに格納される。メッセージは以前に格納されたメッセージの後に置かれ、その後に空白文字が格納される。

- (マウス) パッチ・ウィンドウがロックされた状態で、**Text**オブジェクトをダブルクリックすれば、テキスト・ウィンドウが開き、内容の確認や編集ができる。テキスト・ウィンドウが開いている状態では、そのテキストを変えるメッセージを入力しても、無視される。**capture**オブジェクトとは違い、テキスト・ウィンドウの内容を変えれば、**Text**オブジェクトの内容も変更される。

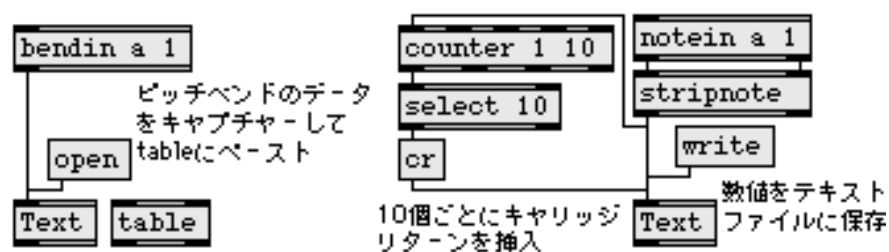
## アーギュメント

- シンボル オブジェクトが読み込まれた時に、同時に読み込むテキスト・ファイルの名前を指定する。

## 出力

- set** **line**メッセージを受け取れば、指定された行番号の内容が、**set**メッセージを先に付けて、出力される。出力される内容は、そのままメッセージ・ボックスを設定するように利用できる（または、**set**メッセージが利用できる、他のオブジェクトに入力することができる）。

## 利用例



他へペーストしたり、テキスト・ファイルに保存するために、テキストとしてメッセージを格納する

## 参照

**capture**  
**filein**  
**spell**  
**sprintf**  
**table**  
**textedit**

数値を格納し、表示・編集する  
ファイルからバイナリー・データを読み出す  
入力をASCIIコードに変換する  
単語や数値のメッセージをフォーマットする  
数値の配列をグラフィカルに編集し、格納する  
パッチ内でユーザ入力できるテキスト

## 入力

- (キーボード) **textedit**オブジェクトがハイライト状態の時に、テキスト表示領域に直接キーボードから入力することができ、その内容を変更できる。**textedit**オブジェクトがread-onlyモード (**readonly**メッセージを参照のこと) に設定されている場合は、キーボード入力はできない。入力された文字のアスキー値は、第2アウトレットから出力される。
- (マウス) パッチ・ウィンドウがロックされている状態で、**textedit**オブジェクトをマウスでクリックすれば、**textedit**オブジェクトの文字か単語が第3アウトレットから出力される。これはclickモードによって動作が異なる。(clickmodeメッセージを参照のこと)
- bang** キーボード入力や格納された**textedit**オブジェクトのバッファの内容を出力する。
- append** **append**に続くメッセージによって、**textedit**オブジェクトのバッファにそのメッセージを追加する。この時、出力は行われない。
- autoscroll** **autoscroll**メッセージに続く0または1の数値で、テキスト表示領域の自動スクロールを切り替える。**autoscroll 1**メッセージは、行数が1であり、ワード・ラップ (**wordwrap**メッセージを参照のこと) が無効であるテキスト表示領域に表示されるテキストを、カーソル・キーやマウスの操作でスクロールできるようにする。デフォルト値は0である (自動スクロールしない)。
- clear** **textedit**オブジェクトのバッファの内容を消去する。
- clickmode** **clickmode**メッセージに続く0か1の数値によって、**textedit**オブジェクトがテキスト表示領域をクリックした時の動作を指定する。**clickmode 0**メッセージは、テキスト表示領域をクリックした位置にある文字のアスキー値を、第3アウトレットから出力するように設定する。**clickmode 1**メッセージは、テキスト表示領域をクリックした位置にある単語を、第3アウトレットから出力するように設定する。デフォルト値は0である (文字をクリックしたことになる)。
- keymode** **keymode**メッセージに続く0または1の数値によって、テキスト表示領域でキーボード入力中に**return**キーを押した時の動作を設定する。**clickmode 0**メッセージは、テキスト入力が可能であり、通常の改行として表示するよう設定する。**keymode 1**メッセージなら、**return**キーによって、現在のバッファの内容を出力するよう設定する。デフォルト値は0である。



- lines** **lines**メッセージに続く数値によって、**textedit**オブジェクトに表示するテキストの最大行数を指定する。**line 0**なら、行数の制限がなくなる。数値入力やダイアログで用いるような単語入力のために**textedit**オブジェクトを用いる場合は、**line 1**を指定すると良い。デフォルトでは行数制限はない。
- outputmode** **outputmode**メッセージに続く**0**か**1**の数値によって、**textedit**オブジェクトがその内容をテキストとして出力するか、1つのシンボルとして出力するかを指定する。**outputmode 0**メッセージなら、バッファの内容をメッセージとして出力する。**outputmode 1**メッセージなら、バッファの内容を1つのシンボルとして出力する。初期値は**0**である（メッセージとして出力）。
- readonly** **readonly**メッセージに続く**0**か**1**の数値によって、**textedit**オブジェクトの**read only**モードを切り替える。**readonly 1**メッセージは、テキスト・ボックスへのユーザーの入力を不可能に設定する。**textedit**オブジェクトのバッファの内容を操作する**clear**、**append**、**separator**などのメッセージは、**readonly**メッセージの影響を受けない。デフォルト値は**0**である（read onlyモードはオフ）。
- set** **set**メッセージに続くあらゆるメッセージは、出力することなく、**textedit**オブジェクトのバッファの内容を設定する。
- select** テキストをハイライト状態にする。read onlyモードがオフであれば、キーボード入力を**textedit**オブジェクトへ受け取るように設定する。
- separator** **separator**メッセージに続く任意のシンボルによって、そのシンボルを行を区切る文字に設定し、バッファの内容を出力する時に、そのシンボルを改行文字として扱う。例えば、バッファの内容が「**red green blue**」であり、オブジェクトが**separateor green**を受け取ると、次に**bang**を受け取った時に、「**red**（改行）**blue**」と出力される。
- wordwrap** **wordwrap**メッセージに続く**0**か**1**の数値によって、テキストが**textedit**オブジェクトのテキスト表示領域を超えた場合の表示方法を設定する。**wordwrap 0**メッセージは、テキスト表示領域の境界における単語のワード・ラップ（自動改行）を行うよう設定する（デフォルト）。**clickmode 1**メッセージは、ワード・ラッピングを無効にする。
- (Fontメニュー) **Font**メニューからフォントやサイズを選択することによって、**textedit**オブジェクトに表示する文字のフォントとサイズを変更できる。

## インスペクター

**textedit**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**textedit**オブジェクトを選択することで、**textedit**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**textedit**オブジェクトを選び、**Object**メニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

*Muximum Lines*と表示されているナンバー・ボックスに数値を入力すると、テキスト表示領域に表示するテキストの最大行数を設定できる。デフォルト値は0である。Optionには**textedit**オブジェクトの振る舞いと出力を設定するチェック・ボックスがある。デフォルトでは、それらはチェックされていない。*Read only*をチェックすれば、オブジェクトをテキストを表示するだけの状態になる。*Return Enters Text*をチェックすれば、returnキーの入力によって、現在のバッファの内容を出力ようになる。*Output as One Symbol*をチェックすれば、**textedit**オブジェクトはバッファの内容をメッセージではなく、1つのシンボルとして出力する。*Word wrap around*オプションをチェックすれば、テキスト表示領域の境界にある単語のワード・ラップを行なう。そして *Automatic scrolling* オプション (デフォルトではオン) は、選択されたテキストを自動的にスクロールさせる。また、**textedit**オブジェクトの出力の動作は *When Clicked...* チェック・ボックスによって設定される。これはテキストをクリックした時に、文字か単語のいずれを出力するかを設定できる。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

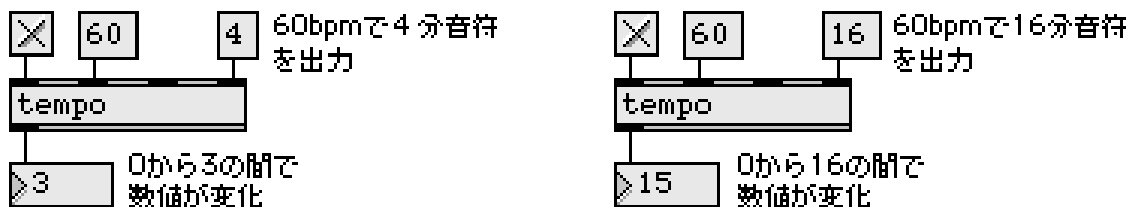
なし。



## 出力

- メッセージ** 第1アウトレット：**bang**メッセージを受け取った時に、**textedit**オブジェクトのバッファに現在格納されている内容を出力する。**textedit**オブジェクトに**keymode 1**メッセージが入力されている場合、returnキーを入力すれば、入力したテキストとバッファの内容が出力される。
- 数値** 第2アウトレット：キーボード入力された文字のアスキー値を出力する。
- メッセージ** 第3アウトレット：テキスト・ボックスをクリックした位置の文字または単語を出力する。

## 利用例



**coll** オブジェクトにテキストを格納する

## 参照

### dialog Text

テキスト入力のためのダイアログ・ボックスを開く  
数値をテキスト・ファイルとしてフォーマットする

**thispatcher**オブジェクトは操作したいパッチ内に置かれる。**thispatcher**オブジェクトは、それを含むパッチに対してメッセージを送る。

## 入力

- loadbang** **thispatcher**オブジェクトに**loadbang**メッセージを送ると、同じパッチ内にある任意の**loadbang**オブジェクトは**bang**を出力する。そのほかのオブジェクトでも（たとえば**preset**オブジェクトのように）内部に初期化のための**loadbang**メッセージを使用しているものは同様に、このメッセージを受信する。
- front** パッチ・ウィンドウを前面に表示させる。パッチが非表示のサブ・パッチの時は、サブ・パッチのウィンドウを開き、さらに前面に表示する。
- wclose** パッチ・ウィンドウを閉じる。そのパッチが編集されている場合は、変更内容を保存するか尋ねられる。
- clean** パッチ・ウィンドウの**dirty**フラグをリセットする。これによりウィンドウを閉じる時に、変更内容を保存するか尋ねられることはなくなる。
- dirty** パッチ・ウィンドウに**dirty**フラグを設定する。これによりウィンドウを閉じる時に、変更内容を保存するか尋ねられるようになる。
- dispose** パッチ・ウィンドウを完全に閉じて、そのメモリーを解放する。**pcontrol**オブジェクトへの**load**メッセージと併用することで、パッチの開閉を自動化することができる。パッチが編集されているなら、変更内容を保存するか尋ねられる。
- offset** (**bpatcher**オブジェクトを使用することで) ボックス内にあるパッチに対して、**offset**メッセージはボックス内にあるパッチの表示部分の左上部の隅の位置を設定する。**offset**メッセージは2つの数値を伴わなければならない。最初の数値は左のオフセットを（ピクセル単位で）、2番目の数値は上部のオフセットを設定する。デフォルトで**bpatcher**ボックス内のパッチはオフセット値が0, 0の地点に表示される。command+shiftキー (Macintosh) あるいはCtrl+shiftキー (Windows) を押したまま**bpatcher**オブジェクトのボックスをドラッグすれば、オフセットはその移動に従って変更される。そして、最新のオフセットは**bpatcher**オブジェクトを含むウィンドウのアシスタンス・エリアに表示される。この数値はオフセット・メッセージのアーギュメントを確定するのに利用できる。

- path** パッチ・ウィンドウがファイルとして保存されているなら、**path**メッセージでこのパッチ・ファイルのあるフォルダーのフル・パス名を**thispatcher**オブジェクトの第2アウトレットから出力させる。
- write** パッチが名前を持っているなら、そのパッチを保存する。名前がない場合、標準ファイル保存ダイアログを開く。
- (その他) **thispatcher**オブジェクトは、新しいオブジェクトを作るためのメッセージに反応する。このようなメッセージの形式は隠され、変更される可能性があるが、パッチ・ファイルをテキスト形式で開き、その内容を調べて**#P**から始まるメッセージを試すことから、さまざまなアイデアを得ることはできる。**#P**の部分を取り除き、残りのメッセージ部分を**thispatcher**に送れば良い。**thispatcher**オブジェクトを、新しいオブジェクトを作るために使うことはサポート外である。
- patcher** 任意のテキストを伴った**patcher**メッセージは、タイトル・バーに表示される名前を置き換える。この新しいウィンドウ名はカッコでくくられているが、これは実際のファイル名ではないことを示し、実際のファイル名は変更されていない。
- window** **window notitle**メッセージはパッチ・ウィンドウのタイトル・バーを隠す。**window title**メッセージはタイトル・バーを表示する。**window flags noclose**メッセージは通常パッチ・ウィンドウのタイトル・バーにあるクローズ・ボックスを隠す。**window flags close**メッセージはクローズ・ボックスを表示する。**window flags nozoom**メッセージは通常タイトル・バーの右端にあるズーム・ボックスを隠す。**window flags nogrow**メッセージはスクロール・バーと通常ウィンドウの右下にあるグロー・ボックスを隠す。**window flags grow**メッセージはスクロール・バーとグロー・ボックスを表示する。**window size**メッセージは4つの数値を伴うが、それぞれ左端、上部、右端、下部のウィンドウ領域に対応させた正確なスクリーン座標を（スクリーンの左上からの距離をピクセル単位で）設定する。左端と上部の座標はタイトル・バーのところではなくてウィンドウ内部の左上に対応する。
- fullscreen 1**メッセージはメニュー・バーを隠し、パッチ・ウィンドウをスクリーン全体に表示させる。その場合、タイトル・バーとスクロール・バーは表示しない。**window fullscreen 0**メッセージはメニュー・バーを表示し、パッチ・ウィンドウを以前のサイズとアピアランスに戻す。

上記の**window**メッセージは、**window exec**メッセージを送るまで効果を生じない。

**window getsize**、**window getflags**、そして**window gettitle**メッセージによって、**thispatcher**オブジェクトは第1アウトレットから現在のウィンドウ情報を出力する。

**savewindow** 0以外の数値を伴った**savewindow**メッセージは、**window flags**メッセージによって設定された特別なウィンドウ設定を、パッチの一部として次回から保存するようになることを示す。**savewindow 0**メッセージは、**window flags**メッセージによるウィンドウ設定の変更内容を、パッチが保存される時に一緒に保存しない。この場合、以前のパッチ・ウィンドウの設定が保存されることになる。

## スクリプト・メッセージ

**thispatcher**オブジェクトへスクリプト・メッセージを送ることで、オブジェクトの生成、削除、大きさ変更、位置変更、そしてパッチングに渡るダイナミックな操作を可能にする。スクリプト・メッセージは機能を表すキーワードを伴って使われる。キーワードの次にくるものはアークギュメントで、これはメッセージがオブジェクトにどのような影響を与えるかを指定する。

下記で説明する各スクリプト・メッセージに関する記述の中で、シンタックスの部分ではそのメッセージに必要とされるアークギュメントを、キーワードの後にくる山形カッコの中に示してある。各メッセージの例もまた用意されている。

variable-nameは新しいオブジェクト名か、または既にあるオブジェクト名を表すシンボルである。variable namesはObjectメニューの**Name...**から設定することができる。または、**new**や**select**といったスクリプト・メッセージによっても設定できる。

## オブジェクトの生成と削除

**new** 新しいオブジェクトをパッチ内に生成し、名前を付ける。

シンタックス : **script new** <variable-name> <creation message>

例 : **script new footog toggle 101 93 15 0**

**toggle**オブジェクトを15ピクセルで101 93の位置に、footogというvariable nameを付けて生成する。

Maxオブジェクトの保存形式は文章ではないので、求めるオブジェクトを生成する適切なメッセージを決定するには、テキスト形式のMaxパッチを調べる必要がある。ほとんどのオブジェクトは下記の標準的な形式で保存されている：

#P クラス名 アーギュメント; (internal UI object)

#P newex クラス名 アーギュメント; (normal internal or external object)

#P user クラス名 アーギュメント; (external UI object)

#Pとセミコロンを削除して、残りのメッセージを新たに生成するオブジェクトのvariable nameの後に置けばよい。

**delete** パッチ・ウィンドウ内のオブジェクトを削除する。

シンタックス : **script delete** <variable-name>

例 : **script delete footog**

**footog**というvariable nameであるオブジェクトを削除する。

### オブジェクトにVariable Nameを割り当てる

**class** アーギュメントの一致する特定のクラスで最初のインスタンスへvariable nameを割り当てる。

シンタックス : **script class** <variable-name> <class-name> <arguments (optional)>

例 : **script class rubadub + 4**

パッチ内でアーギュメントを4にもった+オブジェクトに、**rubadub**というvariable nameを割り当てる。

**nth** 特定のクラスでn番目のインスタンスに対してvariable nameを割り当てる。

シンタックス : **script nth** <variable-name> <class-name> <index>

例 : **script nth yoyo toggle 1**

パッチ内で最初の**toggle**オブジェクトに対して、**yoyo**というvariable nameを割り当てる。

パッチ内のオブジェクトの順番は前面背面の並びによって決定される。他のオブジェクトの背後に表示される背面のパッチは選択される順番としては一番早い。

**selected** 選択されているオブジェクトで最初のオブジェクトに対してvariable nameを割り当てる。

シンタックス : `script selected <variable-name>`

例 : `script selected impo`

選択されたオブジェクトで最初のオブジェクトに**impo**というvariable nameを割り当てる。言うまでもなく、パッチがロック状態の時はオブジェクトを選択できないので、このスクリプト・メッセージはパッチがロックされていない状態の時にしか機能しない。

## オブジェクトの接続と切断

下記の3つの接続メッセージすべてに対して言えることであるが、インレットとアウトレットはインデックスで指定される。0は第1インレットまたは第1アウトレットを意味する。

**connect** 2つのオブジェクトをパッチ・コードで繋ぐ。

シンタックス : `script connect <outlet-variable-name> <outlet-index> <inlet-variable-name> <inlet-index>`

例 : `script connect fooboo 0 bobo 0`

オブジェクトのvariable nameが**fooboo**の第1アウトレットと、variable nameが**bobo**の第1インレットを繋ぐ。

**disconnect** パッチ・コードで繋がれている2つのオブジェクトを切断する。

シンタックス : `script disconnect <outlet-variable-name> <outlet-index> <inlet-variable-name> <inlet-index>`

例 : `script disconnect fooboo 0 bobo 0`

このメッセージは**fooboo**の第1アウトレットと、**bobo**の第1インレットの間の繋がりを解く。

**connectcolor** 存在するパッチ・コードのカラーを変更する。カラーはMaxの16種の標準カラーの内から設定する。

シンタックス : `script connectcolor <outlet-variable-name> <outlet-index> <inlet-variable-name> <inlet-index> <color>`

例 : `script connectcolor rover 0 dover 2 12`



**rover**の第1アウトレットと**dover**の第3インレットとの間に結ばれたコードのカラーを、12番目の標準カラーに変更する。

## オブジェクトのプロパティを変更する

**hide** 表示されているオブジェクトを隠す。

シンタックス : **script hide <variable-name>**

例 : **script hide visigoth**

**visigoth**という名のオブジェクトを隠す。

**show** 隠れているオブジェクトを表示する。

シンタックス : **script hide <variable-name>**

例 : **script hide visigoth**

**visigoth**という名のオブジェクトを表示する。

**ignoreclick** マウス・クリックに反応しないようにオブジェクトを設定する。

シンタックス : **script ignoreclick <variable-name>**

例 : **script ignoreclick visigoth**

**visigoth**という名のオブジェクトがマウス・クリックを無視するようになる。

**respondtoclick** オブジェクトをマウス・クリックに反応するようにする。

シンタックス : **script respondtoclick <variable-name>**

例 : **script respondtoclick visigoth**

**visigoth**という名のオブジェクトがマウス・クリックに反応するようになる。

**bringtofront** オブジェクトを現在あるレイヤーの前面に持ってくる。

シンタックス : **script bringtofront <variable-name>**

例 : **script bringtofront visigoth**



**visigoth**がforegroundのレイヤーにある時、このメッセージはオブジェクトをforegroundのレイヤーの最前面に移動させる。そうでなければ、オブジェクトをbackgroundのレイヤーの最前面に移動させる。

**sendtoback** オブジェクトを現在のレイヤーの最背面へ移動させる。

シンタックス : `script sendtoback <variable-name>`

例 : `script sendtoback visigoth`

**visigoth**がforegroundのレイヤーにある時、このメッセージはオブジェクトをforegroundのレイヤーの最背面に移動させる。そうでなければ、オブジェクトをbackgroundのレイヤーの最背面に移動させる。最背面にあるオブジェクトは、**nth**や**class**の変数割り当てメッセージによって、最初に見つけ出されるオブジェクトである。

**size** オブジェクトのサイズを変更する。サイズに制限のあるオブジェクトもいくつかあるが、一般にそれらは不適切なサイズから守られているわけではないので、このメッセージを使うには注意が必要である。例えば、**toggle**オブジェクトは正方形であることが通常である。長方形に作ろうとすれば、適切に表示できないであろう。

シンタックス : `script size <variable-name> <width> <height>`

例 : `script size togipoo 30 30`

**togipoo**という名のオブジェクトの幅と高さを、それぞれ30ピクセルに変更する。

## オブジェクトへメッセージを送信する

**send** 1つのオブジェクトへメッセージを送信する。このメッセージはセミコロンを使ったメッセージ・ボックスや**send**オブジェクトを使うのと同じであるが、メッセージを受け取るオブジェクトをvariable-nameで特定し、スクリプトを使ってそのオブジェクトへ伝達するのに使える。このメッセージは、**thispatcher**オブジェクトのあるパッチ内のオブジェクトに対してのみ、メッセージを送信する。

シンタックス : `script send <variable-name> <message>`

例 : `script send foobert 666`

**foobert**という名のオブジェクトへ整数666を送信する。**foobert**がナンバー・ボックスならば、表示は666に変更される。

**sendbox** オブジェクト・ボックスへメッセージを送信する。このメッセージはボックスに参照されるオブジェクトに対してというよりも、オブジェクトのボックスそのものへメッセージを送るためのものである。現在のところこのようなオブジェクトは**bpatcher**のみであり、この場合オブジェクト・ボックスとその中身は異なるオブジェクトになっている。ボックスは**bpatcher**であり、オブジェクトはその中のパッチということになる。では、**bpatcher**オブジェクトにどのようなメッセージを送ることができるだろうか。1つの例は、**boxborder**メッセージで、これは**bpatcher**オブジェクト内のパッチに置かれた**thispatcher**オブジェクトへ**border**メッセージを送るのと同じである。他のアイデアを得るために、**bpatcher**オブジェクトのインスペクターの中を覗くと良い。

シンタックス : `script sendbox <variable-name> <message>`

例 : `script sendbox bpbp boxborder 0`

**bpatcher**オブジェクトが**bpbp**という名前だとすると、このスクリプト・メッセージは境界線を描画しないように設定する。

## オブジェクトの移動

**move** パッチ・ウィンドウの左上の位置からの距離として、オブジェクトの位置を移動させる。0,0はアイコン・バーのちょうど下の位置である。

シンタックス : `script move <variable-name> <top> <left>`

例 : `script move molly 0 100`

**molly**という名のオブジェクトをウィンドウの左端で、最上部から100ピクセル下の位置に移動する。

**offset** オブジェクトの現在の位置から一定の距離だけ移動させる。距離の値が正の時、オブジェクトは右下に移動し、負の時は左上に移動する。

シンタックス : `script offset <variable-name> <delta-x> <delta-y>`

例 : `script offset molly 30 - 40`

**molly**という名のオブジェクトを30ピクセル右、40ピクセル上に移動させる。

**offsetfrom** オブジェクトを他のオブジェクトから一定の距離に移動させる。

シンタックス : `script offsetfrom <variable-name-to-move> <target-variable-name> <delta-x> <delta-y> <top-left-flag>`

距離をオブジェクトの左上の端から指定する時、top-left-flagは1である。右下から指定する時、top-left-flagは0である。

例 : `script offsetfrom molly panther -100 -120 1`

**molly**という名のオブジェクトを**panther**という名のオブジェクトに対して、100ピクセル左、120ピクセル上の位置に移動させる。

## アーギュメント

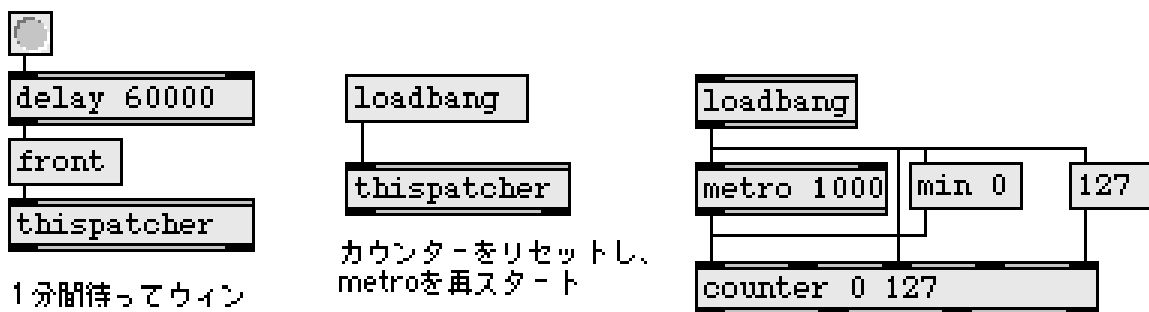
なし。

## 出力

**window** 第1アウトレット : **window getsize**メッセージを受け取る時、**thispatcher** オブジェクトはwindow sizeという言葉の後に、ウィンドウのスクリーン座標 (スクリーンの左上からの距離をピクセル単位で表したものを左、上、右、下の順で出力する。**window gettitle**メッセージを受け取る時、ウィンドウがタイトル・バーを持つかどうかによって、**window title**か**window notitle**メッセージを出力する。**window getflags**メッセージを受け取る時、**hispatcher**オブジェクトはwindow flagsという言葉の後に、スクロール・バーとグロー・ボックスの状態 (growまたはnogrow)、クローズ・ボックスの状態 (closeまたはnoclose)、そしてズーム・ボックスの状態 (zoomまたはnozoom) を出力する。

**symbol** 第2アウトレット : **path**メッセージを受け取った時に、パッチのファイルがあるフォルダー、またはボリュームのフル・パス名を出力する。パッチがまだ保存されていない時、何も出力されない。

## 利用例



自動化されたウィンドウ操作、ファイルの保存、パッチのリセットは **thispatcher** オブジェクトで行える

タイトルバーとスクロールバーが無い小さなウィンドウを  
画面の左上隅に表示

```
window nogrow, window notitle,
window size 0 20 160 140,
window exec, front
```

```
thispatcher
```

パッチのウィンドウは任意のサイズ、位置、アピアランス、を独自に設定できる

## 参照

**bpatcher**  
**pack**  
**patcher**  
**pcontrol**  
**sprintf**

ボックス内に埋め込まれた可視型のサブ・パッチ  
数値やシンボルをひとつのリストに結合させる  
パッチの中でサブ・パッチを作る  
パッチ内のサブ・ウィンドウを開いたり、閉じたりする  
単語や数値のメッセージをフォーマットする

Tutorial 46  
Tutorial 47

Basic Scripting (英文PDF参照)  
Advanced Scripting (英文PDF参照)

## 入力

任意のメッセージ **thisTimeline** オブジェクトがアクション・パッチ内にあり、そのアクション・パッチが **timeline** に使用されている場合、**timeline** が受け取ることができるあらゆるメッセージを、**thisTimeline** オブジェクトに入力することができます、そのメッセージは **timeline** に送られる。

**bang** アクション・パッチに **thisTimeline** オブジェクトを含む **timeline** の現在時刻を出力する。

## アーギュメント

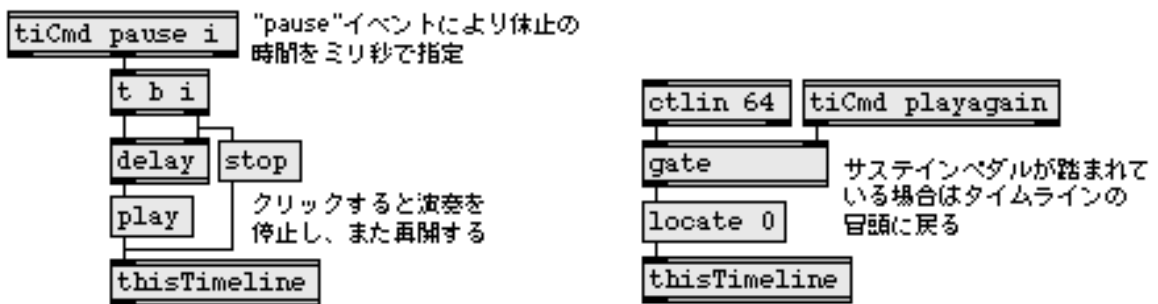
なし。

## 出力

(**timeline**へ) インレットに受け取ったメッセージは、**thisTimeline** オブジェクトを含むアクション・パッチを持つ **timeline** に伝えられる。

**整数** インレットに **bang** を受け取ると、アクション・パッチを含む **timeline** の現在時刻がミリ秒単位で、**thisTimeline** オブジェクトのアウトレットから出力される。

## 利用例



アクション・パッチ内の **thisTimeline** を通して、**timeline** は実際に自分自身をコントロールすることができる

## 参照

**thisTrack** タイムライン・トラックへメッセージを送る  
**tiCmd** **timeline** オブジェクトからメッセージを受け取る  
**timeline** Maxメッセージを時間軸で扱うスコア

Tutorial 41 Timeline of Max messages (英文PDF参照)

## 入力

- 任意のメッセージ **thisTrack**オブジェクトがアクション・パッチ内にあり、そのアクション・パッチが**timeline**に使用されている場合、**thisTrack**オブジェクトが受け取るメッセージは、アクション・パッチを呼び出している**timeline**のトラックに送られる。
- mute** 0以外の数値が続く**mute**メッセージは、**thisTrack**オブジェクトを含むアクション・パッチの**timeline**トラックの動作をミュートする。**mute 0**メッセージは、ミュートを解除する。
- name** **name**メッセージに続く任意のシンボルによって、アクション・パッチを含む**timeline**のトラック (**timeline**のグラフィック・エディター・ウィンドウ内) の名前を設定する。
- height** **height**メッセージに続く0以上の数値で、**timeline**のグラフィック・エディター・ウィンドウに表示されるトラックの高さを、ピクセル単位で設定する。

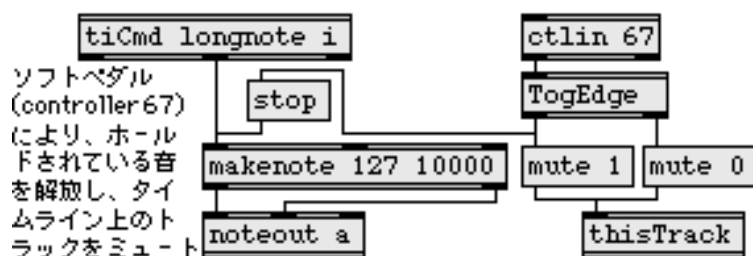
## アーギュメント

なし。

## 出力

- (**timeline**へ) インレットに受け取ったメッセージは、**thisTrack**オブジェクトを含むアクション・パッチの**timeline**トラックに送られる。

## 利用例



**timeline** のアクション・パッチは、**thisTrack** オブジェクトによって、そのトラックの動作をミュートすることができる

## 参照

**thisTimeline**  
**tiCmd**  
**timeline**

タイムラインへメッセージを送る  
**timeline**オブジェクトからメッセージを受け取る  
Maxメッセージを時間軸で扱うスコア

## 入力

**整数または実数** 第1インレット：ある時間内に数値を受け取ると、それぞれの数値がリストとしてまとめられる。数値を受け取る時間間隔が、設定した値を超えると、アウトレットからリストが出力され、新しいリストを作り始める。

第2インレット：数値をまとめてリストとして出力する時間間隔をミリ秒単位で設定する。設定した時間を超えても第1インレットに新しい数値が入力されない場合に、アウトレットからリストが出力され、新しいリストを作り始める。

**リスト** 第1インレット：受け取ったリスト全体が、**thresh**オブジェクトが格納しているリストに追加される。

## アーギュメント

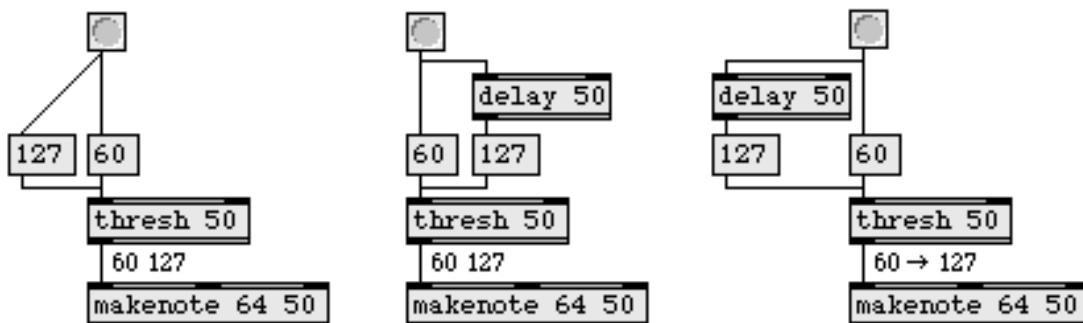
**整数** オプション。時間間隔の初期値を設定する。アーギュメントを指定しなければ、初期値は10ミリ秒となる。

**実数** 整数に変換される。

## 出力

**リスト** 第1インレットに受け取った数値をリストに追加し、格納する。指定した時間が経過しても、新しい数値を受け取ることがない場合は、リストが出力され、新しいリストを作り始める。

## 利用例



一定時間を超えても、新たな数値を受け取ることがない場合、**thresh** は格納しているリストを出力する



## 参照

**bondo**  
**buddy**  
**iter**  
**pack**  
**zl**

一群のメッセージを同期する  
入力されたデータを同期させ、それらを同時に出力する  
リストを数値ごとに分解する  
数値やシンボルをひとつのリストに結合させる  
多目的なリスト処理

Tutorial 37

Data structures (英文PDF参照)

## 入力

**tiCmd**オブジェクトは、**timeline**オブジェクトのトラックとして読み込まれているアクション・パッチ内に置かれるものである。**tiCmd**オブジェクトは、**timeline**のトラックの中で同じ名前のイベント・エディターから入力情報を受け取る。受け取ることのできるメッセージのタイプはアーギュメントとして書き込まれる**i**、**f**、**l**、**b**、**s**あるいは**a**などによって決まる。

**整数** 2番目の（そして最後の）アーギュメントに**i**が入力されると、**tiCmd**オブジェクトは**timeline**のイベント・エディターから整数を受信するようになり、その数値を第2アウトレットから出力する。**timeline**オブジェクトのトラックに置かれるイベント・エディターで整数を送信するのは、次の3タイプである。**int**、**etable**、**efunc**。

**int** イベント・エディターは外観も機能もパッチ内のナンバー・ボックスに似ている。**timeline**オブジェクトが再生され、**int** イベント・エディターに到達すると、そのナンバー・ボックス内の数値を適切な**tiCmd**オブジェクトへ送信し、数値は**tiCmd**オブジェクトの第2アウトレットから出力される。

**etable** イベント・エディターは**table**オブジェクトとよく似ており、グラフィカルに整数の配列を編集できる。**timeline**オブジェクトが再生され**etable** イベント・エディターに到達すると、**timeline**上で**etable**が占有している割合に比例してすべての数値を1つずつ出力する。例えば、128の数量を持つ**etable**が**timeline**のトラック内に1000から9000（ミリ秒）の範囲を占有しているとする、**timeline**がこの8秒間を進行する間、**tiCmd**オブジェクトは1秒間に16の割合で整数を受信することになる。

**efunc** イベント・エディターはx軸、y軸の数値ペアの二次元配列をグラフィカルに編集できる。**timeline**オブジェクトが再生され、**efunc** イベント・エディターに到達すると、(x値の最大値に関係してくる) x値と、**timeline**上で**efunc**が占有している割合に比例して決められる時間で**tiCmd**オブジェクトへ**efunc**のy値を送信する。例えば、**efunc**のx値の最大範囲が1000で、その**efunc**が1000から9000（ミリ秒）の範囲を占めているとする。そして、x軸、y軸のペアが500、127とすると、(y値である) 127は時間が（1000から9000の長さの500 / 1000 の位置にあたる）5000の時に**tiCmd**に送信される。

**実数** 2番目の（そして最後の）アーギュメントに**f**が入力されると、**tiCmd**オブジェクトは**timeline**のイベント・エディターから実数を受信するようになり、その数値を第2アウトレットから出力する。**float** イベント・エディターは外観も機能も、パッチ内のフロート・ナンバー・ボックスに似ている。

**リスト** 2番目のアーギュメントに**l**が入力されるか、2つ以上のアーギュメントが入力された場合、**tiCmd**オブジェクトは**timeline**内のメッセージャー・イベント・エディターからリストを受信する。メッセージャーはイベントの名前（送信先の**tiCmd**オブジェクトの名前）が最初に書かれている以外は、メッセージ・ボックスに似ている（名前は送信されるメッセージには含まれない。ただ、どこにメッセージを送るかを示すだけである）。

**bang** 2番目の（そして最後の）アーギュメントに**b**が入力されると、**tiCmd**オブジェクトは**timeline**のメッセージャーから**bang**を受信する。これはメッセージャー内のメッセージのタイプには関係しない。

**シンボル** 2番目の（そして最後の）アーギュメントに**s**が入力されると、**tiCmd**オブジェクトは**timeline**のメッセージャーからシンボルを受信する。メッセージャー内に1つ以上の言葉が入力されている時、最初の言葉だけが**tiCmd**オブジェクトへ送られる。1つより多くの言葉を含んだメッセージャーがあり、そしてそれらの言葉すべてを1つのシンボルとして**tiCmd**オブジェクトへ送信する時は、バックスラッシュ (\) を言葉の前に置く。

**任意のメッセージ** 2番目の（そして最後の）アーギュメントに**a**が入力されると、**tiCmd**オブジェクトは**timeline**のメッセージャーから任意のメッセージを受信し、何も変えることなく第2アウトレットから出力する。

## アーギュメント

**シンボル** 必須。最初のアーギュメントは**tiCmd**オブジェクトの名前であり、これにより**timeline**トラック内に使用可能なイベントが現れる。1つ以上の**tiCmd**オブジェクトが同じ名前を持っていると、受け取る順番は未定義となるが、各**tiCmd**は同じメッセージを受信する。同じアクション・パッチに同じ名前が存在している**tiCmd**オブジェクトは異なったタイプのアーギュメントを持つ（異なったタイプのメッセージを期待する）ことができる。しかし、**timeline**オブジェクトに現れるイベント・エディターは、最初に読み込まれた**tiCmd**オブジェクトのアーギュメントのタイプによって異なる（このため、いつも同じものとは限らない）。

**i、f、s、l、bまたはa** オプション。最初のアーギュメントの後にある、これらの追加アーギュメントは（常に存在している第1アウトレットと最後のアウトレットに加え）新たなアウトレットを作り、そのアウトレットから指定されたメッセージのタイプが出力される。**i**は整数、**f**は実数、**l**はリスト、**b**は**bang**、**s**はシンボル、**a**は任意のメッセージを表す。このタイプに関するアーギュメントがなければ、真ん中のアウトレットは作られない。しかし、イベントは**timeline**内にメッセージャーとして配置するこ

とができ、**tiCmd**オブジェクトは第1と第2アウトレットから**bang**を出力する。タイプのアーギュメントがfのみならば、**timeline**トラックのイベント・エディターはフロート・ナンバー・ボックスとなる。タイプのアーギュメントがiのみならば、**timeline**トラックのイベント・エディターはナンバー・ボックスと**etable**と**efunc**になる（上記入力メッセージの整数を参照のこと）。タイプのアーギュメントが他のものか、または1つ以上のタイプが記されるなら、**timeline**トラックのイベント・エディターはメッセージャーになる（上記入力メッセージのリストを参照のこと）。

## 出力

**bang** 第1アウトレット：**timeline**上で**tiCmd**オブジェクトと同じ名前を持つイベントに達すると、**tiCmd**オブジェクトは第1アウトレットから**bang**を出力する。

第2アウトレット：アウトレットが**b**のアウトレットとして指定されている時、**bang**は**timeline**でそのイベントに達した時に（第1アウトレットが**bang**を出力したすぐ後に）出力される。**s**のアウトレットで**bang**という言葉が出力された時も同じである。

第3アウトレット：**timeline**上で**tiCmd**オブジェクトと同じ名前を持つメッセージャーの最後に達すると、**tiCmd**オブジェクトは第3アウトレットから**bang**を出力する。

**整数** 第2アウトレット：アウトレットが**i**のアウトレットとして指定されている時、整数は**timeline**でそのイベントに達した時に（第1アウトレットが**bang**を出力したすぐ後に）出力される。（**s**のアウトレットで）整数がシンボルとして出力された時も同じである。

**実数** 第2アウトレット：アウトレットが**f**のアウトレットとして指定されている時、実数は**timeline**でそのイベントに達した時に（第1アウトレットが**bang**を出力したすぐ後に）出力される。（**s**のアウトレットで）小数点を含む数値がシンボルとして出力された時も同じである。

**リスト** 第2アウトレット：アウトレットが**l**のアウトレットとして指定されている時、リストは**timeline**でそのイベントに達した時に（第1アウトレットが**bang**を出力したすぐ後に）出力される。

アーギュメントが2つより多い（名前のアーギュメントに加えて2つ以上の）時、**timeline**からのリストは分けられて、リストの各項目は異なるアウトレットから出力される。

**シンボル** 第2アウトレット：アウトレットが**s**のアウトレットとして指定されている時、シンボルはtimelineでそのイベントに達した時に（第1アウトレットが**bang**を出力したすぐ後に）出力される。しかし、シンボルとして送られてくるものが数値や**bang**の時は、整数、実数、または**bang**として出力する。

**任意のメッセージ** 第2アウトレット：アウトレットが**a**のアウトレットとして指定されている時、そのメッセージはtimelineでイベントに達した時に（第1アウトレットが**bang**を出力したすぐ後に）出力される。

### 利用例



**timeline** は **tiCmd** オブジェクトを介してアクション・パッチと繋がっている

### 参照

- thisTimeline** タイムラインへメッセージを送る
- timeline** Maxメッセージを時間軸で扱うスコア
- Tutorial 41 Timeline of Max messages (英文PDF参照)

**timein**オブジェクトは利用時にOMS（そしてOMS Timing）を必要とする。OMSがインストールされていない場合は**timein**オブジェクトは機能しない。

## 入力

**整数** 第1インレット：0以外の数値は**timein**オブジェクトをスタートさせる。一定の間隔で、**timein**オブジェクトはOMS Timingから得た現在の時間を出力する。OMS Timingを動かすためには、それをサポートするアプリケーションによってスタートされなければならない（これは**timeout**オブジェクトに**start**メッセージを送ることで、Max自体でも行える）。0は**timein**オブジェクトを停止させる。

第2インレット：この数値は**timein**オブジェクトが出力するタイム・コードの形式を指定する。タイム・コードは下記の番号に対応する：

|    |                            |
|----|----------------------------|
| 0  | ミリ秒                        |
| 1  | beats                      |
| 2  | 24 frames/sec bits         |
| 3  | 25 frames/sec bits         |
| 4  | 30 frames/sec Drop bits    |
| 5  | 30 frames/sec bits         |
| 6  | 24 frames/sec              |
| 7  | 25 frames/sec              |
| 8  | 30 frames/sec Drop         |
| 9  | 30 frames/sec              |
| 10 | 29.97 frames/sec Drop bits |
| 11 | 29.97 frames/sec Drop      |
| 12 | 29.97 frames/sec bits      |
| 13 | 29.97 frames/sec           |

第3インレット：この数値は**timeline**オブジェクトが時間を出力する時間間隔であり、ミリ秒として扱われる。5より小さい数値は5に設定される。第3インレット新たな数値が入力されても、次の出力が送られるまで影響はない。

**実数** 整数に変換される。

**bang** 第1インレット：OMS Timingから得られる現在の時間を出力する。

**format** 第1インレット：数値を伴った**format**メッセージは、OMS Timingのタイム・コードを設定し、現在の時間を出力する。タイム・コードの形式に対応する数値は上記を参照のこと。

**beats** 第1インレット：**timein**オブジェクトがbeats形式で出力するようにな



る。この形式では、時間はティックスとして知られる数値で出力される。ティックスは、通常OMS Timingのテンポで四分音符にあたる長さに対して、480ティックスとなる。

- smpte** 第1インレット：**timein**オブジェクトがOMSによって扱われるフレーム・レートであるSMPTE形式で出力される。**qf**メッセージが先に入力されていなければSMPTE形式はデフォルトでビットとして扱われる。
- qf** 第1インレット：このメッセージの後に**smpte**メッセージが入力される時、(ビットではなく)クォーター・フレーム形式で表示される。
- bits** 第1インレット：このメッセージの後に**smpte**メッセージが受け取られる時、ビット形式で表示される。

## アーギュメント

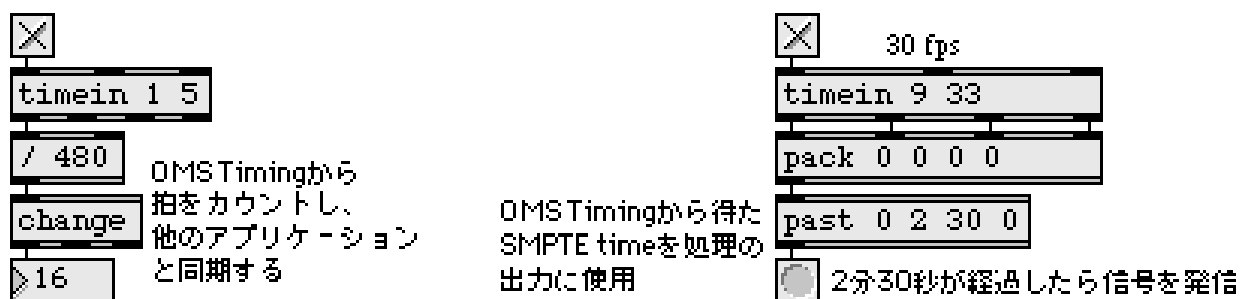
- 整数** オプション。最初のアーギュメントは**timein**オブジェクトが時間を示すタイム・コードの形式を指定する。アーギュメントがない場合、ミリ秒形式に設定される。
- 整数** 2番目のアーギュメントは**timein**オブジェクトが時間を出力してゆく時間間隔の初期値を設定する。アーギュメントがない場合、初期時間間隔は100ミリ秒になる。1より小さい数値は100に設定される。

## 出力

- 整数** 第1アウトレット：現在の時間が一定の間隔で出力される(この間隔は第3インレットに受け取られた数値によって特定される)。時間は(第2インレットによって特定される)タイム・コードの形式によって、ミリ秒、ティック、SMPTEビット形式などで出力される。ビットでないSMPTE形式では、時間は時、分、秒、そしてフレームで出力される。この時、第1アウトレットは時を出力する。
- 第2アウトレット：**timein**オブジェクトがビットでないSMPTE形式の時、このアウトレットからは分が出力される。
- 第3アウトレット：**timein**オブジェクトがビットでないSMPTE形式の時、このアウトレットからは秒が出力される。
- 第4アウトレット：**timein**オブジェクトがビットでないSMPTE形式の時、このアウトレットからはフレーム数が出力される。



## 利用例



OMS Timing は、Max と他のアプリケーションを同期させるための単一クロックを提供する

## 参照

**clocker**  
**tempo**  
**timeout**

一定の間隔で経過時間を出力する  
メトロノームのようなテンポで数値を出力する  
現在の時間をOMSに書き出す

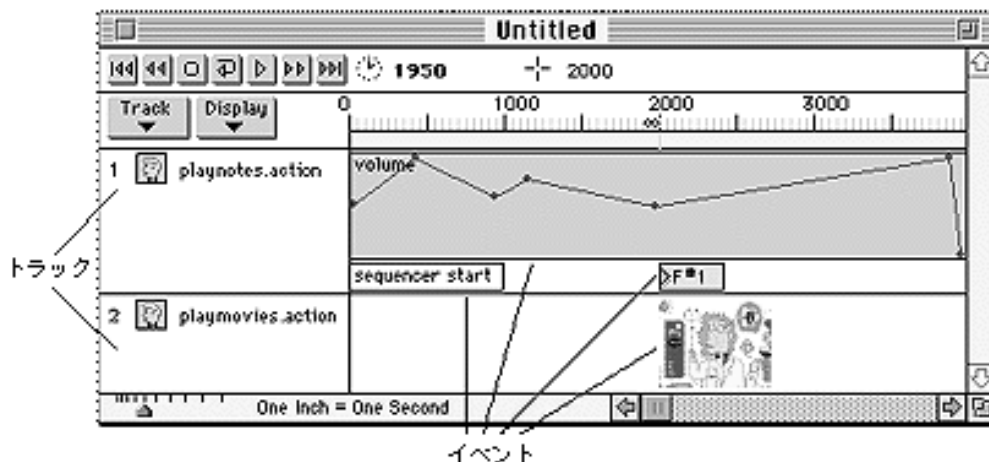
## 入力

- clock** 存在する**setclock**オブジェクトの名前を伴った**clock**メッセージは、**timeline**に対してMaxの内部ミリ秒クロックではなく、**setclock**オブジェクトによって操作されるように設定する。**clock**メッセージのみの場合、**timeline**オブジェクトは通常のミリ秒クロックを扱うように設定する。
- locate** 数値を伴った**locate**メッセージは、**timeline**の時間をミリ秒で特定し、**timeline**オブジェクト内のポインターをその時間に合わせる。**timeline**オブジェクトがすでに再生状態にあり、**locate**メッセージを受け取った時、ポインターの位置は変更されるが、再生は続けられる。
- markers** **timeline**オブジェクトのアウトレット番号を伴った**markers**メッセージは、**timeline**内の各マーカー・イベントの最初の言葉を特定のアウトレットへ出力するが、これは**umenu**オブジェクトのアーギュメントのメッセージとして扱われる（指定されたアウトレットがない場合、エラー・メッセージがMaxウィンドウに表示され、**timeline**オブジェクトからは何も送信されない）。なぜなら、**markers**メッセージは各マーカーを**umenu**オブジェクト内に格納するように意図されたものだからで、このメッセージの最初には**clear**メッセージが送信され、**umenu**オブジェクトが現在格納している内容を消去することになる。そのすぐ後に一連の**append**メッセージが送信され、各マーカーの最初の言葉が**umenu**オブジェクト内に加えられる（こうして**umenu**オブジェクトが出力するテキストは、**prepend search**オブジェクトと一緒に用いて、それを**timeline**オブジェクトのインレットに戻すことにより、ポインターをマーカーの位置に置くことができる。利用例を参照のこと）。
- mute** **timeline**オブジェクト内のトラック番号を伴った**mute**メッセージは、そのトラックをミュートし、アクション・パッチにイベントが送られるのを防ぐ。
- open** **timeline**オブジェクトのウィンドウを表示状態にする。ウィンドウは最前面に表示される。**timeline**オブジェクトをダブルクリックしても同様の効果が得られる。
- play** **timeline**オブジェクトに含まれる時間軸を再生する。
- read** **timeline**ファイルの名前を伴った**read**メッセージは、**timeline**オブジェクトにそのファイルを読み込む。**read**メッセージだけの時は、標準ファイル・ダイアログを呼び出して、**timeline**ファイルを読み込むことができる。

- search** シンボルを伴った**search**メッセージは、**timeline**オブジェクト内のマーカー・イベントでシンボルと正確に一致するものを探す。一致するものがあれば、**timeline**オブジェクトのポインターはそのマーカーの位置に移動する。
- stop** **timeline**の再生を止める。
- timeFormat** 0から4の整数を伴った**timeFormat**メッセージは、**timeline**オブジェクトのエディター・ウィンドウに表示される時間の表示方法を設定する。数値0はミリ秒、1はMIDIクロック、2は24fps（フレーム毎秒）、3は25fps、4は30fpsを表す。他の数値が入力されても、0から4の範囲に制限される。
- unmute** **timeline**オブジェクト内のトラック番号を伴った**unmute**メッセージは、そのトラックのミュートを解除し、アクション・パッチにイベントが送られるのを許す。
- wclose** **timeline**オブジェクトのウィンドウを閉じる。
- write** 標準ファイル保存ダイアログを呼び出し、**timeline**オブジェクト内の情報のみをファイルとして保存することができる。
- zoomLevel** 整数の0から10を伴った**zoomLevel**メッセージは、**timeline**オブジェクトのエディター・ウィンドウが表示する時間軸を拡大する。0は最大縮小（1インチ = 40秒）の意味で、10は最大拡大（1インチ = 0.04秒）となる。**timeline**のウィンドウのデフォルトは4（1インチ = 4秒）である。0から10の範囲を超える数値が入力されても、この範囲に限定される。

**timeline**オブジェクトが作られると、時間軸でMaxメッセージを扱う**timeline**オブジェクトのエディター・ウィンドウが開かれる。**timeline**オブジェクトの中に独立したトラック（これはマルチ・トラック・シーケンサーのトラック、または楽譜の譜表に類似している）として他のパッチも読み込むことができる。そして、メッセージはこのトラックの中に配置して各パッチへ特定の時間にメッセージを送信できる。**timeline**オブジェクトに読み込まれたパッチは、**timeline**オブジェクトからメッセージを受け取るために、少なくとも1つの**tiCmd**オブジェクトを含んでいなければならない。そのようなパッチはアクション・パッチといわれる。**timeline**トラックの中のメッセージはイベントと呼ばれ、それはトラック内に特別なエディター・オブジェクトを配することで生じる。

Timeline Editor window



**timeline** オブジェクトが再生されると、トラック内のイベントは指定されたアクション・パッチの **tiCmd** へ送信され、イベントのメッセージは**tiCmd** オブジェクトのアウトレットから出力される。

## アーギュメント

**シンボル** オプション。最初のアーギュメントは**timeline**オブジェクトが読み込む**timeline**ファイルの名前を指定する。その名前のファイルが見つからないとしても、**timeline**オブジェクトが作られた時にタイトル・バーにその名前が表示された状態で空のエディター・ウィンドウが開かれる。

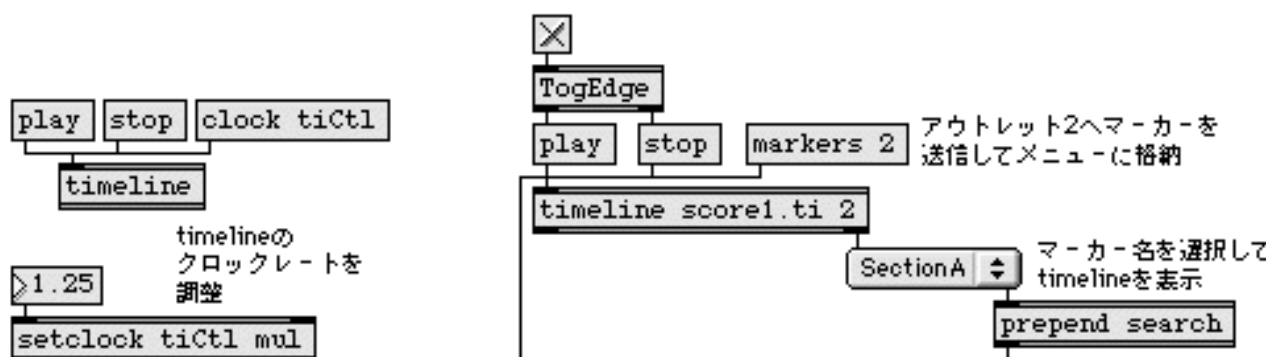
**整数** 2番目に置かれるアーギュメント（または最初のアーギュメントなしでこの整数だけでも良い）は、**timeline**オブジェクトが持つアウトレットの数を設定する。1より小さい数は0に設定される。

## 出力

**任意のメッセージ** **timeline**オブジェクトが正の整数のアーギュメントを持つ場合、オブジェクトはその数分のアウトレットを持つことになる。アクション・パッチ（または**timeline**オブジェクト自身を含んでいるパッチ）が**tiOut**オブジェクトを含んでいる時、**tiOut**オブジェクトが受け取った任意のメッセージは、**timeline**オブジェクトの特定のアウトレットへ送信される。**timeline**オブジェクトがアウトレットを持っていないならば、**tiOut**オブジェクトが読み込まれた時点でMaxウィンドウにエラー・メッセージが表示される。なぜなら、どのようなメッセージも**timeline**オブジェクトに送信することができないからである。

(動作) **timeline** オブジェクトが **play** メッセージを受け取る時、**timeline** オブジェクトのエディター・ウィンドウに置かれているイベントのタイムラインは独自に進行する。タイムライン中のイベントに遭遇すると、そのイベントを特定の **tiCmd** オブジェクト (**timeline** オブジェクトが読み込んだ他のパッチ内にある) に送信し、**tiCmd** オブジェクトは自分自身のアウトレットからそのメッセージを出力する。

## 利用例



*timeline* のスピードを **setclock** オブジェクトで操作する。 **markers** メッセージを使って *timeline* の特定の場所に進む

## 参照

|                     |                                     |
|---------------------|-------------------------------------|
| <b>mtr</b>          | マルチ・トラック・シーケンサー                     |
| <b>setclock</b>     | タイミング関連のオブジェクトのクロック速度をリモート・コントロールする |
| <b>thisTimeline</b> | タイムラインへメッセージを送る                     |
| <b>thisTrack</b>    | タイムライン・トラックへメッセージを送る                |
| <b>tiCmd</b>        | <b>timeline</b> オブジェクトからメッセージを受け取る  |
| <b>tiOut</b>        | <b>timeline</b> オブジェクトにメッセージを送る     |

Tutorial 41      Timeline of Max messages (英文PDF参照)

**timeout**のオブジェクトが動作するためには、OMS（およびOMS Timing）が必要である。OMSがインストールされていない場合は、**timeout**は何も行わない。

## 入力

**start** 第1インレット：OMSタイミング・クロックを開始する。

**stop** 第1インレット：OMSタイミング・クロックを停止する。

**整数** 第1インレット：OMSタイミングの現在時刻を設定する。アーギュメントとして指定したタイム・コードのフォーマットに従って、ミリ秒か、ティックス（4分音符の分数値）か、SMPTEフォーマットでの80フレーム・ビットとして時刻を指定する。**timeout**の設定がSMPTEビット・フォーマットでない場合は、第1インレットは現在時刻の時間を設定する。

第2インレット：**timeout**がSMPTEビット・フォーマットでない場合に、現在時刻の分を設定する。

第3インレット：**timeout**がSMPTEビット・フォーマットでない場合に、現在時刻の秒を設定する。

第4インレット：**timeout**がSMPTEビット・フォーマットでない場合に、現在時刻のフレーム数を設定する。

**format** **format**に続く数値によって、OMS Timingと**timeout**のタイム・コードのフォーマットを設定する。タイム・コードのフォーマットは、0から13までの数値として以下のように表される。

|    |                     |
|----|---------------------|
| 0  | ミリ秒                 |
| 1  | ビート                 |
| 2  | 24フレーム／秒・ビット        |
| 3  | 25フレーム／秒・ビット        |
| 4  | 30フレーム／秒・ドロップビット    |
| 5  | 30フレーム／秒・ビット        |
| 6  | 24フレーム／秒            |
| 7  | 25フレーム／秒            |
| 8  | 30フレーム／秒・ドロップ       |
| 9  | 30フレーム／秒            |
| 10 | 29.97フレーム／秒・ドロップビット |
| 11 | 29.97フレーム／秒・ドロップ    |
| 12 | 29.97フレーム／秒・ビット     |
| 13 | 29.97フレーム／秒         |

- tempo** 第1インレット：**tempo**に続く数値（整数でも実数でも構わない）によって、OMS Timingのビート・テンポ（1分間のビート数、BPM）を設定する。**timeout**がビートのタイム・フォーマットを用いている場合、第1インレットに受け取る数値は、指定したテンポでのビートの分数値（ティックス）を表す。
- granularity** 第1インレット：**granularity**に続く数値によって、4分音符あたりのティックス数を設定する。これは、**timeout**がOMS Timingに時刻を出力する時に用いられる。**granularity**として用いることができる値は、**12、24、48、96、120、160、192、240、320、384、480、640、960、1920**である。これら以外の値では、エラー・メッセージが表示され、設定されない。

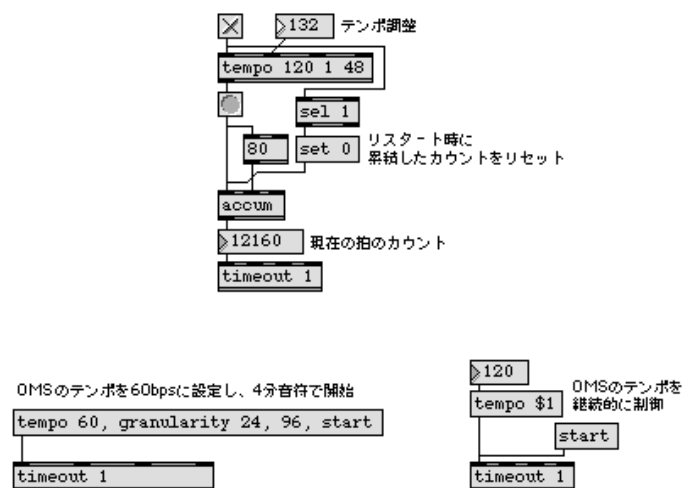
## アーギュメント

- 整数** オプション。時刻を出力する時に用いるタイム・コードのフォーマットを指定する。タイム・コードは先に挙げた0から13までの数値として指定する。アーギュメントを指定しない場合は、デフォルトのタイム・コードとしてミリ秒が用いられる。

## 出力

**timeout**はアウトレットを持たない。**timeout**は、受け取った時刻によって、OMS Timingのクロックをコントロールする。OMS Timingのクロックは、Maxと他のアプリケーション（外部クロック・モードに設定する必要がある）を同期するために用いることができる。

## 利用例



**timeout** は、マスター・テンポを設定するために、あるいは任意の時刻にジャンプするために用いることができる

## 参照

**timein** 外部ソースのタイム・コードを出力する



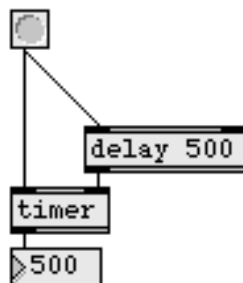
## 入力

- bang** 第1インレット：**timer**オブジェクトを始動または再始動させる。  
第2インレット：**timer**オブジェクトを始動してからの時間を入力させる。
- clock** 第1インレット：既存の**setclock**オブジェクトの名前が続く**clock**メッセージは、Maxの内部クロックでなく、指定された**setclock**オブジェクトによって、**timer**オブジェクトを制御するように設定する。**clock**メッセージだけであれば、Maxの内部クロックによって、**timer**オブジェクトを制御するように設定する。

## 出力

- 実数** **timer**オブジェクトが始動してから、第2インレットに**bang**を受け取るまでの間の経過時間を、ミリ秒単位で出力する。

## 利用例



**bang**を受け取る時間感覚を出力する



ひとつのイベントで時間を入力し、**timer**を再始動させる

## 参照

- clocker** 一定の間隔で経過時間を入力する  
**delay** **bang**を遅らせて出力する  
**setclock** タイミング関連のオブジェクトのクロック速度をリモート・コントロールする

Tutorial 20 Using the computer keyboard (英文PDF参照)

### 入力

任意のメッセージ **tiOut** オブジェクトはアクション・パッチにおいて使用するよう設計されている。アクション・パッチの **tiOut** オブジェクトに受け取られたあらゆるメッセージは、アクション・パッチを使用している **timeline** オブジェクトのアウトレットから出力される。

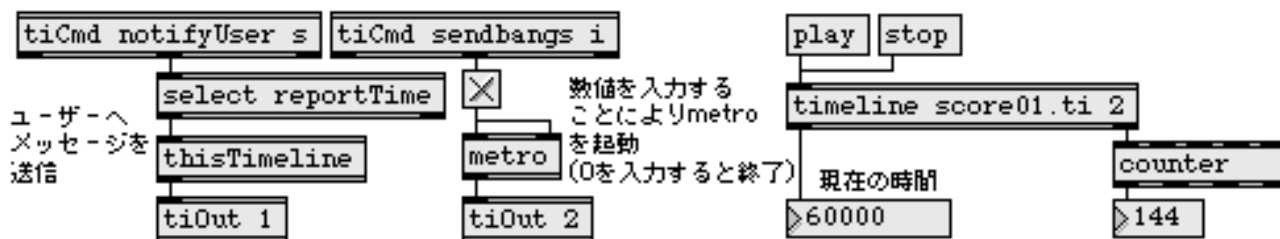
### アーギュメント

**整数** オプション。 **timeline** オブジェクトのどのアウトレットへメッセージを送るかを指定する。アーギュメントを指定しない場合は、 **tiOut** オブジェクトのメッセージは **timeline** オブジェクトの第1アウトレットへ送られる。

### 出力

(**timeline**へ) **tiOut** オブジェクトが受け取った任意のメッセージは、その **tiOut** オブジェクトをアクション・パッチに持つ **timeline** オブジェクトの指定したアウトレットへと送られる。 **timeline** オブジェクトにアウトレットが存在しなければ、 **tiOut** オブジェクトを読み込んだ時点でエラー・メッセージがMaxウィンドウに表示され、 **tiOut** オブジェクトから **timeline** オブジェクトへ何のメッセージも送られない。

### 利用例



メッセージは **tiOut** に入り、 **timeline** の指定したアウトレットから出力される

### 参照

**tiCmd** **timeline** オブジェクトからメッセージを受け取る  
Maxメッセージを時間軸で扱うスコア

Tutorial 41 [Timeline of Max messages \(英文PDF参照\)](#)

## 入力

**整数** 数値は**TogEdge**オブジェクトに格納される。入力する数値が0以外で、以前に入力された数値が0の場合は、**TogEdge**オブジェクトは第1アウトレットから**bang**を出力する。入力する数値が0で以前に入力された数値が0以外の場合は、**TogEdge**オブジェクトは第2アウトレットから**bang**を出力する。それ以外の場合には、何も出力しない。

**実数** **TogEdge**オブジェクトは実数を無視する。

**bang** **TogEdge**オブジェクトに格納されている数値を0から0以外へ、あるいはその逆に変化させ、その変化に応じて、いずれかのアウトレットから**bang**を出力する。

## アーギュメント

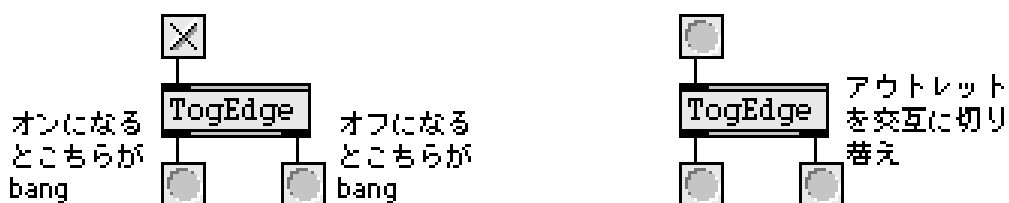
なし

## 出力

**bang** 第1アウトレット：格納している値が0から0以外の数値に変わった時に、**bang**を出力する。

第2アウトレット：格納している値が0以外の数値から0に変わった時に、**bang**を出力する。

## 利用例



オン・オフ状態の検出や、2つのトリガーでスイッチを切り替えるために利用する

## 参照

**change  
led  
toggle**

数値の反復を取り除く  
色によって、オン/オフの状態を表示する  
オンとオフ（1と0）を切り替える



## 入力

**整数** 数値はアウトレットから出力される。この数値が0でない場合には、**toggle**オブジェクトは×印を表示し、オンであることを表す。それが0の場合は、**toggle**オブジェクトは空白を表示し、オフであることを表す。

**実数** 整数に変換される。

**bang** **toggle**オブジェクトがオフの場合にはオンに切り替え、オンの場合にはオフに切り替える。

**toggle**オブジェクト上でマウスのクリックは、そのインレットに**bang**を受け取った時と同じ効果を持つ。

**set** アウトレットから出力せずに、**toggle**オブジェクトのオン・オフを切り替える。0でない値の続く**set**メッセージは**toggle**オブジェクトをオンに設定し、**set 0**メッセージはオフにする。

(マウス) **toggle**オブジェクトをクリックすれば、**bang**メッセージの入力と同じように動作する。

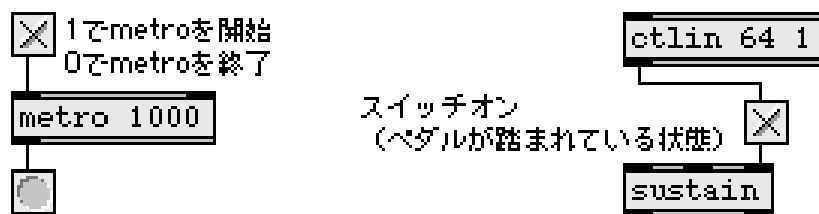
## アーギュメント

なし。

## 出力

**整数** インレットに受け取った数値は、アウトレットから出力される。**bang**かマウスのクリックは、**toggle**オブジェクトがオンまたはオフへ変化した状態に従って、**1**か**0**をアウトレットから出力する。

## 利用例



画面上のコントローラ、または、通過する数値のオン・オフの状態を表示するために使用する



## 参照

|                   |                                        |
|-------------------|----------------------------------------|
| <b>led</b>        | 色によって、オン/オフの状態を表示する                    |
| <b>matrixctrl</b> | マトリックス状のスイッチ・コントロール                    |
| <b>pictctrl</b>   | ピクチャーを用いたコントロール                        |
| <b>radiogroup</b> | ラジオ・ボタンやチェック・ボックスのユーザー・インターフェース・オブジェクト |
| <b>TogEdge</b>    | 0と0以外の数値の変化を出力する                       |
| Tutorial 5        | toggle and comment (英文PDF参照)           |

## 入力

任意のメッセージ **tosymbol** オブジェクトは入力に、任意のメッセージ、数値、リストを受け付け、1つのシンボルを出力する。シンボルは最大2048キャラクターの長さまで可能である。

**separator** **separator** は、連結するために用いられる区切り文字を指定する。アーギュメントを持たない **separator** メッセージはすべてのスペースを取り除き、1つのシンボルを生成する（例えば、**1 2 3 4** は **1234** となる）。**separator** メッセージはパス名を構成するのに用いることができる（例えば **mylaptop max mujunk** というメッセージを **mylaptop:Max:myjunk** とする）。デフォルトの区切り文字は1つのスペースである。

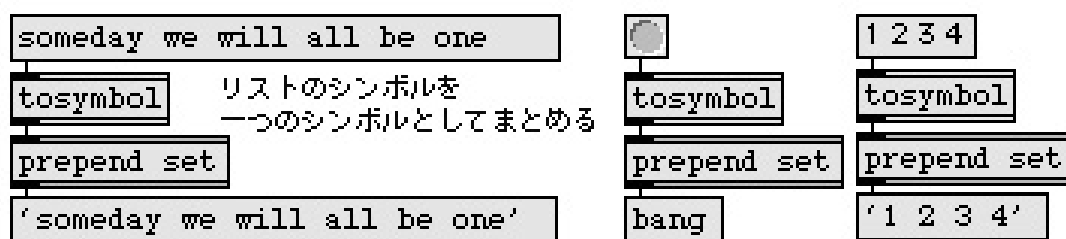
## アーギュメント

なし。

## 出力

シンボル   メッセージ、数値、リストを連結した1つのシンボル。

## 利用例



任意の入力を1つのシンボルへと変換する

## 参照

**fromsymbol**  
**zl**

シンボルを個々の数値かメッセージに変換する  
多目的なリスト処理

## 入力

- (MIDI) **touchin**は、MIDI入力デバイスから受信したMIDIアフタータッチ（チャンネル・プレッシャー）・メッセージを受け取る。
- enable** **enable 0**のメッセージはオブジェクトを使用不能にし、受信するMIDIデータを無視するようになる。**pcontrol**オブジェクトやMIDIEnable/Disableボタンによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを再び受信可能にする。
- port** **port**の後に**a**から**z**の文字を続けるか、MIDI入力デバイスの名前を続けることによって、アフタータッチを受信するポートを設定する。**port**という単語はオプションで、省略することもできる。
- 整数** 数値は、受信したMIDIアフタータッチの値として扱う。第2アウトレットがある場合は、MIDIチャンネル・ナンバーとして0を出力する。受け取った数値は第1アウトレットから出力し、0から127の範囲には限定されない。
- (マウス) **touchin**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

## アーギュメント

- a-z** オプション。アフタータッチを受信するポートを指定する。アーギュメントを指定しない場合、**touchin**はすべてのポートのすべてのチャンネルから受信する。
- (MIDI入力デバイス名) オプション。入力ポートを指定するために、第1アーギュメントにMIDI入力デバイスの名前を使うことができる。
- a-zと整数** 空白文字で区切られた文字と数値の組み合わせは、アフタータッチを受信するポートとMIDIチャンネルを指定する。チャンネル・ナンバーが16より大きい場合は、1から16の範囲に収まるように変換される。
- 整数** 文字と数値の組み合わせの代わりに、数値だけを指定しても良い。チャンネル・ナンバーのアーギュメントの正確な意味は、**MIDI Setup**ダイアログで指定したポートごとのチャンネル・オフセットによって決まる。



## 出力

**整数** アーギュメントで特定のチャンネル・ナンバーを指定している場合は、アウトレットは1個だけ存在する。この場合は、指定したチャンネルとポートから受信したアフタータッチの値である0から127までの数値が出力される。

アーギュメントでチャンネル・ナンバーを指定しない場合は、**touchin**は第2アウトレットを持ち、このアウトレットから受信したアフタータッチ・メッセージのチャンネル・ナンバーを出力する。

## 利用例



アフタータッチ・メッセージは、どこでも受信することができ、ポートのみを指定するか、ポートとチャンネルを指定することができる

## 参照

**touchout**  
**midin**

MIDIアフタータッチを送信する  
生のMIDIデータを受信する

Tutorial 16

More MIDI ins and outs (英文PDF参照)

## 入力

**整数** 第1インレット：数値はMIDIアフタータッチの値として扱われ、指定されたチャンネルとポートから送信される。数値は0から127の間に制限される。

第2インレット：数値はアフタータッチを送信するチャンネル・ナンバーとして格納される。

**実数** 整数に変換される。

**リスト** 第1インレット：リストの1番目の数値はアフタータッチの値として、2番目の数値はチャンネル・ナンバーとして扱われ、指定したチャンネルとポートから送信する。

**enable** **enable 0**メッセージはオブジェクトを使用不可にし、MIDIデータを送信しないように設定する。MIDI Enable/Disableボタンや**pcontrol**オブジェクトによって、パッチ・ウィンドウの全体がMIDIの使用を不可にしている場合でも、0ではない数値が続く**enable**は、オブジェクトを送信可能にする。

**port** **port**の後に文字の**a**から**z**の文字を続けるか、MIDI出力デバイスの名前を続けることによって、アフタータッチを送信するポートを設定する。**port**という単語はオプションで、省略することもできる。

(マウス) **touchout**オブジェクトをダブルクリックすれば、OMSデバイスを選択するダイアログが開く。

## アーギュメント

**a-z** オプション。アフタータッチを送信するポートを指定する。文字のアーギュメントを指定している時に、チャンネル・ナンバーが16より大きければ、1から16の範囲に収まるように変換される。アーギュメントが存在しない場合は、**touchout**は初期状態として、ポートaからMIDIチャンネル1として送信する。

**a-zと整数** 空白で区切られた文字と数値の組み合わせによって、送信するアフタータッチのポートとMIDIチャンネルを指定する。チャンネル・ナンバーが16より大きければ、1から16の範囲に収まるように変換される。

(MIDI出力デバイス名) オプション。MIDI出力デバイスの名前は、ポートを指定する最初のアーギュメントとして用いることができる。

**整数** 文字と数値の組み合わせの代わりに、数値だけを指定しても良い。チャンネル・ナンバーのアーギュメントの正確な意味は、**MIDI Setup**ダイアログで指定したポートごとのチャンネル・オフセットによって決まる。

## 出力

(MIDI) アウトレットは存在しない。オブジェクトに指定したMIDI出力ポートから、MIDIアフタータッチ・メッセージが送信される。

## 利用例



文字のアーギュメントは、送信ポートを限定する

または、数値によってポートとチャンネルを指定する

## 参照

**touchin**  
**midiout**

MIDIアフタータッチを受信する  
生のMIDIデータを送信する

Tutorial 16

More MIDI ins and outs (英文PDF参照)

## 入力

- 整数または実数** 数値は、入力されたアーギュメント（整数、実数、リスト、シンボル（空の場合でも）、**bang**のいずれか）で指定された型に従って、それぞれのアウトレットから出力される。
- bang** **bang**、整数の**0**、実数の**0.**、リストの**0**、空のシンボルのいずれかを、それぞれのアウトレットから出力する。
- リスト** リストは、**1**という文字によって割り当てられたアウトレットから出力される。他のアウトレットではリストは変換され、整数の**0**、実数の**0.**、空のシンボル、**bang**のいずれかとして出力される。
- シンボル** シンボルは、**s**という文字によって割り当てられたアウトレットから出力される。他のアウトレットではシンボルは変換され、整数の**0**、実数の**0.**、リストの**0**、**bang**のいずれかとして出力される。

## アーギュメント

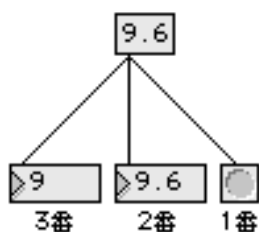
- i, f, b, l, s** オプション。アーギュメントの数はアウトレットの数を決定する。それぞれのアウトレットはアーギュメントに従って整数、実数、**bang**、リスト、シンボルのいずれかを出力する。アーギュメントが指定されていないければ、整数を出力する2つのアウトレットが作られる。
- 任意のメッセージ** オプション。整数、実数、シンボルが指定された場合は、定数としてその値が出力される。

## 出力

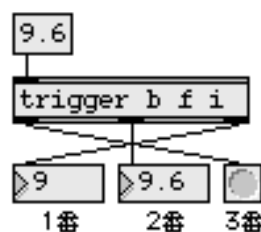
- 整数または実数** インレットに受け取った数値は、右から左への順序で、アウトレットから出力される。その数値は、アウトレットに対応するアーギュメントに従って、整数、実数、リスト、シンボル、**bang**のいずれかに変換されて、出力される。インレットに受け取ったシンボル、リスト、**bang**は、**i**のアーギュメントに対応するアウトレットでは整数の**0**に、**f**のアーギュメントに対応するアウトレットでは実数の**0.**に変換される。
- bang** インレットに受け取ったすべての入力は、**b**のアーギュメントに対応するアウトレットから出力される時に、**bang**に変換される。
- リスト** インレットに受け取ったリストは、**1**のアーギュメントに対応するアウトレットから、そのまま出力される。その他のすべての入力は、**0**という1つの項目を持つリストに変換されて、出力される。

**シンボル** インレットに受け取ったシンボルは、**s**のアーギュメントに対応するアウトレットから、そのまま出力される。その他のすべての入力は、空のシンボル(“”)に変換されて、出力される。注記：この空のシンボルを認識できるのは**print**オブジェクトのみであり、空のメッセージをMaxウィンドウに正確に出力する。他のオブジェクトは、この空のシンボルを無視するか、またはエラー・メッセージをMaxウィンドウに出力する。

## 利用例



通常、右から左への順序で処理される



**trigger** オブジェクトによって、異なる順序を指定できる

## 参照

### bangbang message

順番に複数の**bang**を出力する  
任意のメッセージを送る

### Tutorial 7

Right-to-Left (英文PDF参照)

## 入力

- 整数** 第1インレット：入力値が現在**Trough**オブジェクトに格納されている値より小さければ、それを新しい最小値として格納し、出力する。
- 第2インレット：この数値は、**Trough**オブジェクトに新しい最小値として格納され、出力される。
- 実数** 第1インレット：**Trough**オブジェクトは認識しない。
- 第2インレット：整数に変換される。
- リスト** 第1インレット：2つ目の数値は、新しい最小値として格納され、出力される。そして、1つ目の数値が、第1インレットに受け取られることになる。
- bang** 第1インレット：現在格納されている最小値を、第1アウトレットから出力する。

## アーギュメント

なし。**Trough**オブジェクトに格納される初期値は128である。

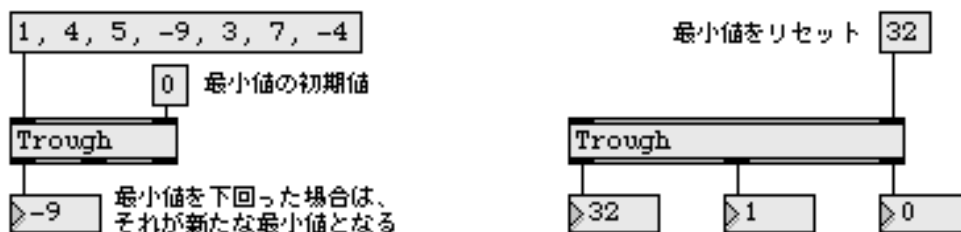
## 出力

- 整数** 第1アウトレット：新しい最小値が出力される（第2インレットに受け取られた数値は、常に新しい最小値となる）。
- 第2アウトレット：受け取られた値が新しい最小値の場合は、**1**が出力される。また、第1インレットに受け取った数値が新しい最小値でない場合は、**0**が出力される。
- 第3アウトレット：受け取られた値が新しい最小値の場合は、**0**が出力される。また、第1インレットに受け取った数値が新しい最小値でない場合は、**1**が出力される。

# Trough

数値が以前の値より  
小さければ、それを入力する

## 利用例



連続する数値の中で最小値を見つける

第2インレットに受け取られた数値は、常に新しい最小値を設定する

## 参照

**minimum**

**Peak**

<

数値のリストの中で最も小さい値を出力する

数値が直前の値より上回る時に出力する

2つの数値を比較し、より小さいことを判断する





## 入力

**bang** **ubutton**オブジェクトは2つのモードで操作することができる。**ubutton**がボタン・モード（デフォルトのモード）であれば、インレットに**bang**を受け取ると、瞬間的にハイライト状態になり、第1アウトレットから**bang**を出力する。**ubutton**がトグル・モードであれば、インレットに**bang**を受け取ると、ハイライト状態になり、第2アウトレットから**bang**を出力する。すでにハイライト状態なら、ハイライト状態を解除し、第1アウトレットから**bang**を出力する。

任意のシンボル **bang**に変換される。

(マウス) ボタン・モードでは、マウス・クリックによって**ubutton**はハイライト状態になり、マウス・ボタンを押している間、ハイライト状態を続ける。マウス・ボタンを押した時に、第2アウトレットから**bang**を出力し、マウス・ボタンを離れた時に、第1アウトレットから**bang**を出力する。トグル・モードでは、マウス・クリックは**bang**の入力と同じように動作する。マウスをクリックした時に、マウス・カーソルが**ubutton**オブジェクトの領域内であれば、**ubutton**は第4アウトレットから**1**を出力し、領域内になければ**0**を出力する。これらのメッセージは、マウス・ボタンを離れた時も、同じよう出力される。**ubutton**がTrack Mouse While Draggingモードであれば、クリック後にマウス・ボタンを押している間中、これらのメッセージを連続して出力する。

**stay** 0でない数値が続く**stay**メッセージは、**ubutton**をボタン・モードに設定し、指定した数値を待つように設定する。その数値をインレットに受け取れば、出力は行われませんが、**ubutton**はハイライト状態になり、他のメッセージを受け取るか、マウス・クリックされるまでハイライト状態を続ける。**stay 0**メッセージは、**ubutton**を通常のボタン・モードに設定し、特定の数値を待たなくなる。

**整数** **ubutton**が特定の数値（Stay-on Valueと呼ばれる）を待っている時、受け取った数値が一致すれば、**ubutton**はハイライト状態になるが、出力は行わない。受け取った数値が一致しない場合、**ubutton**はハイライト状態を解除する。Stay-on Valueが**0**であれば、整数は**bang**と同じように扱われる。

**実数** 整数に変換される。

**dragtrack** 0でない数値が続く**dragtrack**メッセージは、Track Mouse While Draggingモードを有効にする。このモードでは、位置と内側であるか外側であるかを示すメッセージ（先のマウスの項を参照のこと）を、マウス・ボタンをクリックして、マウス・ボタンを押している間、連続的に出力する。**dragtrack 0**メッセージは、このモードを解除する。デフ



オルトでは、このモードは解除されている。このモードは**ubutton**のインスペクターでも設定できる。

- set** **ubutton**がトグル・モードであれば、**set 1**メッセージは**ubutton**のトグルを入れ（ハイライト状態にする）、**set 0**メッセージはトグルを切る（ハイライト状態を解除する）。**set**に他の整数が続く場合は、*Stay-on Value*と比較される数値として**ubutton**は扱う。この場合、出力は行われない。
- toggle** 0でない数値が続く**toggle**メッセージは、**ubutton**をトグル・モードにする。**toggle 0**メッセージは、**ubutton**をボタン・モードにする。

## インスペクター

**ubutton**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**ubutton**オブジェクトを選択することで、**ubutton**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**ubutton**オブジェクトを選び、ObjectメニューからGet Info...を選択することでも、インスペクターを表示することができる。

**ubutton**のインスペクターでは、*Button Mode*（デフォルト）か*Toggle Mode*を指定することができる。*Highlight When Clicked*のチェック・ボックスは、**ubutton**オブジェクトのマウスに対する動作を設定する。*Track Mouse While Dragging*のチェック・ボックスは、マウス・カーソルの位置に関する出力を設定する（*dragtrack*メッセージを参照のこと）。*Stay-on Value*に0でない数値を入力すれば、**ubutton**がボタン・モードにおいて待機する数値を指定することになる。トグル・モードを選択するためには、*Stay-on Value*を**0**に設定する必要がある。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

なし



## 出力

**bang** 第1アウトレット：（*Stay-on Value*が0の時の）ボタン・モードでは、どのような入力があっても、**ubutton**は一瞬ハイライトし、第1アウトレットから**bang**を出力する。マウス・ボタンを離れた時も、第1アウトレットから**bang**を出力する。

**ubutton**オブジェクトがトグル・モードで、すでにハイライト状態であれば、どのような入力があっても、**ubutton**はハイライト状態を解除し、第1アウトレットから**bang**を出力する。

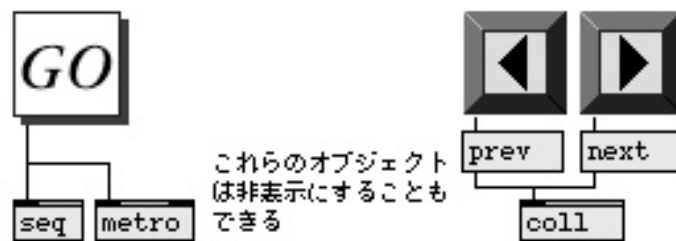
**bang** 第2アウトレット：（*Stay-on Value*が0の時の）ボタン・モードでは、**ubutton**オブジェクトをマウスでクリックすれば、マウス・ボタンを押した時に、第2アウトレットから**bang**を出力する。

**ubutton**オブジェクトがトグル・モードであれば、どのような入力があっても、**ubutton**オブジェクトはハイライト状態になり、第2アウトレットから**bang**を出力する。すでにハイライト状態であれば、ハイライト状態を解除するが、**bang**は出力しない。

**リスト** 第3アウトレット：マウス・ボタンをクリックしたり、離れた時に、**ubutton**オブジェクトは、マウス・カーソルの位置を示す2つの数値のリストを出力する。この位置は、**ubutton**オブジェクトの領域の左上隅からのオフセットとして、ピクセル単位で表される。インスペクターや**dragtrack**メッセージによって、**ubutton**オブジェクトがTrack Mouse While Draggingモードであれば、マウス・ボタンを離すまで、マウスの動きに応じて、新しい位置を出力する。

**整数** 第4アウトレット：マウス・ボタンをクリックしたり、離れた時に、マウス・カーソルが**ubutton**のオブジェクトの領域内であれば、アウトレットから1を出力する。インスペクターや**dragtrack**メッセージによって、**ubutton**オブジェクトがTrack Mouse While Draggingモードであれば、マウス・ボタンを押している間に、マウス・カーソルが**ubutton**オブジェクトの領域外に移動すれば、0を出力する。

## 利用例



コメントやピクチャの上に**ubutton** オブジェクトを配置すれば、それらはマウス・クリックに反応する



## 参照

|                   |                                        |
|-------------------|----------------------------------------|
| <b>bangbang</b>   | 順番に複数の <b>bang</b> を出力する               |
| <b>button</b>     | 任意のメッセージによって動作し、 <b>bang</b> を送信する     |
| <b>fpic</b>       | 画像ファイルより画像を表示する                        |
| <b>led</b>        | 色によって、オン/オフの状態を表示する                    |
| <b>matrixctrl</b> | マトリックス状のスイッチ・コントロール                    |
| <b>pictctrl</b>   | ピクチャーを用いたコントロール                        |
| <b>radiogroup</b> | ラジオ・ボタンやチェック・ボックスのユーザー・インターフェース・オブジェクト |

Tutorial 19      Screen aesthetics (英文PDF参照)



## 入力

- 整数** 数値はメニュー項目を出力し、その項目を**umenu**に表示する。メニュー項目の番号は0から始まる。
- メニュー項目は、ポップアップ・メニューとしてマウスで選択することができる。
- append** **append**に続くメッセージによって、メニューの最後に新しい項目を追加することができる。
- autosize** **autosize**に続く**1**か**0**によって、最も長い項目の横幅に合わせてポップアップ・メニューのサイズを変更する機能を設定または解除する。**autosize**がオフであれば、メニューの横幅はオブジェクトの領域自体の長さになる。
- bang** 現時点で表示されているメニュー項目を出力する。
- brgb** **brgb**に続く0から255までの3つの数値によって、**umenu**オブジェクトの色をRGBフォーマットで設定する。デフォルト値は、211 211 211である。
- checkitem** **checkitem**の後にメニュー番号と**1**か**0**の数値が続き、**1**ならばメニュー項目にチェック・マークを付け、**0**ならチェック・マークを外す。
- clearchecks** **clearchecks**は、すべてのメニュー項目のチェック・マークを外す。
- clear** すべてのメニュー項目を**umenu**から消去する。
- color** **color**に続く0から15までの数値によって、テキストの色を設定する。この数値は、標準でプリセットされている色の番号である。
- delete** **delete**に続く数値によって、その番号のメニュー項目を**umenu**から削除する。
- evalitemtext** **evalitemtext**に続く**1**か**0**によって、Evaluate Item Textモードのオン・オフを切り替える。このモードがオンであれば、メッセージやマウス・クリックによってメニューの値が変わった時に、そのアイテムのテキストによるメッセージを第2アウトレットから出力する。
- frgb** **frgb**に続く0から255までの3つの数値によって、**umenu**オブジェクトのテキストの色をRGBフォーマットで設定する。デフォルト値は、0 0 0である。
- labelclick** **labelclick**に続く**1**か**0**によって、Label Clickモードのオン・オフを切り替える。このモードがオンであり、オブジェクトがLabelモードであれ





ば、オブジェクトの領域をクリックすることで、現在のメニューの値を第1アウトレットから出力することができる。この場合、現在のメニュー項目のテキストはアンダーライン付きで表示される。

- maxitems** **maxitems**に続く数値によって、**umenu**のメニュー項目の最大数を設定する。これは、**umenu**オブジェクトのインスペクター（下記のインスペクターの項目を参照のこと）の*Maximum number of items*の設定と同じである。
- mode** **mode**に続く1、2、3の数値によって、**umenu**の表示方法と動作を設定する。**mode 1**は通常のポップアップ・メニュー方式であり、**mode 2**ではスクロール・メニュー方式、**mode 3**はメニューの代わりにラベルを表示する。この設定は、**umenu**オブジェクトのインスペクター（下記のインスペクターの項目を参照のこと）での*Mode*の設定と同じである。
- rgb2** **rgb2**に続く0から255までの3つの数値によって、**umenu**オブジェクトのメニュー項目の枠の上側の明るい色（3Dメニュー項目のハイライト部分）をRGBフォーマットで設定する。デフォルト値は、255 255 255である。
- rgb3** **rgb3**に続く0から255までの3つの数値によって、**umenu**オブジェクトのメニュー項目の枠の上側の暗い色（3Dメニュー項目のシャドウ部分）をRGBフォーマットで設定する。デフォルト値は221 221 221である。
- rgb4** **rgb4**に続く0から255までの3つの数値によって、**umenu**オブジェクトのメニュー項目の枠の下部の明るい色（3Dメニュー項目のハイライト部分）をRGBフォーマットで設定する。デフォルト値は170 170 170である。
- rgb5** **rgb5**に続く0から255までの3つの数値によって、**umenu**オブジェクトのメニュー項目の枠の下部の暗い色（3Dメニュー項目のシャドウ部分）をRGBフォーマットで設定する。デフォルト値は119 119 119である。
- rgb6** **rgb6**に続く0から255までの3つの数値によって、**umenu**オブジェクトのメニュー項目の角のドットの色をRGBフォーマットで設定する。色がついたオブジェクトやパネルを背景にして**umenu**のオブジェクトを使う場合、背景のオブジェクトの色に合う色を設定する必要がある。デフォルト値は187 187 187である。
- set** **set**に続く数値またはシンボルによって、**umenu**が表示するメニュー項目を指定する。ただし、出力は行わない。**set**のアーギュメントがシンボルであれば、そのシンボルで始まるメニュー項目を探して表示する。
- setcheck** Macintoshのみ。**setcheck**に続く0から255までの数値によって、チェック・マークとして用いる文字を設定する。**setcheck 0**はデフォルトの文字を使うように設定する。



- setitem** **setitem**に続くメニュー項目の番号とメッセージによって、そのメニュー項目の内容を設定する。
- setrgb** **setrgb**に続く0から255までの6つの数値によって、最初の3つの数値は前景（テキスト）の色を、残りの3つの数値は背景の色（オブジェクトの色）をRGBフォーマットで設定する。
- showchecked** 現在表示しているメニュー項目にチェック・マークが付いていれば、何も変化しない。そうでなければ、チェック・マークが付いている最初の項目を表示する。どの項目もチェック・マークが付いていなければ、何も変化しない。
- シンボル** シンボルのアーギュメントが続く**set**メッセージと同じであるが、シンボルに一致した項目の番号を出力する（Evaluate Item Textモードであれば、その項目のテキストを第2アウトレットから出力する）。

## インスペクター

**umenu**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**umenu**オブジェクトを選択することで、**umenu**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**umenu**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

*Menu Text*にはメニューに表示する項目を、コンマで区切って入力する。*Maximum Items*はメニュー項目の最大数を指定する。32から2000までのメニュー項目を持つことができる（デフォルト値は64）。ポップアップ・メニューの*Mode*では、**umenu**オブジェクトの表示方法とユーザー・インターフェースとしての動作を指定することができる。*Normal*（デフォルト）モードは、クリックしている間に、すべてのメニュー項目を一覧として見ることができる標準的なポップアップ・メニューである。*Scrolling*モードでは、一度には一つの項目しか表示しないが、マウスを上下にドラッグすることで個々のメニュー項目をスクロールすることができる。*Label*モードは、周囲に境界線を表示せずに、選択されたメニュー項目のテキストを表示し、マウスには反応しない。*Auto Size*をチェックすれば、**umenu**のオブジェクトのポップアップ・メニューの横幅は、最も長いメニュー項目の横幅に合うように調節される。*Evaluate Item Text*をチェックすれば、メニュー項目を選択した時に、そのテキストを第2アウトレットから出力する。

*Color*のオプションでは、**swatch**オブジェクトによるカラー・ピッカーやRGBの数値によって、**umenu**のテキストや背景の色を指定できる。*Texts*は表示するメッセージの色を設定し（デフォルト値は0 0 0）、*Background*は、選択肢として表示するメッセージの背景色を設定する





(デフォルト値は221 221 221)。Upper Frame Light、Upper Frame Dark、Lower Frame light、Lower Frame Darkは、メニュー項目の枠にあるハイライト部分とシャドウ部分の色を設定する。デフォルト値は、枠の上部のハイライト部分は255 255 255、枠の上部のシャドウ部分は221 221 221、枠の下部のハイライト部分170 170 170、枠の下部のシャドウ部分は119 119 119である。Corner Dotsはメニュー項目の角のドットの色を設定する。デフォルト値は187 187 187である。

Revertボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

**umenu**のフォントと大きさはFontメニューで変えることができる。

## アーギュメント

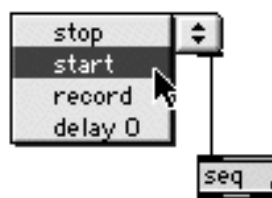
なし

## 出力

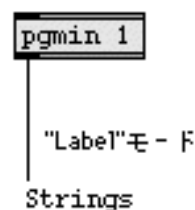
**整数** 第1アウトレット：選択されたメニュー項目の番号を出力する。メニュー項目の番号は0から始まる。

**任意のメッセージ** 第2アウトレット：Evaluate Item Textモードであれば、選択されたメニュー項目のテキストをメッセージとして出力する。

## 利用例



コマンドを送るために使用する



または、受け取った数値と一致するテキストを表示する

## 参照

**coll** 様々な種類のメッセージを格納し、編集する

Tutorial 37 Data structures (英文PDF参照)

## 入力

クラス・シンボル **universal** オブジェクトは、オブジェクト・クラスの名前（例えば、**table**や**dspstate~**など）に続いて、メッセージの指定と、そのメッセージのアーギュメントとなる任意の数値を入力することを想定している。このメッセージと（もしあれば）アーギュメントは、同じパッチ（および、そのサブ・パッチ）内の、指定したクラスのすべてのインスタンスへ送信される。

**sendmessage** 予約されているMaxメッセージ名（**int**や**float**のような）をクラス名として持つオブジェクトにメッセージを送る場合は、そのメッセージを**sendmessage**で始める必要がある。**sendmessage**メッセージは、任意のクラスに使うことができる。

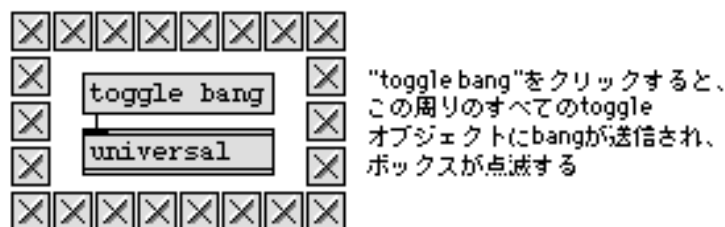
## アーギュメント

**int** オプション。アーギュメントとして1が指定されている場合には、**universal** オブジェクトは、受け取ったメッセージを、その**universal** オブジェクトを含んでいるパッチと同じように、サブ・パッチ内のクラスのオブジェクトにも送信する。

## 出力

なし。このオブジェクトはアウトレットを持たないが、このオブジェクトが送信するメッセージを受信するオブジェクトは、それらのアウトレットから何らかの出力を生じることになる。ただし、メッセージがオブジェクトに送られる順序は確定できない。このことは、**send** オブジェクトや**receive** オブジェクトを用いた場合でも、同じである。

## 利用例



同じクラスのすべてのオブジェクトに、一度にメッセージを送信する

## 参照

- forward** 様々なオブジェクトにパッチ・コードなしでメッセージを送信する
- receive** パッチ・コードなしでメッセージを受信する
- send** パッチ・コードなしでメッセージを送信する
- value** 他のオブジェクトと格納されている値を共有する

## 入力

- リスト** このリストの（アウトレットの数までの）項目は、リスト内の位置に対応するアウトレットから出力される。
- 整数** この数値は第1アウトレットから出力される。
- 実数** 第1アーギュメントに実数を指定している場合以外は、整数に変換される。この数値は第1アウトレットから出力される。
- シンボル** このシンボルは第1アウトレットから出力される。第1アーギュメントにシンボルを指定している場合以外は、第1アウトレットから0が出力される。
- bang** 格納されているリストの項目を、対応しているアウトレットから出力する。

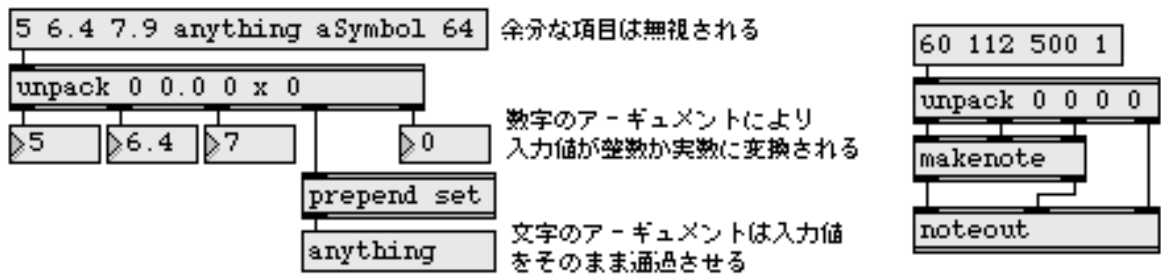
## アーギュメント

- 任意** オプション。アウトレットの数はアーギュメントの数によって決定される。アーギュメントは、整数、実数、シンボルの任意の組み合わせとして指定できる。整数や実数のアーギュメントに対応するアウトレットでは、入力された項目は、必ず常に数値の型に変換される。アーギュメントでシンボルが指定されたアウトレットは、入力された項目に変化させずに出力する。アーギュメントが指定されていない場合は、**unpack**オブジェクトは整数のアウトレットを2つ持つ。

## 出力

- 整数** インレットに受け取ったリストの項目は、対応するアウトレットから出力される。リストの1つ目の項目は第1アウトレットから出力され、以下それに続くよう出力される。実数やシンボルは、アーギュメントで整数が指定されたアウトレットから出力される時に、整数に変換される（シンボルは整数の0へと変換される）。
- 実数** リスト内の整数やシンボルは、アーギュメントで実数が指定されているアウトレットから出力される時に、実数に変換される（シンボルは実数の0.へと変換される）。
- シンボル** 入力されたリスト内のシンボルは、アーギュメントでシンボルが指定されたアウトレットから、対応する順序で出力される。整数や実数が指定されたアウトレットからは、シンボルは0または0.0へと変換されて、出力される。

## 利用例



リスト内の項目は、異なる場所へと出力される

## 参照

**iter**  
**pack**  
**zl**

リストを数値ごとに分解する  
数値やシンボルをひとつのリストに結合させる  
多目的なリスト処理

Tutorial 30

Number groups (英文PDF参照)

## 入力

- bang** 第1インレット：0から指定された最大値より1小さい数までの、すでに出力されていない数値を出力する。
- clear** 第1インレット：出力された数値のリストを消去する。
- 整数** 第2インレット：すでに出力された数値のリストを消去して、乱数ジェネレーターの最大値を指定する。出力される数値は、0から最大値より1小さい数までの範囲である。
- seed** 第1インレット：**seed**メッセージに続く数値で、乱数ジェネレーターのシード値を設定する。それは、疑似乱数の特定の（再現可能な）系列を発生させる。数値が**0**であれば、システム起動時からの経過時間（これは予測不可能な値と考えられる）をシード値として設定する。**urn**オブジェクトが作成される時は、このシード値を初期値として用いる。

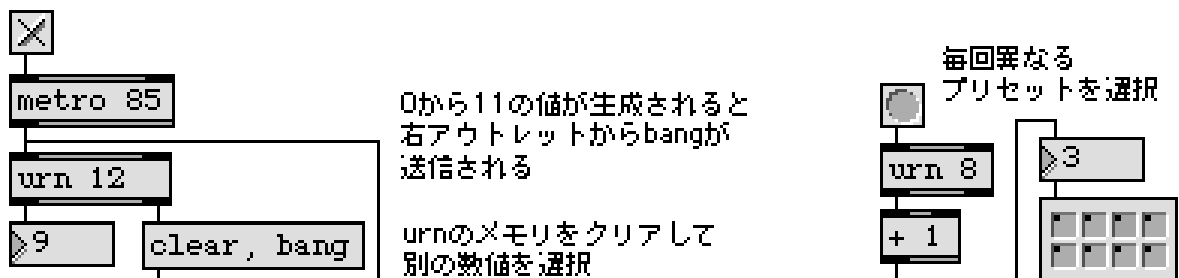
## アーギュメント

- 整数** オプション。乱数ジェネレーターが取り得る値の数を指定する。アーギュメントを指定しない場合は、取り得る値の数は**1**である。

## 出力

- 整数** 第1アウトレット：**clear**メッセージを受け取った以降に、現在設定されている範囲で、まだ出力されていない数値があれば、**urn**オブジェクトは、0から最大値より1小さな数までのランダムな数値を出力する。
- bang** 第2アウトレット：現在設定されている範囲のすべての数値が出力されれば、**urn**オブジェクトは第1アウトレットから数値を出力する代わりに、第2アウトレットから**bang**を出力する。

## 利用例



繰り返しのないランダムを発生させなさい

## 参照

**decide**  
**drunk**  
**random**

オンあるいはオフ (1あるいは0) をランダムに選択する  
ある範囲を移動するようなランダムな数値を出力する  
ランダムな数値を発生させる



## 入力

- 整数** 数値が入力されれば、グラフィカルに **uslider** オブジェクトが動き、アウトレットからその数値が出力される。オプションとして、**uslider** オブジェクトは、アウトレットから出力する前に、数値に一定数値を掛け算し、オフセットを足し算することができる。
- (マウス) **uslider** オブジェクトは、マウスのクリックやドラッグによって、数値を出力する。
- 実数** 整数に変換される。
- bang** **uslider** オブジェクトに格納されている現在の数値を出力する。
- color** **color** メッセージに続く0から15までの数値によって、**uslider** オブジェクトの中央部分の色を指定する。Objectメニューの**Color**から選ぶことでも、色を指定することができる。
- local** **local** メッセージに続く0以外の数値によって、**uslider** オブジェクトはマウス操作が可能になる（デフォルト）。**local 0** なら、マウス操作は不可能になり、インレットからの入力にのみ応じる。
- min** **min** メッセージに続く数値で、**uslider** オブジェクトが出力する数値の最小値を指定する。デフォルト値は0である。
- mult** **mult** メッセージに続く数値によって、**uslider** オブジェクトから出力する数値に対して掛け算を行なう乗数を指定する。掛け算は、オフセットの足し算の前に行われる。乗数のデフォルト値は1である。
- resolution** **resolution** メッセージに続く数値によって、**uslider** オブジェクトから出力する数値の時間間隔をミリ秒単位で指定する。これは、**uslider** オブジェクトの表示を更新する時間間隔でもある。
- set** **set** メッセージに続く数値によって、**uslider** オブジェクトの値と表示を変えるが、出力は行わない。
- size** **size** メッセージに続く数値によって、**uslider** オブジェクトのサイズ（範囲）を指定する。このデフォルト値は128である。サイズを1に設定すると、**uslider** オブジェクトのスライダーは動かなくなる（1つの値を表示するだけであるため）。1未満の数値を設定しても、サイズは2になる。

## インスペクター

**uslider** オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**uslider** オブジェクトを選択することで、**uslider** オブジ





エクトのインスペクターをフローティング・ウィンドウに表示することができる。**uslider**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

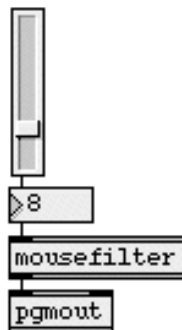
**uslider**オブジェクトが受け取る数値は、0からインスペクターのSlider Rangeで設定した値より1小さい数値までに制限される。この設定値のデフォルト値は128である。Offsetには、掛け算の後に足し算する数値を設定する。このデフォルト値は0である。Multiplierでは、出力する数値に掛け算をする数値を設定する。掛け算は、オフセット値を足し算する前に行われ、そのデフォルト値は1である。

Revertボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

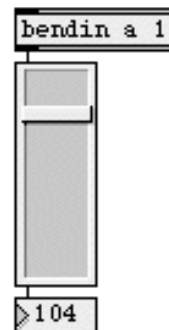
## 出力

**整数** インレットに受け取った数値や、マウスのクリックやドラッグによって設定された数値は、最初に乗数を掛け算し、次にオフセット値を足し算した上で、アウトレットから出力される。

## 利用例



画面上をドラッグして出力する



または、通過する数値を表示する

## 参照

- dial** ダイアル操作によって数値を出力する
- hslider** 画面上のスライダーを動かすことによって数値を出力する
- kslider** 画面上のキーボードから数値を出力する
- pictctrl** ピクチャーを用いたコントロール
- pictslider** ピクチャーを用いたスライダー・コントロール
- rslider** 数値の範囲を変更し、それを表示する
- slider** 画面上でスライダーを動かすことによって数値を出力する

## 入力

- bang** 第1インレット：次から次へと、**bang**メッセージを一気に出力させる。出力される**bang**メッセージの数は、インレットに受け取った最後の数値によって決まる。
- 整数** 第1インレット：**bang**メッセージを出力する数を設定し、その数だけ、次から次へと、一気に**bang**メッセージを出力させる。
- 第2インレット：**bang**メッセージを出力する数を設定する。**bang**メッセージは出力しない。
- pause** 第1インレット：**Uzi**オブジェクトの出力を途中で止める（**Uzi**オブジェクトは一気に**bang**を出力するので、**Uzi**オブジェクト自身の出力に従って何らかの方法で、このメッセージを出力しなければならない）。**Uzi**オブジェクトは、何個の**bang**を出力したかを記憶しているので、**pause**メッセージによって、**Uzi**オブジェクトの出力を途中で止めた後に、**resume**メッセージや**continue**メッセージによって、残りの**bang**メッセージを出力することができる。
- break** **pause**と同じである。
- resume** 第1インレット：**Uzi**オブジェクトが、**pause**メッセージによって、出力を途中で止められている場合に、残りの**bang**を出力する。
- continue** **resume**メッセージと同じである。

## アーギュメント

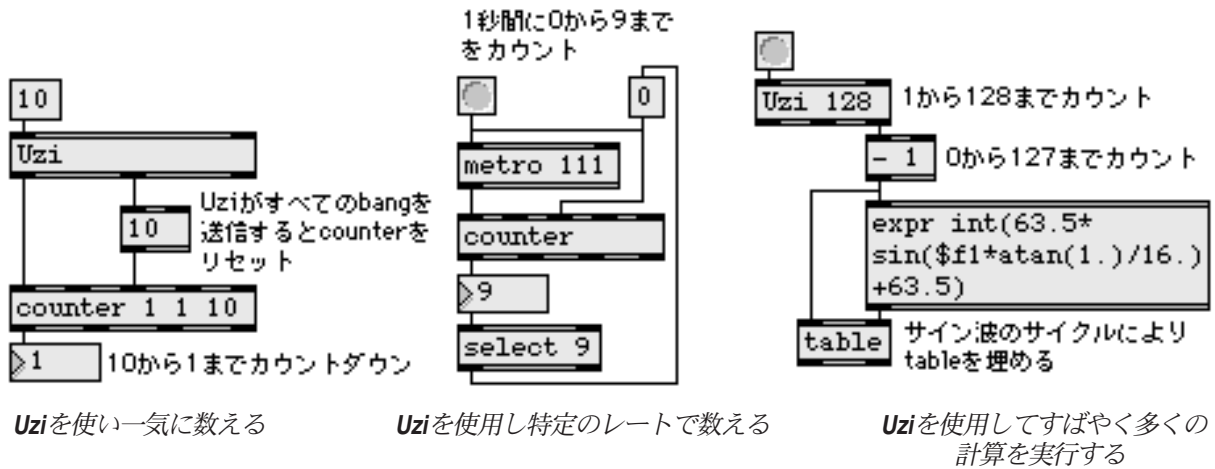
- 整数** オプション。**bang**を出力する個数の初期値を指定する。アーギュメントに何も指定しない場合は、**Uzi**オブジェクトは1つしか**bang**を出力しない。

## 出力

- bang** 第1アウトレット：**Uzi**オブジェクトに**bang**または整数を入力された時に、指定された数だけ**bang**メッセージを一気に出力する。出力する**bang**の数は、第1インレット、第2インレットに入力された数値によって決まる。
- 第2アウトレット：最後の**bang**が出力された時に、第2アウトレットから**bang**が出力される。これは**counter**オブジェクトの**carry**アウトレットに似ていて、すべての**bang**が送られたという合図として使用できる。

**整数** 第3アウトレット：出力されるbangの番号を出力する。整数またはbangを第1インレットに受け取る度に、出力されるbangには、1から始まる番号が付けられる。Uziオブジェクトがresumeメッセージまたはcontinueメッセージによって出力を再開する時には、最後に出力された番号の次から出力が始まる。

## 利用例



## 参照

**counter**  
**line**  
**metro**

bangメッセージをカウントし、回数を入力する  
ある値から異なる値へと、補完しながら数値を連続して出力する  
一定の時間間隔でbangメッセージを出力する

## 入力

任意のメッセージ メッセージは格納され、他のすべての同じ名前の**value**オブジェクトと共有される。それは、他のパッチにあって構わない。同じ名前を持つ他の**value**オブジェクトが受け取ったメッセージによっても、格納されている内容が変化する。

**bang** 格納されているメッセージを出力する。

(マウス) **value**オブジェクトをダブルクリックすると、同じ名前の**value**オブジェクトを持つすべてのパッチ・ウィンドウが開く。

## アーギュメント

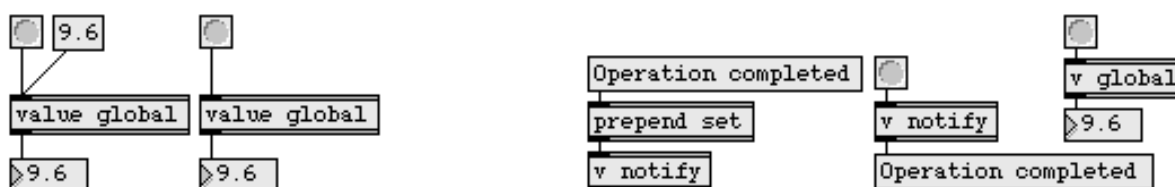
**symbol** 必須。**value**オブジェクトの名前を決める。

任意のメッセージ オプション。第2アーギュメントとして、**value**オブジェクトに格納するメッセージを指定する。アーギュメントを指定しなければ、**value**オブジェクトの内容は空である。

## 出力

任意のメッセージ **bang**をインレットに受け取ると、格納しているメッセージを出力する。

## 利用例



1つの値または任意のタイプのメッセージが、同じ名前を持つすべての**value**の間で共有される

## 参照

**float** 実数を格納する  
**int** 整数を格納する  
**pv** パッチとそのサブ・パッチ内に限定して、変数を共有する  
**send** パッチ・コードなしでメッセージを送信する  
**receive** パッチ・コードなしでメッセージを受信する

**vdp**オブジェクトは、Pioneer 4200または8000 standardと互換性のある、シリアル制御に対応したビデオディスク・プレーヤーを扱う（これらは販売が終了している）。**vdp**オブジェクトが受け取ったそれぞれのコマンドは、オブジェクトの第1アウトレットから数値のストリームとして出力されるが、これは**serial**オブジェクトに入力することを意図している。後述するそれぞれのコマンドについての説明は、コマンドがプレーヤーに対してどのような効果を持つかについて述べており、**vdp**によって出力される文字列ストリームそのものに関してではない。

ビデオディスク・プレーヤーは、バッファがほとんどないシリアル・インターフェースを持っている。このため、**vdp**オブジェクトは受け取ったそれぞれのコマンドをキューに入れ、プレーヤーが直前のコマンドの処理を終えた時だけに出力する。この仕様により、**vdp**オブジェクトにコマンドが送られてから、シリアル・ポートに実際に出力されるまでの間に時間的な遅れが生じるかもしれない。

第2インレットに受け取った任意のメッセージは、第1インレットで受け取った場合と、まったく同じように振る舞うが、コマンドはキューの先頭に格納され、プレーヤーに対して次に出力されるコマンドになる。

## 入力

- clear** 第1インレット：待機中になっているコマンドをキューから取り除き、オブジェクトをリセットする。
- control** 第1インレット：1つの数値を伴う**control**は、ビデオディスク・プレーヤーに、次の操作の内の1つを実行するように伝える。

### 数値 操作

- |    |                   |
|----|-------------------|
| 0  | プレーヤーを初期化し、リセットする |
| 1  | ディスクをイジェクトする      |
| 2  | オーディオをオフにする       |
| 3  | オーディオ1をオンにする      |
| 4  | オーディオ2をオンにする      |
| 5  | ステレオをオンにする        |
| 6  | 画面をオンにする          |
| 7  | 画面をオフにする          |
| 8  | フレーム数の表示をオンにする    |
| 9  | フレーム数の表示をオフにする    |
| 11 | フレーム・アクセス・モード     |
| 12 | タイム・アクセス・モード      |
| 13 | チャプター・アクセス・モード    |

- fps** 第1インレット：再生速度を設定する。**fps**メッセージは、1つの数値（1秒あたりのフレーム数）または1つの単語を伴う。これらの単語と数値は、（少なくともPioneer 4200に関しては）同じ意味を持つ。

|               |                                                                                                                                                                                                                                                                |     |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
|               | slowest                                                                                                                                                                                                                                                        | 1   |
|               | slower                                                                                                                                                                                                                                                         | 10  |
|               | slow                                                                                                                                                                                                                                                           | 15  |
|               | normal                                                                                                                                                                                                                                                         | 30  |
|               | fast                                                                                                                                                                                                                                                           | 60  |
|               | faster                                                                                                                                                                                                                                                         | 90  |
|               | fastest                                                                                                                                                                                                                                                        | 120 |
| <b>frame</b>  | 第1インレット：プレーヤーに現在のフレーム番号を問い合わせ、第3アウトレットから（第2インレットで受け取った）結果を出力する。                                                                                                                                                                                                |     |
| <b>play</b>   | 第1インレット：アーギュメントがない場合には、 <b>play</b> は現在の位置から現在の速度で、ディスクの最後まで（またはプレーヤーが他のコマンドを受け取るまで）の再生を開始する。1つのアーギュメント（フレーム番号）が与えられた場合には、 <b>play</b> は指定されたフレーム番号を検索し、ディスクの最後に向かって再生を開始する。2つのアーギュメントが与えられた場合には、 <b>play</b> は最初の数値によって指定されたフレーム番号を検索し、2番目のフレーム番号に到達するまで再生する。 |     |
| <b>整数</b>     | 第1インレット：指定されたフレーム番号からディスクの最後まで再生する。                                                                                                                                                                                                                            |     |
|               | 第2インレット： <b>vdp</b> は、 <b>serial</b> オブジェクトが受信するプレーヤーからの応答を、第2インレットに入力することを想定している。プレーヤーが受信したことを示す応答（改行文字を伴う文字 <b>R</b> ）を、 <b>vdp</b> が受け取ると、待機中のコマンドのキューから、次のコマンドを出力する。利用例に <b>vdp</b> オブジェクトと <b>serial</b> オブジェクトの接続方法を示す。                                  |     |
| <b>scan</b>   | 第1インレット：早送り（fast forward）または巻き戻し（rewind）操作を開始する。 <b>scan forward</b> であれば早送りとなり、 <b>scan backward</b> であれば巻き戻しを行う。                                                                                                                                             |     |
| <b>search</b> | 第1インレット：最初のアーギュメントは、検索するフレーム番号を指定する。オプションの2番目のアーギュメントは、0でなければ、検索中に画像を表示したままにするようにプレーヤーに指示する。現在の位置から大きく離れたフレームを検索する場合には、プレーヤーは画面を空白のままにすることができないかもしれない。プレーヤーが目的のフレームに到達すると、そのフレームの（静止）画像を表示する。                                                                  |     |
| <b>step</b>   | 第1インレット： <b>-1</b> を伴う <b>step</b> は（再生中であれば）プレーヤーを一時停止し、前のフレームを表示する。 <b>1</b> を伴う <b>step</b> は（再生中であれば）プレーヤーを一時停止し、次のフレームを表示する。                                                                                                                              |     |



- stop** 第1インレット：プレーヤーを一時停止する。
- cmd** 第1インレット：**cmd**メッセージは、ASCIIコードで構成されたプリミティブなコマンドを、プレーヤーに出力するために用いられる。コマンドは、数値アーギュメントに続き、通常2文字のコードから構成される。例えば、フレーム5000の検索は、cmd 5000 SEというメッセージになる。詳しくは、プレーヤーのオーナーズ・マニュアルを参照のこと。特にPioneer 8000プレーヤーは、**vdp**オブジェクトの通常のメッセージではサポートされない多くの特別な機能を持つので、**cmd**メッセージが役立つだろう。
- setskip** 第1インレット：1つの数値を伴う**setskip**は、現在のフレーム番号から**skip**メッセージを用いて（順方向または逆方向に）ジャンプするフレームの数を設定する。
- skip** 第1インレット：**-1**を伴う**skip**は、**setskip**メッセージで指定されたフレーム数だけ巻き戻す。**1**を伴う**skip**は、**setskip**メッセージで指定されたフレーム数だけ先に進める。

## アーギュメント

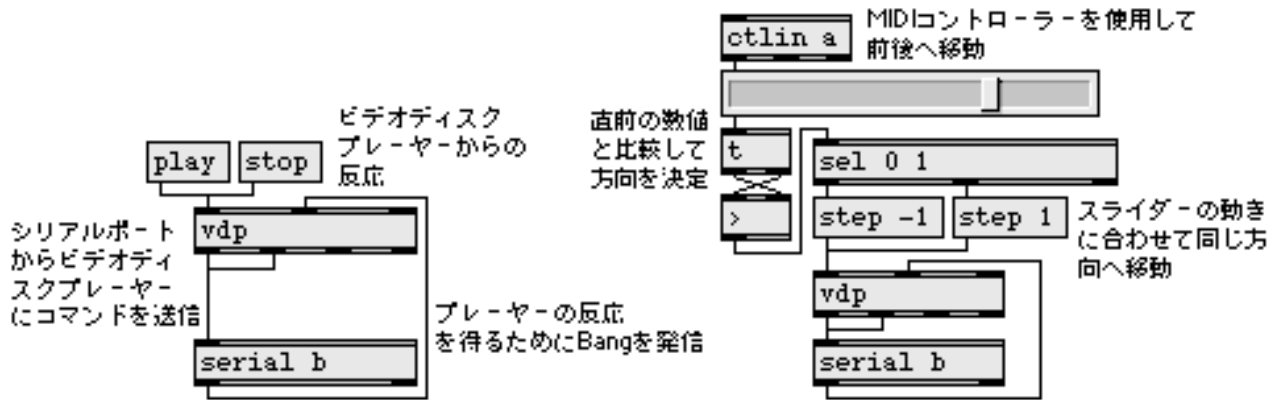
なし。

## 出力

- 整数** 第1アウトレット：それぞれのコマンドに対応した、ビデオディスク・プレーヤーへのコード化されたコマンドの文字列。これらの数値は、**serial**オブジェクトの第1インレットに入力して、シリアルポートから送信することを想定している。
- bang** 第2アウトレット：第1アウトレットからコマンドを出力した後、**vdp**はプレーヤーからの応答を待つために、このアウトレットから約20ミリ秒ごとに**bang**を出力することによって、**serial**オブジェクトの定期的な監視を始める。これは、**vdp**が第2インレットに受信したことを示す応答を受け取るまで続く（**serial**オブジェクトに送られた**bang**メッセージによって、シリアル・ポートから受け取った任意の文字が、**serial**オブジェクトのアウトレットから出力される）。
- 整数** 第3アウトレット：**frame**メッセージによって、プレーヤーから受け取った現在のフレーム番号。
- 整数** 第4アウトレット：未実装。



### 利用例



**vdp** オブジェクトと **serial** オブジェクトの基本的な組み合わせ

**slider** または **MIDI** コントローラーによるスクラブ

### 参照

**appledvd**  
**serial**

Apple DVD Playerアプリケーションを制御する  
シリアル・ポートとカードから文字を送受信する

**vexpr**オブジェクトは**expr**オブジェクトと同じような演算をリストに行なう。  
詳細については**expr**オブジェクトを参照のこと。

## 入力

**リスト** **vexpr**オブジェクトは、インレットにリストを受け取ることができる。リストのそれぞれの数値は、変数のアーギュメントに代入され数式によって左から右へ計算される。リストが第1インレットに入力される時、最初にリストの始めの項目を評価する。次にはそれぞれのリストの2番目の項目が評価される。そして、これらの評価の結果はリストとして出力される。

**整数または実数** 整数や実数は1つのリストとして入力される。

**bang** 第1インレット：最後に入力されたリストによる演算の結果を出力する。

## アーギュメント

必須。**expr**オブジェクトを参照。

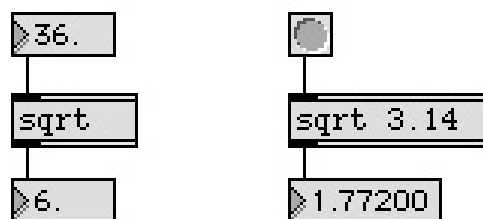
## 出力

**リスト** リストを第1インレットに受け取った時、**vexpr**オブジェクトそれぞれのインレットに受け取ったリストが変数のアーギュメントに代入され、リスト同士で演算をする。そして、その演算は最も短いリストの最後の項目が計算されるまで、同様に2番目の項目から処理される。そして、1つのリストになって出力される。

**整数** インレットにリストではなく1つの整数を受け取った場合は、演算の結果も1つの整数で出力される。

**実数** インレットに1つの実数を受け取った場合は、演算の結果も1つの実数で出力される。

## 利用例



入力値の全体のリストに同じ計算を実行する

## 参照

**expr** 数式の値を求める

Tutorial 38 **expr** and **if** (英文PDF参照)

## 入力

**整数** 数値は、**midin**オブジェクトや**seq**などから受け取る7ビット値のMIDIメッセージである。MIDIピッチ・ベンド・メッセージは**xbendin**オブジェクトによって認識され、ピッチ・ベンド値は高い精度で出力される。

## アーギュメント

**整数** オプション。ピッチ・ベンド・メッセージとして認識するMIDIチャンネル・ナンバーを指定する。アーギュメントを指定しない場合は、全てのチャンネルが認識され、右に追加されたアウトレットからチャンネル・ナンバーが出力される。

**xbendin2** オプション。通常、**xbendin**オブジェクトは、14ビットのピッチ・ベンド値を第1アウトレットから出力する。オブジェクト名を**xbendin2**とすると、アウトレットが追加され、ピッチ・ベンド値の上位バイトと下位バイトを、それぞれ第1アウトレット、第2アウトレットから出力する。

## 出力

**整数** 第1アウトレット：**xbendin**オブジェクトの時は14ビットでピッチ・ベンド値が出力される。**xbendin2**の時は上位バイトが出力される。

第2アウトレット：**xbendin**オブジェクトでアーギュメントにチャンネル・ナンバーを指定しない場合は、チャンネル・ナンバーが出力される。**xbendin2**の場合は、ピッチ・ベンド値の下位バイトが出力される。

第3アウトレット：**xbendin2**でアーギュメントにチャンネル・ナンバーを指定しない場合は、チャンネル・ナンバーが出力される。

## 利用例



ピッチ・ベンド・メッセージを14ビットで出力するか、あるいは上位バイトと下位バイトに分けて出力する

## 参照

- bendin** MIDIピッチ・ベンド・メッセージを受信する
- midin** 生のMIDIデータを受信する
- xbendout** MIDIピッチ・ベンド・メッセージ (14ビット) を作成する

## 入力

**整数** 第1インレット：14ビットのピッチ・ベンド値を入力すれば、**xbendout** オブジェクトは完全なMIDIピッチ・ベンド・メッセージにフォーマットする。

第2インレット：**xbendout** オブジェクトが出力するMIDIピッチ・ベンド・メッセージのMIDIチャンネル・ナンバーとして格納される。数値が16より大きい場合は、1から16までの範囲に変更される。

**リスト** リストの最初の値は、14ビットのピッチ・ベンド値で、2番目はチャンネル・ナンバーである。それぞれの数値は内部に格納され、MIDIピッチ・ベンド・メッセージとしてフォーマットされて、アウトレットから出力される。

**bang** 既に格納されているMIDIピッチ・ベンド・メッセージを送信する。

## アーギュメント

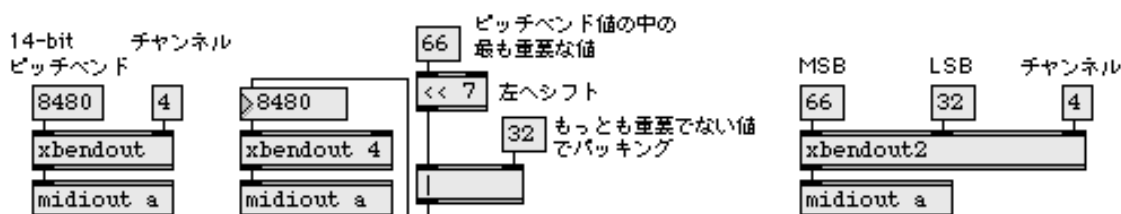
**xbendout2** オブジェクトの名前を**xbendout2**とすると、3つのインレットが作られ、第1インレットにはピッチ・ベンド・メッセージの上位バイト、第2インレットには下位バイトを入力する。

**整数** オプション。ピッチ・ベンド・メッセージのMIDIチャンネル・ナンバーの初期値を設定する。アーギュメントを指定しない場合の初期値は1である。

## 出力

**整数** ピッチ・ベンド値を第1インレットに受け取ると、完全なMIDIピッチ・ベンド・メッセージが、バイトごとにアウトレットから出力される。

## 利用例



14ビットのピッチ・ベンド値がMIDIメッセージとしてフォーマットされ、バイトごとに出力される

## 参照

|                |                                  |
|----------------|----------------------------------|
| <b>bendout</b> | MIDIピッチ・バンド・メッセージを送信する           |
| <b>midout</b>  | 生のMIDIデータを送信する                   |
| <b>xbendin</b> | MIDIピッチ・バンド・メッセージ (14ビット) を解釈する  |
| Tutorial 34    | Managing raw MIDI data (英文PDF参照) |

## 入力

**整数** **midin**オブジェクトから送られるMIDIデータの個々のバイトを入力する。ベロシティが0であるノート・オンは、ほとんどの場合、ノート・オフを示すものとして扱われる。しかし、**xnotein**オブジェクトでは、ノート・オフ・コマンドとそのリリース・ベロシティも認識する。

## アーギュメント

**整数** オプション。MIDIノート・オンとノート・オフのMIDIメッセージを検出するチャンネル・ナンバーを指定する。チャンネル・ナンバーが16より大きければ、1から16までの範囲に変更される。アーギュメントを指定しない場合は、**xnotein**オブジェクトは、全てのチャンネルのノート・オンとノート・オフのMIDIメッセージを認識し、第4アウトレットから、チャンネル・ナンバーを出力する。

## 出力

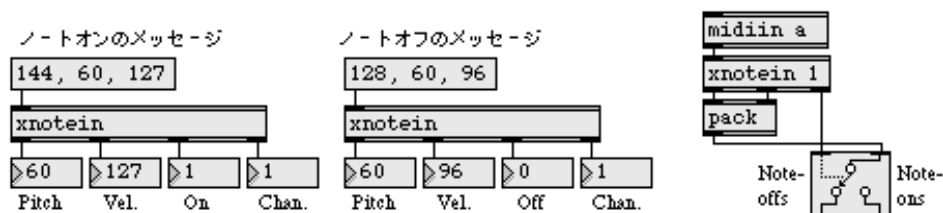
**整数** 第1アウトレット：ノート・オンまたはノート・オフのMIDIノート・ナンバーの値を出力する。

第2アウトレット：ノート・オンまたはノート・オフのベロシティを出力する。

第3アウトレット：ノート・オンの場合は1を出力し、ノート・オフまたはベロシティ0のノート・オンの場合は、0を出力する。

アーギュメントを指定しなければ、**xnotein**オブジェクトに第4アウトレットが作られ、入力されたメッセージのチャンネル・ナンバーを出力する。

## 利用例



キー・ダウンやキー・アップのベロシティとともに、ノート・オンとノート・オフのメッセージが解釈される

## 参照

- notein** MIDIノート・メッセージを出力する
- midin** 生のMIDIデータを受信する
- xnoteout** ノート・オフ・ベロシティを持つMIDIノート・メッセージを作成する



## 入力

- 整数** 第1インレット：ノート・オンとノート・オフのMIDIノート・ナンバーを入力し、MIDIメッセージをMIDIバイトごとに出力する。
- 第2インレット：数値はベロシティとして格納される。数値を受け取っていない場合は、ノート・オンのベロシティは**64**、ノート・オフのベロシティは**0**となる。
- 第3インレット：出力するメッセージがノート・オンであるかノート・オフであるかを区別する識別子として数値を格納する。数値が0以外の場合は、**xnoteout**オブジェクトはノート・オン・メッセージを出力する。**0**の場合は、**xnoteout**オブジェクトはノート・オフ・ベロシティ付きのノート・オフ・メッセージを出力する。何も数値を受け取っていないければ、初期状態は**1**（ノート・オン）である。
- 第4インレット：数値は**xnoteout**オブジェクトが出力するMIDIメッセージのチャンネルとして格納される。数値が16よりも大きい場合は、1から16までの範囲に収まるように変換される。
- 実数** 第1インレット：**xnoteout**オブジェクトは、実数を理解しない。
- その他のインレット：整数に変換される。
- リスト** リストの始めの値はMIDIノートナンバーである。2番目はベロシティである。3番目はノート・オン、ノート・オフインディケータ（ノート・オンは0以外、ノート・オフは**0**）である。4番目はチャンネル番号である。数値は**xnoteout**オブジェクトに格納され、MIDIノート・オンまたはノート・オフメッセージが出力される。
- bang** 現在**xnoteout**オブジェクトに格納されているMIDIメッセージを出力する。

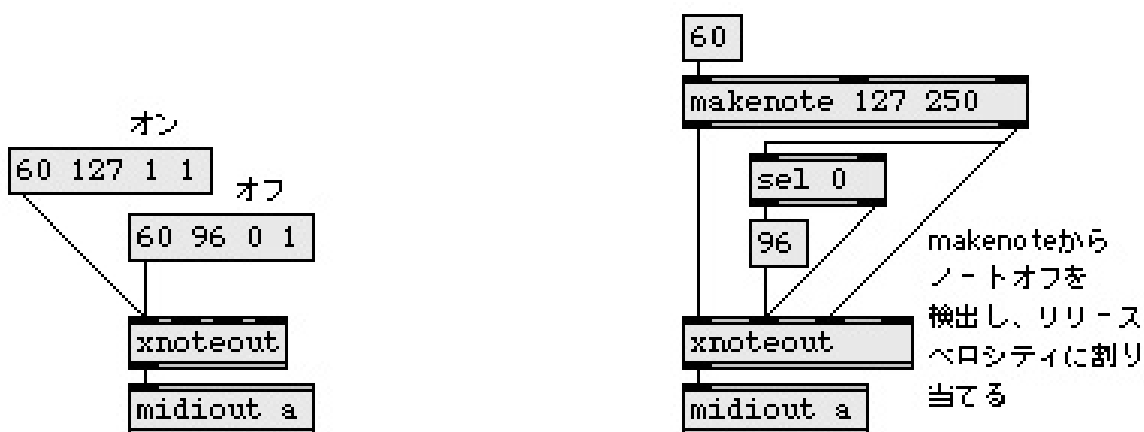
## アーギュメント

- 整数** オプション。メッセージが送信されるMIDIチャンネルの初期値を設定する。16より大きい数のチャンネル番号は1~16の範囲に変換される。アーギュメントに何も指定しなければ、デフォルトのチャンネル番号は1である。

## 出力

**整数** MIDIノート・ナンバーを受け取った時に、完全なMIDIノート・オン・メッセージ、またはノート・オフ・メッセージをMIDIバイトごとにアウトレットから出力する。ベロシティが0であるノート・オン・メッセージをノート・オフを示すために用いるのが一般的であるが、**xnoteout**はノート・オフのコマンドを出力し、ノート・オフ・ベロシティも出力する。

## 利用例



数値は MIDI ノート・オンまたはノート・オフ・メッセージにフォーマットされ、MIDI バイトごとに出力される

## 参照

- noteout** MIDIノート・メッセージを送信する
- midiout** 生のMIDIデータを送信する
- xnotein** ノート・オフ・ベロシティ付きのMIDIノート・メッセージを解釈する

Tutorial 34 Managing raw MIDI data (英文PDF参照)

**zl**オブジェクトはいくつかの種類のリスト処理関数を実行する。キーワードとなるアーギュメントによって機能を設定でき、また、**mode**メッセージによって実行する機能を変更できる。**zl**オブジェクトのインレットやアウトレットの振る舞いや、期待されるあるいは処理されるメッセージのタイプは、**zl**オブジェクトのモードによって異なる。続く説明で簡潔になるように、任意のMaxメッセージ、整数、シンボルおよび実数のような単一の要素、およびシンボルから始まるメッセージをリストと呼ぶ（Maxでのリストは数から始まるメッセージである）。

## 入力

**mode** **mode**には、**group**、**iter**、**join**、**len**、**reg**、**rev**、**rotate**、**sect**、**slice**、**union**、これらのうち1つのシンボルが続き、**zl**オブジェクトの現在のモードを設定する。いくつかのモードでは、設定されたモードの機能を指定するために第1インレットに受け取られたリストをアーギュメントとして扱う。出力するのに十分な長さでないメッセージの項目は、格納されたリストに加えられる。そして、格納されたリストが十分な長さを持った時、第1アウトレットから出力される。

**mode group**はリストの要素のサイズを指定する1つの数値のアーギュメントを持つ。第1インレットに受け取られたリストは格納され、リストの長さは第2インレットに受け取られた数値、または**zl**オブジェクトのアーギュメントと比較される。リストの項目が第2インレットまたはアーギュメントによって決められた数値より長くなれば、その決められた数の項目のリストは第1アウトレットから出力される。その他の残ったシンボル、またはリストの要素は格納される。

**mode iter**はリストの要素のサイズを指定する1つの数値のアーギュメントを持つ。第1インレットに受け取られたリストは格納され、アーギュメント、または第2インレットに受け取られた数値によって指定された項目数で構成されたリストの連なりとして第1アウトレットから出力される。**zl**オブジェクトに格納された内容次第で、最後のリストの出力は指定された項目数より少なくなることもある。

**mode join**は両方のインレットからリストを受け取り、両方から入力されたリストを結合した1つのリストを第1アウトレットから出力する。

**mode len**は第1インレットからリストを受け取り、リストの要素数を第1アウトレットから出力する。

**mode reg**はリストを保持するレジスタとして機能する。第1インレットに受け取られたリストは直ちに第1アウトレットから出力される。第2インレットから受け取られたリストは格納される。**bang**で、格納されたリストを第1アウトレットから出力する。

**mode rev**は第1インレットからリストを受け取り、第1アウトレットから逆順にしたリストを出力する。

**mode rotate**はリストの内容を回すために使われる。アーギュメントは回されるリストの項目の場所の数-正の値はリストを右に、負の値はリストを左に回す-を指定するために使われる。この値はこのモードの時の第2インレットからの入力によっても指定できる。

**mode sect**は両方のインレットから入力を受け取り、第1アウトレットから両方のリストに共通して含まれる値のリストを出力する。

**mode slice**は1つのリストを2つのリストに分割するために使われる。このモードはリストの要素のサイズを指定する1つの数値のアーギュメントを持つ。この値はこのモードの時の第2インレットからの入力によっても指定できる。第1インレットに受け取られたリストは2つのリストに分割される。この時、1つ目のリストはアーギュメントによって指定された項目数に抑えられ、第1アウトレットから出力される。その他の残ったリストの要素はオブジェクトの第2アウトレットから出力される。注記：リストは第2アウトレットからはじめに出力される。

**mode union**は両方のインレットからリストを受け取り、入力されたリスト両方を含んだ1つのリストを第1アウトレットから出力する。第1および第2インレットが任意の共通する項目を含む場合、1つのシンボルだけが出力される。

**リスト** 第1インレット：**zl**オブジェクトが第1インレットにリストを受け取った時の振る舞いは、上記の**mode**メッセージの説明で述べたように、**zl**オブジェクトの**mode**によって異なる。

**リスト** 第2インレット：**zl**オブジェクトのいくつかのモードは第2インレットからのリストの入力を受け付け、以下のように振舞う。

**モード**     **振舞い**

**join**        このリストは第1インレットに受け取ったリストと結合され、第1インレットに**bang**かリストが受け取られた時に出力される。

**reg**         このリストは格納され、第1インレットに**bang**が受け取られた時に第1アウトレットから出力される。

**sect**        このリストは格納される。**bang**かリストが第1インレットに送られた時、両方のリストに共通の要素が、第1アウトレットから出力される。

---

|               |                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>union</b>  | このリストは格納される。 <b>bang</b> からリストが第1インレットに送られた時、両方のリストに共通する要素の繰り返しを除いた、一つの結合されたリストが第1アウトレットから出力される。                      |
| <b>bang</b>   | 第1インレット：第1または第1および第2アウトレットから以下の様にリストを出力する。                                                                            |
| <b>モード</b>    | <b>振舞い</b>                                                                                                            |
| <b>group</b>  | 最も最近に格納された $N$ 個の項目を第1アウトレットから出力する。ただし、 $N$ はアーギュメントか、第2インレットに受け取られた数値によって指定される。                                      |
| <b>iter</b>   | 最も最近に格納された項目のアーギュメントか、第2インレットに受け取られた数値によって指定されたサイズのグループを出力する。                                                         |
| <b>join</b>   | 第1および第2インレットに受け取られたリストの結合を、第1アウトレットから出力する。                                                                            |
| <b>reg</b>    | 現在格納されているリストを第1アウトレットから出力する。                                                                                          |
| <b>rev</b>    | 現在格納されているリストを逆順にして第1アウトレットから出力する。                                                                                     |
| <b>rotate</b> | 現在格納されているリストを $N$ 個分内容を循環させ第1アウトレットから出力する。ただし、 $N$ はアーギュメントか、第2インレットに受け取られた数値によって設定される。                               |
| <b>sect</b>   | 第1および第2インレットかに受け取られたリストの共通する項目を第1アウトレットから出力する。                                                                        |
| <b>slice</b>  | 現在格納されているリストを2つに分割し、はじめの $N$ 個の項目を第1アウトレットから出力する。また、残りの項目は第2アウトレットから出力する。ただし、 $N$ はアーギュメントか第2インレットに受け取られた数値によって設定される。 |
| <b>union</b>  | 第1および第2インレットに受け取られた両方のリストで共通する項目の繰り返しを除いた、両方のリストの項目からなる1つのリストを出力する。                                                   |

|        |                                                                                                                                |
|--------|--------------------------------------------------------------------------------------------------------------------------------|
| 整数     | 第2インレット： <b>zl</b> オブジェクトのいくつかのモードは第2インレットからの整数の入力を受け付け、以下のように振舞う。                                                             |
| モード    | 振舞い                                                                                                                            |
| group  | 最も最近に格納されたリストの出力する項目数を指定する。                                                                                                    |
| iter   | <b>zl</b> オブジェクトに現在格納されている内容は、この値によって指定された項目数から成るリストの連なりとして出力される。 <b>zl</b> オブジェクトに格納された内容次第で、最後のリストの出力は指定された要素の数より少なくなることもある。 |
| rotate | 現在格納されているリストを回す位置を指定する。正の値はリストを右に回させ、負の値は左に回させる。                                                                               |
| slice  | <b>zl</b> オブジェクトの第1アウトレットから出力するリストの項目数を指定する。このインレットにより指定された数値を超えた残りのリストの要素は、第2アウトレットから出力される。                                   |

## アーギュメント

|       |                                                                                                                                                                                                                                                    |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| シンボル  | オプション。 <b>group</b> 、 <b>iter</b> 、 <b>join</b> 、 <b>len</b> 、 <b>reg</b> 、 <b>rev</b> 、 <b>rotate</b> 、 <b>sect</b> 、 <b>slice</b> 、または <b>union</b> という単語は <b>zl</b> オブジェクトのモードを設定する任意のアーギュメントとして使われる。上記の異なるモードについての <b>mode</b> メッセージの説明を参照のこと。 |
| 整数    | オプション。いくつかの操作の <b>mode</b> では、設定された <b>mode</b> の機能を指定するために第2インレットに受け取られたリストをアーギュメントとして扱う。                                                                                                                                                         |
| モード   | 振舞い                                                                                                                                                                                                                                                |
| group | 最も最近格納されたリストの出力する項目数を指定する。                                                                                                                                                                                                                         |
| iter  | <b>zl</b> オブジェクトに現在格納されている内容は、この値によって指定された項目数から成るリストの連なりとして出力される。 <b>zl</b> オブジェクトに格納された内容次第で、最後の                                                                                                                                                  |



リストの出力は指定された要素の数より少なくなることもある。

- rotate** 現在格納されているリストを回す位置を指定する。正の値はリストを右に回させ、負の値は左に回させる。
- slice** **zl**オブジェクトの第1アウトレットから出力するリストの項目数を指定する。このインレットにより指定された数値を超えた残りのリストの要素は、第2アウトレットから出力される。

## 出力

リスト 第1アウトレットの出力：

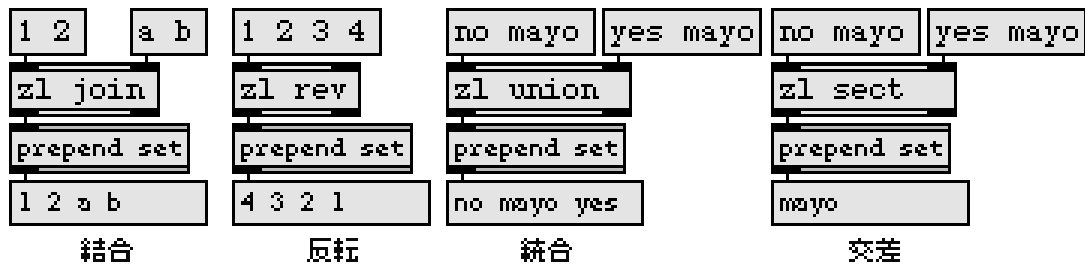
- group** モードでは、アーギュメントで指定された要素数からなるリスト。
- iter** モードでは、アーギュメントで指定された要素数をもつ複数個のリスト。**zl**オブジェクトに格納された内容次第で、最後のリストの出力は指定された要素の数より少なくなることもある。
- join** モードでは、両方のインレットから受け取ったリスト、すべての要素からなるリスト。
- len** モードでは、リストの項目数と一致する数値。
- reg** モードでは、入力のリストか、第2インレットに受け取ったもっとも最近の値。
- rev** モードでは、逆順にした入力のリスト。
- rotate** モードでは、入力のリストはアーギュメントで指定された正、または負の値に従って、右、または左に回転させられる。
- sect** モードでは、両方のインレットで受け取られたリストで共通する要素からなるリスト。
- slice** モードでは、アーギュメントで指定された要素数からなるリスト。



**union** モードでは、両方のリストで共通する項目の繰り返しを除いた、両方のリストの項目からなる1つのリスト。第1および第2インレットからの入力が必要な共通する項目を含む場合、1つのシンボルだけが出力される。

**リスト** 第2アウトレットの出力：**slice**モードでは、アークギュメントで指定された要素数を越えたリスト要素からなるリスト。

## 利用例



`zl` オブジェクトはリストに対する万能ナイフ

## 参照

|                   |                           |
|-------------------|---------------------------|
| <b>fromsymbol</b> | シンボルを個々の数値かメッセージに変換する     |
| <b>maximum</b>    | 数値のリストの中で最も大きい値を出力する      |
| <b>minimum</b>    | 数値のリストの中で最も小さい値を出力する      |
| <b>pack</b>       | 数値やシンボルをひとつのリストに結合させる     |
| <b>swap</b>       | 2つの数値の一連の順番を逆にする          |
| <b>thresh</b>     | 短時間に受け取った数値をリストとしてまとめる    |
| <b>tosymbol</b>   | メッセージ、数値、リストを1つのシンボルに変換する |

注記：このオブジェクトは、Max 4.1以前のバージョンでは使用できない

## 入力

**整数** 実数に変換される。

**実数** 第1インレット：入力値は、アーギュメントあるいは他のインレットから受け取られた数値により指定されたマッピングに基づいて拡大縮小される。

第2インレット：入力値の下限を設定する。もしこの値が第3インレットの値よりも大きい場合は、上限／下限の関係が逆転し、正しく範囲を指定するように変更される。

第3インレット：入力値の上限を設定する。もしこの値が第2インレットの値よりも小さい場合は、上限／下限の関係が逆転し、正しく範囲を指定するように変更される。

第4インレット：出力値の下限を設定する。もしこの値が第5インレットの値よりも大きい場合は、上限／下限の関係が逆転し、正しく範囲を指定するように変更される。

第5インレット：出力値の上限を設定する。もしこの値が第4インレットの値よりも小さい場合は、上限／下限の関係が逆転し、正しく範囲を指定するように変更される。

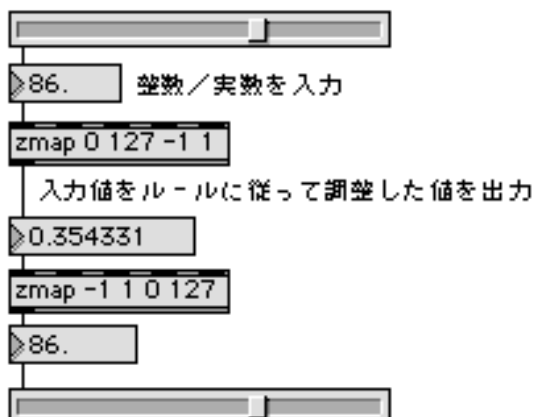
## アーギュメント

**整数または実数** オプション。1つ目のアーギュメントは入力値の最小値を、2つ目のアーギュメントは入力値の最大値を表す。同様に3つ目と4つ目のアーギュメントはそれぞれ出力値の最小値と最大値を表す。もし、最小値に最大値より大きな値を入力した場合（あるいはその逆）は、これら2つの値は入れ替わり、上限／下限の関係を保つように変更される。

## 出力

**実数** 第1インレットに値を受け取ると、その値は指定された出力範囲内で呼応するポジションになるように変換され、出力される。

## 利用例



整数のスライダー値を、実用的な実数に変換し、また元の数値に戻した利用例

## 参照

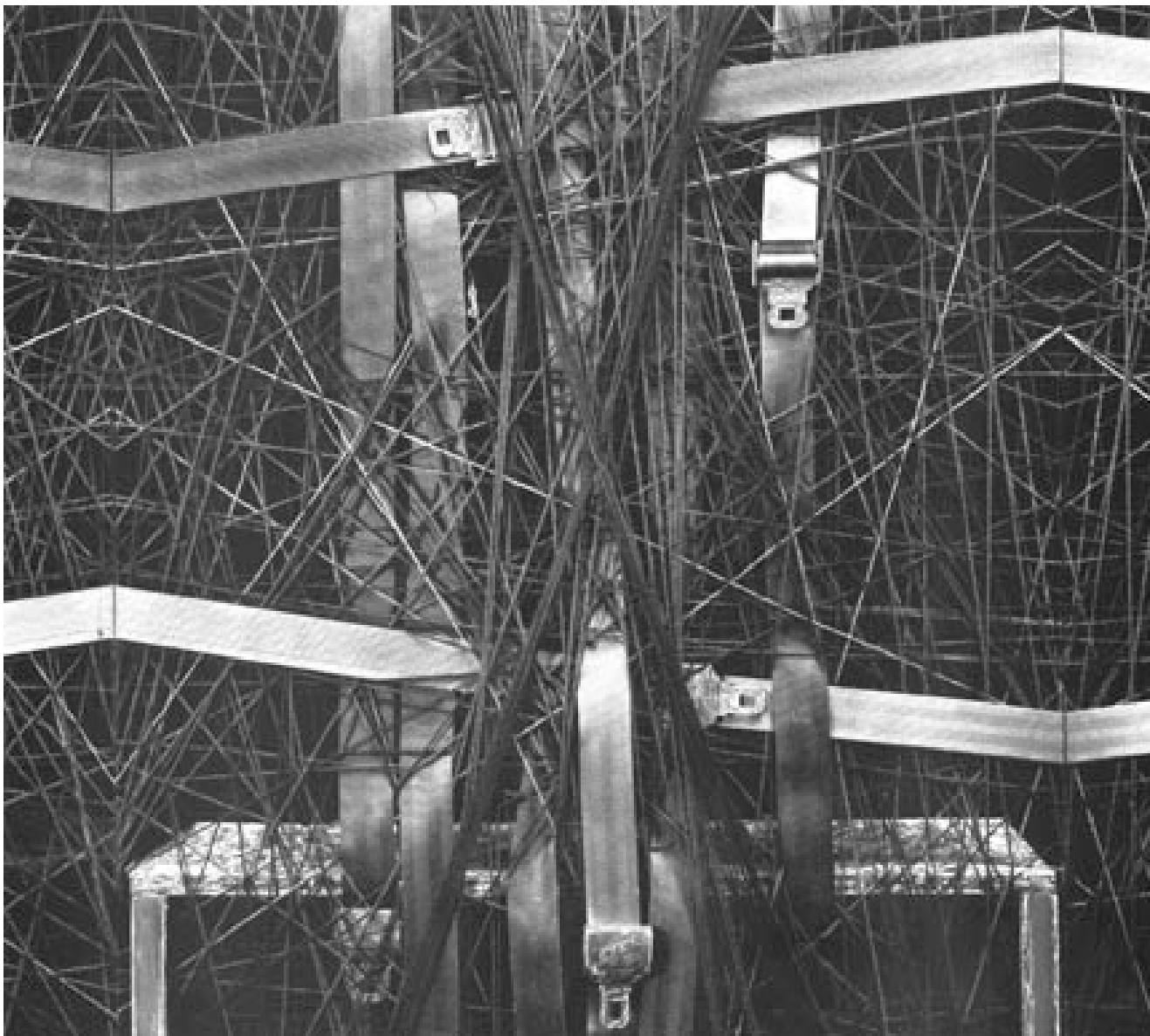
### scale

入力値を指定した数値の範囲にマッピングし、指定した出力範囲に拡大縮小および再マッピングして出力する

### expr

数式の値を求める

MSP



117アレンス

!~オブジェクトのように機能するが、インレットの順序が逆になっている。

## 入力

**シグナル** 第1インレット：第2インレットから入力されたシグナルまたは定数の値からこのシグナルを減算する。

第2インレット：第1インレットから入力されたシグナルまたは定数値をこのシグナルから減算する。

**実数または整数** 第1インレット：第2インレットから入力されたシグナルから、この数値を減算する。第1インレットにシグナルが入力された場合は、この数値は無視される。

第2インレット：この数値から、第1インレットから入力されたシグナルを減算する。第2インレットにシグナルを入力した場合は、この数値は無視される。

## アーギュメント

**実数または整数** オプション。第2インレットから入力されたシグナルから減算する初期値をセットする。第1インレットからシグナルが入力された場合は、アーギュメントは無視される。アーギュメントが存在せず、第1インレットにシグナルが入力されない場合は、デフォルトとして初期値は0になる。

## 出力

**シグナル** 2つの入力の差。

## 利用例



!~オブジェクトとはインレットの順序が逆

## 参照

+~ シグナルの加算

!/~オブジェクトは/~オブジェクトのように機能するが、インレットの順序が逆になっている。

注記：除算は、計算的に効率的な演算ではない。/~オブジェクトは、最初のアーギュメントか、第1インレットに受け取った整数または実数の逆数と、第2インレットに受け取ったシグナルを掛けるように最適化されている。しかし、両方のインレットにシグナルが入力された場合は、!/~は、著しく非効率的な除算方法をとる。

## 入力

**シグナル** 第1インレット：このシグナルは除数として使われ、第2インレットから入力されたシグナルまたは定数の値をこのシグナルで除算する。

第2インレット：このシグナルは、第1インレットから入力されたシグナルまたは定数の値に除算される。

**実数または整数** 第1インレット：第2インレットから入力されたシグナルを除算する数。第1インレットにシグナルが入力された場合は、実数または整数は無視される。

第2インレット：第1インレットから入力されたシグナルに除算される数。第2インレットにシグナルが入力された場合は、実数または整数は無視される。

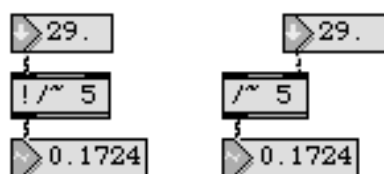
## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルを除算する初期値をセットする。第2インレットからシグナルが入力された場合は、アーギュメントは無視される。アーギュメントが存在せず、第2インレットにシグナルが入力されない場合は、デフォルトとして初期値は1になる。

## 出力

**シグナル** 2つの入力の比、すなわち、第2インレットを第1インレットで割ったもの。

## 利用例



!~ オブジェクトとはインレットの順序が逆

## 参照

\*~

シグナルの乗算



## 入力

**シグナル** 第1インレット：第2インレットから入力されたシグナルか定数値と、このシグナルを比較する。第2インレットの値と等しくない場合は1、等しい場合は0を出力する。

第2インレット：このシグナルを、第1インレットから入力されたシグナルと比較する。

**実数または整数** 第2インレット：この数値は、第1インレットから入力されたシグナルと比較される。第2インレットにシグナルが入力された場合は、この数値は無視される。

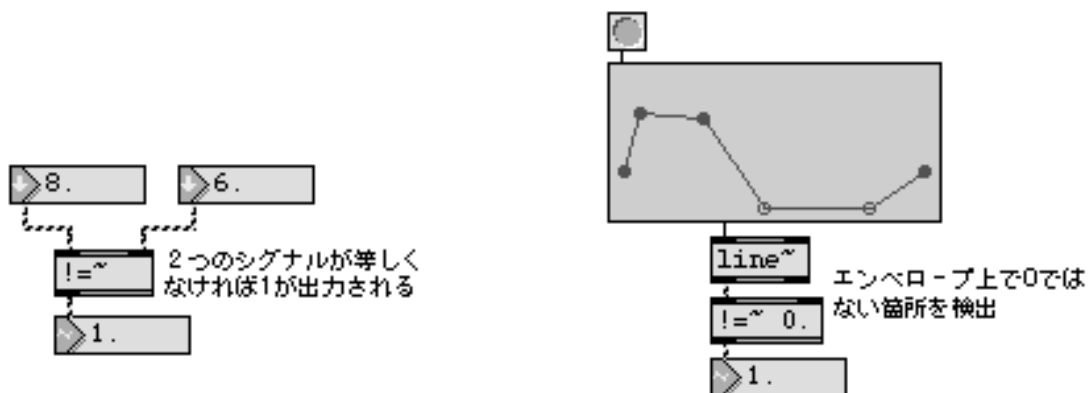
## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルと比較する初期値を指定する。シグナルがアーギュメントと等しくない場合は1、等しい場合は0を出力する。第2インレットにシグナルが入力された場合は、このアーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として0が用いられる。

## 出力

**シグナル** 第1インレットのシグナルと第2インレットの値が等しくない場合は1、等しい場合は0を出力する。

## 利用例



!=~を用いて、シグナルやエンベロープの値が0になるのを検出する

**!=~**

2つのシグナルの比較を行い、  
等しくなければ1を出力する

---

## 参照

|                |                               |
|----------------|-------------------------------|
| <b>==~</b>     | 2つのシグナルの比較を行い、等しければ1を出力する     |
| <b>&lt;~</b>   | 2つのシグナルの比較を行い、小さければ1を出力する     |
| <b>&lt;=~</b>  | 2つのシグナルの比較を行い、小さいか等しければ1を出力する |
| <b>&gt;~</b>   | 2つのシグナルの比較を行い、大きければ1を出力する     |
| <b>&gt;=~</b>  | 2つのシグナルの比較を行い、大きいか等しければ1を出力する |
| <b>change~</b> | シグナル値の変化方向を出力する               |
| <b>edge~</b>   | シグナルの論理的变化を検出する               |

## 入力

**シグナル** 第1インレット：このシグナルを第2インレットから入力されたシグナルか定数値で割り、その余りをアウトレットから出力する。

第2インレット：第1インレットから入力されたシグナルか定数値を、このシグナルで割る。

**実数または整数** 第1インレット：この数値を、第2インレットから入力されたシグナルで割る。第1インレットにシグナルが入力された場合は、この数値は無視される。

第2インレット：第1インレットから入力されたシグナルを、この数値で割る。第2インレットにシグナルが入力された場合は、この数値は無視される。

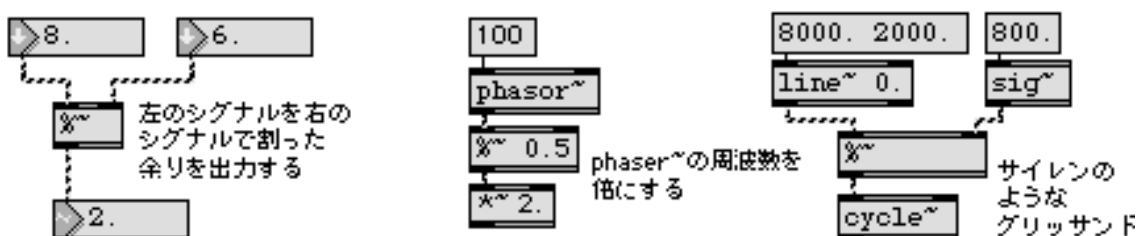
## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルを除算する初期値を指定する。第2インレットにシグナルが入力された場合は、アーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として1が用いられる。

## 出力

**シグナル** 第1インレットのシグナルを第2インレットで割った余りをアウトレットから出力する。%はモジュロ演算子と呼ぶ。

## 利用例



## 参照

!/~ シグナルの除算（インレットが逆）  
/~ シグナルの除算

## 入力

**シグナル** 第1インレット：このシグナルに、第2インレットから入力されたシグナルか定数値を掛ける。

第2インレット：第1インレットから入力されたシグナルか定数値に、このシグナルを掛ける。

**実数または整数** 第1インレット：第2インレットから入力されたシグナルに掛ける乗数。第2インレットにシグナルが入力された場合は、この数値は無視される。

第2インレット：第1インレットから入力されたシグナルに掛ける乗数。第2インレットにシグナルが入力された場合は、アーギュメントは無視される。

## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルに乗算する初期値を指定する。第2インレットにシグナルが入力された場合は、このアーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として0が用いられる。

## 出力

**シグナル** 2つの入力の乗算の結果。

## 利用例



シグナルの振幅に、定数値や変化する値、他のオーディオ・シグナルをかける

## 参照

/~ シグナルの除算  
 !/~ シグナルの除算（インレットが逆）

Tutorial 2 Fundamentals: Adjustable oscillator（英文PDF参照）  
 Tutorial 8 Synthesis: Tremolo and ring modulation（英文PDF参照）

-~

## 入力

**シグナル** 第1インレット：このシグナルから、第2インレットから入力されたシグナルか定数値を減算する。

第2インレット：第1インレットから入力されたシグナルか定数値から、このシグナルを減算する。

**実数または整数** 第1インレット：この数値から、第2インレットから入力されたシグナルを減算する。第1インレットにシグナルを入力した場合は、この数値は無視される。

第2インレット：第1インレットから入力されたシグナルから、この数値を減算する。第2インレットにシグナルが入力された場合は、この数値は無視される。

## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルから減算する初期値を指定する。第2インレットにシグナルが入力された場合は、アーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として0が用いられる。

## 出力

**シグナル** 2つの入力の差。

## 利用例



負の値のDC オフセット加算前に、シグナルを反転するために減算する

## 参照

+~ シグナルの加算  
!~ シグナルの減算（インレットが逆）

## +~

注記：シグナルの入力を受け付けるMSPオブジェクトのインレットは、そのインレットに入力されたすべてのシグナルを自動的に加算するようになっている。しかし、+~オブジェクトは、シグナルの加算を視覚的に明示したり、実数や整数のオフセットをシグナルに加算するために必要となる。

## 入力

**シグナル** 第1インレット：このシグナルを、第2インレットから入力されたシグナルか定数値に足す。

第2インレット：第1インレットから入力されたシグナルか定数値に、このシグナルを足す。

**実数または整数** 第1インレット：この数値（オフセット）を、第2インレットから入力されたシグナルに足す。第1インレットにシグナルを入力した場合は、この数値は無視される。

第2インレット：第1インレットから入力されたシグナルに、この数値を足す。第2インレットにシグナルが入力された場合は、この数値は無視される。

## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルに加算する初期値を指定する。第2インレットにシグナルが入力された場合は、このアーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として0が用いられる。

## 出力

**シグナル** 2つの入力の和。

## 利用例



複数のシグナルをミックスする、またはシグナルにDC オフセットを加える

## 参照

|      |                  |
|------|------------------|
| +==~ | 結果を格納し、シグナルの加算   |
| --~  | シグナルの減算          |
| !-~  | シグナルの減算（インレットが逆） |



## 入力

**シグナル** 入力されたそれぞれのサンプル値は、現在の加算合計の値に加算され、出力される。例えば、合計値が0である時に、シグナルが1,1,1,1と入力された場合、出力は1, 2, 3, 4となる。

**bang** 合計値を0にリセットする

**set** setメッセージに数値を続け、合計値を設定する。

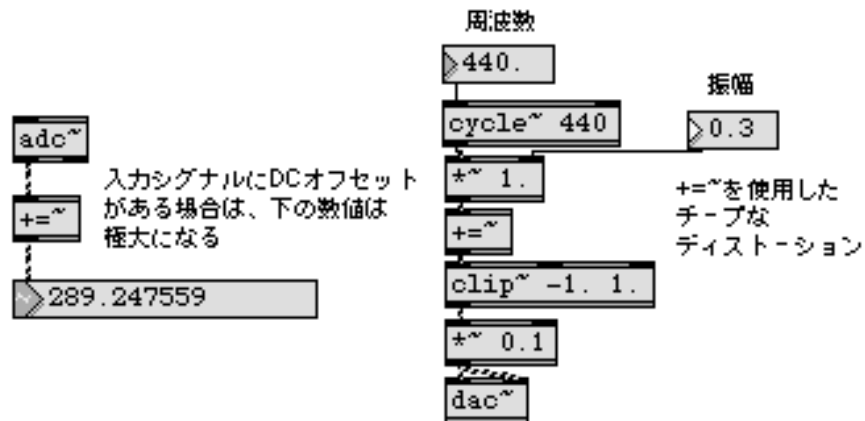
## アーギュメント

**実数** オプション。合計値の初期値を指定する。デフォルト値は0である。

## 出力

**シグナル** これまでに入力されたすべてのサンプル値の合計の値が出力される。

## 利用例



## 参照

**+=~** シグナルの加算

注意：除算は計算的に効率のよい演算ではない。`/~`オブジェクトは最適化されており、第1インレットに入力されたシグナルに、アーギュメントで指定した初期値か、第2インレットに入力された整数または実数の逆数を乗算している。また、2つのシグナルが入力された場合、`/~`オブジェクトは大幅に非効率的な除算処理を行うことになる。

## 入力

**シグナル** 第1インレット：このシグナルを、第2インレットから入力されたシグナルか定数値で割る。

第2インレット：第1インレットから入力されたシグナルか定数値を、このシグナルで割る。

**実数または整数** 第1インレット：この数値を、第2インレットから入力されたシグナルで割る。第1インレットにシグナルが入力された場合は、この数値は無視される。

第2インレット：第1インレットから入力されたシグナルを、この数値で割る。第2インレットにシグナルが入力された場合は、この数値は無視される。

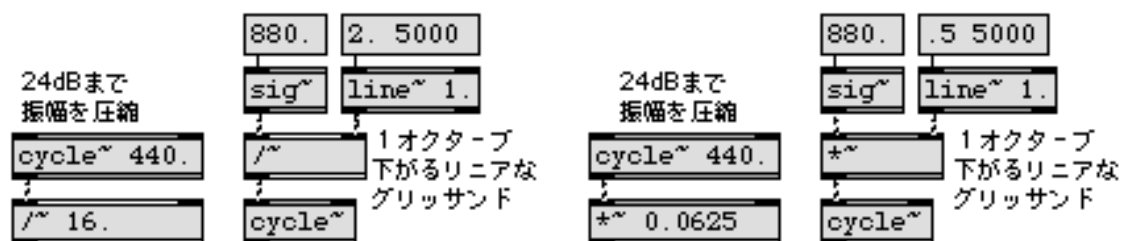
## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルを除算する初期値を指定する。第2インレットにシグナルが入力された場合は、このアーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として1が用いられる。

## 出力

**シグナル** 2つの入力の比、すなわち、第1インレットの入力を第2インレットの入力で割った結果が出力される。

## 利用例



可能な場合は、乗算を用いた方が、より効率的に計算を行うことができる

---

## 参照

|     |                       |
|-----|-----------------------|
| !/~ | シグナルの除算（インレットが逆）      |
| *~  | シグナルの乗算               |
| %~  | 2つのシグナルの除算を行い、余りを出力する |

## 入力

**シグナル** 第1インレット：このシグナルは、第2インレットから入力されたシグナルか定数値と比較される。第2インレットの値よりも小さければ1を、そうでなければ0を出力する。

第2インレット：第1インレットから入力されたシグナルと、このシグナルを比較する。

**実数または整数** 第2インレット：第1インレットから入力されたシグナルと比較される数値。第2インレットにシグナルが入力された場合は、この数値は無視される。

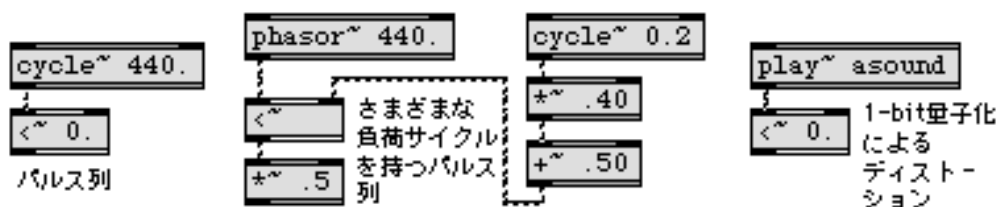
## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルと比較する初期値を指定する。シグナルがアーギュメントより小さい場合は1を、そうでない場合は0を出力する。第2インレットにシグナルが入力された場合は、アーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として0が用いられる。

## 出力

**シグナル** 第1インレットのシグナルが第2インレットの値よりも小さい場合は1を、そうでない場合は0を出力する。

## 利用例



どのようなシグナルも、1か0の値に変換される

## 参照

- <==~ 2つのシグナルの比較を行い、小さいか等しければ1を出力する
- >~ 2つのシグナルの比較を行い、大きければ1を出力する
- >==~ 2つのシグナルの比較を行い、大きい等しければ1を出力する
- ==~ 2つのシグナルの比較を行い、等しければ1を出力する
- !=~ 2つのシグナルの比較を行い、等しくなければ1を出力する

## 入力

**シグナル** 第1インレット：このシグナルは、第2インレットから入力されたシグナルか定数値と比較される。第2インレットの値以下であれば1を、そうでなければ0を出力する。

第2インレット：第1インレットから入力されたシグナルと、このシグナルを比較する。

**実数または整数** 第2インレット：第1インレットから入力されたシグナルと比較される数値。第2インレットにシグナルが入力された場合は、この数値は無視される。

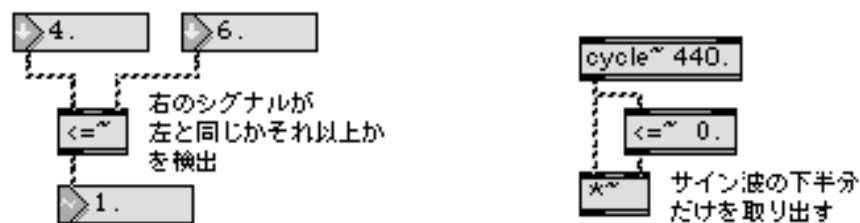
## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルと比較する初期値を指定する。シグナルがアーギュメント以下の場合は1を、そうでない場合は0を出力する。第2インレットにシグナルが入力された場合は、アーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として0が用いられる。

## 出力

**シグナル** 第1インレットのシグナルが第2インレットの値以下の場合は1を、そうでない場合は0を出力する。

## 利用例



## 参照

- < 2つのシグナルの比較を行い、小さければ1を出力する
- > 2つのシグナルの比較を行い、大きければ1を出力する
- >= 2つのシグナルの比較を行い、大きい等しければ1を出力する
- = 2つのシグナルの比較を行い、等しければ1を出力する
- != 2つのシグナルの比較を行い、等しくなければ1を出力する

## 入力

**シグナル** 第1インレット：第2インレットから入力されたシグナルか定数値と、このシグナルを比較する。第2インレットの値と等しい場合は**1**、等しくない場合は**0**を出力する。

第2インレット：このシグナルは、第1インレットから入力されたシグナルとの比較に利用される。

**実数または整数** 第2インレット：この数値は、第1インレットから入力されたシグナルと比較される。第2インレットにシグナルが入力された場合は、この数値は無視される。

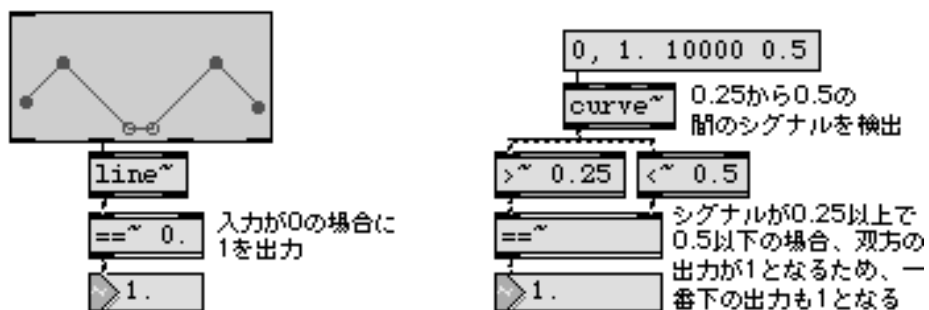
## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルと比較する初期値を指定する。シグナルがアーギュメントと等しい場合は**1**、等しくない場合は**0**を出力する。第2インレットにシグナルが入力された場合は、このアーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として0が用いられる。

## 出力

**シグナル** 第1インレットのシグナルと第2インレットの値が等しい場合は**1**、等しくない場合は**0**を出力する。

## 利用例



シグナルが特定の値であるか、または2つのシグナルが同じ値であるか、を検出する

## 参照

|                |                               |
|----------------|-------------------------------|
| <~             | 2つのシグナルの比較を行い、小さければ1を出力する     |
| <=~            | 2つのシグナルの比較を行い、小さいか等しければ1を出力する |
| >~             | 2つのシグナルの比較を行い、大きければ1を出力する     |
| >=~            | 2つのシグナルの比較を行い、大きいか等しければ1を出力する |
| !=~            | 2つのシグナルの比較を行い、等しくなければ1を出力する   |
| <b>change~</b> | シグナル値の変化方向を出力する               |
| <b>edge~</b>   | シグナルの論理的变化を検出する               |



## 入力

**シグナル** 第1インレット：このシグナルは、第2インレットから入力されたシグナルか定数値と比較される。第2インレットの値よりも大きければ1を、そうでなければ0を出力する。

第2インレット：第1インレットから入力されたシグナルと、このシグナルを比較する。

**実数または整数** 第2インレット：第1インレットから入力されたシグナルと比較される数値。第2インレットにシグナルが入力された場合は、この数値は無視される。

## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルと比較する初期値を指定する。アーギュメントがアーギュメントより大きい場合は1を、そうでない場合は0を出力する。第2インレットにシグナルが入力された場合は、このアーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として0が用いられる。

## 出力

**シグナル** 第1インレットのシグナルが第2インレットの値よりも大きい場合は1を、そうでない場合は0を出力する。

## 利用例



どのようなシグナルも、1か0の値に変換される

## 参照

- <~ 2つのシグナルの比較を行い、小さければ1を出力する
- <=~ 2つのシグナルの比較を行い、小さいか等しければ1を出力する
- >=~ 2つのシグナルの比較を行い、大きい等しければ1を出力する
- ==~ 2つのシグナルの比較を行い、等しければ1を出力する
- !=~ 2つのシグナルの比較を行い、等しくなければ1を出力する
- sah~ サンプル&ホールド

## 入力

**シグナル** 第1インレット：このシグナルは、第2インレットから入力されたシグナルか定数値と比較される。第2インレットの値以上であれば1を、そうでなければ0を出力する。

第2インレット：第1インレットから入力されたシグナルと、このシグナルを比較する。

**実数または整数** 第2インレット：第1インレットから入力されたシグナルと比較される数値。第2インレットにシグナルが入力された場合は、この数値は無視される。

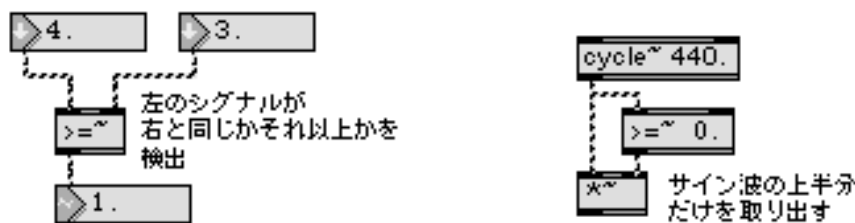
## アーギュメント

**実数または整数** オプション。第1インレットから入力されたシグナルと比較する初期値を指定する。シグナルがアーギュメント以上の場合は1を、そうでない場合は0を出力する。第2インレットにシグナルが入力された場合は、このアーギュメントは無視される。アーギュメントを設定しないか、第2インレットにシグナルを入力しない場合は、デフォルトの初期値として0が用いられる。

## 出力

第1インレットのシグナルが第2インレットの値以上の場合は1を、そうでない場合は0を出力する。

## 利用例



## 参照

- <~ 2つのシグナルの比較を行い、小さければ1を出力する
- <=~ 2つのシグナルの比較を行い、小さいか等しければ1を出力する
- >~ 2つのシグナルの比較を行い、大きければ1を出力する
- =~ 2つのシグナルの比較を行い、等しければ1を出力する
- !=~ 2つのシグナルの比較を行い、等しくなければ1を出力する
- sah~ サンプル&ホールド

## 入力

**シグナル** 第1インレット：0から1へ連続的に変化するシグナルを入力して、**buffer~**オブジェクト内の特定範囲のサンプルを読み出す。**phasor~**の出力シグナルを**2d.wave~**をオシレーターとしてコントロールするのに使えるほか、**buffer~**をウェーブテーブルとする場合に読み出すサンプル範囲を調整することなどに使える。しかし、**2d.wave~**の第1インレットに接続している**phasor~**の周波数を変更した時は、注意が必要である。その場合、複数の波形や繰り返し波形をウェーブテーブルとして利用していると、**2d.wave~**から出力されるシグナルのピッチが**phasor~**の設定周波数に完全に同期しているとは限らないからだ。**phasor~**の出力シグナルを正負反転させたものを**2d.wave~**に入力すれば、波形の逆再生が行える。

第2インレット：0から1へ連続的に変化するシグナルを入力して、**rows**メッセージで分割された列のどれを利用するかを決める。**phasor~**の正負反転した出力シグナルによって、再生する列の順番を逆転することができる。

第3インレット：**buffer~**オブジェクトのサンプル・メモリに対し、読み出す開始位置を波形の先頭からのオフセット値としてミリ秒単位で指定する。

第4インレット：**buffer~**オブジェクトのサンプル・メモリに対し、読み出す終了位置を波形の先頭からのオフセット値としてミリ秒単位で指定する。

**実数または整数** 第3および第4インレット：シグナルの代わりに波形の開始点と終了点をコントロールするのに使う数値を入力できる。これらのインレットにシグナルのパッチコードが接続されていない場合のみ、数値を受け付ける。

**rows** **rows**に続いて整数を指定する。これは**buffer~**に保持されるサンプル・データを分割する数であり、分割された各サンプル・データは、2次元行列の各列として扱われる。第2インレットに入力するシグナルはどの列のデータを再生するかを決定する。デフォルト値は0である。

**set** **set**に続いて**2d.wave~**がウェーブテーブルとして利用する**buffer~**の参照名をシンボルで指定する。この参照名のシンボルに続き、読み出し開始および終了位置の数値をオプションとして指定することができる。これらの値が指定されていない場合、デフォルトの開始および終了位置（**buffer~**内の最初と最後のサンプル）が使われる。第3および第4インレットにシグナルのパッチコードが接続されていると、**set**メッセージでのこれらの数値は無視される。

## アーギュメント

**シンボル** 必須。ウェーブテーブルとして利用する **buffer~** の参照名を指定する。**2d.wave~** は **buffer~** 内のデータのコピーしたものを出力するわけではないため、**buffer~** 内のデータが変更されると、**2d.wave~** の出力も変化することに注意が必要である。**buffer~** 内部に複数チャンネルのデータがある場合、**2d.wave~** は常に **buffer~** 内の最初の  $n$  個のチャンネルのデータを読み出す。この  $n$  の数値は **2d.wave~** オブジェクトの出力チャンネル数である。デフォルトの出力チャンネル数は1だが、第3アーギュメントでチャンネル数を指定することができる。

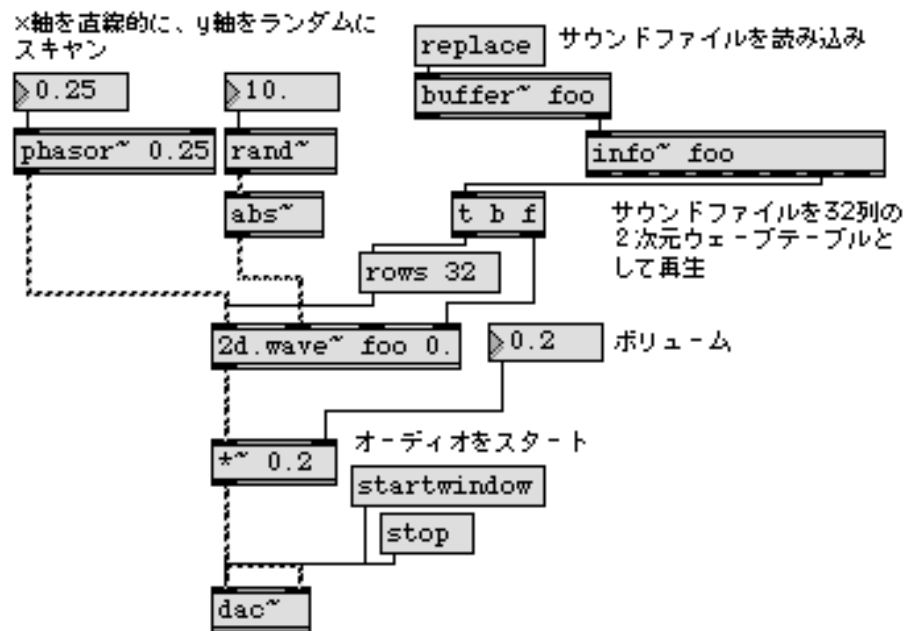
**実数または整数** オプション。**buffer~** の参照名に続き、波形の読み出し開始および終了位置を指定することができる。どちらの値も、**buffer~** オブジェクト内のサンプル・メモリの先頭からミリ秒単位のオフセット値として指定する。デフォルトでは開始位置は0、終了位置はサンプル・データの最後の位置となる。開始位置を0でない値、終了位置をサンプル・データの最後のサンプルとする場合、アーギュメントは **buffer~** の後ろに1つの数値だけを指定する。第3インレットにシグナルのパッチコードが接続されている場合は、アーギュメントでの **buffer~** 参照名の後の開始位置の指定は無視される。同じく、第4インレットにシグナルのパッチコードが接続されていると、終了位置のアーギュメントは無視される。追加オプションとして、これらの数値の後に **buffer~** 内のチャンネル数を指定することができる。

**整数** オプション。出力チャンネル数を設定する。これは **2d.wave~** オブジェクトが持つアウトレットの数となる。8チャンネルまでチャンネル数を指定することができる。デフォルト値は1である。再生するオーディオ・ファイルのチャンネルが **2d.wave~** で設定したチャンネル数より多い場合、余ったチャンネルのデータは再生されない。オーディオ・ファイルのほうがチャンネル数が少ない場合、**2d.wave~** オブジェクトの余分なアウトレットについては無音出力となる。

## 出力

**シグナル** **2d.wave~** オブジェクトで設定された読み出し開始位置と終了位置に相当する **buffer~** 内の一部のデータを、**2d.wave~** の第1インレットに入力された0から1の範囲のシグナルで走査する。そして、その **buffer~** 内の相当するデータを **2d.wave~** のアウトレットから出力する。**2d.wave~** への入力シグナルが **phasor~** オブジェクトが出力するノコギリ波のような繰り返し波形である場合、それに対する出力波形は入力シグナルに対応する周波数で繰り返すものとなる（1波形自体は **buffer~** 内の一部のデータである）。

## 利用例



サンプルの一部を利用することで、可変サイズのウェーブテーブルとして繰り返し波形を出力

## 参照

|                |                  |
|----------------|------------------|
| <b>buffer~</b> | オーディオ・サンプルの格納    |
| <b>groove~</b> | 速度可変のループ再生       |
| <b>phasor~</b> | ノコギリ波の生成         |
| <b>play~</b>   | ポジション・ベースのサンプル再生 |
| <b>wave~</b>   | サイズ可変のウェーブテーブル再生 |

Tutorial 15

Sampling: Variable-length wavetable (英文PDF参照)

## 入力

シグナル 任意のシグナル。

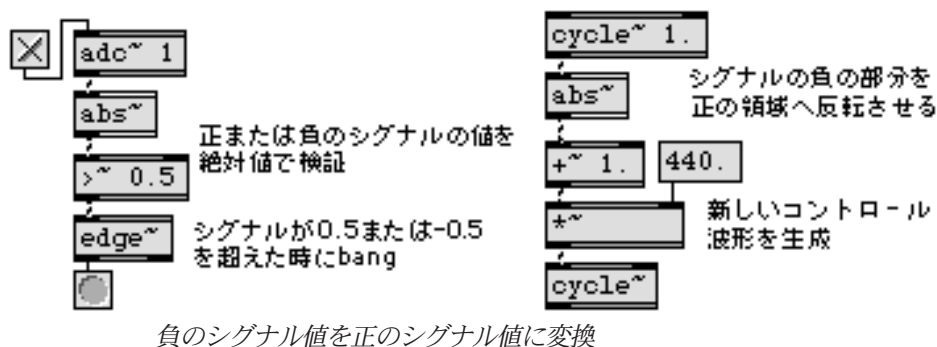
## アーギュメント

なし。

## 出力

シグナル 入力されたシグナルの絶対値（つまり、負ではない値）であるサンプル値から成るシグナル。

## 利用例



## 参照

avg~ シグナルの平均値

## 入力

シグナル アーク・コサイン関数への入力。

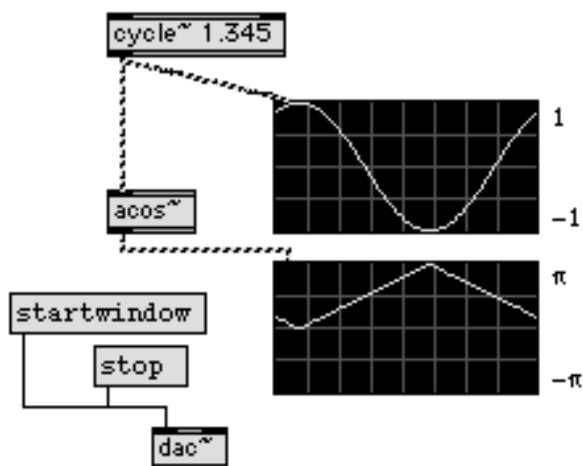
## アーギュメント

なし。

## 出力

シグナル ラディアン単位での入力のアーク・コサイン値。

## 利用例



acos~ を使ってラディアン単位 ( $-\pi$ – $\pi$ ) の逆リニア・ランプをつくる

## 参照

|               |                         |
|---------------|-------------------------|
| <b>asin~</b>  | シグナルのアーク・サイン関数          |
| <b>atan~</b>  | シグナルのアーク・タンジェント関数       |
| <b>atan2~</b> | シグナルのアーク・タンジェント関数 (2変数) |
| <b>cos~</b>   | シグナル (0-1の範囲) のコサイン関数   |
| <b>cosh~</b>  | シグナルの双曲線コサイン関数          |
| <b>cosx~</b>  | シグナルのコサイン関数             |
| <b>sinh~</b>  | シグナルの双曲線サイン関数           |
| <b>sinx~</b>  | シグナルのサイン関数              |
| <b>tanh~</b>  | シグナルの双曲線タンジェント関数        |
| <b>tanx~</b>  | シグナルのタンジェント関数           |



## 入力

シグナル 双曲線アーク・コサイン関数への入力。

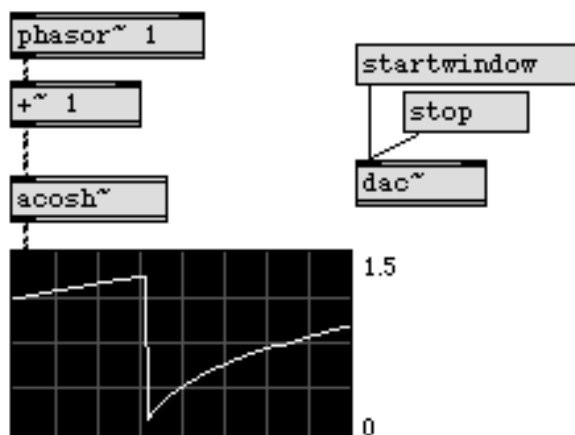
## アーギュメント

なし。

## 出力

シグナル 入力の双曲線アーク・コサイン値。

## 利用例



## 参照

|               |                         |
|---------------|-------------------------|
| <b>acos~</b>  | シグナルのアーク・コサイン関数         |
| <b>asin~</b>  | シグナルのアーク・サイン関数          |
| <b>asinh~</b> | シグナルの双曲線アーク・サイン関数       |
| <b>atan~</b>  | シグナルのアーク・タンジェント関数       |
| <b>atanh~</b> | シグナルの双曲線アーク・タンジェント関数    |
| <b>atan2~</b> | シグナルのアーク・タンジェント関数 (2変数) |
| <b>cos~</b>   | シグナル (0-1の範囲) のコサイン関数   |
| <b>cosh~</b>  | シグナルの双曲線コサイン関数          |
| <b>cosx~</b>  | シグナルのコサイン関数             |
| <b>sinh~</b>  | シグナルの双曲線サイン関数           |
| <b>sinx~</b>  | シグナルのサイン値               |
| <b>tanh~</b>  | シグナルの双曲線タンジェント関数        |
| <b>tanx~</b>  | シグナルのタンジェント関数           |

## 入力

- 整数** 0でない数は、読み込まれたすべてのパッチのオーディオ処理をオンにする。0は、読み込まれたすべてのパッチのオーディオ処理をオフにする。
- open** DSPステータス・ウィンドウを開く。
- set** **set**に二つの数を続ける。最初の数は、アウトレットのナンバーを定義するものであり、1であれば一番左のアウトレットから出力する。2番目の数は、論理チャンネル数（1から512）であり、0にした場合はアウトレットからはゼロ・シグナルが出力される。
- start** 読み込まれたすべてのパッチのオーディオ処理をオンにする。
- stop** 読み込まれたすべてのパッチのオーディオ処理をオフにする。
- startwindow** **adc~**オブジェクトの置かれたパッチと、そのパッチのサブパッチのオーディオ処理をオンにする。また、その他のパッチのオーディオ処理をオフにする。
- wclose** DSPステータス・ウィンドウが開いている場合は閉じる。
- (マウス) **adc~**オブジェクトをダブルクリックすれば、DSPステータス・ウィンドウが開く。

## アーギュメント

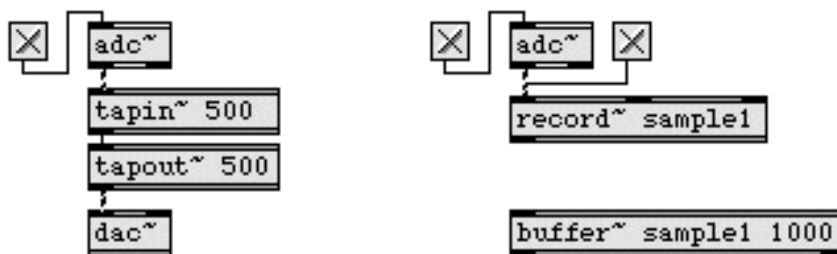
- 整数** オプション。**adc~**オブジェクトは、1つ以上、512までのオーディオ入力チャンネルを持つことができる。これらのチャンネル・ナンバーは、対応する論理チャンネルであるが、DSPステータス・ウィンドウのI/Oマッピング、または**input**というアーギュメントを持った**adstatus**オブジェクトで、ドライバの物理デバイス・チャンネルに、ダイナミックに割り当てを変更することができる。Sound Managerを使用している場合は、2つの入力チャンネルをもつ。その他のオーディオ・ドライバを使用している場合は、それ以上のチャンネル数に対応することもできるだろう。**adc~**がアーギュメントを持たない場合、最初に2つのアウトレットが作られ、入力の論理チャンネルは1と2にセットされる。

## 出力

- シグナル** コンピュータに入力されたシグナルを出力する。1つのアウトレットにつき1チャンネル分が出力される。アーギュメントがない場合は、チャンネルは1と2となり、左のアウトレットから右のアウトレットへ

と番号付けられる。アーギュメントがある場合は、チャンネルはアーギュメントで指定された順番に出力される。

### 利用例



オーディオの入力処理とレコーディング

### 参照

**adstatus**  
**ezadc~**  
**dac~**

オーディオ・ドライバーの出力チャンネルにアクセスする  
 オーディオの入力とオン/オフ・ボタン  
 オーディオ出力とオーディオ処理のオン・オフ

Tutorial 13

Sampling: Recording and playback (英文PDF参照)

## 入力

**set** 2つの数値が後に続く**set**は、あるオーディオ・ドライバーの出力チャンネルを、**adoutput~**オブジェクトのシグナル・アウトレットの1つに割り当てる。最初の数値はアウトレットのインデックス番号で、1という値は左のアウトレットにあてはまる。2番目の数値はオーディオ・ドライバーの出力デバイス・チャンネルで、1は最初のチャンネルにあてはまる。2番目の数値が0の場合には、指定されたアウトレットは機能せず、無音を出力する。

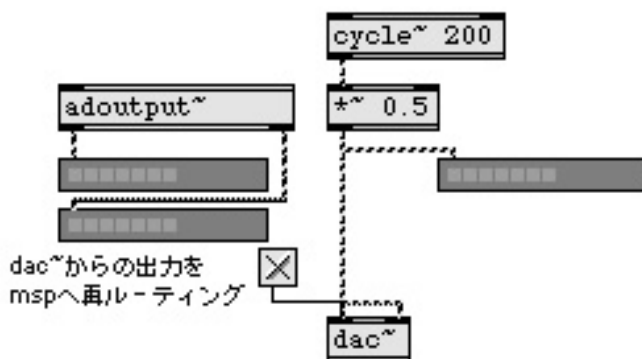
## アーギュメント

**整数** オプション。アーギュメントは、カレントのオーディオ・ドライバーの出力チャンネルを指定する。指定することのできるチャンネル数には制限がない。デフォルトでは、**adoutput~**は2つのアウトプットを生成し、カレントのオーディオ・ドライバーのチャンネル1と2からのオーディオ出力をアサインする。注意すべきことは、これらのチャンネル番号は、**dac~**と**adc~**オブジェクトで使用される論理的なチャンネル番号ではなく、いかなる再割当が行われた後でも、ドライバーの物理的な出力チャンネルを表す、ということである。**dac~**での論理的なチャンネルとオーディオ・ドライバーの実際のチャンネルの間の関連は、DSP StatusウィンドウのI/O Mappingsサブ・ウィンドウで設定を行う。

## 出力

**シグナル** **adoutput~**のそれぞれのアウトレットは割り当てられたオーディオ・ドライバーのチャンネルからのシグナルを出力し、カレントのシグナル・ベクター・サイズのサンプル数分だけ遅れる。

## 利用例



パッチの出力を録音(あるいは再処理)するために、物理的なDACチャンネルの出力を取得する

## 参照

**adstatus**  
**dac~**

オーディオ・ドライバーの出力チャンネルにアクセスする  
オーディオ出力とオーディオ処理のオン・オフ

**adstatus**オブジェクトは、アーギュメントを利用し、様々なオーディオの設定を行うことができる。使用できるアーギュメントは、後記のアーギュメントのセクションにリストアップされている。

## 入力

**bang** 第1インレット：現在の設定の状態を出力する。多くの場合、**adstatus**オブジェクトの第1アウトレットからメッセージが出力される。このメッセージは、**menu**オブジェクトのポップアップ・メニューに現在の状態を表示させるための**set**メッセージを伴っている。このような場合には、第2アウトレットから設定の数値が出力される。詳しい動作については、後述の出力の項目を参照のこと。

**override** 第1インレット：**override 1**は、**adstatus**オブジェクトに対応する設定のオーバーライド・モードをオンにする。オーバーライド・モードがオンの場合は、設定を変更しても、MSP Preferencesファイルに保存されない。**override 0**は、オーバーライド・モードをオフにする。デフォルトでは、すべての設定に対してオーバーライド・モードはオフになっている。しかし、いくつかの設定はオーディオ・ドライバー特有のものであり、ドライバーにより保存されない場合がある。

**整数** 第1インレット：設定を変更する。ほとんどの場合、この数値は**menu**オブジェクトのインデックス値に対応しており、そのインデックス値は**adstatus**への**bang**メッセージにより設定される。

第2インレット：**adstatus**オブジェクトで、**input**、**iovs**、**output**、**sigvs**、**sr**を設定する場合、第2インレットに整数を入力することにより、設定ができる。これは第1インレットのようにメニューのインデックス値として設定を行うものではない（次の**reset**または**loadbang**メッセージを参照）。その他の設定を行う場合は、第2インレットは第1インレットと同じように機能する。

**set** 第1インレット：setに1から512の数字を続け、**adstatus input**または、**adstatus output**オブジェクトに対応する論理チャンネルの変更をする。現在物理的に使用しているオーディオ・ドライバーの入出力チャンネルが、オブジェクトのアウトレットから出力される。これは新しい論理チャンネルの設定に用いられる。

**float** 整数と同じ。

## resetまたはloadbang

**adstatus**オブジェクトは、ポップアップ・メニューとともに使われるが、**reset**または**loadbang**メッセージは、**adstatus**オブジェクトをコントロールするためのポップアップ・メニューを作るのに必要なメッセー

ジを出力する。**clear**メッセージが最初に出力され、続けてそれぞれのメニューのアイテム、さらに続けて、設定の現在の値をもとにしたメニューの値の表示をセットするための**set**メッセージが出力される。

## アーギュメント 動作

|                   |                                                                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>cpu</b>        | なし。                                                                                                                                         |
| <b>cpulimit</b>   | 最大CPU占有率を設定する。CPU占有率が最大CPU占有率を上回った場合は、オーディオ処理を一時停止する。0を設定した場合は、最大CPU占有率を設けない。                                                               |
| <b>driver</b>     | <b>adstatus driver</b> によって作られた使用可能なオーディオ・ドライバーのメニューのインデックス値として扱われる値。この値により、メニューのインデックス値に対応するドライバーをロードする。                                   |
| <b>info</b>       | なし。                                                                                                                                         |
| <b>input</b>      | <b>adstatus input</b> によって作られた、使用可能なオーディオ入力チャンネルのメニューのインデックス値として扱われる値。オブジェクトに割り当てられた論理チャンネルを、メニューのインデックスに対応するチャンネルからの入力を受け入れるために、この値で設定をする。 |
| <b>iovs</b>       | <b>adstatus iovs</b> によって作られた利用可能なI/Oベクター・サイズのメニューのインデックス値として扱われる数値。この数値を使って、ドライバーのI/Oベクターサイズを、メニューのインデックスにより指定されたアイテムの値に設定する。              |
| <b>latency</b>    | なし。                                                                                                                                         |
| <b>numinputs</b>  | なし。                                                                                                                                         |
| <b>numoutputs</b> | なし。                                                                                                                                         |
| <b>optimize</b>   | 0を受け取ると最適化モードをオフ、1を受け取るとオンにする。                                                                                                              |
| <b>option</b>     | <b>adstatus option</b> によって作られた特定のオプションを選択するためのメニューのインデックスとして扱われる数値。この数値を使って、オプションをメニ                                                       |

|                   |                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | ユーのインデックスに対応する値に設定する。                                                                                                                             |
| <b>optionname</b> | なし。                                                                                                                                               |
| <b>output</b>     | <b>adstatus output</b> によって作られた利用可能なオーディオ・アウトプット・チャンネルのメニューのインデックス値として扱われる値。オブジェクトに割り当てられた論理チャンネルを、メニューのインデックスに対応するチャンネルからの出力を受け入れるために、この値で設定をする。 |
| <b>overdrive</b>  | 0を受け取るとオーバードライブ・モードをオフ、1を受け取るとオンにする。                                                                                                              |
| <b>sigvs</b>      | <b>adstatus sigvs</b> によって作られた利用可能なシグナル・ベクター・サイズメニューのインデックス値として扱われる値。この数値を使って、現在のシグナル・ベクター・サイズをメニューのインデックスにより指定されたアイテムの値に設定する。                    |
| <b>sr</b>         | <b>adstatus sr</b> によって作られた利用可能なサンプリング・レートのメニューのインデックス値として扱われる値。この数値を使って、現在のサンプリング・レートをメニューのインデックスにより指定されたアイテムの値に設定する。                            |
| <b>switch</b>     | 0を受け取るとDSPをオフ、1を受け取るとオンにする。                                                                                                                       |
| <b>takeover</b>   | 0を受け取るとオーディオ・インタラプト・モードをオフ、1を受け取るとオンにする。                                                                                                          |
| <b>timecode</b>   | 0を受け取るとタイムコードの出力をオフ、1を受け取るとオンにする。                                                                                                                 |

## アーギュメント

様々なアーギュメント 最初のアーギュメントはシンボルであり、これで**adstatus**オブジェクトで何を設定するかを定義する。設定によっては、他にもアーギュメントを必要とする場合がある。設定することができるものを以下に挙げる。

**cpu** 現在のCPU占有率を得る。



|                   |                                                                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>cpulimit</b>   | パーセンテージ (0から100) として、最大CPU占有率を制限、取得する。                                                                                                    |
| <b>driver</b>     | 現在使用可能なオーディオ・ドライバー をリストアップする。また、使用するオーディオ・ドライバーを変更することができる。                                                                               |
| <b>info</b>       | トップレベルのDSPチェーンで使われている関数とシグナルの数を得る                                                                                                         |
| <b>input</b>      | <b>input</b> に続けて論理チャンネル (これは <b>adc~</b> オブジェクトで使用される) を定義するアーギュメント (1から512) を必要とする。利用可能なオーディオ・ドライバーの入力チャンネルをリストアップしたり、現在の設定を変更することができる。 |
| <b>iovs</b>       | 現在使用しているオーディオ・ドライバーのI/Oベクター・サイズを取得、変更する。                                                                                                  |
| <b>latency</b>    | オーディオ・ドライバーがサポートしている場合、ドライバーの入出力のレイテンシーを得る。                                                                                               |
| <b>numinputs</b>  | 現在使用しているオーディオ・ドライバーの入力チャンネルの数を得る。                                                                                                         |
| <b>numoutputs</b> | 現在使用しているオーディオ・ドライバーの出力チャンネルの数を得る。                                                                                                         |
| <b>optimize</b>   | 最適化をオン/オフにする。Macintoshでは、これはAltivec (G4のプロセッサ) の最適化をコントロールするのに用いられる。                                                                      |
| <b>option</b>     | オプション番号を指定する第2アーギュメントを必要とする (オプション番号は0から始まる)。現在のオーディオ・ドライバーが、番号のつけられたオプションを利用する場合は、オプションの値の設定として利用可能な選択肢を出力する。                            |
| <b>optionname</b> | オプション番号を指定する第2アーギュメントを必要とする (オプション番号は0から始まる)。現在のオーディオ・ドライバーが、番号のつけられたオプションを利用する場合は、オプションの名前を出力する。                                         |
| <b>output</b>     | <b>output</b> に続けて論理チャンネル (これは <b>dac~</b> オブジ                                                                                            |

エクトで使用される) を定義するアーギュメント (1から512) を必要とする。利用可能なオーディオ・ドライバー の出力チャンネルをリストアップしたり、現在の設定を変更することができる。

|                  |                                                |
|------------------|------------------------------------------------|
| <b>overdrive</b> | オーバードライブ・モードの設定を行う (オーバードライブ・モードは、割り込み優先度が高い)。 |
| <b>sigvs</b>     | 現在のシグナル・ベクター・サイズを取得、設定の変更をする。                  |
| <b>sr</b>        | 現在使用可能なサンプリング・レートを取得、設定の変更をする。                 |
| <b>switch</b>    | DSPのオン/オフを切り替える。                               |
| <b>takeover</b>  | オーディオ・インタラプト・モードにおけるスケジューラーの設定を行う。             |
| <b>timecode</b>  | オーディオ・ドライバー にサポートされている場合、現在のタイムコード値を得る。        |

## 出力

様々なメッセージ 第1アウトレット：多くの設定においては、**reset**または**loadbang**メッセージを**adstatus**オブジェクトに受け取った時に、ポップアップ・メニューオブジェクトを設定するためのメッセージが、第1アウトレットから出力される。詳しくは、入力の項目の**reset**メッセージを参照のこと。**driver**、**input**、**iovs**、**optimize**、**output**、**sigvs**、**sr**、**switch**、**takeover**の設定では、メニュー形式で出力される。

|               |                                                                                                                                                          |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>set</b>    | 第1アウトレット： <b>bang</b> メッセージを受け取った場合や、メニューとしての設定の値を変更した時に、 <b>set</b> に続けてメニューのアイテムのインデックス値 (0から始まる) が出力される。特定の設定時に第1アウトレットから出力される、メニュー形式の出力の詳細は下記の通りである。 |
| <b>driver</b> | 現在選択可能なすべてのオーディオ・ドライバーのリスト。                                                                                                                              |
| <b>input</b>  | 現在使用しているオーディオ・ドライバーのオーディオ入力チャンネルのリスト。                                                                                                                    |

|                   |                                                                       |
|-------------------|-----------------------------------------------------------------------|
| <b>iovs</b>       | 現在使用しているオーディオ・ドライバーのI/Oベクター・サイズのリスト。                                  |
| <b>optimize</b>   | 最適化モードのオン/オフを切り替えるためのOn/Offメニューを作る。                                   |
| <b>option</b>     | 特定のオプションの為の選択肢のリストを作る。                                                |
| <b>optionname</b> | ラベル・モードのポップアップ・メニューオブジェクトと利用する場合のために、特定のオプションの名前のメニューをセットする。          |
| <b>output</b>     | 現在使用しているオーディオ・ドライバーのオーディオ出力チャンネルのリスト。                                 |
| <b>overdrive</b>  | オーバードライブ・モードのオン/オフを切り替えるためのOn/Offメニューを作る。                             |
| <b>sigvs</b>      | 現在使用しているオーディオ・ドライバーのシグナル・ベクター・サイズのリスト。                                |
| <b>sr</b>         | 現在使用しているオーディオ・ドライバーのサンプリング・レートのリスト。                                   |
| <b>switch</b>     | DSPのオン/オフを切り替えるためのOn/Offメニューを作る。                                      |
| <b>takeover</b>   | オーディオ・インタラプト・モードにおけるスケジューラーのオン/オフを切り替えるためのOn/Offメニューを作る。              |
| <b>整数または実数</b>    | 第1アウトレット : 次に挙げるオブジェクトは、メニュー・スタイルの出力を取らず、それぞれの設定の現在の値を第1アウトレットから出力する。 |
| <b>cpu</b>        | 現在のCPU占有率をパーセンテージ（通常は0から100）として得る。                                    |
| <b>cpulimit</b>   | 最大CPU占有率の制限値を得る。                                                      |
| <b>info</b>       | トップレベルのDSPチェーンで使われている関数とシグナルの数を得る。                                    |
| <b>latency</b>    | オーディオ・ドライバー にサポートされている場合、ドライバーの入力のレイテンシーを得る。                          |

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>numinputs</b>  | 現在使用しているオーディオ・ドライバーの入力チャンネルの数を取得する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>numoutputs</b> | 現在使用しているオーディオ・ドライバーの出力チャンネルの数を取得する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>timecode</b>   | オーディオ・ドライバーにサポートされている場合、現在のタイムコード値を取得する。タイムコードは下記のフォーマットのリストとして出力される。 <ol style="list-style-type: none"><li>1. 速度（実数、デフォルトでは1）</li><li>2. タイムコードのサンプルカウントの最上位ワード（整数）</li><li>3. タイムコードのサンプルカウントの最下位ワード（整数）</li><li>4. タイムコードのサブフレーム（整数、80ビット/秒）</li><li>5. LTCと実際のサンプル値との間のサンプルオフセット（整数）</li><li>6. タイムコード・フラグ</li><li>7. フレーム・レート（整数、下記にリストアップされた値）<ul style="list-style-type: none"><li>0 (24 fps)</li><li>1 (25 fps)</li><li>2 (30 fps drop)</li><li>3 (30 fps)</li><li>4 (29.97 fps drop)</li><li>5 (29.97 fps)</li></ul></li></ol> |
| <b>整数または実数</b>    | 第2アウトレット：次に挙げるオブジェクトは、オブジェクトのアウトレットから何らかの値を出力する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>info</b>       | トップレベルのDSPチェーンで使われている関数とシグナルの数を取得する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>iovs</b>       | 現在のI/Oベクター・サイズを取得する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>sigvs</b>      | 現在のシグナル・ベクター・サイズを取得する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>option</b>     | オプションの現在の値のメニュー・アイテムのインデックスを取得する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>switch</b>     | 現在、DSPがオンかオフかを取得する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>takeover</b>   | 現在、テイク・オーバー・モードの設定がオンかオフかを取得する。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

|                   |                                                |
|-------------------|------------------------------------------------|
| <b>input</b>      | 特定の論理チャンネルの現在のインプット・チャンネルを得る。                  |
| <b>output</b>     | 特定の論理チャンネルの現在のアウトプット・チャンネルを得る。                 |
| <b>overdrive</b>  | 現在、オーバードライブ・モードの設定がオンかオフかを得る。                  |
| <b>sr</b>         | 現在のサンプリング・レートを得る。                              |
| <b>numinputs</b>  | 現在使用しているオーディオ・ドライバーの入力チャンネルの数を取得 (第1インレットと同じ)。 |
| <b>numoutputs</b> | 現在使用しているオーディオ・ドライバーの出力チャンネルの数を取得 (第1インレットと同じ)。 |

## 利用例



**adstatus** オブジェクトを使用すれば、パッチからオーディオのパラメーターを変更したり、モニターすることができる

## 参照

**dspstate~** 現在のDSPの設定を出力する  
**adoutput~** オーディオ・ドライバー設定の出力とコントロール

## 入力

**シグナル** 第1インレット：フィルター処理を行う任意のシグナル。このフィルターは次の式に従って現在の入力サンプルを以前の出力サンプルとミックスする：

$$y_n = -g x_n + x_{n-(DR/1000)} + g y_{n-(DR/1000)}$$

$R$ はサンプリング・レートで、 $D$ はミリ秒単位でのディレイ時間である。

第2インレット：過去の出力サンプルを現在の出力に加算する場合のミリ秒単位でのディレイ時間 ( $D$ )

第3インレット：入力と出力のサンプルの総計を出力に送る場合にスケーリングするためのゲイン係数 ( $g$ )

**実数または整数** 第2と第3のインレットのフィルター・パラメーターは、シグナルではなく実数または整数で指定することもできる。そのインレットにシグナルが入力されている場合は、実数または整数での指定は無視される。

**clear** **allpass~** オブジェクトの以前の出力のメモリーをクリアーして、0にリセットする

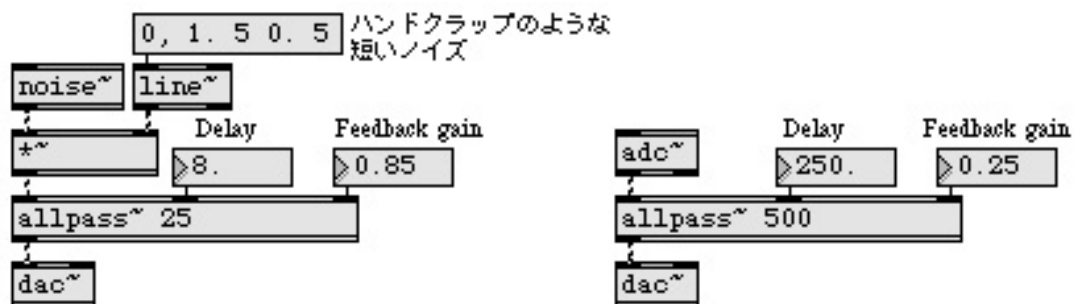
## アーギュメント

**実数** オプション。最大で3つの数値で、最大ディレイ時間とディレイ時間 $D$ とゲイン係数 $g$ の初期値を設定する。いずれかのインレットにシグナルが入力されている場合には、そのインレットにアーギュメントとして与えられた係数は無視される。アーギュメントが存在しない場合には、最大ディレイ時間はデフォルト値の10ミリ秒となる。

## 出力

**シグナル** フィルター処理を行ったシグナル

## 利用例



フィードバック付きの短いディレイは入力サウンドをぼやけさせ、長いディレイは離散的なエコーになる

## 参照

|                |                                           |
|----------------|-------------------------------------------|
| <b>biquad~</b> | 2ポール、2ゼロのフィルター                            |
| <b>comb~</b>   | コム・フィルター                                  |
| <b>lores~</b>  | レゾナンス付きローパス・フィルター                         |
| <b>reson~</b>  | レゾナンス付きバンドパス・フィルター                        |
| <b>teeth~</b>  | フィードフォワードとフィードバックにディレイ・コントロールが付いたコム・フィルター |



## 入力

シグナル アーク・サイン関数への入力。

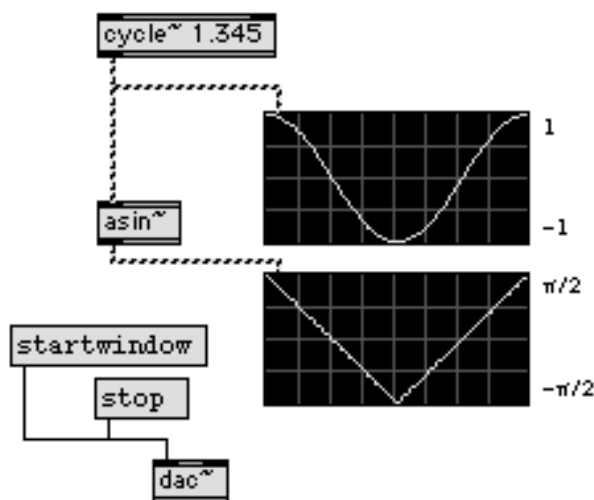
## アーギュメント

なし。

## 出力

シグナル ラディアン単位での入力のアーク・サイン値。

## 利用例



asin~ を使って、ラディアン単位で範囲が $-\pi/2 \sim \pi/2$ のリニア・ランプをつくる

## 参照

|               |                         |
|---------------|-------------------------|
| <b>acos~</b>  | シグナルのアーク・コサイン関数         |
| <b>atan~</b>  | シグナルのアーク・タンジェント関数       |
| <b>atan2~</b> | シグナルのアーク・タンジェント関数 (2変数) |
| <b>cos~</b>   | シグナル (0-1の範囲) のコサイン関数   |
| <b>cosh~</b>  | シグナルの双曲線コサイン関数          |
| <b>cosx~</b>  | シグナルのコサイン関数             |
| <b>sinh~</b>  | シグナルの双曲線サイン関数           |
| <b>sinx~</b>  | シグナルのサイン値               |
| <b>tanh~</b>  | シグナルの双曲線タンジェント関数        |
| <b>tanx~</b>  | シグナルのタンジェント関数           |

## 入力

シグナル 双曲線アーク・サイン関数への入力。

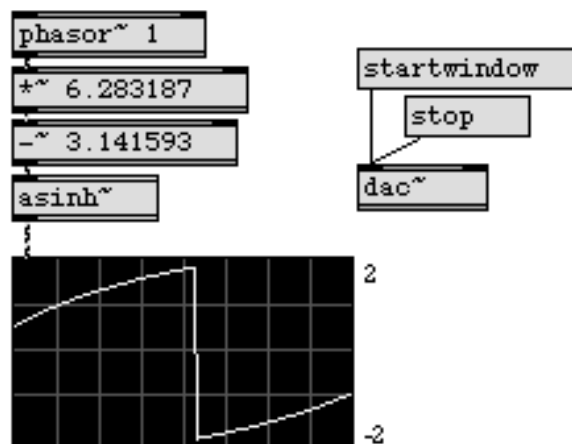
## アーギュメント

なし。

## 出力

シグナル ラディアン単位での入力の双曲線アーク・サイン値。

## 利用例



## 参照

- acos~** シグナルのアーク・コサイン関数
- acosh~** シグナルの双曲線アーク・コサイン関数
- asin~** シグナルのアーク・サイン関数
- atan~** シグナルのアーク・タンジェント関数
- atan2~** シグナルのアーク・タンジェント関数 (2変数)
- cos~** シグナル (0-1の範囲) のコサイン関数
- cosh~** シグナルの双曲線コサイン関数
- cosx~** シグナルのコサイン関数
- sinh~** シグナルの双曲線サイン関数
- sinx~** シグナルのサイン関数
- tanh~** シグナルの双曲線タンジェント関数
- tanx~** シグナルのタンジェント関数

## 入力

シグナル アーク・タンジェント関数への入力。

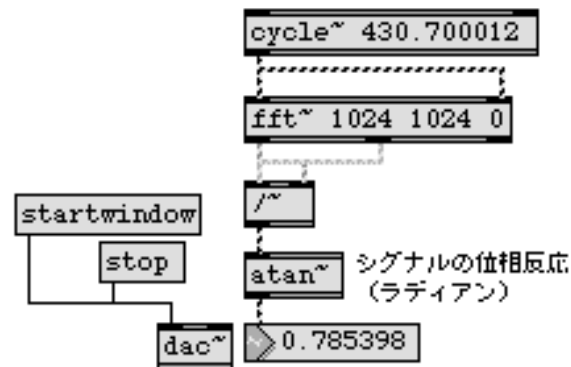
## アーギュメント

なし。

## 出力

シグナル 入力のアーキ・タンジェント値。

## 利用例



atan~を用いて、シグナルのアーキ・タンジェント関数を計算する

## 参照

|               |                         |
|---------------|-------------------------|
| <b>acos~</b>  | シグナルのアーキ・コサイン関数         |
| <b>asin~</b>  | シグナルのアーキ・サイン関数          |
| <b>atan2~</b> | シグナルのアーキ・タンジェント関数 (2変数) |
| <b>cos~</b>   | シグナル (0-1の範囲) のコサイン関数   |
| <b>cosh~</b>  | シグナルの双曲線コサイン関数          |
| <b>cosx~</b>  | シグナルのコサイン関数             |
| <b>sinh~</b>  | シグナルの双曲線サイン関数           |
| <b>sinx~</b>  | シグナルのサイン値               |
| <b>tanh~</b>  | シグナルの双曲線タンジェント関数        |
| <b>tanx~</b>  | シグナルのタンジェント関数           |

## 入力

シグナル 第1インレット：アーク・タンジェント関数へ入力する $x$ 値。

第2インレット：アーク・タンジェント関数へ入力する $y$ 値。

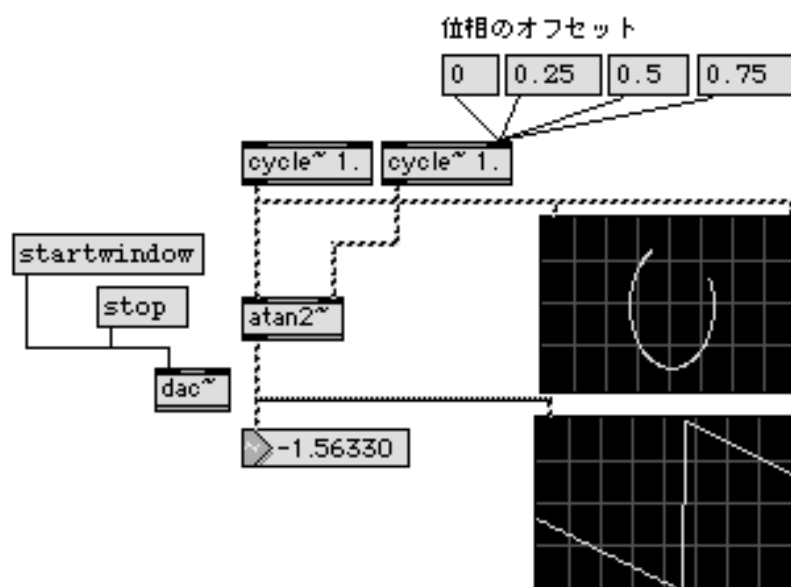
## アーギュメント

なし。

## 出力

シグナル 入力のアーク・タンジェント値(つまり、 $y/x$ のアークタンジェント値)。

## 利用例



原点(0, 0)の周りの2点の角度をラディアン単位で計算する

## 参照

|              |                       |
|--------------|-----------------------|
| <b>acos~</b> | シグナルのアーク・コサイン関数       |
| <b>asin~</b> | シグナルのアーク・サイン関数        |
| <b>atan~</b> | アーク・タンジェント関数          |
| <b>cos~</b>  | シグナル (0-1の範囲) のコサイン関数 |
| <b>cosh~</b> | シグナルの双曲線コサイン関数        |
| <b>cosx~</b> | シグナルのコサイン関数           |
| <b>sinh~</b> | シグナルの双曲線サイン関数         |
| <b>sinx~</b> | シグナルのサイン値             |
| <b>tanh~</b> | シグナルの双曲線タンジェント関数      |
| <b>tanx~</b> | シグナルのタンジェント関数         |

## 入力

シグナル 双曲線アーク・タンジェント関数への入力。

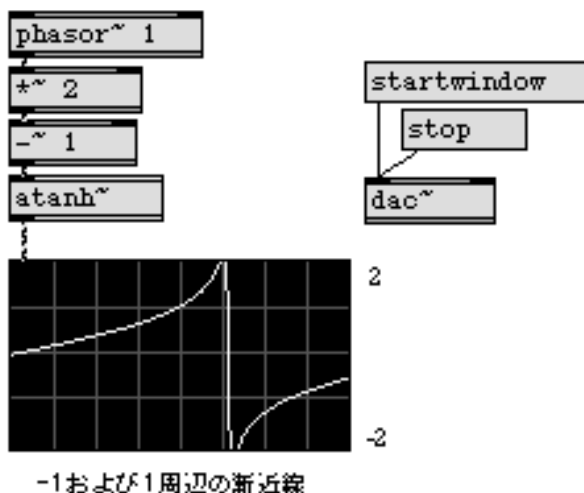
## アーギュメント

なし。

## 出力

シグナル 入力の双曲線アーク・タンジェント値。

## 利用例



-1および1周辺の漸近線

## 参照

- acos~** シグナルのアーク・コサイン関数
- acosh~** シグナルの双曲線アーク・コサイン関数
- asin~** シグナルのアーク・サイン関数
- asinh~** シグナルの双曲線アーク・サイン関数
- atan~** シグナルのアーク・タンジェント関数
- atan2~** シグナルのアーク・タンジェント関数 (2変数)
- cos~** シグナル (0-1の範囲) のコサイン関数
- cosh~** シグナルの双曲線コサイン関数
- cosx~** シグナルのコサイン関数
- sinh~** シグナルの双曲線サイン関数
- sinx~** シグナルのサイン値
- tanh~** シグナルの双曲線タンジェント関数
- tanx~** シグナルのタンジェント関数

## 入力

|          |                                                                                                                                                                                               |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| シグナル     | 平均化しようとするシグナル                                                                                                                                                                                 |
| 整数       | シグナル平均化の3つのモードで用いられるインターバルをサンプル単位で設定する                                                                                                                                                        |
| bipolar  | バイポーラー平均化モード（デフォルト）に設定する。バイポーラー・モードではサンプルの数値が平均化される。                                                                                                                                          |
| absolute | 絶対値平均化モードに設定する。このモードは入力されるサンプルの絶対値を平均化する。                                                                                                                                                     |
| rms      | 二乗平均平方根（RMS）平均化モードに設定する。このモードはサンプルの値を二乗した平均の平方根を計算する。 <b>average~</b> オブジェクトのRMSモードはバイポーラーや絶対値モードと比較してよりCPUの負荷が高い。RMSの値はシグナルのレベルを計測するためによく用いられるが、多くのレベル検出タスクにおいて絶対値モードもRMSモードと同じように適した機能を果たす。 |

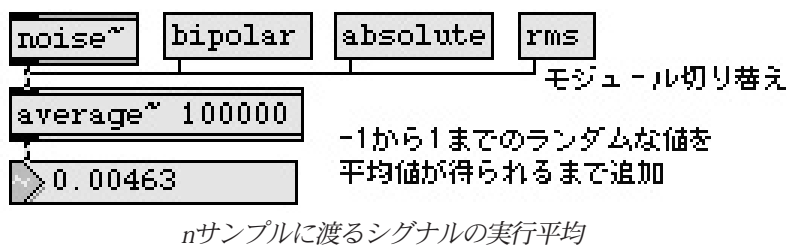
## アーギュメント

|        |                                                    |
|--------|----------------------------------------------------|
| 整数     | オプション。最大の平均化インターバルをサンプル単位で設定する。デフォルト値は100である。      |
| symbol | オプション。上記のように平均化モードを設定する。デフォルトは <b>bipolar</b> である。 |

## 出力

|    |                                 |
|----|---------------------------------|
| 実数 | 指定された数のサンプルを通して平均化された入力信号の実行平均値 |
|----|---------------------------------|

## 利用例



## 参照

|                |                                |
|----------------|--------------------------------|
| avg~<br>meter~ | シグナルの平均値<br>視覚的なピーク・レベル・インジケータ |
|----------------|--------------------------------|

## 入力

**bang** 前回に**bang**を受け取った時から現在までに入力されたシグナルの振幅の平均値（絶対値）を出力する。また次の平均値の出力に備えて、**avg~**オブジェクトのメモリを消去する。

**シグナル** 平均値を求めるシグナル。

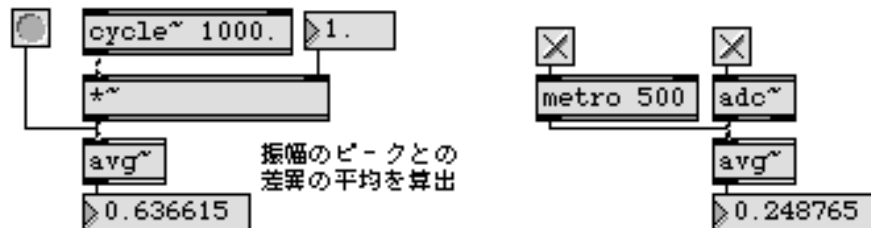
## アーギュメント

なし。

## 出力

**実数** インレットに**bang**を受け取ると、前回に**bang**を受け取った時から現在までに入力されたシグナルの振幅の平均値を出力する。

## 利用例



シグナルの振幅の平均値（絶対値）を出力する

## 参照

**average~**  
**meter~**

マルチ・モードのシグナル平均化  
視覚的なピーク・レベル・インジケータ



# begin~

シグナル・ネットワークの切り替え可能な部分を定義する

## 入力

なし。

## アーギュメント

なし。

## 出力

**シグナル** **begin~**は値が0である一定のシグナルを出力するオブジェクトで、不必要な時に処理を止めたいシグナル・ネットワークの一部分の開始部分を定義するために用いる。**begin~**のアウトレットをオブジェクトのシグナル・インレットに接続することで、シグナル・ネットワークの開始部分が定義され、**gate~**または**selector~**まで続くことになる。1個の**begin~**は、**gate~**または**selector~**のシグナル・インレットに対して有効である。**gate~**や**selector~**に入ってくるシグナルが遮断されると、**begin~**と**gate~**または**selector~**の間のシグナル・ネットワークにあるすべてのオブジェクトの処理は行われなくなる。

## 利用例



## 参照

### selector~ gate~

複数の入力を切り替えて、1つのアウトレットから出力する  
複数アウトレットの1つへシグナルを切り替える

Tutorial 5

Fundamentals: Turning signals on and off (英文PDF参照)

## 入力

**シグナル** 第1インレット：フィルター処理を行うシグナル。このフィルターは現在の入力サンプルと、2つの以前の入力サンプルと、2つの以前の出力サンプルを次の計算式に従ってミックスする。

$$\text{計算式: } y_n = a_0x_n + a_1x_{n-1} + a_2x_{n-2} - b_1y_{n-1} - b_2y_{n-2}$$

第2インレット：出力にダイレクトに通過させる現在の入力の量をスケールリングするための振幅係数 $a_0$

第3インレット：出力に加算される1つ前の入力サンプルの量をスケールリングするための振幅係数 $a_1$

第4インレット：出力に加算される2つ前の入力サンプルの量をスケールリングするための振幅係数 $a_2$

第5インレット：現在の出力に加算される1つ前の出力の量をスケールリングするための振幅係数 $b_1$

第6インレット：現在の出力に加算される2つ前の出力サンプルの量をスケールリングするための振幅係数 $b_2$

**実数** インレット2からインレット6までの係数は、シグナルのかわりに実数でも指定することができる。そのインレットにシグナルが入力されている場合は、実数での指定は無視される。

**リスト** 5つの係数は第1インレットにリストとして入力することもできる。リストの最初の数値は $a_0$ 、次は $a_1$ というようになる。いずれかのインレットにシグナルが入力されている場合は、そのインレットに対してリストで与えた係数は無視される。

**clear** **biquad~**オブジェクトの以前の入力と出力のメモリーを消去し、 $x_{n-1}$ 、 $x_{n-2}$ 、 $y_{n-1}$ そして $y_{n-2}$ を0にリセットする。

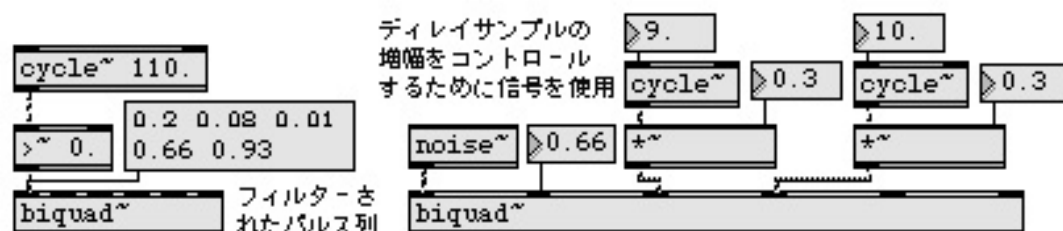
## アーギュメント

**実数** オプション。最大で5つの数値で、係数 $a_0$ 、 $a_1$ 、 $a_2$ 、 $b_1$ そして $b_2$ の初期値を設定する。いずれかのインレットにシグナルが入力されている場合は、そのインレットに対してアーギュメントとして与えられた係数は無視される。

## 出力

シグナル フィルター処理を行ったシグナル。

## 利用例



フィルター係数は、数値または変動するシグナルとして指定することができる

## 参照

|                     |                                           |
|---------------------|-------------------------------------------|
| <b>buffir~</b>      | <b>buffer~</b> を利用したFIRフィルター              |
| <b>comb~</b>        | コム・フィルター                                  |
| <b>filtergraph~</b> | グラフィカル・フィルター・エディター                        |
| <b>lores~</b>       | レゾナンス付きローパス・フィルター                         |
| <b>onepole~</b>     | 単極のローパス・フィルター                             |
| <b>reson~</b>       | レゾナンス付きバンドパス・フィルター                        |
| <b>teeth~</b>       | フィードフォワードとフィードバックにディレイ・コントロールが付いたコム・フィルター |

**bitand~**オブジェクトは、2つの入力される浮動小数点シグナルに対して、生の32ビット・データまたは整数値として、ビット単位のAND演算を行う。出力は両方の数値で1であるビットからなる浮動小数点シグナルである。

## 入力

**シグナル** 第1インレット：バイナリー形式で第2インレットからの浮動小数点シグナルと比較される浮動小数点シグナル。このシグナルは浮動小数点シグナルとしても整数としても扱うことができる。

第2インレット：第1インレットのシグナルと比較される浮動小数点シグナル。このシグナルは浮動小数点シグナルとしても整数としても扱うことができる。

生の浮動小数点シグナルのビットの値は次の形式で表現される。

<1符号ビット> <8指数ビット> <23仮数ビット>

**整数** 第2インレット：適当なモードが設定されると、**bitand~**オブジェクトの第2インレットに入力された整数値は、ビットマスクとして用いられる。

**bits** 第1インレット：32個の1または0を含むリストを伴う**bits**は、**bitand~**で用いられるビットマスクを指定する。この代わりに、ビットマスクの値は、第2インレットに入力された整数値を用いて設定することもできる。

**mode** 第1インレット：0から3までの数値を伴う**mode**は、ビット単位での演算を行う場合に、浮動小数点シグナルまたは浮動小数点の値を、生の32ビット浮動小数点の値と変換された整数値のどちらとして処理するのかを指定する。演算のモードは以下の通り。

| モード | 説明                                                 |
|-----|----------------------------------------------------|
| 0   | 両方の浮動小数点シグナル入力を、生の32ビット値として扱う（デフォルト）。              |
| 1   | 両方の浮動小数点シグナル入力を、整数の値に変換する。                         |
| 2   | 第1インレットの浮動小数点シグナルを生の32ビット値として扱い、第2インレットの値を整数値として扱う |
| 3   | 第1インレットの浮動小数点シグナルを整数値に変換し、第2インレットを生の32ビット値として扱う。   |

注記：浮動小数点シグナルを整数に変換した後で元に戻した場合には、結果の浮動小数点の値は24ビット分の整数分解能しか持つことができない。

## アーギュメント

**整数** オプション。**bitand~**で用いられるビットマスクを設定する。デフォルト値は0である。整数値はモードに関係なくビットマスクとして用いることができる。この場合、整数値のバイナリー表現がビットマスクとなる。

**整数** オプション。ビット単位での演算を行う場合に、浮動小数点シグナルまたは浮動小数点の値を、生の32ビット浮動小数点の値と変換された整数値のどちらとして処理するのかを指定する。演算のモードは以下の通り。

| モード | 説明                                                  |
|-----|-----------------------------------------------------|
| 0   | 両方の浮動小数点シグナル入力を、生の32ビット値として扱う（デフォルト）。               |
| 1   | 両方の浮動小数点シグナル入力を、整数の値に変換する。                          |
| 2   | 第1インレットの浮動小数点シグナルを生の32ビット値として扱い、第2インレットの値を整数値として扱う。 |
| 3   | 第1インレットの浮動小数点シグナルを整数値に変換し、第2インレットを生の32ビット値として扱う。    |

## 出力

**シグナル** インレットの2つの浮動小数点シグナルまたは整数は、ビットごとに比較される。あるビットが両方の数値で1であれば、出力の数値の該当ビットも1になり、そうでない場合には出力の浮動小数点シグナルの該当ビットは0になる。

## 利用例



## 参照

**bitshift~**

**bitor~**

**bitxor~**

**bitnot~**

浮動小数点シグナルに対するビット・シフト

浮動小数点シグナルに対するビット単位のOR演算

浮動小数点シグナルに対するビット単位のXOR演算

浮動小数点シグナルに対するビットのNOT演算

**bitnot~**オブジェクトは、入力される浮動小数点シグナルに対して、生の32ビット・データまたは整数値としてビットのNOT演算を行う。値が1であるすべてのビット値は0になり、逆もまた同様である。

## 入力

**シグナル** **bitnot~**オブジェクトは浮動小数点シグナルまたは整数としてビットのNOT演算を行う。

浮動小数点シグナルのビット値は次の形式で表現される。

<1符号ビット> <8 指数ビット> <23 仮数ビット>

**mode** 0または1を伴う**mode**は、ビットのNOT演算の場合に浮動小数点シグナルまたは浮動小数点の値を、生の32ビット浮動小数点の値または変換された整数値のどちらとして扱うかを指定する。演算のモードは以下の通り。

| モード | 説明                                 |
|-----|------------------------------------|
| 0   | 浮動小数点シグナル入力を、生の32ビット値として扱う（デフォルト）。 |
| 1   | 浮動小数点シグナル入力を、整数値に変換する。             |

注記：浮動小数点シグナルを整数に変換した後で元に戻した場合には、結果の浮動小数点の値は24ビット分の整数分解能しか持つことができない。

## アーギュメント

**整数** オプション。ビットのNOT演算の場合に、浮動小数点シグナルまたは浮動小数点の値を、生の32ビット浮動小数点の値または変換された整数値のどちらとして扱うかを指定する。演算のモードは以下の通り。

| モード | 説明                                 |
|-----|------------------------------------|
| 0   | 浮動小数点シグナル入力を、生の32ビット値として扱う（デフォルト）。 |
| 1   | 浮動小数点シグナル入力を、整数値に変換する。             |

## 出力

**シグナル** ビットのNOT演算後の浮動小数点シグナル

## 利用例



bitnot~の例

## 参照

**bitshift~**

浮動小数点シグナルに対するビット・シフト

**bitor~**

浮動小数点シグナルに対するビット単位のOR演算

**bitxor~**

浮動小数点シグナルに対するビット単位のXOR演算

**bitand~**

浮動小数点シグナルに対するビット単位のAND演算



**bitor~**オブジェクトは2つの入力される浮動小数点シグナルを、生の32ビット値または整数値として。ビット単位のOR演算を行う。両方の入力シグナルはビットごとに比較され、2つビットのうちどちらかの値が1であれば1が出力となる。出力は結果のビット・パターンからなる浮動小数点シグナルである。

## 入力

**シグナル** 第1インレット：第2インレットの浮動小数点シグナルとバイナリー形式で比較される浮動小数点シグナル。シグナルは浮動小数点シグナルとしても整数としても扱うことができる。

第2インレット：第1インレットのシグナルと比較される浮動小数点シグナル。シグナルは浮動小数点シグナルとしても整数としても扱うことができる。

生の浮動小数点シグナルは次の形式で表現される。

<1符号ビット> <8指数ビット> <23仮数ビット>

**整数** 第2インレット：適当なモードが設定されると、**bitor~**オブジェクトの第2インレットに入力された整数値はビットマスクとして用いられる。

**bits** 第1インレット：32個の1または0を含むリストを伴う**bits**は、**bitor~**で用いられるビットマスクを指定する。この代わりに、ビットマスクの値は、第2インレットに入力された整数値を用いて設定することもできる。

**mode** 第1インレット：0から3までの数値を伴う**mode**は、ビット単位での演算を行う場合に、浮動小数点シグナルまたは浮動小数点の値を生の32ビット浮動小数点の値と変換された整数値のどちらとして処理するかを指定する。演算のモードは以下の通り。

| モード | 説明                                                  |
|-----|-----------------------------------------------------|
| 0   | 両方の浮動小数点シグナル入力を、生の32ビット値として扱う（デフォルト）。               |
| 1   | 両方の浮動小数点シグナル入力を、整数の値に変換する。                          |
| 2   | 第1インレットの浮動小数点シグナルを生の32ビット値として扱い、第2インレットの値を整数値として扱う。 |
| 3   | 第1インレットの浮動小数点シグナルを整数値に変換し、第2インレットを生の32ビット値として扱う。    |

注記：浮動小数点シグナルを整数に変換した後で元に戻した場合には、結果の浮動小数点の値は24ビット分の整数分解能しか持つことができない。

## アーギュメント

**整数** オプション。**bitor~**で用いられるビットマスクを設定する。デフォルト値は0である。整数値はモードに関係なくビットマスクとして用いることができる。この場合、整数値のバイナリー表現がビットマスクとなる。

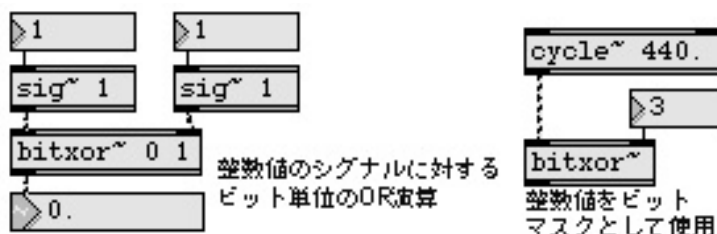
**整数** オプション。ビット単位での演算を行う場合に浮動小数点シグナルまたは浮動小数点の値を生の32ビット浮動小数点の値と変換された整数値のどちらとして処理するのかを指定する。演算のモードは以下の通り。

| モード | 説明                                                  |
|-----|-----------------------------------------------------|
| 0   | 両方の浮動小数点シグナル入力を、生の32ビット値として扱う（デフォルト）。               |
| 1   | 両方の浮動小数点シグナル入力を、整数の値に変換する。                          |
| 2   | 第1インレットの浮動小数点シグナルを生の32ビット値として扱い、第2インレットの値を整数値として扱う。 |
| 3   | 第1インレットの浮動小数点シグナルを整数値に変換し、第2インレットを生の32ビット値として扱う。    |

## 出力

**シグナル** インレットの2つの浮動小数点シグナルまたは整数は、ビットごとに比較される。数値のうちどちらか一方のビットが1であれば、出力の数値の該当ビットも1になり、そうでない場合には出力の数値の該当ビットは0になる。出力は結果のビット・パターンからなる浮動小数点シグナルである。

## 利用例



## 参照

- bitshift~** 浮動小数点シグナルに対するビット・シフト
- bitand~** 浮動小数点シグナルに対するビット単位のAND演算
- bitxor~** 浮動小数点シグナルに対するビット単位のXOR演算
- bitnot~** 浮動小数点シグナルに対するビットのNOT演算

## 入力

**シグナル** **bitshift~** オブジェクトは浮動小数点シグナルに対して、生の32ビットデータまたは整数値としてビット・シフトを行う。

浮動小数点シグナルのビットの値は次の形式で表現される。

<1符号ビット> <8 指数ビット> <23 仮数ビット>

**mode** 第1インレット：0または1を伴うmodeは、ビット・シフトを行う場合に、浮動小数点シグナルまたは浮動小数点の値を、生の32ビット浮動小数点の値と変換された整数値のどちらとして処理するのかを指定する。演算のモードは以下の通り。

| モード | 説明                                    |
|-----|---------------------------------------|
| 0   | 両方の浮動小数点シグナル入力を、生の32ビット値として扱う（デフォルト）。 |
| 1   | 両方の浮動小数点シグナル入力を、整数の値に変換する。            |

注記：浮動小数点シグナルを整数に変換した後で元に戻した場合には、結果の浮動小数点の値は24ビット分の整数分解能しか持つことができない。

**shift** 第1インレット：正または負の数値を伴う**shift**は、入力される浮動小数点シグナルを何ビットシフトするかを指定する。正の数値はビット数分だけ左にシフトすることに相当する（すなわち数値を左に $n$ 桁シフトすることは2を掛けるのと同じ）。負の数値はビット数分だけ右にシフトすることに相当する（すなわち右に $n$ 桁シフトすることは $2^n$ で割るのと同じ）。

## アーギュメント

**整数** オプション。入力される浮動小数点シグナルを何ビットシフトするかを設定する。正のシフト値は数値のビットを左にシフトすることに相当し、負のシフト値は数値のビットを右にシフトすることに相当する。

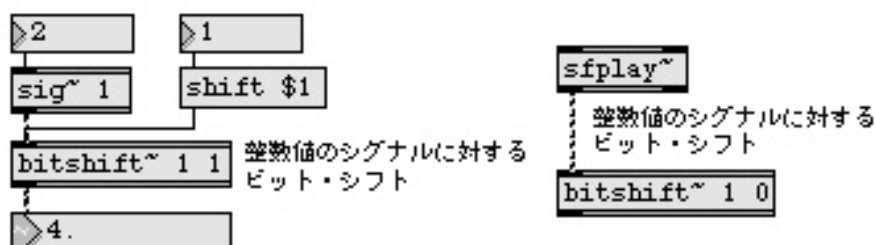
**整数** オプション。ビット・シフトを行う場合に、浮動小数点シグナルまたは浮動小数点の値を生の32ビット浮動小数点の値と変換された整数値のどちらとして処理するのかを指定する。演算のモードは以下の通り。

| モード | 説明                                    |
|-----|---------------------------------------|
| 0   | 両方の浮動小数点シグナル入力を、生の32ビット値として扱う（デフォルト）。 |
| 1   | 両方の浮動小数点シグナル入力を、整数の値に変換する。            |

## 出力

シグナル ビットシフトを行った浮動小数点シグナル

## 利用例



## 参照

**bitand~**  
**bitor~**  
**bitxor~**  
**bitnot~**

浮動小数点シグナルに対するビット単位のAND演算  
浮動小数点シグナルに対するビット単位のOR演算  
浮動小数点シグナルに対するビット単位のXOR演算  
浮動小数点シグナルに対するビットのNOT演算

**bitxor~** オブジェクトは、2つの入力される浮動小数点シグナルに対して、生の32ビット・データ、または整数値として、ビット単位のXOR演算を行う。両方の入力シグナルは比較され、相当する出力ビットは、2つのビットの値が異なる場合には**1**に、2つの値が等しい場合には**0**に設定される。出力は結果のビット・パターンからなる浮動小数点シグナルである。

## 入力

**シグナル** 第1インレット：バイナリー形式で第2インレットの浮動小数点シグナルと比較される浮動小数点シグナル。このシグナルは浮動小数点シグナルとしても整数としても扱うことができる。

第2インレット：第1インレットのシグナルと比較される浮動小数点シグナル。このシグナルは浮動小数点シグナルとしても整数としても扱うことができる。生の浮動小数点シグナルのビットの値は、次の形式で表現される。

<1符号ビット> <8指数ビット> <23仮数ビット>

**整数** 第2インレット：適当なモードが設定されると、**bitxor~** オブジェクトの第2インレットに入力された整数値は、ビットマスクとして用いられる。

**bits** 第1インレット：32個の1または0を含むリストを伴う**bits**は、**bitxor~** で用いられるビットマスクを指定する。この代わりに、ビットマスクの値は、第2インレットに入力された整数値を用いて設定することもできる。

**mode** 第1インレット：0から3までの数値を伴う**mode**は、ビット単位での演算を行う場合に、浮動小数点シグナルまたは浮動小数点の値を生の32ビット浮動小数点の値と変換された整数値のどちらとして処理するかを指定する。演算のモードは以下の通り。

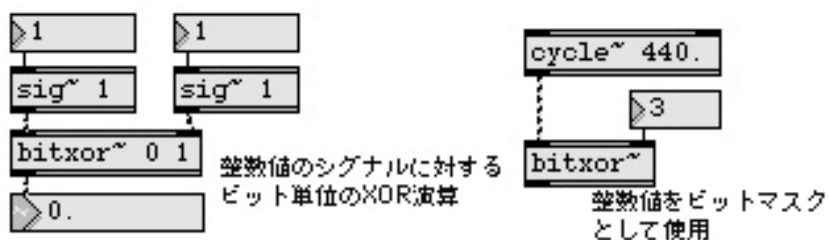
| モード | 説明                                                  |
|-----|-----------------------------------------------------|
| 0   | 両方の浮動小数点シグナル入力を、生の32ビット値として扱う（デフォルト）。               |
| 1   | 両方の浮動小数点シグナル入力を、整数の値に変換する。                          |
| 2   | 第1インレットの浮動小数点シグナルを生の32ビット値として扱い、第2インレットの値を整数値として扱う。 |
| 3   | 第1インレットの浮動小数点シグナルを整数値に変換し、第2インレットを生の32ビット値として扱う。    |

注記：浮動小数点シグナルを整数に変換した後で元に戻した場合には、結果の浮動小数点の値は24ビット分の整数分解能しか持つことができない。

## 出力

**シグナル** インレットの2つの浮動小数点シグナルまたは整数はビットごとに比較される。2つのビットの値が異なれば1が出力となり、同じであれば0が出力となる。出力は結果のビット・パターンからなる浮動小数点シグナルである。

## 利用例



## 参照

**bitshift~****bitand~****bitor~****bitnot~**

浮動小数点シグナルに対するビット・シフト  
 浮動小数点シグナルに対するビット単位のAND演算  
 浮動小数点シグナルに対するビット単位のOR演算  
 浮動小数点シグナルに対するビットのNOT演算



## 入力

- clear** **buffer~**オブジェクトに格納されたデータを消去する。
- filetype** 第1インレット：**filetype**に続き、読み込むオーディオ・ファイルのフォーマット名を表すシンボルを指定する。フォーマットのデフォルトはAIFFである。サポートされているファイル・タイプは次の通りである。
- |             |                                               |
|-------------|-----------------------------------------------|
| <b>aiff</b> | Apple Interchange File Format 形式のファイル (デフォルト) |
| <b>sd2</b>  | Sound Designer II 形式のファイル (Macintoshのみ)       |
| <b>wave</b> | WAVE 形式のファイル                                  |
| <b>raw</b>  | raw 形式のファイル                                   |
| <b>au</b>   | NeXT/Sun で利用されてきた au 形式のファイル                  |
- import** QuickTimeの機能を利用して、オーディオ・メディアを含むファイル (MP3を含む) から、オーディオ・データを**buffer~**のメモリへ読み込む。**import**に続いて**buffer~**に読み込むファイル名を指定した場合、ファイルがMaxのサーチ・パス内にあれば即座にデータが読み込まれる。**import**に続いてファイル名を指定しない場合、標準ファイル・ダイアログが表示されるので、それに従ってファイルを選択する。インポートされたデータは元のファイルが持つサンプリング・レートとワード・サイズ (量子化ビット数) を保持するが、ループ・ポイントやマーカーなどのデータはインポートされない。ファイル名を指定した時には、それに続いて実数を指定することで、データを読み込む開始点をミリ秒単位で指定することができる (ファイルの先頭位置は0)。但し、**import**メッセージはQuickTimeの機能を利用して内部的に1/600秒時間単位で処理されるため、ミリ秒単位で設定した開始位置のオフセットが保証されるわけではない。データの開始点に続いてミリ秒を表す1つの実数を設定することもできる。この値は**buffer~**に読み込むデータの開始点からのデュレーション (時間長) となる。このデュレーションは現在のオブジェクトが確保しているサンプル用のメモリ・サイズを上書きする。デュレーション値を負の数で設定した場合、**buffer~**はファイルのすべてのデータを読み込み、サンプル用のメモリをそのサイズに合わせて確保し直す。デュレーション値が0または与えられなかった場合は、読み込むオーディオ・ファイルのサイズが、現在の**buffer~**内部のメモリサイズより大きくても、サイズ変更はしない。デュレーション値の後には、読み込むチャンネル数を指定することができる。チャンネル数を指定しない場合、オーディオ・ファイルのヘッダに示されているチャンネル数でファイルを読み込む。**read**メッセージにてチャンネル数を指定するかしないかに関わらず、元々**buffer~**で設定されていたチャンネル数は変更され、新しいファイルに従ったチャンネル数となる。
- name** **name**に続き、**buffer~**を参照する時の名前となるシンボルを指定する。これは**buffer~**に関連するオブジェクト (例：**cycle~**、**groove~**、



**lookup~**、**peek~**、**play~**、**record~**、**wave~**など)が、参照先の**buffer~**を区別するためのシンボルとなる。**buffer~**側でシンボルを設定し直した場合、古いシンボルで**buffer~**を参照していたオブジェクトは**buffer~**への繋がりを失う。それぞれの**buffer~**オブジェクトは、個別の参照名を設定すべきである。仮に既に存在する**buffer~**と同じ名前のシンボルを設定しても、その**buffer~**は既に存在する**buffer~**と関連付けられるわけではない。

- open** **buffer~**内のサンプル・データの波形表示ウィンドウが開く。既に波形表示ウィンドウが開かれている場合は、一番手前に表示される。
- read** サポートするサウンド・ファイル (AIFF形式、Sound DesignerII形式 (Macintoshのみ)、NeXT/Sun形式 (au形式)、WAV形式) を**buffer~**オブジェクト内部のサンプル・メモリへ読み込む。**read**に続いて読み込むファイル名を指定でき、Maxのサーチ・パスにそのファイルが存在すれば、即座に読み込まれる。ファイル名を指定しない場合は、標準ファイル・ダイアログが開くので、そこでファイルを選択する。そのファイル名を指定する場合は、その後の実数を指定し、データを読み込む開始点をミリ秒単位で指定することができる (ファイルの先頭位置は0)。そのデータ開始点に続いてミリ秒を表す1つの実数を設定することもできる。この値は**buffer~**に読み込むデータの開始点からのデュレーション (時間長) となる。このデュレーションは現在のオブジェクトが確保しているサンプル用のメモリ・サイズを変更する。デュレーション値を負の数で設定した場合、**buffer~**はファイルのすべてのデータを読み込み、サンプル用のメモリをそのサイズに合わせて確保し直す。デュレーション値が0または与えられなかった場合は、読み込むオーディオ・ファイルのサイズが現在の**buffer~**内部のメモリサイズより大きくてもサイズ変更はしない。デュレーション値の後には、読み込むチャンネル数を指定することができる。チャンネル数を指定しない場合、オーディオ・ファイルのヘッダに示されているチャンネル数に従ってファイルを読み込む。**read**メッセージによってチャンネル数を指定するかしないかに関わらず、元々**buffer~**で設定されていたチャンネル数は変更され、新しいファイルに従ったチャンネル数となる。
- readagain** 一番最近に読み込んだファイルからサウンド・データを読み込む (ここでは**read**または**replace**メッセージで1つ前に読み込まれたデータとなる)。
- replace** 負の数でデュレーションを指定した**read**メッセージと同様の働きをする。**replace**の後にシンボルを指定することで、それをファイル名とするファイルがMaxサーチ・パス内であれば、そのファイルを読み込む。シンボルの記述がない場合、標準ファイル・ダイアログが開き、選択可能なオーディオファイルを表示する。開始点、デュレーション、チャンネル数の指定などは**read**メッセージと同様に利用できる。
- samptype** 第1インレット : **samptype**に続きサンプルのデータ型を表すシンボルを

指定する。このデータ型はオーディオ・ファイルのサンプル・データをどのように解釈するかを表すものとなる（実際のサンプル・データ型を無視することになる）。

次のサンプル・データ型を表すシンボルがサポートされている。

|                |                     |
|----------------|---------------------|
| <b>int8</b>    | 8bit 整数型            |
| <b>int16</b>   | 16bit 整数型           |
| <b>int24</b>   | 24bit 整数型           |
| <b>int32</b>   | 32bit 整数型           |
| <b>float32</b> | 32bit 浮動小数点型        |
| <b>float64</b> | 64bit 浮動小数点型        |
| <b>mulaw</b>   | 8bit $\mu$ -law 符号型 |
| <b>alaw</b>    | 8bit a-law 符号型      |

- set** **set**に続き、**buffer~**を参照する時の名前となるシンボルを指定する。これは**buffer~**に関連するオブジェクト（例：**cycle~**、**groove~**、**lookup~**、**peek~**、**play~**、**record~**、**wave~**など）が、参照先の**buffer~**を区別するためのシンボルとなる。**buffer~**側でシンボルを設定し直した場合、古いシンボルで**buffer~**を参照していたオブジェクトはその**buffer~**への繋がりを失う。それぞれの**buffer~**オブジェクトは個別の名前を設定すべきである。仮に既に存在する**buffer~**と同じ名前のシンボルを設定しても、その**buffer~**は既に存在する**buffer~**と関連付けられるわけではない。
- size** **size**に続いてミリ秒単位で表したデュレーション（時間長）を指定する。このデュレーション値は、**buffer~**が確保するサンプル・メモリのサイズを設定する。これは1つの**buffer~**オブジェクトで格納できるデータ量を制限するものである。この制限は、**replace**メッセージを入力するか、**read**メッセージでのデュレーション値設定が行われるまで有効となる。
- sr** **sr**に続いて**buffer~**内部で処理されるサンプリング・レートの数値を指定する。デフォルトではサンプリング・レートはその時点での出力サンプリング・レート、または一番最近に読み込んだオーディオ・ファイルのサンプリング・レートとなる。
- wclose** サンプル・データの波形表示ウィンドウが表示されている場合に、それを閉じる。
- write** **buffer~**内に格納されているデータをオーディオ・ファイルにセーブする。**write**の後にシンボルが指定されていない場合は、標準ファイル保存ダイアログが表示され、カレント・ディレクトリを示して、保存するファイル名を問い合わせてくる。その標準ファイル保存ダイアログで保存するファイルのフォーマットを特に指定しない限り、セーブされるのは**filetype**メッセージでもっとも最近に設定されたフォーマットか、もっとも最近に開いたオーディオ・ファイルのフォーマットとなる。デフォルトでは**buffer~**はAIFF形式でセーブする。

- writeaiff** **buffer~**内に格納されているデータをAIFFファイルとして保存する。**writeaiff**に続いてファイル名となるシンボルを指定しない場合は、標準ファイル保存ダイアログが開く。シンボルを指定した場合は、現在のデフォルト・ディレクトリに指定したファイル名で保存される。
- writeau** **buffer~**内に格納されているデータをNeXT/Sun形式として保存する。**writeau**に続いてファイル名となるシンボルを指定しない場合は、標準ファイル保存ダイアログが開く。シンボルを指定した場合は、現在のデフォルト・ディレクトリに指定したファイル名で保存される。
- writeraw** **buffer~**内に格納されているデータをヘッダ情報のないraw形式として保存する。セーブするデータ型のデフォルトは16bitだが、**samptime**メッセージで出力のデータ型を指定することができる。**writeraw**に続いてファイル名となるシンボルを指定しない場合は、標準ファイル保存ダイアログが開く。シンボルを指定した場合は、現在のデフォルト・ディレクトリに指定したファイル名で保存される。
- writesd2** Macintoshのみ。**buffer~**内に格納されているデータをSound Designer II形式として保存する。**writesd2**に続いてファイル名となるシンボルを指定しない場合は、標準ファイル保存ダイアログが開く。シンボルを指定した場合は、現在のデフォルト・ディレクトリに指定したファイル名で保存される。
- writewave** **buffer~**内に格納されているデータをWAV形式として保存する。**writewave**に続いてファイル名となるシンボルを指定しない場合は、標準ファイル保存ダイアログが開く。シンボルを指定した場合は、現在のデフォルト・ディレクトリに指定したファイル名で保存される。
- (remote) **peek~**オブジェクトや**record~**オブジェクトによって、データを変更することができる。
- (マウス) **buffer~**オブジェクトをダブルクリックすると、波形表示ウィンドウが開き、そのデータの中身を見ることができる。

## アーギュメント

- シンボル 必須。第1アーギュメントは、他のオブジェクトが**buffer~**を区別するための参照名を指定する。
- シンボル オプション。第1アーギュメントの参照名に続き、**buffer~**オブジェクトが生成された時点で読み込むオーディオファイルを指定することができる。
- 実数または整数 オプション。オプションのファイル名を指定した後に、ミリ秒で表されるデュレーション値を指定することができる。この値はオブジェクト内で格納するデータの量を制限するものとなる（新しいデュレーシ

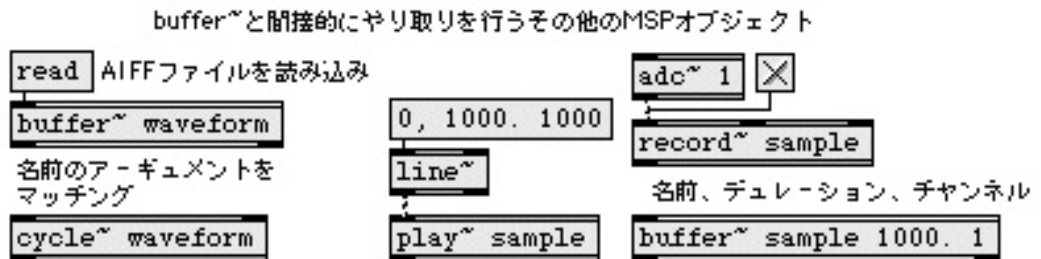
ョン値は、**read**メッセージで設定し直すこともできる)。デューレション値を指定しない場合、**buffer~**オブジェクトはサンプル用のメモリを確保しないが、読み込むオーディオ・ファイルのサイズを制限するものではない。

**整数** オプション。デューレション値に続いて**buffer~**に格納するオーディオ・チャンネル数を指定することができる（これは**buffer~**オブジェクトにどれだけのメモリを初期的に確保するかを伝えるものだが、読み込むオーディオ・ファイルがより多くのチャンネルを持っていた場合、そのチャンネル数に応じたメモリを割り当てる）。チャンネル数の最大値は4、デフォルト値は1である。

## 出力

**実数** **buffer~**オブジェクトの波形表示ウィンドウ上でマウス・クリックかドラッグをすることで、カーソルのある位置が時間軸で、どの位置にあるかをミリ秒単位でアウトレットから出力する。

## 利用例



**buffer~**はオシレータとして利用するための波形テーブルとして、またはサンプル用バッファとして利用できる

## 参照

|                  |                                     |
|------------------|-------------------------------------|
| <b>2d.wave~</b>  | 2次元のウェーブテーブル                        |
| <b>buffir~</b>   | <b>buffer~</b> を利用したFIRフィルター        |
| <b>cycle~</b>    | テーブル参照可能なオシレーター                     |
| <b>groove~</b>   | 速度可変のループ再生                          |
| <b>lookup~</b>   | <b>buffer~</b> を利用したテーブル参照機能        |
| <b>peek~</b>     | サンプル値の読み出しと書き込み                     |
| <b>play~</b>     | ポジション・ベースのサンプル再生                    |
| <b>record~</b>   | サウンドを <b>buffer~</b> のサンプル・メモリへ録音する |
| <b>sfplay~</b>   | ディスクからオーディオ・ファイルを再生する               |
| <b>sfrecord~</b> | ディスクへオーディオ・ファイルとして録音する              |
| <b>wave~</b>     | サイズ可変のウェーブテーブル再生                    |

|             |                                              |
|-------------|----------------------------------------------|
| Tutorial 3  | Fundamentals: Wavetable oscillator (英文PDF参照) |
| Tutorial 12 | Synthesis: Waveshaping (英文PDF参照)             |
| Tutorial 13 | Sampling: Recording and playback (英文PDF参照)   |



**buffir~**オブジェクトは有限インパルス応答（FIR：Finite Impulse Response）フィルターを実装したもので、**buffer~**内に格納されたフィルター係数のセットと入力シグナルとの畳み込み積分を行う。その畳み込み積分は次の式で示される。

$$y_n = b_0x_n + b_1x_{n-1} + b_2x_{n-2} + \dots + b_qx_{n-q}$$

$$y_n = \sum_{j=0}^q b_j x_{n-j}$$

## 入力

|         |                                                                                                                                                                               |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| シグナル    | 第1インレット：フィルター係数と畳み込む（フィルター処理する）入力シグナル。<br><br>第2インレット： <b>buffer~</b> 内のフィルター係数を読み出す開始位置のオフセット・サンプル数。<br><br>第3インレット： <b>buffer~</b> からフィルター係数として利用するサンプルの数（係数の数に相当）。最大値は128。 |
| 整数および実数 | 第2インレット： <b>buffer~</b> 内のフィルター係数を読み出す開始位置のオフセット・サンプル数。<br><br>第3インレット：フィルター用 <b>buffer~</b> から係数として利用するサンプル数（係数の数に相当）。最大値は128。 <b>clear</b> フィルター処理の内部状態データを消去する（0にする）。      |
| set     | <b>set</b> に続き <b>buffer~</b> オブジェクトの参照名、オフセット・サンプルの整数を指定する。オプションとして、それらに続いて <b>buffer~</b> のチャンネルを整数で指定することもできる。                                                             |

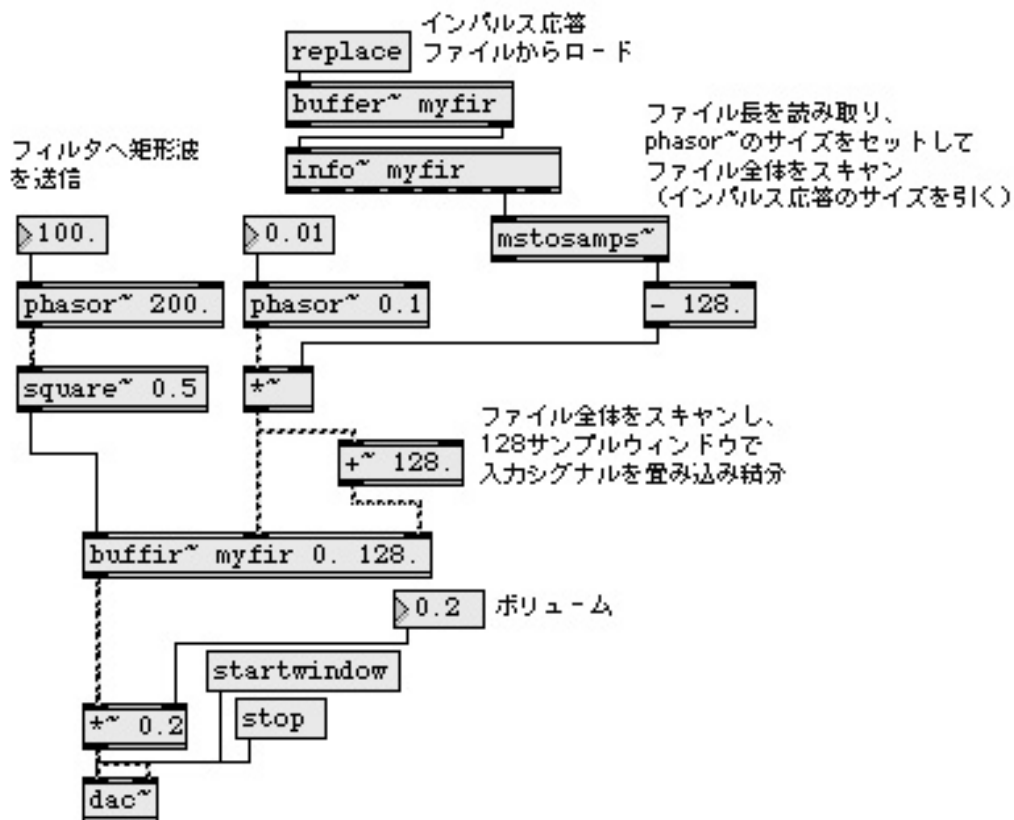
## アーギュメント

|         |                                                                    |
|---------|--------------------------------------------------------------------|
| シンボル    | 必須。フィルター係数として利用するシグナルが格納されている <b>buffer~</b> の参照名。                 |
| 整数および実数 | オプション。 <b>buffer~</b> 内のフィルター係数を読み出す開始位置のオフセット・サンプル数。              |
| 整数および実数 | オプション。フィルター用 <b>buffer~</b> から係数として利用するサンプル数（係数の数に相当）。デフォルト値は0である。 |

## 出力

シグナル 入力シグナルと**buffir~**のフィルター係数とを畳み込み積分したシグナル。

## 利用例



**buffir~** は、**buffer~** オブジェクトをFIRフィルターのためのインパルス応答のデータ格納場所として使用する

## 参照

**biquad~** 2ポール、2ゼロのフィルター  
**buffer~** オーディオ・サンプルの格納

## 入力

- シグナル** シグナルの一部を、閲覧、編集、ファイル保存するためにテキスト形式で格納する（格納するデータ・サイズはオブジェクトのアーギュメントとして指定することができる）。
- write** **capture~**内に格納されたデータをテキスト・ファイルとして保存する。ファイル名を指定するために、標準ファイル保存ダイアログが開く。**write**に続いて保存ファイル名のシンボルを指定すれば、そのファイル名でデータが保存される。その時、保存されるフォルダは**capture~**オブジェクトが含まれるMaxパッチのあるフォルダとなる。パッチが保存されていない場合、**capture~**内のデータはMaxのアプリケーションがあるフォルダに保存される。
- clear** **capture~**内のデータを消去する。
- open** **capture~**が格納するデータを閲覧および編集するために、ウィンドウを開く。ウィンドウはフロント・ウィンドウとして開かれる。
- wclose** **capture~**オブジェクトに関連付けられているウィンドウを閉じる。
- (マウス)** **capture~**オブジェクトをダブル・クリックすることで、オブジェクトに格納されているデータを閲覧、編集するためのウィンドウが開く。編集ウィンドウに表示される数字列は、コピーして**buffer~**の波形編集ウィンドウにペーストすることができる。

## アーギュメント

- f** オプション。第1アーギュメントに**f**を指定した場合、**capture~**は受け取ったシグナルの最初のサンプルを格納していく。格納バッファがいっぱいになれば、その後のサンプルは無視され、格納されない。**f**を指定しない場合、**capture~**は最近に入力されたシグナルのデータを格納していき、必要であれば古いデータを捨てる。
- 整数** オプション。**capture~**で格納するサンプル数の容量を制限する。数字を指定しない場合、**capture~**は4096個のサンプルを格納する。サンプル数の最大数は、Maxのアプリケーションが確保しているメモリ量にのみ制限される。2つめの数値のアーギュメントによって、データ編集ウィンドウで表示される各サンプルの正確さ（小数点以下第何位までか）を指定することができる。

## 出力

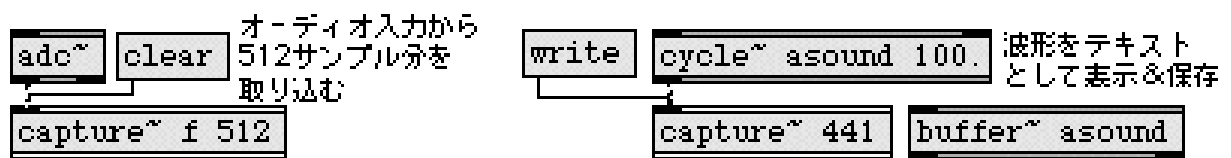
なし。



# capture~

シグナルを  
テキストとして格納する

## 利用例



シグナルの一部を閲覧、保存、コピー&ペーストなどできるようテキスト・データとして格納する

## 参照

**scope~** シグナル・オシロスコープ

## 入力

**シグナル** 第1インレット：周波数領域でのシグナルの実数部分（例えば**fft~**、または**fftin~**オブジェクトで作られるもの）を入力し、振幅と位相のからなるペアの極座標のシグナルに変換される。

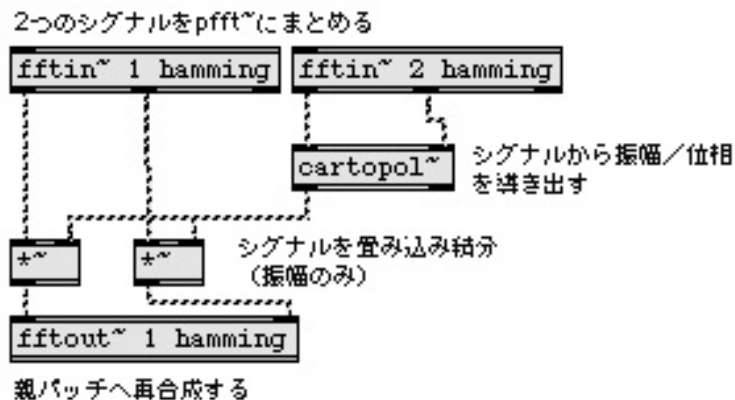
第2インレット：周波数領域でのシグナルの虚数部分（例えば**fft~**、または**fftin~**オブジェクトで作られるもの）を入力し、振幅と位相のからなるペアの極座標のシグナルに変換される。アーギュメントなし。

## 出力

**シグナル** 第1アウトレット：現在の入力シグナルから求められた周波数ビンごとの大きさ（振幅）。

第2インレット：現在の入力シグナルから求められた周波数ビンごとの位相で、ラディアン単位で表される。第1インレットだけにパッチコードが繋がれている場合は、計算の負荷を減らすために位相計算は無視される。

## 利用例



**fftin~** オブジェクトからペアで出力される実数/虚数データから振幅/位相を求めるために、**cartopol~** オブジェクトを使用する

## 参照

|                    |                                  |
|--------------------|----------------------------------|
| <b>cartopol</b>    | 直交座標を極座標に変換する                    |
| <b>fft~</b>        | 高速フーリエ変換                         |
| <b>fftin~</b>      | <b>pfft~</b> に読み込まれるパッチにおける入力    |
| <b>fftinfo~</b>    | <b>pfft~</b> に読み込まれるパッチに関する情報の取得 |
| <b>pfft~</b>       | 周波数領域処理を行うパッチの管理                 |
| <b>fftout~</b>     | <b>pfft~</b> に読み込まれるパッチにおける出力    |
| <b>frameaccum~</b> | 連続した位相変化データのフレームから位相を計算          |
| <b>framedelta~</b> | 連続したFFT処理フレーム間の位相変化を計算           |
| <b>ifft~</b>       | 逆高速フーリエ変換                        |

**pfft~**

**poltoocar**

**poltoocar~**

**vectral~**

周波数領域処理を行うパッチの管理

極座標を直交座標に変換する

シグナルについて極座標を直交座標に変換する

ベクター・ベースのエンベロープ・フォロワー

Tutorial 26

Frequency Domain Signal Processing with **pfft~** (英文PDF参照)

注記：このオブジェクトは、Max/MSP4.2以降のバージョンでは使用されない

MSPの**cd~**オブジェクトは、Maxの**cd**オブジェクト（CD-ROMドライブに入れたオーディオCDを再生する）と似ているが、**cd~**はMSPを通してオーディオCDを再生する。これは、外部のDSP機器やオーディオカードを通すことで、CDをデジタル再生できることを意味する。または、内部の不十分なD/Aコンバーターの使用を避けて、CDの高品質なアナログ再生ができることを意味する（**cd**オブジェクトを利用すると、一旦アナログへ変換されたCDのオーディオがサウンド・マネージャーを通して入力され、再びデジタルへ再変換されることになり、結果的に再現性に劣る音質となる）。

**cd~**オブジェクトは、**sfplay~**オブジェクトのように、サンプル単位で正確にトリガーするためのキューを定義できる。**sfplay~**と同じく、**cd~**は各キューの始まりのオーディオ・データをあらかじめ読み込んでおくことができ、CDの続きのデジタル・データは、キューがあらかじめ読み込んだデータ（約1秒分）から読み込まれる。また、**sfplay~**のように、キュー番号のリストを**cd~**に送ることで、即座に変化する再生の連なりを作り出すことも可能である。

## 入力

- init** オーディオCDドライバーの再初期化を行う。このメッセージまたは**toc**メッセージは、CDドライブに新たにオーディオCDを挿入した後、**cd~**オブジェクトにCDの認識をさせるために利用する必要がある。
- 整数** 0は現在再生しているすべてのトラックまたはキューを停止する。1はCDの最初のトラックを始めから再生する。1より大きな数字は、その数字のキューを再生する。
- リスト** 2つ以上の整数から成るリストで、再生するキューの順番を指定する。
- toc** オーディオCDドライバーの再初期化を行う。オーディオCDがドライブにあれば、そのCDの内容に関する情報が第2アウトレット以降から4つの項目からなるリストとして出力される。そのリストはトラック番号に続いて、CDの開始点からのそのトラックの開始分、開始秒、ブロック（1/75秒）から成る。
- trackdurs** トラック番号とその時間的な長さをミリ秒単位の実数で表した、2つの項目からなるリストを出力する。
- pause** 現在再生中のトラックまたはキューを一時停止する。
- resume** 一時停止している場合に、再生を再開する。
- preload** トラック情報を用いずキューを定義する。第1アーギュメント（必須）はキュー番号を表し、2以上の数字を指定する必要がある。第2アーギュメント（オプション）ではCDの再生開始時間をミリ秒単位で定義

する。第2アーギュメントが与えられない場合、キューはCDの最初から開始することになる。第3アーギュメント（オプション）はキューの長さをミリ秒単位で定義する。第3アーギュメントが与えられない場合、デュレーションはオーディオCDの再生時間すべてとなる。

- preloadtrack**   トラック情報を用いてキューを定義する。第1アーギュメント（必須）はキュー番号を表し、2以上の数字を指定する必要がある。第2アーギュメント（必須）はトラック番号である。第3アーギュメント（オプション）はキューの指定トラック中での再生開始時間をミリ秒単位で定義する。第3アーギュメントが与えられない場合、キューは指定トラックの最初から再生開始となる。第4アーギュメント（オプション）はキューの長さをミリ秒単位で定義する。第4アーギュメントが与えられない場合、そのキューの長さは指定トラックの開始点からの残り時間となる。
- clear**   キューの定義を消去する。**clear**に続くアーギュメントがない場合は、すべてのキューの定義を消去する。**clear**の後に1つ以上の数字が指定されている場合は、その指定番号のキューが消去される。  
注記：**clear**メッセージはオーディオ処理がオンでないと機能しない。
- stitch**   **stitch**に続いて1か0の数値を指定し、サウンドのブロック・ステッチ機能のオン・オフを切り替える。いくつかの古いCD-ROMドライブやポータブルCD-ROMドライブなどでは周期的なノイズが見られることがあり、このブロック・ステッチ機能はこのノイズを除去する機能である。キューの開始時や終了時のサンプル単位の正確さに対しては重要なものとなる。デフォルトではこの機能はオンとなっている。利用するドライブにこの機能が必要かどうか、オン・オフを切り替えて試してみるべきである。
- bang**   キューまたはトラックの再生中に**bang**メッセージを入力すると、現在再生中のトラック番号とミリ秒単位の再生位置をリストとして一番右側のアウトレットから出力する。
- eject**   ドライブ中にあるCDに対し、使用中でなければドライブからイジェクトする。この機能はすべてのドライブで機能するとは限らない。
- embed**   **embed**に続いて1か0を指定することで、**cd~**オブジェクトがキューの定義をパッチ内部に保存するかどうかを定義する。デフォルトではキューの定義はパッチ内に保存されない（**embed 0**）。
- play**   あるトラック中の特定の箇所からCDの再生を行う。第1アーギュメント（必須）はトラック番号を指定する。第2アーギュメント（オプション）は再生開始位置をトラック先頭からの時間としてミリ秒で指定する。第2アーギュメントが与えられない場合、開始位置はトラックの先頭となる。第3アーギュメント（オプション）は再生する長さをミリ秒で指定する。これを指定しない場合はトラック全体が再生される。

## アーギュメント

**整数** **cd~**オブジェクトは2つのアーギュメントを指定することができ、その両方ともオプションである（ただし、第2アーギュメントを記述するには両方とも記述する必要がある）。第1アーギュメントはドライブのデバイスIDである。使用マシンにCDドライブが1つしかない場合は、0から7までのどの数値をIDとして指定しても、**cd~**はドライブを特定できる。第2アーギュメントは、**cd~**オブジェクトがどの程度の頻度で再生中のドライブの時間位置を取得するかを決める。第2アーギュメントに0を指定した場合、**cd~**オブジェクトに**bang**メッセージを入力するまで、再生中のドライブの時間情報を取得しない。

## 出力

**シグナル** 第1および第2アウトレット：CDの左右のチャンネルがそれぞれ出力される。オーディオ・データはCDから直接デジタル・データとして読み出される。**cd~**はサンプリング・レートの変換を行わないため、サンプリング・レートが44.1kHzの時のみ、正しいピッチとしてデータが利用できる、という点に注意が必要である。

**list** 右側から3つ目のアウトレット：**bang**メッセージに対して、再生中のトラック番号およびそのトラックの先頭からの再生位置をミリ秒で出力する。

**list** 右側から2つ目のアウトレット：**toc**メッセージに対して、CDの内容に関する情報を4つの項目からなるリストを出力する。リストはトラック番号、CDの先頭からのトラックの開始分、秒、およびブロック（1/75秒単位）からなる。

**list** 右側から2つ目のアウトレット：**trackdurs**メッセージに対して、トラック番号とそのトラックのミリ秒単位のデュレーションの2つの項目からなるリストを出力する。

**bang** 一番右のアウトレット：**preloadtrack**または**preload**メッセージの処理が終了した時に、**bang**メッセージを出力する。

## 参照

**cd**                   オーディオCDをコントロールする  
**sflist~**             オーディオ・ファイルのキューを格納する  
**sfplay~**           ディスクからオーディオ・ファイルを再生する

# change~

シグナル値の  
変化方向を出力する

## 入力

シグナル 任意のシグナル。

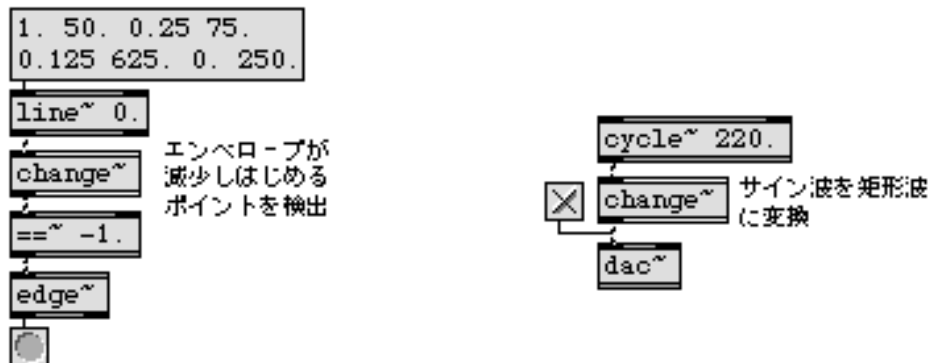
## アーギュメント

なし。

## 出力

シグナル 現在のサンプルがひとつ前のサンプルよりも大きい場合、サンプル値として**1**を出力する。現在のサンプルがひとつ前のサンプルと同じ値の場合、サンプル値として**0**を出力する。現在のサンプルがひとつ前のサンプルより小さい場合、サンプル値として**-1**を出力する。

## 利用例



シグナルの値が増加したか、減少したか、または同じ値のままか、を検出

## 参照

**edge~** シグナルの論理的变化を検出する  
**thresh~** 設定したレベルよりも大きなシグナルを検出する  
**zerox~** ゼロ地点の通過を検出する



## 入力

- bang** **click~** オブジェクトのアウトレットからインパルスを出力する。デフォルトでは、インパルスは1.0の値の後に0.0が続くサンプルから成る。
- set** **set** に続けて、0.0から1.0までの範囲の実数値から成るリストをもつ。このリストは、言い換えれば、インパルスを小さなウェーブ・テーブルとして定義するものである。ウェーブ・テーブルの長さは、リストの要素の数により決定される。リストの項目の最大数は、256である。

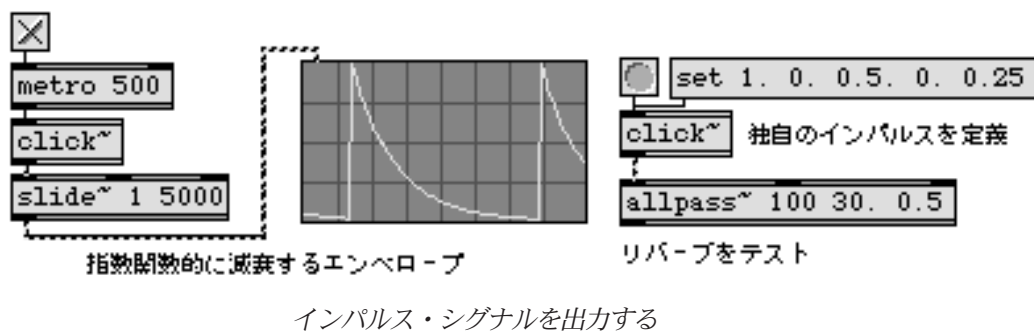
## アーギュメント

- リスト** オプション。インパルスのウェーブ・テーブルを定義することができる (**set** メッセージを参照のこと)。アーギュメントの最大値は256である。

## 出力

- シグナル** インパルス。

## 利用例



## 参照

- buffer~** オーディオ・サンプルの格納
- buffir~** **buffer~** を利用したFIRフィルター
- line~** シグナル・ランプ・ジェネレーター

## 入力

**シグナル** 第1インレット：入力シグナル。このシグナルは、第2インレットおよび第3インレットに入力された振幅の最小値と最大値によって制限をかけられる。

第2インレット：出力シグナルの振幅範囲の最小値

第3インレット：出力シグナルの振幅範囲の最大値

**実数または整数** 第2および第3インレットは、シグナルの代わりに実数か整数で振幅最小値および最大値を入力することができる。

## アーギュメント

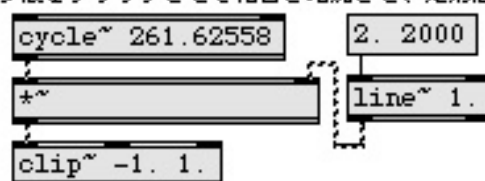
**実数** オプション。出力シグナルの振幅範囲に対し、その最小値および最大値の初期値を指定する。アーギュメントで指定しない場合は、最小値・最大値ともに0に設定される。第2インレットまたは第3インレットに数値が入力された場合は、それらに対応するアーギュメントは無視される。

## 出力

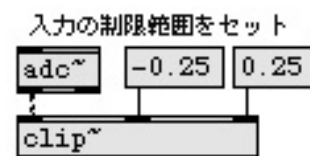
**シグナル** 入力シグナルの各サンプルに対し、設定された振幅範囲で制限のかかったシグナルが出力される。入力シグナルで範囲を超えた値は、すべてその制限値として出力される。

## 利用例

サイン波をクリップさせて倍音を増加させ、矩形波に近づける



入力に対してクリップされたシグナルを出力する



入力の制限範囲をセット

## 参照

- <~ 2つのシグナルの比較を行い、小さければ1を出力する
- >~ 2つのシグナルの比較を行い、大きければ1を出力する

## 入力

**シグナル** 第1インレット：フィルター処理を行うシグナル。フィルターは現在の入力サンプルを以前の入力サンプルと出力サンプルのどちらか、または両方と、次の式に従ってミックスする：

$$y_n = ax_n + bx_{n-(DR/1000)} + cy_{n-(DR/1000)}$$

$R$ はサンプリング・レートで $D$ はミリ秒単位のディレイ時間

第2インレット：現在の出力に加算する過去のサンプル用のミリ秒単位のディレイ時間 ( $D$ )。

第3インレット：出力に送る入力サンプルの量をスケールリングするための振幅係数 ( $a$ )。

第4インレット：出力に送る遅らせた過去の入力サンプルの量をスケールリングするための振幅係数 ( $b$ )。

第5インレット：出力に送る遅らせた過去の出力サンプルの量をスケールリングするための振幅係数 ( $c$ )。

**実数または整数** 第2インレットから第5インレットまでのフィルター・パラメーターはシグナルの代わりに実数で指定することもできる。シグナルがインレットに入力されている場合には、実数での指定は無視される。

**リスト** 3つのパラメーターは第1インレットにリストとして与えることもできる。リストの1番目の数値はディレイ時間 $D$ 、次の数値は係数 $a$ 、そして3番目の数値は係数 $b$ である。シグナルがインレットに入力されている場合には、リストでそのインレットに与えられた係数は無視される。

**clear** **comb~** オブジェクトの以前の出力のメモリーを消去し、**0**にリセットする。

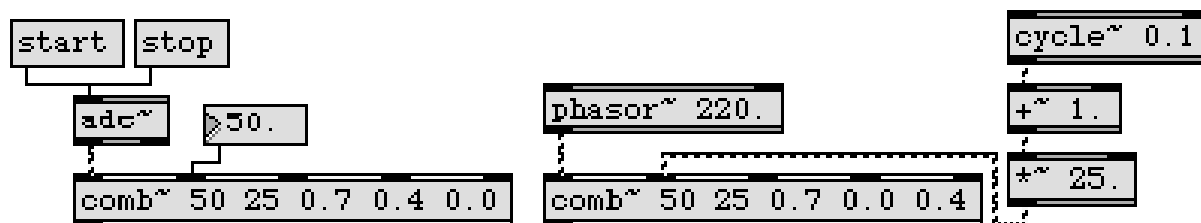
## アーギュメント

**実数** オプション。5つまでの数値で最大ディレイ時間とディレイ時間 $D$ と係数 $a$ 、 $b$ 、 $c$ の初期値を設定する。シグナルがインレットに入力されていれば、そのインレットにアーギュメントとして与えられた係数は無視される。何もアーギュメントがなければ、最大ディレイ時間はデフォルトの10ミリ秒になり、他の値はデフォルトの0になる。

## 出力

**シグナル** フィルター処理を行ったシグナル。

## 利用例



フィルター・パラメーターは実数またはシグナルとして与えられる

## 参照

**allpass~**

オールパス・フィルター

**delay~**

サンプル単位のディレイ・ライン

**reson~**

レゾナンス付きバンドパス・フィルター

**teeth~**

フィードフォワードとフィードバックにディレイ・コントロールが付いたコム・フィルター

## 入力

**シグナル** コサイン関数への入力。入力は一周期の割合 (一般的には、範囲は0から1) として表される。入りに $2\pi$ を掛けた値がコサイン関数で用いられる。

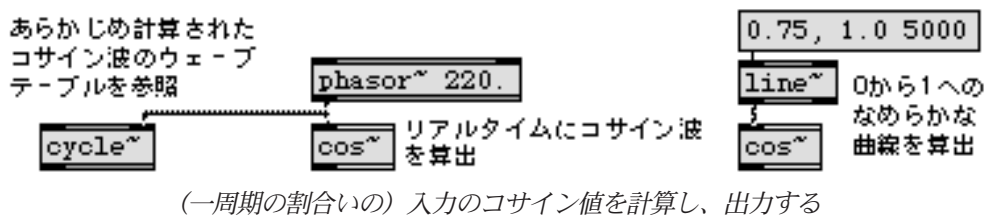
## アーギュメント

なし。

## 出力

**シグナル** 入力値を $2\pi$ 倍にすることで得るコサイン値。このオブジェクトで使われている、コサインを直接計算する方法は、通常は、**cycle~**オブジェクトに格納されているコサインを使うよりも、非効率的である。

## 利用例



## 参照

- acos~** シグナルのアーク・コサイン関数
- asin~** シグナルのアーク・サイン関数
- atan~** シグナルのアーク・タンジェント関数
- atan2~** シグナルのアーク・タンジェント関数 (2変数)
- cosh~** シグナルの双曲線コサイン関数
- cosx~** シグナルのコサイン関数
- cycle~** テーブル参照可能なオシレーター
- phasor~** ノコギリ波の生成
- asinh~** シグナルの双曲線アーク・サイン関数
- sinh~** シグナルの双曲線サイン関数
- sinx~** シグナルのサイン関数
- tanh~** シグナルの双曲線タンジェント関数
- tanx~** シグナルのタンジェント関数
- trapezoid~** 台形波のウェーブテーブル
- triangle~** 三角波のウェーブテーブル
- wave~** サイズ可変のウェーブテーブル再生
- 2d.wave~** 2次元のウェーブテーブル

## 入力

シグナル 双曲線コサイン関数への入力。

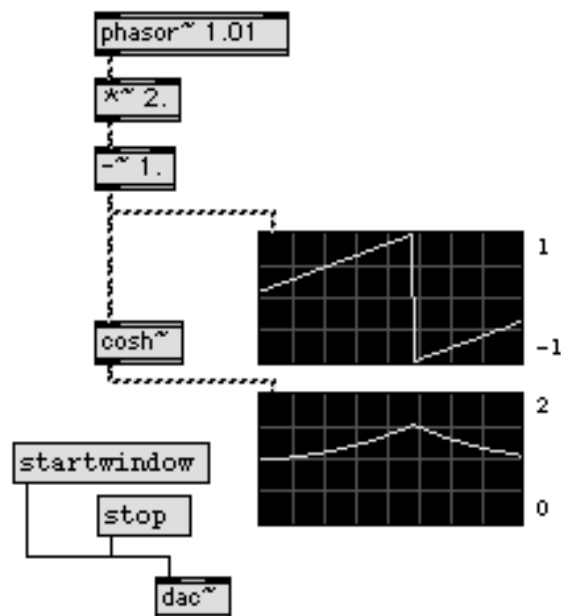
## アーギュメント

なし。

## 出力

シグナル 入力の双曲線コサイン値。

## 利用例



**cosh~** オブジェクトを用いて、面白い波のような形のオーディオ・コントロール・シグナルをつくる

## 参照

|               |                         |
|---------------|-------------------------|
| <b>acos~</b>  | シグナルのアーク・コサイン関数         |
| <b>asin~</b>  | シグナルのアーク・サイン関数          |
| <b>atan~</b>  | シグナルのアーク・タンジェント関数       |
| <b>atan2~</b> | シグナルのアーク・タンジェント関数 (2変数) |
| <b>cos~</b>   | シグナル (0-1の範囲) のコサイン関数   |
| <b>cosx~</b>  | シグナルのコサイン関数             |
| <b>sinh~</b>  | シグナルの双曲線サイン関数           |
| <b>sinx~</b>  | シグナルのサイン関数              |
| <b>tanh~</b>  | シグナルの双曲線タンジェント関数        |
| <b>tanx~</b>  | シグナルのタンジェント関数           |

## 入力

**シグナル** コサイン関数の出力。**cos~**オブジェクトとは違い、**cosx~**オブジェクトは実際に円周率を元にした関数である。**cos~**オブジェクトの出力は1周期を元にしており、**phasor~**オブジェクトを用い、参照テーブルとして使用するためのものである。

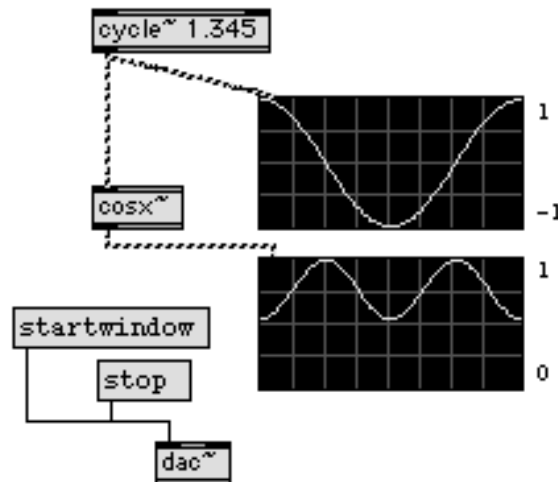
## アーギュメント

なし。

## 出力

**シグナル** 入力のコサイン値。

## 利用例



**cosx~**を用いて、より振幅が浅く、より隆起の激しいオーディオ・コントロール・シグナルを作ることができる

## 参照

- acos~** シグナルのアーク・コサイン関数
- asin~** シグナルのアーク・サイン関数
- atan~** シグナルのアーク・タンジェント関数
- atan2~** シグナルのアーク・タンジェント関数 (2変数)
- atanh~** シグナルの双曲線アーク・タンジェント関数
- cos~** シグナル (0-1の範囲) のコサイン関数
- cosh~** シグナルの双曲線コサイン関数
- sinh~** シグナルの双曲線サイン関数
- sinx~** シグナルのサイン値
- tanh~** シグナルの双曲線タンジェント関数
- tanx~** シグナルのタンジェント関数



## 入力

- bang** オーディオ処理がオンの時、現在の最小値からシグナルのサンプル数をカウントし始める。既にシグナルをカウントしている場合は、最小値にリセットしてカウント・アップし続ける。
- 整数** 第1インレット：新しい最小値を設定する。出力シグナルはこの値からカウント・アップしていく。
- 第2インレット：最大値を設定する。カウント値がこの値まで来た場合、次のサンプルで最小値からカウントし直す。0（デフォルト値）を指定した場合、最大値は設定されず、リセットなしにカウント・アップし続ける。
- リスト** 第1インレット：4つの数字から成るリストで、**count~**オブジェクトの振る舞いを指定することができる。1つ目と2つ目の数字はカウンターの最小値および最大値を指定する。3つ目の数字は**count~**がオフ（0）であるかオン（1）であるかの初期状態を指定する。4つ目の数字はオート・リセットのフラグ（**autoreset**の欄を参照）を設定する。
- 実数** 任意のインレット：整数に変換される。
- autoreset** 第1インレット：**autoreset**に続いて0でない数値を指定することで、オーディオ処理をオンにした時にカウンターを最小値にリセットする。
- min** 第1インレット：**min**に続く数値を指定して、次のループでの最小値を設定する。ただし、出力はすぐに変更されるわけではない。
- set** 第1インレット：**set**に続く数値を指定して、次のループでの最小値を設定する。ただし、出力はすぐに変更されるわけではない。
- stop** 第1インレット：**count~**内の現在の最小値をシグナルとして出力する。

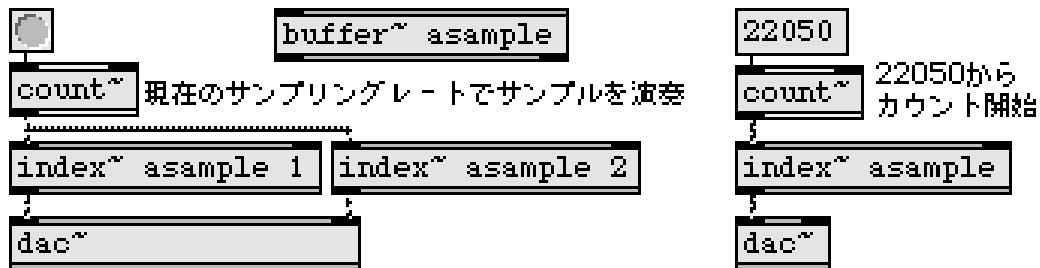
## アーギュメント

- 整数** オプション。第1アーギュメントはカウンターの最小値の初期値を設定する。そのデフォルト値は0である。第2アーギュメントはカウンターの最大値の初期値を設定する。そのデフォルト値は0で、それは最大値がないことを意味する。第3アーギュメントは**count~**オブジェクトがオフ（0）であるかオン（1）であるかの初期状態を指定する。第4アーギュメントはオート・リセットの状態（上記**autoreset**メッセージを参照）を設定する。

## 出力

**シグナル** オーディオ処理を最初にオンにした時、**count~**は常に現在の最小値を出力する。**bang**か整数が入力されると、現在の最小値からカウント・アップし始める。

## 利用例



通過したサンプル数を与えた数からカウントして出力する

## 参照

**index~** 補間機能なしのサンプル・プレイバック  
**mstosamps~** ミリ秒をサンプル数に変換  
**sampstoms~** サンプル数をミリ秒に変換  
**+=~** 結果を格納し、シグナルの加算

Tutorial 13

Sampling: Recording and playback (英文PDF参照)

## 入力

**リスト** 最初の数値は目標値を指定する。2番目の数値はその値に到達するのにどれだけの時間がかかるかをミリ秒単位で指定する。そして最後のオプションである3番目の数値は曲線のパラメーターを指定する。そのパラメーターが0から1までの場合は指数曲線を作り出し、-1から0の場合は対数曲線を作り出す。曲線のパラメーターが0に近づくにつれて、曲線はより直線に近づいていく。また、曲線のパラメーターが0から離れるにつれて、曲線は階段の形に近づいていく。指定された時間の値の中では、**curve~**オブジェクトは現在の格納された値から目標値へ指数勾配を持つシグナルを生成する。**curve~**オブジェクトは連続する指数勾配を生成するために、目標と時間に関する3個1組のパラメーターを最大で42個受け取ることができる。(例えば、**0 1000 .5 1 1000 -.5**というメッセージは現在の値から1秒間かけて0に到達、次の1秒間で1に到達する。) 勾配の中の一つが目標値に到達すると、次のものがスタートする。新たに第1インレットからリスト、実数、または整数が入力された場合は、まだ生成されていないすべての勾配は消去される。

**実数、または整数** 第1インレット：第2インレットから入力される数値によって指定された時間内に到達する目標値。最後の目標値への時間が指定されない場合は、到達時間は0と考えられて、出力シグナルは即座に目標値となる。

第2インレット：出力シグナルが目標値に到達する時間をミリ秒で指定する。

第3インレット：曲線のパラメーターを設定する。0から1の数値は指数曲線を作り出し、-1から0の数値は対数曲線を作り出す。曲線のパラメーターが0に近づくにつれて、曲線はより直線に近づいていく。また、曲線のパラメーターが0から離れるにつれて、曲線は階段の形に近づいていく。

## アーギュメント

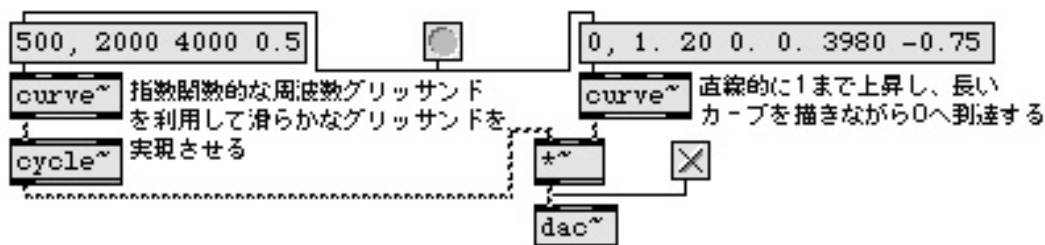
**実数、または整数** オプション。最初のアーギュメントは出力シグナルの初期値を設定する。二番目のアーギュメントは曲線のパラメーターの初期値を設定する。出力シグナルと曲線のパラメーターの初期値はデフォルト値では0である。

## 出力

**シグナル** 第1アウトレット：現在の目標値、または最も最後に受信した目標値、到達時間、曲線のパラメーターによって目標値に向かって変化する指数曲線の値を出力する。

**bang** 第2アウトレット：**curve~**オブジェクトがすべての勾配を生成し終えた時、**bang**を出力する。

## 利用例



曲線の勾配を、周波数と振幅のシグナルをコントロールするために使用する

## 参照

**line~**

シグナル・ランプ・ジェネレーター

**cycle~**オブジェクトは、512サンプルのウェーブテーブルを一波形として繰り返し読み出すことで、補間機能のあるオシレーターとして働く。デフォルト波形はコサイン波形だが、**buffer~**オブジェクトを利用することで、他のオシレーターとして動作することができる。ウェーブテーブル (**buffer~**) 上の513番目のサンプルデータは512番目のサンプルの次のサンプルへの補間に使われる。波形を繰り返すことを考慮すると、513番目のサンプルは波形の終わりから次の始まりにかけての不連続点がないことから1番目のサンプルと同じ値であることが望ましい。512個のサンプルしか有効でない場合、513番目のサンプルは1番目のサンプルと同じであるとみなして動作する。これは**cycle~**のデフォルト波形であるコサイン波形などがあてはまる。他の波形に適用しようとする場合、513番目の値は512番目の値と同じにするか、または513番目を省略するべきである。

## 入力

**シグナル** 第1インレット：オシレーターとしての周波数を指定する。負の数も入力可能である。

第2インレット：0から1の値によって、1波形を移動させる位相を入力する。0から1の範囲外の数値の場合はその範囲に丸め込まれる。オシレーターとして機能させるもうひとつの方法は、周波数を0として、このインレットに**phasor~**を入力することである。周波数が0以外の場合に、このインレットに**cycle~**や他の周期的なオブジェクトを入力すれば、周波数変調に似た形で位相変調を作り出すことができる。

**実数または整数** 第1インレット：オシレーターの周波数を設定する。このインレットにシグナルが入力される場合は、この数値は無視される。

第2インレット：0から1の範囲の値でオシレーターの位相を設定する。この範囲以外の数値を入力した場合は、その範囲内に丸め込まれる。周波数の設定が同じである他の**cycle~**や**phasor~**オブジェクトがあり、周波数が固定のままである場合は、それらオブジェクトと同期して動作するよう位相変化は一定に保たれる。このインレットにシグナルが入力される場合は、この数値は無視される。

**set** **set**に続き、**buffer~**オブジェクトの参照名を指定する。この**buffer~**は**cycle~**が利用するウェーブテーブルの参照先として使われる。オプションとして、この参照名に続いて**buffer~**内でのサンプル・メモリの読み出すオフセット・サンプル数を整数で指定することができる。マルチ・チャンネルを持っている**buffer~**については、ウェーブテーブルとして1つ目のチャンネルしか使うことができない。

**set**に続いてアーギュメントがない場合は、デフォルトのコサイン波形を出力する。

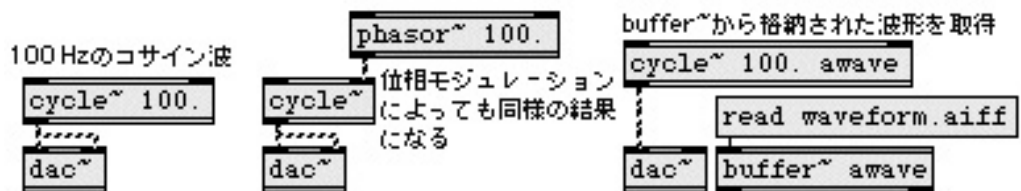
## アーギュメント

- 実数または整数** オプション。オシレーターとしての周波数の初期値を与える。このアーギュメントがない場合、周波数の初期値は0に設定される。
- シンボル** オプション。オシレーターが利用するウェーブテーブルを格納する **buffer~** オブジェクトの参照名を指定することができる。実数または整数による周波数のアーギュメントがある場合は、その後に **buffer~** 参照名を指定する。 **buffer~** 参照名が指定されない場合は、内部に持つコサイン波形が利用される。
- 整数** オプション。 **buffer~** 参照名が与えられている場合、その後に最後のアーギュメントとして **buffer~** のサンプル・メモリから読み出すサンプルのオフセット値を指定することができる。

## 出力

- シグナル** 指定した周波数および位相で繰り返し波形（デフォルトはコサイン波形）を出力する。

## 利用例



連続したコサイン波形またはその他の波形の出力

## 参照

- buffer~** オーディオ・サンプルの格納
- buffir~** **buffer~** を利用したFIRフィルター
- cos~** シグナル（0-1の範囲）のコサイン関数
- line~** シグナル・ランプ・ジェネレーター
- phasor~** ノコギリ波の生成
- trapezoid~** 台形波のウェーブテーブル
- triangle~** 三角波のウェーブテーブル
- wave~** サイズ可変のウェーブテーブル再生
- 2d.wave~** 2次元のウェーブテーブル

- Tutorial 2 Fundamentals: Adjustable oscillator (英文PDF参照)
- Tutorial 3 Fundamentals: Wavetable oscillator (英文PDF参照)



## 入力

**シグナル** **dac~**のインレットに入力されたシグナルは、対応した出力チャンネルからオーディオ出力される。シグナルは-1から1の間の値でなければならず、その範囲外の値はクリッピングされる。

**open** DSPステータス・ウィンドウを開く。

**set** 任意のインレット：**set**に続いて数字を指定することで、**set**メッセージを入力したインレットに論理的な出力チャンネル番号を割り当てる。例えば、**set 3**というメッセージを**dac~**の第1インレットに入力した場合、第1インレットに入力したシグナルは論理出力チャンネル3に設定される。

ただし、MSPのオーディオ処理がオンの間は論理チャンネルを使うために、**dac~**に**set**メッセージを入力してもすぐさま反映はされず、指定チャンネルからは無音が出力されるのみである。オーディオ処理を一度オフにして、再度オンにした時に初めて設定が反映される。**1 2 3 4**というアーギュメントを持つ**dac~**オブジェクトがある場合、シグナルのパッチ・コードが第1および第2インレット（チャンネル1および2）のみに接続されている場合、**set 1 3**というメッセージは、オーディオ処理がオンの時には、左端のチャンネルが即座に論理チャンネル3として出力の経路変更が反映されるわけではない。これを即座に反映できるようにうまく処理するには、**set**メッセージを使う予定の**dac~**の各チャンネルに対して**sig~ 0**の出力シグナルを接続しておく方法がある。しかし、これについては、**dac~**にオーディオ・シグナルのデータを入力する前に、**matrix~**や**switch~**などのオブジェクトを利用してシグナルの経路変更を行うほうが良い。

**start** 読み込まれているすべてのパッチのオーディオ処理をオンにする。

**startwindow** 入力する**dac~**オブジェクトが存在するパッチおよびそれに含まれるサブパッチのみ、オーディオ処理をオンにする。それ以外の他のパッチはすべてオーディオ処理をオフにする。

**stop** 読み込まれているすべてのパッチのオーディオ処理をオフにする。

**wclose** DSPステータス・ウィンドウが開かれていれば、それを閉じる。

**整数** 0でない数字の場合は**start**メッセージと同様に働き、0の場合は**stop**メッセージと同様に働く。

(マウス) **dac~**をダブル・クリックすることでDSPステータス・ウィンドウが開く。



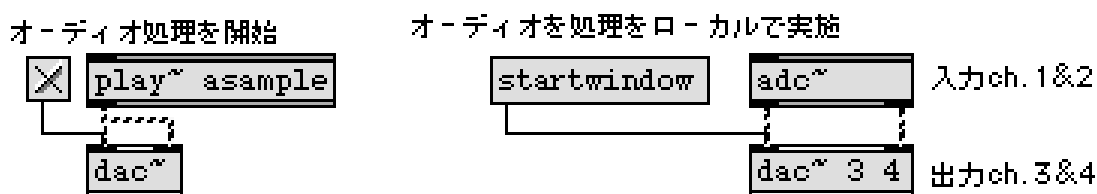
## アーギュメント

**整数** オプション。1から512の間の数字を指定して、1つ以上の出力チャンネル番号を持つ**dac~**オブジェクトを作成できる。これらの番号は論理チャンネルとして扱われ、物理デバイスのチャンネルとの割り当てについては動的に変更可能である。物理デバイスのチャンネルはDSPステータス・ウィンドウのI/O Mappingsサブウィンドウ、または**adstatus**オブジェクトの出力キーワード・アーギュメントで、個々のドライバーに対して割り当てられるものである。サウンド・マネージャーを使う場合は2つの出力チャンネルのみ利用可能である。他のオーディオ・ドライバーを使う場合には、それ以上のチャンネル数を使うことができる場合がある。**dac~**に対して何もアーギュメントを指定しない場合、インレットは2つとなり、それらは出力チャンネル1と2に割り当てられる。

## 出力

なし。インレットに入力されたシグナルは、対応する論理的な出力チャンネルに送られ、DSPステータス・ウィンドウでその論理チャンネルに対して割り当てられた物理デバイスの出力チャンネルから出力される。

## 利用例



オーディオ処理のオン・オフを切り替え、オーディオ出力へシグナルを出力する

## 参照

**adc~**                   オーディオの入力とそのオン/オフ  
**adstatus**             オーディオ・ドライバーの出力チャンネルにアクセスする  
**ezadc~**               オーディオの入力とオン/オフ・ボタン  
**ezdac~**               オーディオの出力とオン/オフ・ボタン

Tutorial 1               Fundamentals: Test tone (英文PDF参照)

## 入力

- シグナル** 第1インレット：品質を落とすためのシグナルを入力する。
- 実数** 第2インレット：入力シグナルのサンプリング・レートを落としてリサンプルする周波数比を指定する。この周波数比はシステムのサンプリング・レートで割られるリサンプリング・レートとなる。例えば、MSPのサンプリング・レートが44100Hzで、周波数比が0.75である場合、出力シグナルとして処理されるサンプリング・レートは33075Hzとなる。
- 整数** 第3インレット：入力シグナルに対する量子化ビット数を指定する。この値は1から24の範囲内でなければならない。より少ないビット数はシグナルの品質低下を意味する。

## アーギュメント

- 実数** オプション。第1アーギュメントは上で述べたリサンプルの周波数比を設定する。アーギュメントがない場合はデフォルトの1.0となる。
- 整数** オプション。第2アーギュメントは入力シグナルに対する量子化ビット数を設定する。アーギュメントがない場合はデフォルトの24となる。

## 出力

- シグナル** 出力シグナルは、リサンプルおよび量子化された後の入力シグナルである。このオブジェクトはリサンプリング時に故意に補間処理は行わず、量子化時についてもディザリングは行わない。ローファイなエフェクトを作り出すものと見るべきである。

注記：量子化ビット数が少ないとやかましいノイズ的なシグナルとなるため、このオブジェクトの出力を聞く場合は注意が必要である。

## 利用例



シグナルのサンプリング・レートや量子化ビット数を変更する

## 参照

- downsamp~** シグナルのダウンサンプル
- round~** 入力されたシグナル値の端数の切り捨て／切り上げを行う

## 入力

**シグナル** 第1インレット：ディレイ処理を行うシグナル。

**整数** 第2インレット：サンプル数によるディレイ時間。ディレイ時間は0（ディレイなし）未満でもアーギュメントで指定した数字より大きくてもいけない。

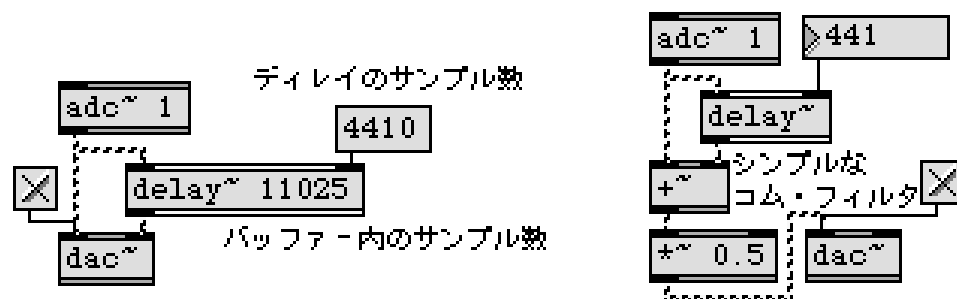
## アーギュメント

**整数** オプション。第1アーギュメントはサンプル数による最大ディレイ時間を設定する。これによってディレイ処理のために確保されるメモリ量が決まる。デフォルト値は512である。第2アーギュメントはサンプル数によるディレイ時間の初期値を設定する。デフォルト値は0である。

## 出力

**シグナル** 指定されたサンプル数だけディレイ処理された入力シグナルが出力される。**delay~**と**tapin~/tapout~**の違いについては、まず**delay~**のディレイ時間はミリ秒ではなくサンプル数で指定するという点が挙げられ、サンプリング・レートが変更されるとディレイの時間長が変わることになる。次に、**delay~**オブジェクトはベクター・サイズよりも小さなサンプル数でのディレイ処理が可能な点が挙げられる。さらに、**delay~**には**tapin~**および**tapout~**のような入力シグナルによるフィードバック処理ができないという点が挙げられる。ショート・ディレイでのフィードバック処理を行いたい場合は、**comb~**オブジェクトを利用することが考えられる。

## 利用例



シグナルに対するサンプル数単位でのディレイ処理によるエコー処理やフィルター・エフェクト

## 参照

**comb~** コム・フィルター  
**tapin~** ディレイ・ラインへの入力  
**tapout~** ディレイ・ラインからの出力

## 入力

シグナル 任意のシグナルを入力する。

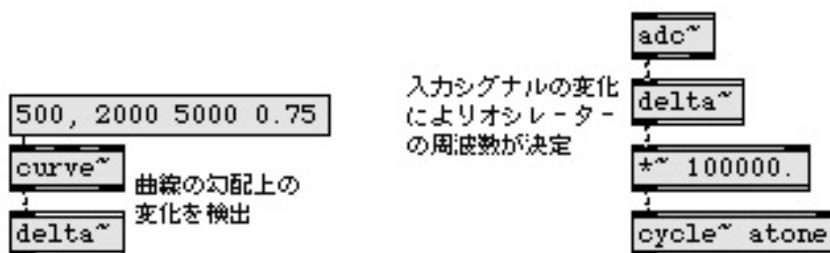
## アーギュメント

なし。

## 出力

シグナル 現在のサンプルと1つ前のサンプルとの差を値とするシグナルを出力する。入力シグナルが 1、0.5、2、0.5 というデータを含んでいるとすれば、その部分に対する出力は 1、-0.5、1.5、-1.5 となる。

## 利用例



あるサンプルとその直前のサンプルとの差を計算する

## 参照

**average~**  
**avg~**

マルチ・モードのシグナル平均化  
シグナルの平均値

**deltaclip~**は入力シグナルのサンプル間の変化量に制限をかける。**clip~**オブジェクトと似た動作だが、振幅値そのものに制限をかけるのではなく、振幅の変化量に制限をかける働きをする。

## 入力

**シグナル** 第1インレット：入力シグナル。

**実数または整数** 第2インレット：出力シグナルの振幅変化量の最小値を指定する。最小値は一般的には負の数を用いる。

第3インレット：出力シグナルの振幅変化量の最大値を指定する。最大値は一般的には正の数を用いる。

## アーギュメント

**実数** オプション。出力シグナルの振幅変化量に対する最小値および最大値の初期値を指定する。アーギュメントで指定しない場合は、最小値と最大値はともに0に設定される。第2インレットまたは第3インレットに数値が入力された場合は、それらに対応するアーギュメントは無視される。

## 出力

**シグナル** 入力シグナルの各サンプル変化量に対して、制限のかかったシグナルが出力される。

## 利用例



## 参照

**clip~** シグナルの振幅に制限をかける

## 入力

**シグナル** 第1インレット：ダウンサンプルするシグナル。**downsamp~**オブジェクトは第1インレットに入力したシグナルに対して、アーギュメントとして設定した値、または第2インレットに入力した値をサンプル数として扱い、サンプル&ホールド処理を行う。出力に対する補間処理は行わない。

第2インレット：入力シグナルの値をサンプル数として扱い、ダウンサンプルする場合のサンプリング・レートとする。

**整数または実数** 第2インレット：第1インレットに入力するシグナルに対してダウンサンプルするサンプル・レートを設定する。実数でサンプル数を指定することもできるが、**downsamp~**はできるだけ現在のサンプリング・レートでサンプルしようとする。

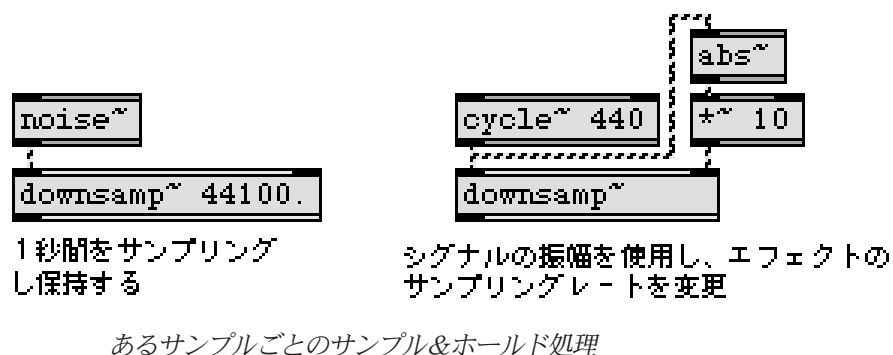
## アーギュメント

**整数または実数** オプション。サンプル・レートを設定する。

## 出力

**シグナル** アーギュメントまたは第2インレットで指定したサンプリング・レートでリサンプルした入力シグナルを出力する。

## 利用例



## 参照

**degrade~** シグナルの品質を低下させる  
**sah~** サンプル&ホールド

## 入力

**bang** **dspstate~** オブジェクトのアウトレットから、オーディオがオンかオフかどうか、現在のサンプリング・レート、シグナル・ベクター・サイズ、I/Oベクターサイズを出力する。

(オン/オフ) オーディオがオンまたはオフになった時に、DSPの設定情報を出力する。

**シグナル** **dspstate~** オブジェクトにシグナルを入力した場合は、シグナルのサンプリング・レート、ベクター・サイズが出力される。これは、グローバルなサンプリング・レートやシグナル・ベクター・サイズと異なる。

## アーギュメント

なし。

## 出力

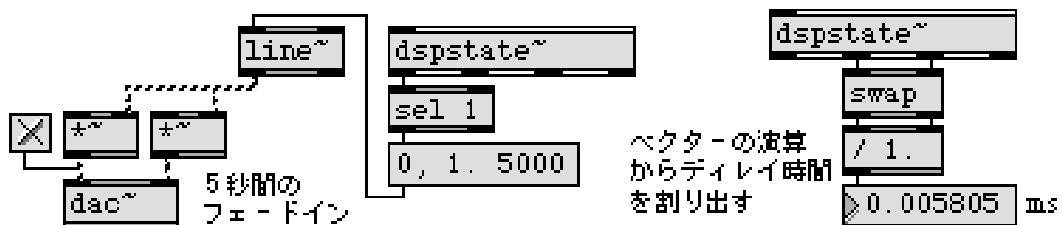
**整数** 第1アウトレット：オーディオがオンである場合、またはオンになった場合、1が出力される。オーディオがオフである場合、またはオフになった場合、0が出力される。

**実数** 第2アウトレット：入力されたシグナルのサンプリング・レート、またはグローバル・サンプリング・レート。

**整数** 第3アウトレット：現在のDSPシグナル・ベクター・サイズ

**整数** 第4アウトレット：現在のI/Oシグナル・ベクター・サイズ

## 利用例



オーディオ処理をオンまたはオフにした時に、サンプリング・レートを用いてタイミングの計算をする

## 参照

**sampstoms~** サンプル数をミリ秒に変換  
**mstosamps~** ミリ秒をサンプル数に変換

Tutorial 20 MIDI control: Sampler (英文PDF参照)  
 Tutorial 25 Analysis: Using the FFT (英文PDF参照)



## 入力

**bang** **dsptime~**に**bang**を入力した時点で、それまでに処理したオーディオ・サンプル数から求めた処理時間を、ミリ秒単位で出力する。

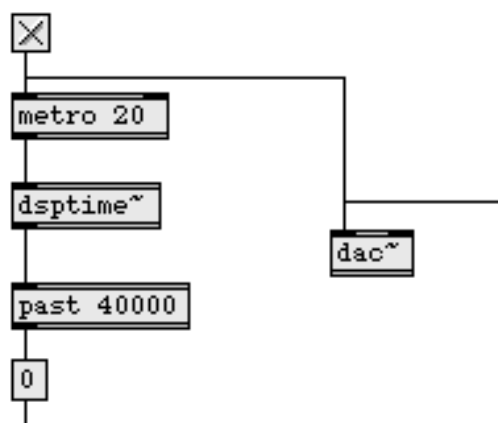
## アーギュメント

なし。

## 出力

**実数** それまでに処理されたオーディオ・サンプル数から求めた処理時間を、ミリ秒単位で出力する。値は処理されたオーディオ・サンプル数から計算されるものであり、実際のミリ秒単位でのタイマー出力ではない。これは、非リアルタイムのオーディオ・ドライバーに対して一種の連動クロックとして用いることができる。

## 利用例



オーディオ処理開始40秒後に自動的に処理を停止する

## 参照

**adstatus**  
**dspstate~**

オーディオ・ドライバーの出力チャンネルにアクセスする  
現在のDSPの設定を出力する

## 入力

**シグナル** 比較演算オペレータの出力シグナルのような、0値と非0値の間で変化するシグナルを入力する。

## アーギュメント

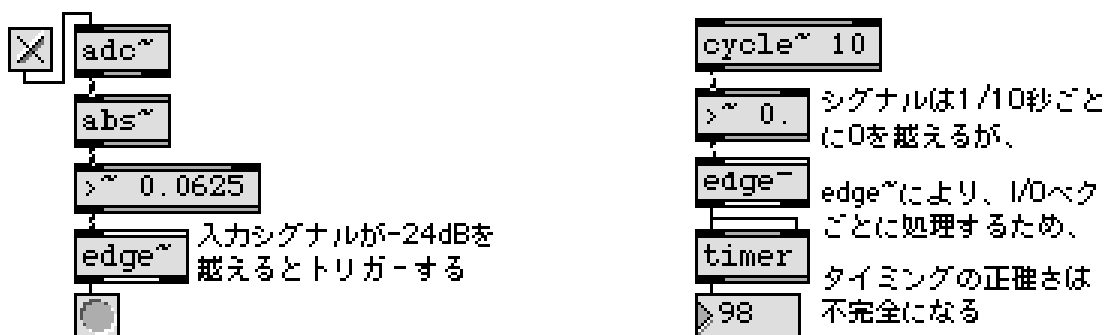
なし。

## 出力

**bang** 第1アウトレット：入力シグナルが0から非0の値へ変化した時に出力する。**bang**メッセージ間の最小時間はMSPのスケジューラの最小インターバルより短くはならない。その最小インターバルは一般的にはシグナル・ベクター・サイズと同じであるが、スケジューラのオーディオ割り込みモードが無効になっている場合はシグナル・ベクター・サイズより大きくなることもある。

第2アウトレット：入力シグナルが非0から0へ値が変化した時に出力する。I/Oベクター・サイズで設定されたサンプル数よりも短い間隔で出力は起こらない。

## 利用例



シグナルの特定の瞬間でトリガーとなるMaxメッセージを出力する

## 参照

**change~** シグナル値の変化方向を出力する  
**thresh~** 設定したレベルよりも大きなシグナルを検出する  
**zerox~** 0地点の通過を検出する



## 入力

- (マウス) **ezadc~**をクリックすることで、オーディオ処理のオン/オフを切り替える。オーディオがオンの場合は、**ezadc~**オブジェクトは反転表示される。
- 整数** 0でない数は、読み込まれたすべてのパッチのオーディオ処理をオンにする。0は、読み込まれたすべてのパッチのオーディオ処理をオフにする。
- local** **local 1**メッセージを送り、**ezadc~**をクリックしてオンにした場合は、**startwindow**メッセージを送った時と同じように動作する。**local 0**メッセージを送ると、**ezadc~**はデフォルト・モードに戻る。**ezadc~**をクリックしてオンにした場合は、**start**メッセージを送った時と同じように動作する。
- open** DSPステータス・ウィンドウを開く。このウィンドウは最前面に表示される。
- start** 読み込まれたすべてのパッチのオーディオ処理をオンにする。
- startwindow** **ezadc~**オブジェクトの置かれたパッチと、そのパッチのサブパッチのオーディオ処理をオンにする。また、その他のパッチのオーディオ処理をオフにする。
- stop** 読み込まれたすべてのパッチのオーディオ処理をオフにする。
- wclose** DSPステータス・ウィンドウを閉じる。

## アーギュメント

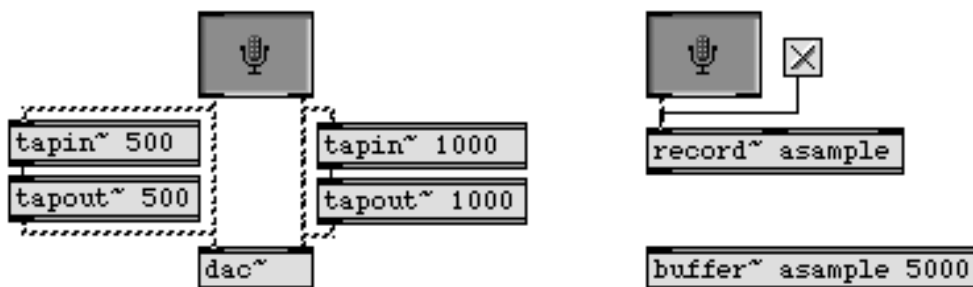
なし。

## 出力

- シグナル** 第1インレット：チャンネル1から入力されたオーディオを出力。  
第2インレット：チャンネル2から入力されたオーディオを出力。



## 利用例



オーディオ入力の処理、レコーディング

## 参照

**adstatus****ezdac~****adc~**

オーディオ・ドライバーの出力チャンネルにアクセスする

オーディオの出力とオン/オフ・ボタン

オーディオの入力とそのオン/オフ



## 入力

- シグナル** 第1インレット：オーディオ出力チャンネル1から出力されるシグナル。デジタル・オーディオ・コンバータのクリッピングを避けるため、それぞれのインレットのシグナルは、-1から1の範囲でなければならない。
- 第2インレット：オーディオ出力チャンネル2から出力されるシグナル。
- (マウス)** **ezdac~**をクリックし、オーディオ処理のオン/オフを切り替える。オーディオがオンの場合は、**ezdac~**オブジェクトは反転表示される。
- 整数** 0でない数は、読み込まれたすべてのパッチのオーディオ処理をオンにする。0は、読み込まれたすべてのパッチのオーディオ処理をオフにする。
- local** **local 1**メッセージを送り、**ezdac~**をクリックしてオンにした場合は、**startwindow**メッセージを送った時と同じように動作する。**local 0**メッセージを送ると、**ezdac~**はデフォルト・モードに戻る。**ezdac~**をクリックしてオンにした場合は、**start**メッセージを送った時と同じように動作する。
- open** DSPステータス・ウィンドウを開く。このウィンドウは最前面に表示される。
- start** 読み込まれたすべてのパッチのオーディオ処理をオンにする。
- startwindow** **ezdac~**オブジェクトの置かれたパッチと、そのパッチのサブパッチのオーディオ処理をオンにする。また、その他のパッチのオーディオ処理をオフにする。
- stop** 読み込まれたすべてのパッチのオーディオ処理をオフにする。
- stopwindow** **ezdac~**オブジェクトの置かれたパッチと、そのパッチのサブパッチのオーディオ処理をオフにする。
- wclose** DSPステータス・ウィンドウを閉じる。

## アーギュメント

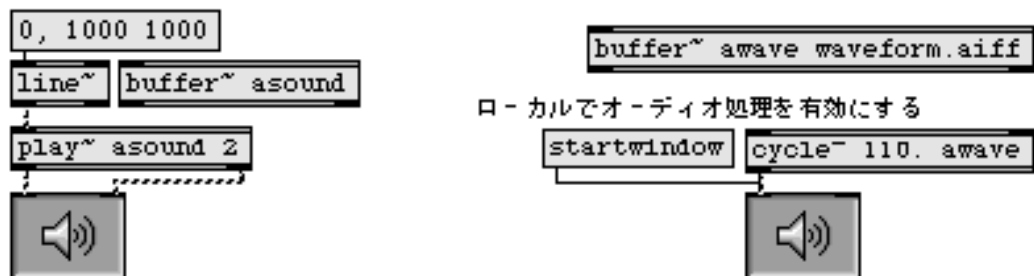
なし。



## 出力

なし。インレットに受け取ったシグナルを、対応するオーディオ出力チャンネルに送る。

## 利用例



オーディオのオン/オフの切り替えと、シグナルのオーディオ出力

## 参照

**adstatus**  
**ezadc~**  
**adc~**

オーディオ・ドライバーの出力チャンネルにアクセスする  
オーディオの入力とオン/オフ・ボタン  
オーディオの入力とそのオン/オフ

Tutorial 3

Fundamentals: Wavetable oscillator (英文PDF参照)

**fffb~**オブジェクトはバンドパス・フィルター・オブジェクトのバンクを実装するが、それぞれのフィルターは**reson~**フィルター・オブジェクトに類似したものである。入力シグナルは、すべてのフィルターに適用され、それぞれのフィルターの出力は独立して利用できる。このオブジェクトは、数多くの**reson~**オブジェクトを使うよりも効率的であるが、スピードを重視するためにパラメーター変更のためのシグナルを受け付けない。

## 入力

**シグナル** 第1インレットのシグナルは、すべてのフィルターに送られる

**freq** 第1インレット：1つの整数と1つ以上の実数からなるリストを伴う**freq**は、最初の数値によって与えられたインデックスのフィルターから始まるフィルターの中心周波数を設定する。このフィルターの周波数はリストの2番目の数値に設定される。リストの中のその後に続く任意の数値は、最初に指定されたものに続くフィルターの周波数を設定する。インデックスは0から始まる。

例えば、**freq 3 1974.0 333.0 1234.0**というメッセージは4番目のフィルターの周波数を1974Hzに、5番目のフィルターを333Hz、6番目のフィルターを1234Hzに設定する。

**freqAll** 第1インレット：実数を伴う**freqAll**は、すべてのフィルターの中心周波数を与えられた浮動小数点の値に設定する。

**freqRatio** 第1インレット：2つまたはそれ以上の数値からなるリストを伴う**freqRatio**は、最初のフィルターの中心周波数をリストの中の最初の値に設定し、残りのフィルターの周波数を最初の値に2番目の値を繰り返しかけ算した値に設定し、後に続くフィルターの周波数の比が2番目の値になるように設定する。例えば、**freqRatio 440. 2.**というメッセージは、最初のフィルターの周波数を440Hzに、2番目の周波数を880Hzに、3番目の周波数を1760Hzに設定する。

**gain** 第1インレット：1つの整数と1つまたはそれ以上の実数からなるリストを伴う**gain**は、最初の数値で与えられたインデックスのフィルターから始まるフィルターのゲインを設定する。フィルターのゲインはリストの2番目の数値に設定される。リストの後に続く任意の数値は、最初に指定されたフィルターの後に続くフィルターのゲインを設定する。インデックスは0から始まる。

**gainAll** 第1インレット：実数を伴う**gainAll**は、すべてのフィルターのゲインを与えられた浮動小数点の値に設定する。

**Q** 第1インレット：1つの整数と1つまたはそれ以上の実数からなるリストを伴う**Q**は、最初の数値で与えられたインデックスのフィルターか



ら始まるフィルターのQを設定する。フィルターのQはリストの2番目の数値に設定される。リストの後に続く任意の数値は、最初に指定されたフィルターの後に続くフィルターのQを設定する。インデックスは0から始まる。

**QAII** 第1インレット：実数を伴うQAIIは、すべてのフィルターのQを与えられた浮動小数点の値に設定する。

## アーギュメント

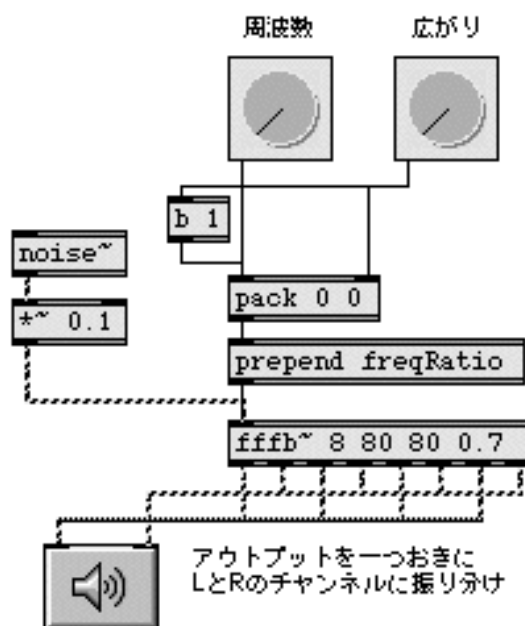
**整数** 必須。最初のアーギュメントはフィルターの数を指定する。

**実数** オプション。追加の3つの実数アーギュメントは最初のフィルターの周波数、後に続くフィルターの周波数の比、そして、すべてのフィルターのQ係数を設定するのに使われる。

## 出力

**シグナル** 個々のアウトレットに割り当てられたそれぞれのフィルターの出力。第1アウトレットが最初のフィルターの出力となる。

## 利用例



基本周波数と周波数比を変えることで、ステレオに広げる

## 参照

**reson~**

レゾナンス付きバンドパス・フィルター

## 入力

**シグナル** 第1インレット：高速フーリエ変換される複素数シグナルの実部。

第2インレット：高速フーリエ変換される複素数シグナルの虚部。

第1インレットおよび第1アウトレットのみに、シグナルのパッチ・コードが接続されている場合、実部のみのFFT（高速フーリエ変換）が実行される。それ以外の場合では複素数FFTが実行される。

## アーギュメント

**整数** オプション。第1アーギュメントは、FFT処理を行うサンプル・ポイント数を指定する。この数値は2の累乗でなければならない。デフォルト値は512サンプルである。第2アーギュメントは、次のFFT処理フレームまでのサンプル数を指定する。この数値も2の累乗のサンプル・ポイント数を指定する。デフォルトの間隔は512サンプルである。第3アーギュメントは、FFT処理を実行するシグナル・ベクター内の処理の開始地点を、サンプル数のオフセット値として指定する。この数値は0またはシグナル・ベクター・サイズの倍数の、どちらかを指定する。**fft~**は間違ったアーギュメントを指定した場合、訂正を行うが、**fft~**オブジェクトを作成した後にシグナル・ベクター・サイズやオフセット値を変更した場合は、ベクター・サイズの倍数ではなくなる可能性があるため、オーディオ処理を開始しても、正常に動作しないことがある。

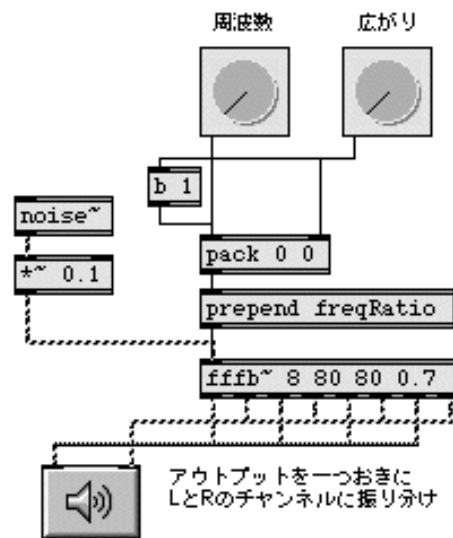
## 出力

**シグナル** 第1アウトレット：入力シグナルに対するフーリエ変換後の実部データを出力する。この出力はFFT処理すべき入力シグナルが、すべて入力された時点で出力される。

第2アウトレット：入力シグナルに対するフーリエ変換後の虚部データを出力する。この出力はFFT処理すべき入力シグナルがすべて入力された時点で出力される。

第3アウトレット：0で始まりFFT出力ポイント数より1小さい数字までの数値列を、シグナル・データとしてFFT出力に同期して出力する。このシグナルは**index~**に入力して利用することで、周波数領域での計算に利用することができる。FFTの結果が出力されない場合（FFTを実行する時間間隔がFFT処理サイズより大きい場合など）は、このシグナルは0として出力される。

## 利用例



オーディオ・シグナルに対する高速フーリエ変換

## 参照

|                    |                                  |
|--------------------|----------------------------------|
| <b>cartopol</b>    | 直交座標を極座標に変換する                    |
| <b>cartopol~</b>   | シグナルについて直交座標を極座標に変換する            |
| <b>fftin~</b>      | <b>pfft~</b> に読み込まれるパッチにおける入力    |
| <b>fftinfo~</b>    | <b>pfft~</b> に読み込まれるパッチに関する情報の取得 |
| <b>fftout~</b>     | <b>pfft~</b> に読み込まれるパッチにおける出力    |
| <b>frameaccum~</b> | 連続した位相変化データのフレームから位相を計算          |
| <b>framedelta~</b> | 連続したFFT処理フレーム間の位相変化を計算           |
| <b>ifft~</b>       | 逆高速フーリエ変換                        |
| <b>index~</b>      | 補間機能なしのサンプル・プレイバック               |
| <b>pfft~</b>       | 周波数領域処理を行うパッチの管理                 |
| <b>poltoCAR</b>    | 極座標を直交座標に変換する                    |
| <b>poltoCAR~</b>   | シグナルについて極座標を直交座標に変換する            |
| <b>vectral~</b>    | ベクター・ベースのエンベロープ・フォロワー            |

Tutorial 25

Analysis: Using the FFT (英文PDF参照)

## 入力

**シグナル** **begin~**オブジェクトから繋ぐためのダミーのインレット。**fftin~**のシグナル入力は、そのオブジェクトを含むサブパッチが**pfft~**で読み込まれた場合、その**pfft~**でのインレットとなる。

## アーギュメント

**整数** 必須。**pfft~**において対応するインレット番号を指定する。**pfft~**のインレットに入力されたシグナルのFFT処理結果は、該当番号の**fftin~**オブジェクトから出力される。インレットの割り当ては1から始まり、1は**pfft~**の第1インレットに対応する。**fftin~**オブジェクトを複数個作成する場合は、明示的に異なる番号を割り当てる必要がある。

**シンボル** オプション。入力シグナルに対してオーバーラップしたFFT処理を行う場合に、オーバーラップに用いる窓関数を指定できる。ここでは、**square**（矩形窓、窓処理を行わない）、**hanning**（ハニング窓、デフォルト）、**hamming**（ハミング窓）が窓関数として利用できる。ただし、**nofft**というシンボルを指定した場合は、**fftin~**はFFT処理結果を出力せず、**pfft~**への入力シグナルそのものを出力する。これによって、**pfft~**を通して親パッチからの生のコントロール・シグナルを入力することができる。

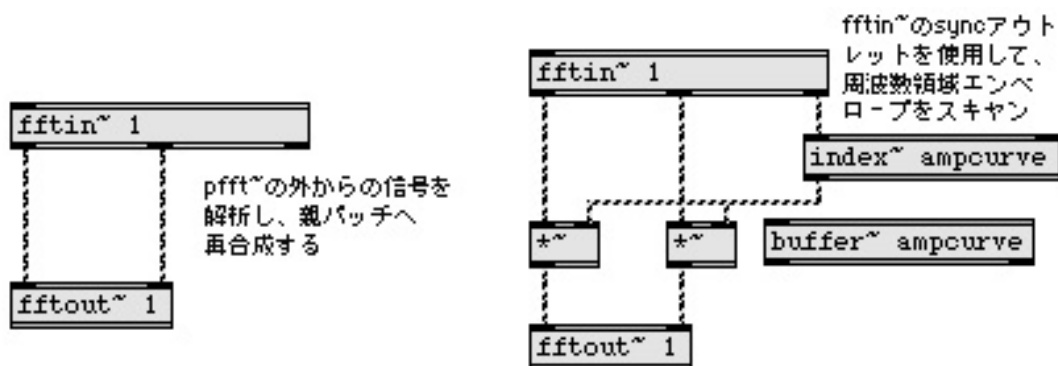
## 出力

**シグナル** 第1アウトレット：対応する**pfft~**のインレットに入力されたシグナルに施された、FFT処理結果の実部が出力される。**fft~**オブジェクトと同様に、**fftin~**も解析の窓サイズに従った出力の遅れが生じる。この解析の窓サイズは、パッチを読み込んだ**pfft~**オブジェクトで指定した数値である。

第2アウトレット：対応する**pfft~**のインレットに入力されたシグナルに施された、FFT処理結果の虚部が出力される。**fft~**オブジェクトと同様に、**fftin~**も解析の窓サイズに従った出力の遅れが生ずる。この解析の窓サイズは、パッチを読み込んだ**pfft~**オブジェクトで指定した数値である。この**fftin~**の第1アウトレットにのみシグナルが入力されている場合は、実部のみのFFTが実行されて、その結果が出力される。この場合は、複素数FFTの処理に比べ、計算にかかるCPUの負荷が減ることになる。

第3アウトレット：第1および第2インレットから出力される現在の周波数データ列に対応したサンプルのインデックス番号が出力される。数値としては、0からFFTサイズより1小さい値までが順に並んだものとなる。

## 利用例



fftin~は、周波数領域のシグナルのペア（実部および虚部）と、ビン番号を示す同期シグナルを出力する

## 参照

|                    |                          |
|--------------------|--------------------------|
| <b>cartopol</b>    | 直交座標を極座標に変換する            |
| <b>cartopol~</b>   | シグナルについて直交座標を極座標に変換する    |
| <b>fft~</b>        | 高速フーリエ変換                 |
| <b>fftinfo~</b>    | pfft~に読み込まれるパッチに関する情報の取得 |
| <b>fftout~</b>     | pfft~に読み込まれるパッチにおける出力    |
| <b>frameaccum~</b> | 連続した位相変化データのフレームから位相を計算  |
| <b>framedelta~</b> | 連続したFFT処理フレーム間の位相変化を計算   |
| <b>ifft~</b>       | 逆高速フーリエ変換                |
| <b>pfft~</b>       | 周波数領域処理を行うパッチの管理         |
| <b>poltoCAR</b>    | 極座標を直交座標に変換する            |
| <b>poltoCAR~</b>   | シグナルについて極座標を直交座標に変換する    |
| <b>vectral~</b>    | ベクター・ベースのエンベロープ・フォロワー    |

## 入力

**bang** FFTフレーム・サイズ、その半分のサイズ（すなわち、シグナル・ベクター・サイズ）、そしてFFTのホップ・サイズを各アウトレットから出力する。

## アーギュメント

なし

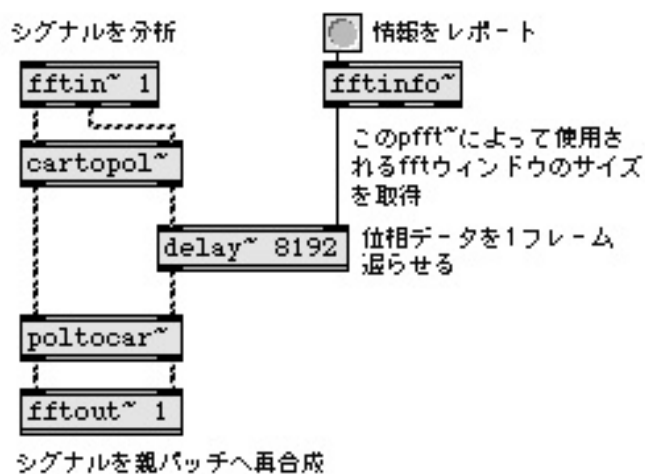
## 出力

**整数** 第1アウトレット：**pfft~**オブジェクトで設定された現在のFFTフレーム・サイズ。

第2アウトレット：現在のFFTフレーム・サイズの半分のサイズ。

第3アウトレット：現在のFFTホップ・サイズ。

## 利用例



**fftinfo~**は、それが含まれるFFT処理サブパッチに関する情報を出力する

## 参照

**cartopol**  
**cartopol~**

直交座標を極座標に変換する  
シグナルについて直交座標を極座標に変換する  
高速フーリエ変換

**fft~**  
**fftin~**

**pfft~**に読み込まれるパッチにおける入力

**fftout~**

**pfft~**に読み込まれるパッチにおける出力

**frameaccum~**

連続した位相変化データのフレームから位相を計算

**framedelta~**

連続したFFT処理フレーム間の位相変化を計算

**ifft~**

逆高速フーリエ変換

**pfft~**

周波数領域処理を行うパッチの管理

**poltocar**

極座標を直交座標に変換する

**poltocar~**

シグナルについて極座標を直交座標に変換する

**vectral~**

ベクター・ベースのエンベロープ・フォロワー

Tutorial 25

Analysis: Using the FFT (英文PDF参照)

Tutorial 26

Frequency Domain Signal Processing with pfft~ (英文PDF参照)



## 入力

**シグナル** 第1インレット：周波数領域から時間領域のデータへ逆変換するシグナルの実部。ifft~オブジェクトと同様に、fftout~は解析の窓サイズに従った遅れが生じる。この窓サイズはパッチを読み込んだpfft~で指定した数値である。

第2インレット：周波数領域から時間領域のデータへ逆変換するシグナルの虚部。ifft~オブジェクトと同様に、fftout~は解析の窓サイズに従った遅れが生じる。この窓サイズはパッチを読み込んだpfft~で指定した数値である。第1インレットのみにシグナルが入力されている場合は、実部のみのIFFTが実行される。この場合は、複素数のIFFTの実行に比べて計算にかかるCPU負荷が減ることになる。

## アーギュメント

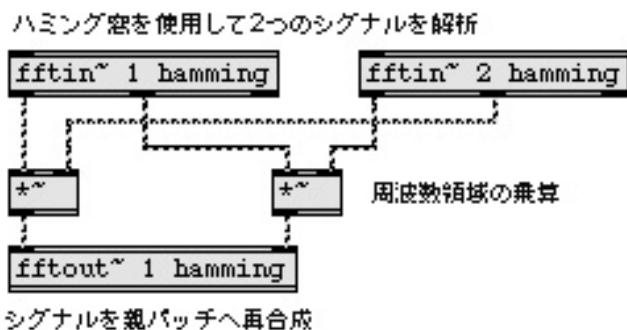
**整数** 必須。pfft~において対応するアウトレット番号を指定する。fftout~に入力されたシグナルはIFFT処理され、pfft~の対応するアウトレットから出力される。アウトレットの割当は1から始まり、1がpfft~の第1アウトレットとなる。fftout~オブジェクトを複数個生成する場合は、明示的に異なるアウトレット番号を割り当てる必要がある。

**シンボル** オプション。オーバーラップしたIFFT処理を行う場合、オーバーラップに用いる窓関数を指定できる。ここでは、**square**（矩形窓、窓処理を行わない）、**hanning**（ハニング窓、デフォルト）、**hamming**（ハミング窓）が窓関数として利用できる。ただし、**nofft** というシンボルを指定した場合は、IFFT処理を行わず、fftout~へ入力されたシグナルがそのままpfft~から出力される。これによって、pfft~を通して親パッチへ生のコントロール・シグナルを出力することができる。

## 出力

**シグナル** fftout~オブジェクトは周波数領域のデータを時間領域のデータに変換し直す。fftout~オブジェクトが含まれるサブパッチを読み込んだpfft~オブジェクトの、対応するアウトレットからデータが出力される。

## 利用例



ffftout~は、周波数領域での信号のペア（実部および虚部）を、時間領域の信号に変換してpfft~へ送る

## 参照

|                    |                          |
|--------------------|--------------------------|
| <b>cartopol</b>    | 直交座標を極座標に変換する            |
| <b>cartopol~</b>   | 信号について直交座標を極座標に変換する      |
| <b>fft~</b>        | 高速フーリエ変換                 |
| <b>fftin~</b>      | pfft~に読み込まれるパッチにおける入力    |
| <b>fftinfo~</b>    | pfft~に読み込まれるパッチに関する情報の取得 |
| <b>frameaccum~</b> | 連続した位相変化データのフレームから位相を計算  |
| <b>framedelta~</b> | 連続したFFT処理フレーム間の位相変化を計算   |
| <b>ifft~</b>       | 逆高速フーリエ変換                |
| <b>pfft~</b>       | 周波数領域処理を行うパッチの管理         |
| <b>poltoCAR</b>    | 極座標を直交座標に変換する            |
| <b>poltoCAR~</b>   | 信号について極座標を直交座標に変換する      |
| <b>vectral~</b>    | ベクター・ベースのエンベロップ・フォロワー    |

Tutorial 25  
Tutorial 26

Analysis: Using the FFT（英文PDF参照）  
Frequency Domain Signal Processing with pfft~（英文PDF参照）



**filtergraph~**オブジェクトはシグナル・オブジェクトではなく、それ自体でオーディオ・シグナルに対して処理を行うことはないが、MSPの現在のサンプリング・レートに従って、周波数、振幅、レゾナンス (Q) といった高水準のパラメーターから、**biquad~**オブジェクトのためのフィルター係数を生成する。

**filtergraph~**オブジェクトは、**biquad~**オブジェクトを用いて実装される様々な種類の2次フィルター (2つの極と2つの零を持つ) のための表示とグラフィカル・ユーザー・インターフェースの両方として設計されている。**filtergraph~**オブジェクトの横軸は周波数 (これはリニアまたは対数スケールで表示することができる) を表し、縦軸は振幅を表す。表示された曲線は現在のフィルター・モデルの周波数特性を表す。周波数特性は、本質的にはオーディオ・シグナルの中にあるそれぞれの周波数成分を、フィルターが増幅または減衰させる量である。**biquad~**オブジェクトは、**filtergraph~**が計算してリストとして出力する係数に基づいて実際のフィルター処理を行う。係数そのものと直接関わる必要はないかもしれないが、高水準のパラメーターとの関係を記述するために、数学の方程式も用意されている。

カットオフ周波数は、フィルターの活動の中心周波数である。その明確な意味はフィルター・タイプごとに異なるが、一般的にはグラフの曲線の転換点と見なすことができる。カットオフ周波数は色付きの矩形として表示されるが、その幅はフィルターのバンド幅に相当する。バンド幅とはフィルターの効果の範囲であり、その中心がカットオフ周波数である。Qはフィルターの幅と同じパラメーターのための別の用語であるが、別の単位 (単なるHzの代わりにオクターブの逆数として) で記述される。さらにレゾナンス、スロープ、S、transitional bandといった名前がバンド幅を参照するのに用いられる。ほとんどの部分において、**filtergraph~**はバンド幅またはQを用いるが、これはお互いに反比例の関係となる。

ゲイン・パラメーターの解釈はフィルターのタイプによって異なるが、一般的にはフィルター・バンドまたはスペクトラムの全体に渡るリニア・スケーリングを意味する。

高水準のフィルター・パラメーターは、オブジェクトの上でクリック&ドラッグすることで変えることができる。デフォルトでは、水平方向のマウス・ドラッグはカットオフ周波数に割り当てられ、垂直方向のマウス操作は (**gainmode**が有効であれば) ゲインに割り当てられる。しかしながら、カーソルがフィルター・バンドの端にちょうど位置していれば、バンドの矩形がハイライトされ、クリック&ドラッグによって、*x*方向の動きがカットオフ周波数の代わりにフィルターのバンド幅の調整に割り当てられることを示す。

特に小さな**filtergraph~**オブジェクトでは、エディットのためにバンド幅のラインを選択することができず、クリックするには幅が狭すぎるような高いQ値を作り出してしまうこともあるだろう。こうした場合などのため、クリックやドラッグの操作をしている間のマウスの働きを、ダブル・クリックによって反転できるようになっている。従って、狭いフィルターのバンド幅のラインをダブルクリックすれば、レゾナンスではなくカットオフ周波数を、マウスを使って設定できる。



## 入力

**実数** 第1から第5インレット：表示のみモードにおいて、最初の5つのインレットの実数は対応する**biquad~**のフィルター係数（それぞれ $a_0$ 、 $a_1$ 、 $a_2$ 、 $b_1$ 、 $b_2$ ）の現在の値を変更し、これらの値に基づいてフィルターの周波数特性を再計算し、そのフィルター係数を第1インレットからリストとして出力する。

第6インレット：フィルターの中心またはカットオフ周波数パラメーターを設定し、フィルター係数を出力する。

第7インレット：フィルターのゲイン・パラメーターを設定し、フィルター係数を出力する。

第8インレット：フィルターのQ（レゾナンス）またはS（スロープ）を設定し、フィルター係数を出力する。

注記：任意の1つのインレットに対する入力は、現在のフィルターのグラフを再計算し、フィルター係数を出力する。

**整数** 実数に変換される。

**リスト** 第1インレット：**biquad~**のフィルター係数に対応する5つの整数値からなるリストは、**filtergraph~**オブジェクト内部のフィルター係数の値を設定し、第1アウトレットからリストを出力する。**filtergraph~**が表示のみモードであれば、これらの係数から得られるフィルターの周波数特性を表示する。

**bang** 第1インレット：表示のみモードであれば、**bang**は、**filtergraph~**オブジェクトの内部に格納されている**biquad~**用のフィルター係数を、第1アウトレットから出力する。さらに、インタラクティブ・フィルター・モードであれば、**bang**によって現在のフィルター・パラメーターを、それぞれのアウトレットから出力する（出力の項を参照のこと）。

**aconstrain** 第1インレット：2つの実数を伴う**aconstrain**は、指定した範囲内に振幅値を制限する。これはクリック&ドラッグによって得られる値を制限するために有効である。**aconstrain 0. 0.**と指定することで、制限は取り除かれる。

**amp** 第1インレット：**amp**メッセージは周波数振幅表示を設定する。これは**spectrum 0**と同義である。

**autoout** 第1インレット：読み込み時に自動的に出力する機能を切り替える。**autoout 1**は**filtergraph~**にパッチが読み込まれた時に、自動的にその係数とパラメーターを出力するよう設定する。**filtergraph~**は現在の状態



をパッチに保存する。**autoout 0**はこの機能を無効にする。デフォルト値は1である。

- bandpass** 第1インレット：**bandpass**は、**filtergraph~**オブジェクトのフィルター・タイプをバンドパス・モードに設定する。これは**mode 3**メッセージと同じである。このフィルターの周波数特性は、cf（中心周波数、またはカットオフ周波数）、ゲイン、Q（レゾナンス）の3つのパラメーターに基づく。
- bandstop** 第1インレット：**bandstop**は、**filtergraph~**オブジェクトのフィルター・タイプをバンドストップ・モードに設定する。これは**mode 4**メッセージと同じである。このフィルターの周波数特性は、cf（中心周波数、またはカットオフ周波数）、ゲイン、Q（レゾナンス）の3つのパラメーターに基づく。
- brgb** 第1インレット：0から255までの3つの数値を伴う**brgb**は、**filtergraph~**オブジェクトの背景（すなわちフィルター曲線よりも上の部分）の色を、RGBフォーマットで設定する。デフォルト値は210 210 210である。
- cascade** 第1インレット：**cascade**メッセージは、表示のみモードだけで動作する。24グループまでのフィルター係数に対応する5つの実数値を伴う**cascade**は、**biquad~**フィルターのグループをカスケードに接続した複合フィルターの特性を表示する。
- color** 第1インレット：0から15までの数値を伴う**color**は、**filtergraph~**オブジェクトのカラーを、Objectメニューの**Color**サブメニューと同じ16個のオブジェクト・カラーの中の1つに設定する。
- display** 第1インレット：**display**は**filtergraph~**オブジェクトのフィルター・タイプを表示のみモードに設定する。これは**mode 0**メッセージと同義である。表示のみモードでは、**filtergraph~**は単に1組の5つの**biquad~**フィルター係数の周波数特性を表示する。
- displaydot** 第1インレット：**0**または**1**を伴う**displaydot**メッセージは、表示のみモードでの2つの赤丸の表示を切り替える。これは**filtergraph~**を、独自のフィルターの設計と表示のためのインターフェースとして使用したい時に、とりわけ役に立つだろう。デフォルト値は1である。
- domain** 第1インレット：Hz単位で周波数を表した2つの整数値を伴う**domain**メッセージは、**filtergraph~**の周波数表示の範囲を変更することができる。デフォルトの表示範囲は0 Hzからサンプリング・レート（ナイキスト周波数）の半分までである。





- fgrb** 第1インレット：0から255までの3つの数値を伴う**fgrb**は、**filtergraph~**オブジェクトの前景（すなわちフィルター曲線よりも下の部分）の色をRGBフォーマットで設定する。デフォルト値は170 170 170である。
- fconstrain** 第1インレット：2つの実数を伴う**fconstrain**は、周波数の値を指定した範囲に制限することができる。これはクリック&ドラッグによって得られる周波数の値を制限したい場合に有効である。**fconstrain 0. 0.**を入力すれば、制限は解除される。
- fullspect** 第1インレット：0または1を伴う**fullspect**は、ハーフ・スペクトラム表示かフル・スペクトラム表示のどちらかを選択することができる。**fullspect 0**（デフォルト）は、0 Hzからナイキスト周波数（サンプリング・レートの半分）までのハーフ・スペクトラムを指定する。**fullspect 1**は、-ナイキスト周波数から+ナイキスト周波数まで（スペクトラムは0 Hzで左右対称になる）のフル（左右対称）スペクトラムを指定する。フル・スペクトラム・モードでは、DC（0 Hz）に赤いマーカーが表示される。
- gainmode** 第1インレット：0または1を伴う**gainmode**は、ローパス、ハイパス、バンドパス、そしてバンドストップ・フィルターのためのゲイン・パラメーターを切り替える。これらのフィルターの伝統的な定義ではゲインは1.0に固定されているが、ゲインを有効にすることにより、フィルターの出力にシグナルのかけ算を追加することなく、振幅特性に倍率を掛けることができる。デフォルト値は0（無効）である。
- highorder** **highorder**メッセージは表示のみモードだけで動作する。 $n$ グループの**biquad~**フィルターの $a$ 係数と、 $n-1$ グループの**biquad~**フィルターの $b$ 係数を伴う**highorder**は、 $n$ 次フィルターの特性を表示する。
- highpass** 第1インレット：**highpass**は、**filtergraph~**オブジェクトのフィルター・タイプをハイパス・モードに設定する。これは**mode 2**メッセージと同じである。このフィルターの周波数特性は、 $cf$ （中心周波数、またはカットオフ周波数）、ゲイン、 $Q$ （レゾナンス）の3つのパラメーターに基づく。
- highshelf** 第1インレット：**highshelf**は、**filtergraph~**オブジェクトのフィルター・タイプをハイシェルフ・モードに設定する。これは**mode 7**メッセージと同じである。このフィルターの周波数特性は、 $cf$ （中心周波数、またはカットオフ周波数）、ゲイン、 $s$ （スロープ）の3つのパラメーターに基づく。
- lin** 第1インレット：**lin**メッセージはリニアな周波数表示スケールに設定する。これは**scale 0**メッセージと同じである。



- linmarkers** 第1インレット：64個までの整数のリストを伴う**linmarkers**は、リニアな周波数表示（**markers**メッセージを参照のこと）のマーカーを設定する。デフォルトでは、マーカーは±サンプル・レート/4、サンプル・レート/2、(3 \* サンプル・レート)/4に設定されている。
- log** 第1インレット：**log**メッセージは、対数的な周波数表示スケールに設定する。これは**scale 1**メッセージと同じである。
- logamp** 第1インレット：**0**または**1**を伴う**logamp**メッセージは、振幅表示スケールを設定する。**scale 0**はリニアな振幅表示（デフォルト）に設定し、**scale 1**は対数的な振幅表示スケールに設定する。
- logmarkers** 第1インレット：64個までの整数値のリストを伴う**logmarkers**は、対数的な周波数表示（**markers**メッセージを参照）のマーカーを設定する。デフォルトでは、マーカーは±50Hz、500Hz、5kHzと44.1kHzに設定されている。これらの値は任意のサンプル・レートで±0.007124、0.071238、そして0.712379ラジアンに対応する。
- lowpass** 第1インレット：**lowpass**は、**filtergraph~**オブジェクトのフィルター・タイプをローパス・モードに設定する。これは**mode 1**メッセージと同じである。このフィルターの周波数特性は、*cf*（中心周波数、またはカットオフ周波数）、ゲイン、*Q*（レゾナンス）の3つのパラメーターに基づく。
- lowshelf** 第1インレット：**lowshelf**は、**filtergraph~**オブジェクトのフィルター・タイプをローシェルフ・モードに設定する。これは**mode 6**メッセージと同じである。このフィルターの周波数特性は、*cf*（中心周波数、またはカットオフ周波数）、ゲイン、*S*（スロープ）の3つのパラメーターに基づく。
- markers** 第1インレット：64個までの周波数の値のリストを伴う**markers**は、グラフの後ろのこれらの周波数に視覚的マーカー（垂直線）を配置する。**markers**メッセージはリニアと対数の両方の周波数表示で使用されるマーカーを設定する。
- mode** 第1インレット：0-7の数値を伴う**mode**は、現在のフィルター・タイプを設定する。数値とそれに対応するフィルター・タイプは以下の通りである。

| 数値 | フィルター・タイプ          |
|----|--------------------|
| 0  | display only（表示のみ） |
| 1  | lowpass（ローパス）      |
| 2  | highpass（ハイパス）     |
| 3  | bandpass（バンドパス）    |
| 4  | bandstop（バンドストップ）  |





- 5 peaknotch (ピーク/ノッチ)
- 6 lowshelf (ローシェルフ)
- 7 highshelf (ハイシェルフ)

表示のみモードでは、**filtergraph~**は5つの**biquad~**フィルター係数の1組の周波数特性を表示する。他のモードでは、*cf* (中心周波数、またはカットオフ周波数)、ゲイン、*Q* (レゾナンス) または*S* (スロープシェルフ・フィルターの場合に用いられる) の3つのパラメータに基づくフィルターの周波数特性をグラフ表示する。

**mousemode** 第1インレット：2つの整数を伴う**mousemode**は、水平と垂直のマウスの動きの解釈を指定する。アーギュメントが1つの場合には、水平方向のマウス・モードだけに影響する。マウス・モードの値は両方の軸に対して同じであり、0=オフ、1=ノーマル、2=オルタネートである。

(第1アーギュメントで指定された) 水平方向の動きに関しては、通常の振る舞いはフィルター・バンドの上でクリックして水平方向にドラッグすると、フィルターのカットオフ周波数を変更することを意味する。オルタネート・マウス・モード (2) に設定すると、水平方向の動きは*Q*またはレゾナンスに影響する。オフ (0) に設定すると、マウスの*x*軸に関する動きは効果を持たない。

(第2アーギュメントで指定された) 垂直方向の動きに関しては、通常の振る舞いはクリック&ドラッグ操作をしている間は*y*軸がゲインに割り当てられることを意味する。オルタネート・マウス・モード (2) に設定すると、垂直方向の動きは代わりに*Q* (レゾナンス) 設定を変更する。オフ (0) に設定すると、垂直方向のマウスの動きは効果を持たない。

**params** 第1インレット：周波数、ゲイン、*Q*を指定する3つの数値を伴う**params**は、フィルターのパラメータを設定して出力を行う。

**peaknotch** 第1インレット：**peaknotch**は、**filtergraph~**オブジェクトのフィルター・タイプをピーク/ノッチモードに設定する。これは**mode 5**メッセージと同じである。このフィルターの周波数特性は、*cf* (中心周波数、またはカットオフ周波数)、ゲイン、*Q* (レゾナンス) の3つのパラメータに基づく。

**phase** 第1インレット：**phase**メッセージは周波數位相表示を設定する。これは**spectrum 1**メッセージと同じである。

**qconstrain** 第1インレット：2つの実数値を伴う**qconstrain**は、指定した範囲に*Q* (レゾナンス) の値を制限することができる。これは特定の状況にお



いて不適切になり得るQの設定を防ぐために有効である。最小値と最大値を同じ値に設定すれば（例えば**qconstrain 0.4 0.4**）、Qを特定の値に固定することができる。**qconstrain 0. 0.**を入力すれば、制限は解除される。

- query** 第1インレット：実数を伴う**query**は、現在のフィルターのその周波数での振幅と位相特性を、**filtergraph~**オブジェクトの第6アウトレットからリストとして出力する。
- range** 第1インレット：0より大きな実数を伴う**range**メッセージは、**filtergraph~**の振幅表示幅を設定する。振幅はグラフの垂直軸にそって0から範囲として設定した値（デフォルト値は2.0）まで表示される。
- rgb** 第1インレット：0から255までの3つの数値を伴う**rgb**は、**filtergraph~**の表示の色を設定する。オブジェクト表示の背景色は自動的に選択される。**brgb**、**frgb**、**rgb2**、**rgb3**、そして**rgb4**メッセージで**filtergraph~**オブジェクトの表示の個々の部分の色を設定することができる。
- rgb2** 第1インレット：0から255までの3つの値を伴う**rgb2**は、**filtergraph~**オブジェクトの曲線ライン（すなわちフィルター・カーブの上と下の部分に分けるライン）の色を、RGBフォーマットで設定する。デフォルト値は0 0 0（黒）である。
- rgb3** 第1インレット：0から255までの3つの値を伴う**rgb3**は、**filtergraph~**の表示マーカーの色を、RGBフォーマットで設定する。デフォルト値は0 0 0（黒）である。
- rgb4** 第1インレット：0から255までの3つの値を伴う**rgb4**は、**filtergraph~**オブジェクトの外枠の色を、RGBフォーマットで設定する。デフォルト値は0 0 0（黒）である。
- scale** 第1インレット：0または1を伴う**scale**メッセージは、周波数表示スケールを設定する。**scale 0**はリニアな周波数表示（デフォルト）に設定し、**scale 1**は対数的な周波数表示スケールに設定する。
- set** 第1インレット：**biquad~**フィルター係数に対応する5つの実数値を伴う**set**は、**filtergraph~**オブジェクト内部のフィルター係数の値を設定するが、出力は行わない。**filtergraph~**が表示のみモードの時、これらの係数から得られたフィルターの周波数特性を表示する。
- spectrum** 第1インレット：0または1を伴う**spectrum**は、周波数に対して振幅と位相のどちらを表示するのかを指定する。**spectrum 0**は周波数振幅表示（デフォルト）に設定し、**spectrum 1**は位相周波数表示に設定する。



- (loadbang) 第1インレット：**filtergraph~**はパッチが読み込まれた時に、**filtergraph~**に送られた**loadbang**メッセージに反応する（**autoout**メッセージを参照のこと）。
- (Get Info...) **filtergraph~**オブジェクトのインスペクター・ウィンドウを開く。
- (preset) **preset**オブジェクトを用いて、**filtergraph~**の設定を保存したり、呼び出したりすることができる。

## インスペクター

**filtergraph~**オブジェクトの振る舞いは、インスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている場合は、任意の**filtergraph~**オブジェクトを選択することで、フローティング・ウィンドウに**filtergraph~**のインスペクターを表示することができる。**filtergraph~**オブジェクトを選択して、Objectメニューから**Get Info...**を選ぶことでも、インスペクターを表示することができる。

**filtergraph~**のインスペクターは、以下のアトリビュートを設定することができる。

*Filter Type*ポップアップ・メニューは、**filtergraph~**オブジェクトによって表示されるフィルター・タイプを設定する。フィルター・タイプは**Display**（表示のみ、デフォルト）、**Lowpass**（ローパス）、**Highpass**（ハイパス）、**Bandpass**（バンドパス）、**Bandstop**（バンドストップ）、**Peak/Notch**（ピーク/ノッチ）、**Low Shelf**（ローシェルフ）、そして**High Shelf**（ハイシェルフ）である。表示のみモードの時には、**Display Dot Enabled**チェックボックスで、2つの赤丸を有効にすることができる。他のフィルター・モードの場合には、**Gain-Enabled**チェックボックスで表示のゲイン・スケーリングを有効にすることができる。

*Frequency Display*オプションで、表示の最小と最大周波数の範囲を設定することができる（デフォルト値は0 Hzと22050 Hz）、リニア（デフォルト）と対数表示スケール、そしてフル・スペクトラム表示を選択することができる。

*Amplitude Display*オプションでは、最小と最大の表示幅（デフォルト値は0.0625と16.0）を設定することができる。また、ラジオ・ボタンで**Amplitude Response**（振幅特性、デフォルト）または**Phase Response**（位相特性）を選択することもできる。振幅表示を選択している場合



には、値の表示をLinear（リニア、デフォルト）とLogarithmic（対数）から選択することもできる。

*Output Coefficients on Load*チェックボックスをチェックすると、フィルターを設定するメッセージをオブジェクトが受け取った時に、**filtergraph~**オブジェクトが**biquad~**フィルターの係数を出力するように設定する。

*Colors*ポップアップ・メニューは、swatchカラー・ピッカーまたはRGB値を用いて、**filtergraph~**オブジェクトの表示で用いられる色を指定することができる。これらは入力の項で説明されている。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

なし。

## 出力

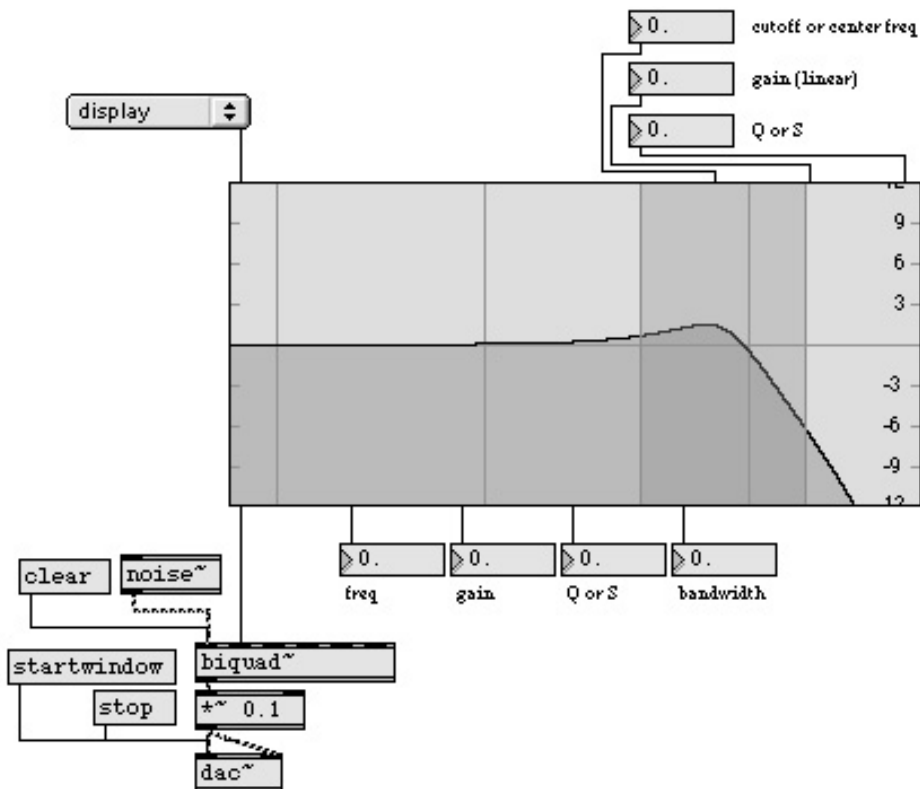
**リスト** 第1アウトレット：**biquad~**オブジェクトのための5つの浮動小数点のフィルター係数のリスト。これらの係数は、マウス・クリックされた時や、フィルター・パラメーターのインレットに入力があった時に出力される。このリストはオーディオ処理が開始された時にも出力され、自動出力のオプションが有効である場合（**autoout**メッセージを参照のこと）にも出力となる。

第6アウトレット：2つの浮動小数点の値（振幅、位相）のリストで、**query**メッセージが入力された時に出力される（**query**メッセージを参照のこと）。

**実数** 第2アウトレットから第5アウトレットまで：**filtergraph~**オブジェクトのクリックに反応して周波数、ゲイン（直線）、レゾナンス（Q）とバンド幅を出力する。



利用例



filtergraph~ オブジェクトは、biquad~ オブジェクトの設定を大幅に単純化する

参照

- allpass~** オールパス・フィルター
- biquad~** 2ポール、2ゼロのフィルター
- delay~** サンプル単位のディレイ・ライン
- lores~** レゾナンス付きローパス・フィルター
- reson~** レゾナンス付きバンドパス・フィルター
- teeth~** フィードフォワードとフィードバックにディレイ・コントロールが付いたコム・フィルター

## 入力

シグナル 累積計算するデータを入力。

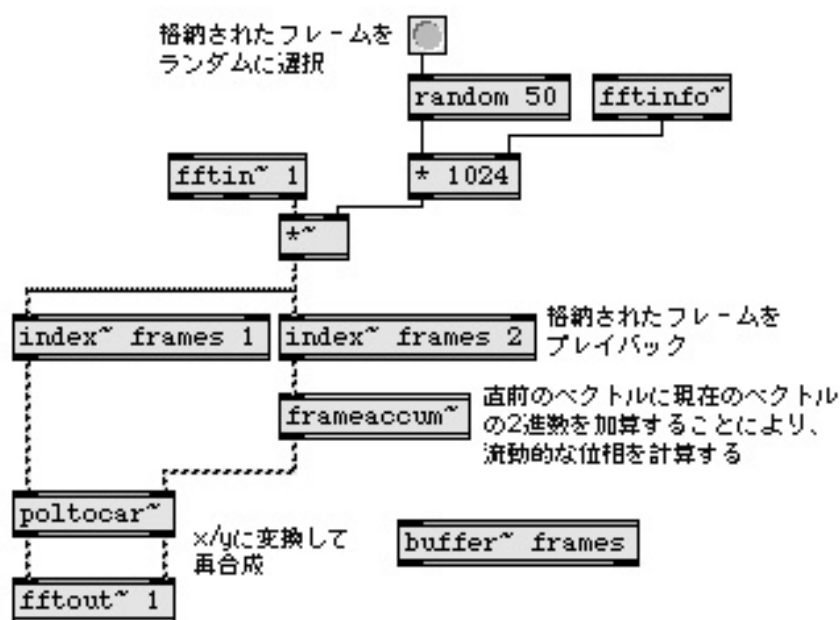
## アーギュメント

なし。

## 出力

シグナル **frameaccum~** オブジェクトは、入力されるシグナル・ベクターのデータ位置ごとに加算をし続けることで、その時点での位相を計算する。言い換えれば、出力シグナルの1サンプル目は、それまでに入力されたシグナル・ベクターの1サンプル目をすべて加算した結果であり、同じように2サンプル目は、それまでのすべての加算結果に2サンプル目が加算されたものである。残りのサンプルも同様に処理される。**pfft~** オブジェクト内部のサブパッチで**frameaccum~**を利用する場合、FFT処理サイズがシグナル・ベクター・サイズと同じであるため、FFTにおけるその時点での位相を計算し続けることとなる。

## 利用例



**frameaccum~** は、周波数データのフレーム間からその時点の位相を計算する

## 参照

### framedelta~

連続したFFT処理フレーム間の位相変化を計算

### Tutorial 26

Frequency Domain Signal Processing with **pfft~** (英文PDF参照)



## 入力

シグナル 変化量を計算したいシグナルを入力する。

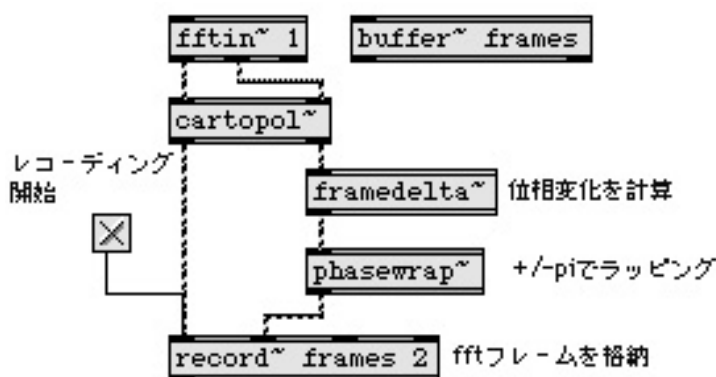
## アーギュメント

なし。

## 出力

シグナル **framedelta~** オブジェクトは、現在のシグナル・ベクターから、1回前に入力されたシグナルル・ベクターを減算することで、その時点での位相変化を計算する。言い換えれば、出力シグナルの1サンプル目は、現在のシグナルル・ベクターの1サンプル目から、1回前に入力されたシグナルル・ベクターの1サンプル目を引き算し、2サンプル目は2サンプル目同士で引き算をする、という具合にシグナル・ベクター間での差分処理が行われる。**pfft~** オブジェクト内部のサブパッチで **framedelta~** を利用する場合は、FFT処理サイズがシグナル・ベクター・サイズと同じであるため、FFTフレーム間のその時点での位相変化を計算することとなる。

## 利用例



**framedelta~** は、連続したFFTデータのフレーム間から差分を計算する

## 参照

**frameaccum~** 連続した位相変化データのフレームから位相を計算

Tutorial 26 Frequency Domain Signal Processing with **pfft~** (英文PDF参照)



## 入力

**実数または整数** 周波数の値。対応するMIDIノート・ナンバー（音高値、0から127）が、アウトレットから出力される。

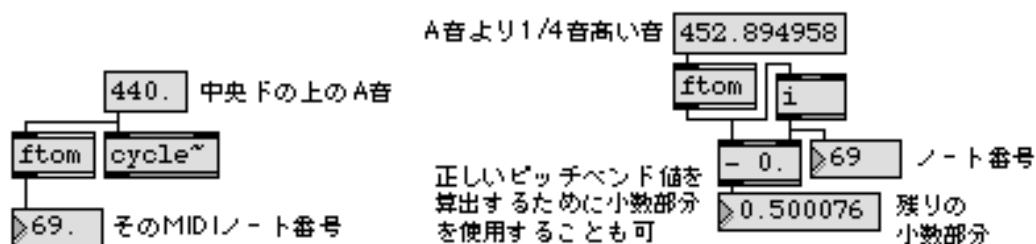
## アーギュメント

**実数** オプション。実数値をアーギュメントとしている場合は、**ftom**オブジェクトは実数を出力する。デフォルトでは、整数値を出力する。

## 出力

**整数または実数** 入力された周波数に対応するMIDIでの音高値。入力された周波数が、平均律の音程でちょうど2音の間になる場合、デフォルトの整数モードの場合は、近い整数値に四捨五入される。**ftom**がオプションの実数モードの場合、端数部分は小数として表される。端数部分を利用して、MIDIピッチ・ベンドで付加的にピッチ補正をすることができる。

## 利用例



MSPのオシレーターと同じ高さの音を演奏するためにMIDIキーナンバーを調べる

## 参照

**expr**  
**mtof**

数式の値を求める

MIDIノート・ナンバーを周波数に変換



## 入力

(マウス) マウスを用いて線分関数の中に点を描画することができる。また、描き終わった関数はMSPのコントロール・シグナルとして、**line~**オブジェクトに送ることができる。**function**オブジェクトの中の空白部分をクリックするとブレークポイントが追加され、そのブレークポイントはドラッグするとすぐに動かしはじめることができる (**function**オブジェクトに**clickadd 0**メッセージを送らない限り)。ブレークポイントをクリックするとドラッグによって動かすことができる (**function**オブジェクトに**clickmove 0**メッセージを送らない限り)。ブレークポイントのXとYの値はオブジェクトのボックスの上の部分に表示される。ブレークポイントをシフトを押しながらクリックすると、関数からそのポイントを削除する。ブレークポイントをcommandキー (Macintosh) あるいはCtrlキー (Windows) を押しながらクリックすれば、ポイントのサステーン特性を切り替えることができる。サステーン・ポイントは白で縁取りされる。マウスでの編集操作が完了した時には、常に**bang**が第2アウトレットから出力される。

Yの値が0のポイントは縁取り付きの円で、他のポイントは黒べたである。これによって、一目で $Y = 0$ で開始または終了しているかどうかを判別することができる。

**実数または整数** 値はXの値として解釈され、対応するYの値が第1アウトレットから出力される。Yの値は関数のリニア浮動小数点補間によって求められる。Xの値が最初または最後のブレークポイントよりも外側であれば、Yの値は0になる。

**bang** 第2アウトレットから現在のブレークポイントのリストを、**line~**オブジェクトで使用する形式で出力する。例えば、 $X = 1, Y = 0, X = 10, Y = 1$ 、そして $X = 20, Y = 0$ というブレークポイントを含む関数の場合には、出力は**0 1 9 0 10**となる。オプションの**output mode**が有効であれば、出力は**0 0 1 9 0 10**になる。

関数の中に1つでもサステーン・ポイントがあれば、**bang**はサステーン・ポイントまでのすべてのポイントのリストを出力する。次に続くサステーン・ポイントまたは最後のポイントまでの残りのポイントは、**next**メッセージが入力された時に、出力される。詳しくは**next**と**sustain**メッセージの説明を参照のこと。

**clear** **clear**はそれだけで既存のブレークポイントを消去する。**clear**の後に1つまたはそれ以上のブレークポイント・インデックス (0から始まる) を続けて、選択したブレークポイントだけを消去することもできる。

**clickmove** **clickmove 0**メッセージは、ユーザーが既存のブレークポイントをマウスでドラッグすることによって移動することを禁止する。**clickmove 1**はユーザーがブレークポイントをドラッグできるようにする。デフォルトの振る舞いは、ユーザーがブレークポイントをドラッグすること



を許可する。現在の設定はパッチが保存された時にオブジェクトとともに保存される。

- color** 0から15までの数値を伴う**color**は、表示されるブレイクポイントの色を指定したカラーに設定する。インデックスに対応するカラーはMaxメニューの**Color...**ダイアログに表示される。
- (Color...)** アンロックされたパッチ・ウィンドウで**function**オブジェクトを選択して、Maxメニューから**Color...**を選択することで、ブレイクポイントのカラーを変更することができる。
- domain** 実数または整数の値を伴う**domain**は、最大表示X値を設定する。最小値は常に0である。ブレイクポイントの実際の値は変更されないので、このメッセージによって新しい最大表示X値よりも大きなX値のブレイクポイントが見えなくなるかもしれない。
- dump** それぞれのブレイクポイントのXとYの値を含む一連の2つのアイテムからなるリストを、**function**オブジェクトの第2アウトレットから出力する。
- clickadd** **clickadd 0**メッセージはユーザーがクリックによって新しいブレイクポイントを追加することを禁止する。**clickadd 1**はユーザーが新しいブレイクポイントを追加することを許可する。デフォルトの振る舞いは、ユーザーが新しいブレイクポイントを追加することを許可する。現在の設定はパッチが保存された時にオブジェクトとともに保存される。
- fix** ポイントのインデックスを指定する数値と0または1を伴う**fix**は、2番目の数値が1の場合には、ユーザーがポイントを変更することを抑制し、2番目の数値が0の時には、ユーザーがポイントを変更することを許可する。デフォルトでは、すべてのポイントの移動を無効にするために、**clickmove 0**が送られていない限り、すべてのポイントは移動可能である。
- リスト** リストが2つの値を含んでいれば、新しいポイントが関数に追加される。1番目の値がXで2番目の値がYとなる。
- リストが3つの値を含んでいれば、関数の中の既存のポイントが変更される。1番目の値は変更するブレイクポイントのインデックス（0から始まる）、2番目はそのブレイクポイントの新しいX値、そして3番目はそのブレイクポイントの新しいY値である（リスト中のインデックスの数値が存在しないブレイクポイントを参照している場合には、そのメッセージは無視される）。
- next** **next**メッセージは、最後の**bang**または**next**メッセージによる出力が終わったサステイン・ポイントから、リスト出力を再開する。例えば、関数が(a)  $X = 1, Y = 0$ 、(b)  $X = 10, Y = 1$ 、(c)  $X = 20, Y = 0$ というブレ



ークポイントを含んでいて、ポイントbがサスティーン・ポイントだとすると、**bang**メッセージは**0, 1 9**を出力し、それに続く**next**メッセージは**1, 0 10**を出力する。**next**メッセージが最終ポイントに到達すると、それに続く**next**メッセージは**bang**メッセージと同等になる。他にも**next**メッセージが**bang**メッセージと同等になる時があるが、それはサスティーン・ポイントに到達するまでの**bang**メッセージが送られたことがない時か、関数がサスティーン・ポイントを含まない時である。

- nth** 数値を伴う**nth**は、数値をブレイクポイントのインデックス（0から始まる）として用い、そのブレイクポイントのY値を第1アウトレットから出力する。指定したインデックスのブレイクポイントが存在しない場合には、何も出力されない。
- outputmode** 1または0を伴う**outputmode**はオプションのoutput modeを有効（1の場合）または無効（0の場合）にする。オンであれば、オブジェクトが**bang**を受け取った時に、0を伴う1番目のY値の後に、残りのY値を伴う単独のリストを出力する。オフの時には、**bang**メッセージの説明で先述したような数値を出力する。オプションのoutput modeはデフォルトでオフである。
- range** 2つの実数または整数値を伴う**range**は、Y値の最小と最大の表示範囲を設定する。ブレイクポイントの実際の値は変更されないため、このメッセージによってビューからブレイクポイントが見えなくなることがあるかもしれない。
- setrange** 2つの実数または整数値を伴う**setrange**は、Y値の最小と最大の表示範囲を設定し、新しく与えられた範囲で同じ場所に表示されるようにY値を変更する。
- setdomain** 実数または整数値を伴う**setdomain**は、最大表示X値を設定し、新しく与えられたドメインで同じ場所に表示されるように、すべてのブレイクポイントのX値を変更する。
- sustain** ポイントのインデックスを指定する数値と0または1を伴う**sustain**は、2番目の数値が1であればポイントをサスティーン・ポイントに変え、2番目の数値が0であれば通常のポイントに変える。デフォルトでは、ポイントは通常（非サスティーン）である。サスティーン・ポイントの振る舞いは、**bang**メッセージの説明を参照のこと。commandキー（Macintosh）あるいはCtrlキー（Windows）を押しながらクリックすることでも、ポイントのサスティーン特性をトグルすることができる。
- （プリセット） **preset**オブジェクトを用いて**function**のブレイクポイントの設定を保存したり、呼び出したりすることができる。



## アーギュメント

なし。

## 出力

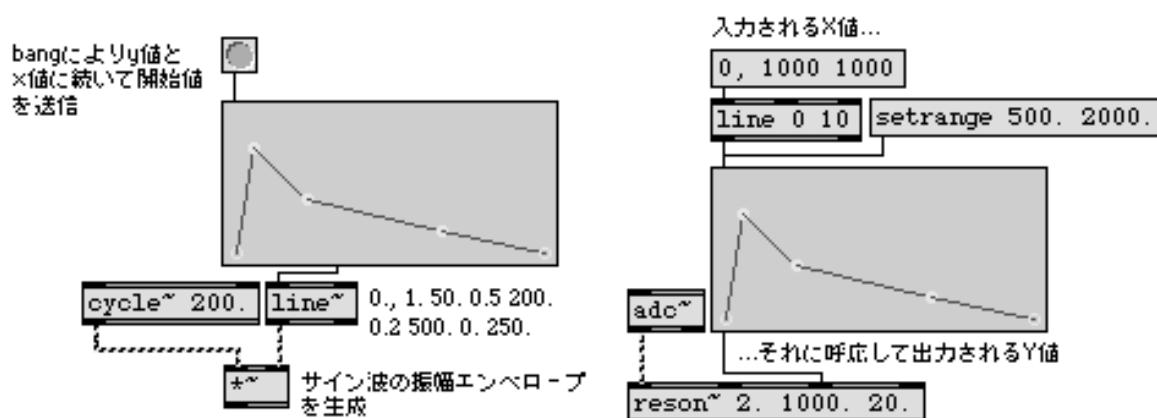
**実数** 第1アウトレット：インレットが受け取った実数または整数のX値に反応して補間されたY値が出力される。または、**nth**メッセージに反応して保存されているY値が出力される。

**リスト** 第2アウトレット：**bang**を受け取ると、保存されているY値の最初のものとして1つの実数が出力され、後に続く保存されているY値のインデックスとトランジション時間（Xと直前のXの間の違い）を示す数値のペアからなるリストが続く。

第3アウトレット：**dump**メッセージを受け取ると、**function**オブジェクトのブレークポイントのそれぞれのX値とY値を含む一連の2つのアイテムのリストが出力される。

**bang** 第4アウトレット：マウスによる編集操作が完了すると、**bang**が出力される。

## 利用例



線分の情報を **line~** に出力する、または個々のY値を参照（そして補間）する

## 参照

**line~** シグナル・ランプ・ジェネレーター

Tutorial 7 Synthesis: Additive synthesis (英文PDF参照)





## 入力

- シグナル** 第1インレット：スライダーでスケーリングする入力シグナル。
- 整数** 第1インレット：スライダーの値を設定し、指定された変化時間で新しい値に対応したレベルに出力シグナルをリニアに変化させ、スライダーの値を第2アウトレットから出力する。
- 実数** 第1インレット：整数に変換される。
- 第2インレット：変化時間をミリ秒単位で指定する。デフォルト値は10ミリ秒である。
- bang** 現在のスライダーの値を第2アウトレットから出力する。
- color** 第1インレット：0から15までの数値を伴う**color**は、**gain~**のストライプ模様を中心部分の色を、16のオブジェクト色の1つに設定する。この色は、Maxメニューの**Color...**から選択することでも設定できる。
- inc** 実数を伴う**inc**は、入力値に基づいて出力倍率を計算するのに用いられる増分値を設定する。デフォルト値は1.071519である。計算方法の説明については、インスペクターの項を参照のこと。
- resolution** 数値を伴う**resolution**は、サンプリング間隔をミリ秒単位で設定する。**gain~**オブジェクトのアウトレットから出力される数値の時間間隔と同時に、表示が更新される時間間隔も設定する。
- scale** 実数を伴う**scale**は、入力値に基づいて出力倍率を計算するのに用いられる基準出力値を設定する。デフォルト値は7.94231である。計算方法の説明については、インスペクターの項を参照のこと。
- set** 第1インレット：数値を伴う**set**は、スライダーの値を設定し、指定された変化時間で新しい値に対応したレベルまで出力シグナルをリニアに変化させるが、スライダーの値を第2アウトレットから出力しない。
- size** 第1インレット：数値を伴う**size**は、**gain~**の範囲をその数値に設定する。スライダーの値は0からレンジの値引く1までの値になる。デフォルト値は158である。

## インスペクター

**gain~**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**gain~**オブジェクトを選択することで、**gain~**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができ



る。**gain~**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**gain~**インスペクターでは、4つのパラメーター—*Range* (範囲)、*Base Value* (基準値)、そして*Increment* (増分) を設定することができる。入力値に基づいて出力倍率を計算する次の計算式で、範囲は $a$ 、基準値は $b$ 、増分は $c$ 、入力は $x$ 、 $e$ は自然対数の底 (約2.718282)、出力は $y$ である。

$$y = b e^{-a \log c} e^{x \log c}$$

これらのパラメーターに関するさらに詳しい情報は、**linedrive**オブジェクトを参照のこと。

範囲 (158) 基準値 (7.94231) 増分 (1.071519) というデフォルト値の場合は、スライダーが128でフル・スケール (1.0で掛算) になり、0は無音を生成し、1は値が127の場合の75.6 dB下になる。スライダーで10変わると、出力で6 dBの変化を生成する。加えて、範囲が158であるため、129から157までのスライダーの値は、17.4 dBのヘッドルームを生成する。スライダーが157の時、出力シグナルは入力シグナルよりも17.4 dB大きくなる。

*Interpolation Time*に数値を入力すると、**gain~**オブジェクトの補間時間を設定することができる。デフォルト値は10ミリ秒である。

**Revert**ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

なし。

## 出力

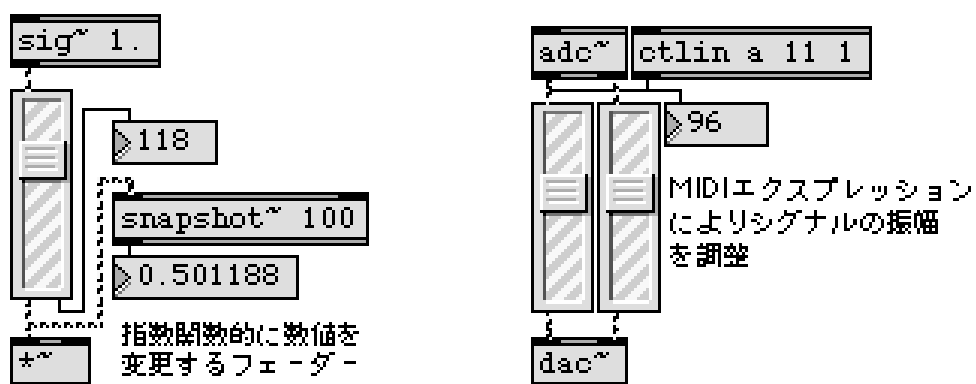
**シグナル** 第1アウトレット：入力シグナルを、現在のスライダーの値を上にした計算式の $x$ としてスケーリングしたシグナル。

**整数** 第2アウトレット：スライダーをマウスでドラッグした時や、整数または実数を第1インレットに入力した時の現在のスライダーの値。





## 利用例



シグナルを指数的または対数的にスケールするために特化したフェーダー

## 参照

**linedrive**

**line~** で用いるために指数的に数値をスケールする

## 入力

**整数** 第1インレット：第2インレットに入力されたシグナルを出力するアウトレットを指定する。0または負の数を指定した場合、第2インレットは閉じられて、無音が出力される。アウトレットの数より大きな数を指定した場合、シグナルは一番右側のアウトレットから出力される。シグナルが第1インレットに入力されている場合、**gate~**は整数および実数の入力を無視する。

**実数** 整数に変換される。

**シグナル** 第1インレット：シグナルが第1インレットに入力されている場合、そのシグナルの値が第2インレットに入力されたシグナルを出力するアウトレットを決定する。第1インレットに入力されたシグナルが0か負の数の場合、第2インレットは閉じられて、無音が出力される。第1インレットの値が1以上2未満の値である場合は、第2インレットの入力シグナルは第1アウトレットへルーティングされ、出力される。第2インレットの値が2以上3未満の値である場合は、第2インレットの値は第2アウトレットへルーティングされ、それ以降も同様の動作となる。第1インレットの値がアウトレットの数より大きい場合は一番右側のアウトレットへルーティングされる。

第2インレット：アウトレットのどれかへ流れるシグナルを入力する。出力されるアウトレットは、第1インレットに入力された整数か実数、またはシグナル値で決定される。

**begin~**オブジェクトを含むシグナルのネットワークが**gate~**の第2インレットに接続され、第1インレットにはシグナルが接続されていない場合、**begin~**のアウトレットから**gate~**までの間のMSPオブジェクトは**gate~**が閉じている間は処理されない。

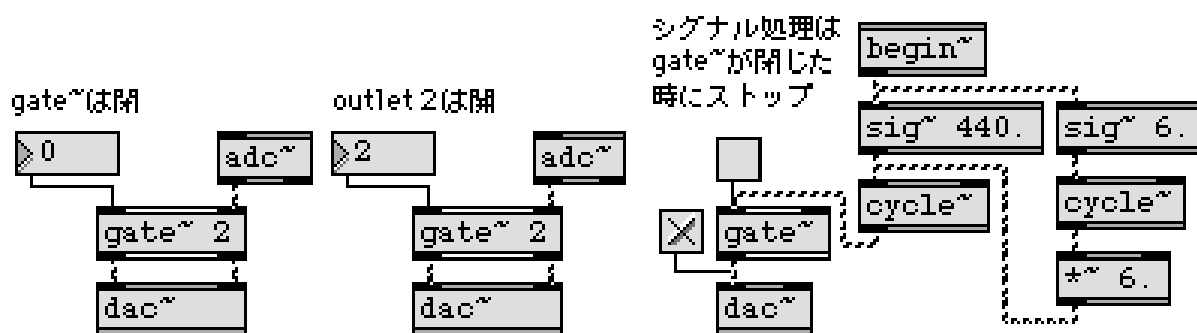
## アーギュメント

**整数** オプション。第1アーギュメントはアウトレットの数を指定する。デフォルト値は1である。第2アーギュメントは入力シグナルが出力されるアウトレットの初期値を指定する。デフォルト値は0であり、シグナル入力は閉じた状態で、すべてのアウトレットからは無音が出力される。第1インレットにシグナルが入力されている場合、第2アーギュメントは無視される。

## 出力

**シグナル** 第1インレットの値（シグナルまたは数値）によって、複数のアウトレットのうち、ひとつのアウトレットから入力シグナルが出力され、残りは無音の出力となる。または、インレットが閉じられていた場合、すべてのアウトレットから0値のシグナルを出力する。

## 利用例



gate~ は、入力シグナルをアウトレットのいずれかへ流すか、すべてを閉じる

## 参照

**selector~**  
**begin~**

複数の入力を切り替えて、1つのアウトレットから出力するシグナル・ネットワークの切り替え可能な部分を定義する

Tutorial 4

Fundamentals: Routing signals (英文PDF参照)

## 入力

**シグナル** 第1インレット：**buffer~**内のサウンドに対する再生速度を指定する。この値が0の場合は再生停止を表し、1の場合は通常速度での再生を、-1の場合は通常速度での逆再生を行う。2は通常速度の2倍の速度を表し、0.5は通常速度の半分による再生を表す。この再生速度を変化させることで、ビブラートやその他の速度変化による音響エフェクトとして利用することができる。

ループ開始点と終了点を指定し、かつ**looping**がオンの場合は、サウンドの再生位置がループ終了点まで来ると次の再生位置をループ開始点に設定することで、現在の再生速度で連続的にループ再生を行う。

第2インレット：ループ開始点をミリ秒単位で指定する。

第3インレット：ループ終了点をミリ秒単位で指定する。

**整数または実数** 第1インレット：サンプル再生位置をミリ秒単位で指定する。0はサンプル・データの最初の位置を表す。

第2インレット：ループ開始点をミリ秒単位で指定する。このインレットにシグナルが入力されている場合は、整数や実数を入力しても無視される。

第3インレット：ループ終了点をミリ秒単位で指定する。このインレットにシグナルが入力されている場合は、整数や実数を入力しても無視される。

**loop** **loop 1** を入力することでループ再生をオンにする。**loop 0** はループ再生をオフにする。デフォルトではループ再生はオフとなっている。

**loopinterp** **loopinterp 1** を入力することで、サンプル間の補間を行うループ再生をオンにする。**loop 0** を指定することで、ループ再生補間をオフにできる。デフォルトでは、ループ再生補間はオフである。

**reset** ループの開始点および終了点をクリアする。

**set** **set**に続き**groove~**が再生に利用する**buffer~**の参照名をシンボルで指定する。

**setloop** **setloop**に続き、ループの開始点および終了点の2つの数値をミリ秒単位で指定する。

**startloop** ループ開始点からループ再生を開始する。ループが指定されていない場合は、データの最初から再生する。

(マウス) **groove~**オブジェクトをダブルクリックすると、波形表示ウィンドウが開き、参照名で関係付けた**buffer~**オブジェクトの中身を見ることができる。

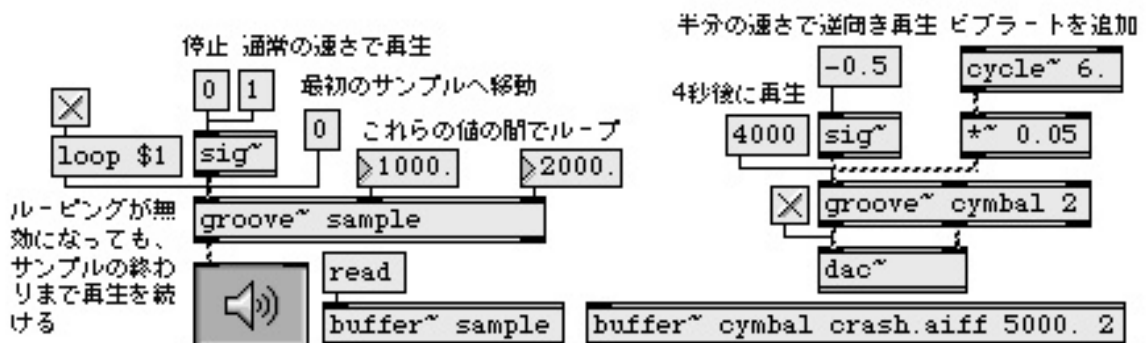
## アーギュメント

- シンボル** 必須。**groove~**がループ再生に利用する**buffer~**の参照名を指定する。
- 整数** オプション。第2アーギュメントで出力チャンネル数 (1, 2, 4) を指定することができる。デフォルトのチャンネル数は1である。指定したチャンネル数より**buffer~**が持つチャンネル数のほうが少ない場合、余分なチャンネルは無音出力となる。**buffer~**側のほうがチャンネル数が多い場合は、チャンネルをミックスして力される。

## 出力

- シグナル** 第1アウトレット：サンプル再生のシグナルが出力される。2つまたは4つの出力チャンネルを持つ場合、第1インレットは左チャンネルのシグナルが出力される。
- 中間のアウトレット：サンプル再生のシグナルが出力される。2つまたは4つの出力チャンネルを持つ場合、第2インレットは左チャンネル以外のシグナルが出力される。
- 右端のアウトレット：同期シグナルが出力される。ループ再生が行われている間、この同期シグナルはループ開始点を0、終了点を1とする値でその間の値となる。

## 利用例



## 参照

**2d.wave~** 2次元のウェーブテーブル  
**buffer~** オーディオ・サンプルの格納  
**play~** ポジション・ベースのサンプル再生  
**record~** サウンドを**buffer~**のサンプル・メモリへ録音する

Tutorial 14 Sampling: Playback with loops (英文PDF参照)

Tutorial 20 MIDI control: Sampler (英文PDF参照)

## 入力

- シグナル** 第1インレット：逆高速フーリエ変換される複素数シグナルの実部。
- 第2インレット：逆高速フーリエ変換される複素数シグナルの虚部。
- 第1インレットおよび第1アウトレットのみにシグナルのパッチ・コードが接続されている場合、実部のみのIFFT（逆高速フーリエ変換）が実行される。それ以外の場合では複素数でのIFFTが実行される。

## アーギュメント

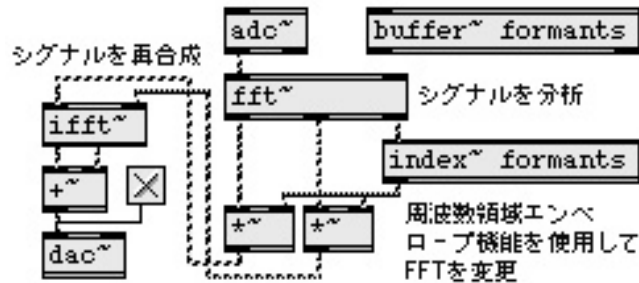
- 整数** オプション。第1アーギュメントは、IFFT処理を行うサンプル・ポイント数を指定する。この数値は2の累乗数でなければならない。デフォルトは512サンプルである。第2アーギュメントは、次のIFFT処理フレームまでのサンプル数を指定する。この数値も2の累乗のサンプル・ポイント数を指定する。デフォルトは512サンプルである。第3アーギュメントは、IFFT処理を実行するシグナル・ベクター内での処理を開始する地点を、サンプル数のオフセット値として指定する。この数値は0またはシグナル・ベクター・サイズの倍数のどちらかを指定する。**ifft~**は間違ったアーギュメントを指定した場合、訂正を試みるが、**ifft~**オブジェクトを作成した後にシグナル・ベクター・サイズやオフセット値を変更した場合は、ベクター・サイズの倍数ではなくなる可能性があるため、オーディオ処理を開始しても、正常に動作しないことがある。

## 出力

- シグナル** 第1アウトレット：入力シグナルに対する逆フーリエ変換後の実部を出力する。この出力はIFFT処理すべき入力シグナルが、すべて入力された時点で出力される。
- 第2アウトレット：逆フーリエ変換後の虚部データを出力する。この出力はIFFT処理すべき入力シグナルが、すべて入力された時点で出力される。
- 第3アウトレット：**0**で始まりFFT出力ポイント数より1小さい数字までの数値列を、シグナルデータとしてIFFT出力に同期して出力する。IFFTの結果が出力されない場合（IFFTを実行する時間間隔がIFFT処理サイズより大きい場合など）は、このシグナルは**0**として出力される。



利用例



fft~とifft~の利用による解析と再合成

参照

- cartopol** 直交座標を極座標に変換する
- cartopol~** シグナルについて直交座標を極座標に変換する
- fft~** 高速フーリエ変換
- fftin~** pfft~に読み込まれるパッチにおける入力
- fftinfo~** pfft~に読み込まれるパッチに関する情報の取得
- fftout~** pfft~に読み込まれるパッチにおける出力
- frameaccum~** 連続した位相変化データのフレームから位相を計算
- framedelta~** 連続したFFT処理フレーム間の位相変化を計算
- pfft~** 周波数領域処理を行うパッチの管理
- poltoCAR** 極座標を直交座標に変換する
- poltoCAR~** シグナルについて極座標を直交座標に変換する
- vectral~** ベクター・ベースのエンベロープ・フォロワー

Tutorial 25

Analysis: Using the FFT (英文PDF参照)

## 入力

なし。

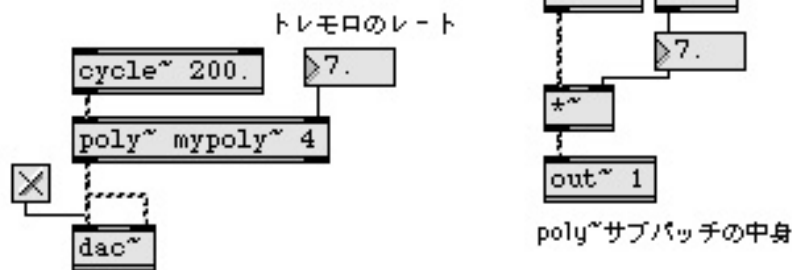
## アーギュメント

**整数** 必須。それぞれのinオブジェクトは、重複しないインデックス番号によって区別され、poly~オブジェクトのどのインレットへのメッセージに対応するかを決定する。1つ目のインレットの番号は1である。

## 出力

**メッセージ** poly~オブジェクトに読み込まれたパッチ内のそれぞれのinオブジェクトは、poly~オブジェクトの上部のインレットとして表れる。poly~オブジェクトの1つ目のメッセージ・インレットに受け取られたメッセージは、読み込まれたパッチ内の1つ目のinオブジェクト（つまり、in 1オブジェクト）に送られる。他の場合でも、同じように処理される。poly~オブジェクトの全体のインレットの数は、読み込まれたパッチ内のin~オブジェクトかinオブジェクトのアーギュメントのより大きな値によって決定される（例えば、読み込まれたパッチが3つのin~オブジェクトと、2つのinオブジェクトを持っていた場合、そのpoly~オブジェクトは3つのインレットを持ち、2つのインレットがシグナルとメッセージの両方を受け付け、3つ目のインレットがシグナルの入力のみを受け付けることになる）。

## 利用例



poly~オブジェクトのメッセージのインレットは、読み込まれたパッチ内のinオブジェクトに対応する

## 参照

- in~ poly~オブジェクトによって読み込まれるパッチのためのシグナルの入力
- out poly~オブジェクトによって読み込まれるパッチのためのメッセージの出力
- out~ poly~オブジェクトによって読み込まれるパッチのためのシグナルの出力
- poly~ パッチのポリフォニック処理とオーディオ処理を管理する
- thispoly~ poly~オブジェクトのボイス割当とミュートのコントロール

## 入力

なし。

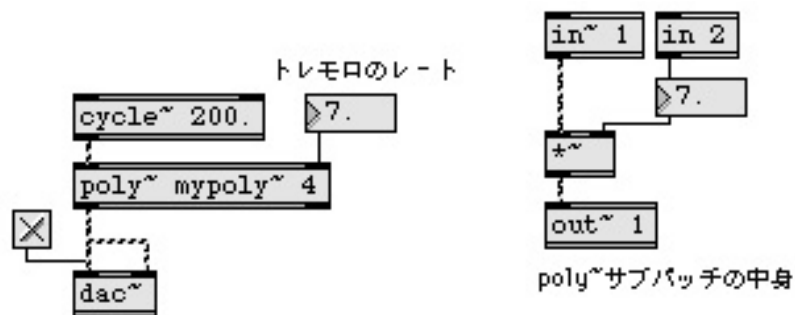
## アーギュメント

**整数** 必須。それぞれのin~オブジェクトは、重複しないインデックス番号によって区別され、poly~オブジェクトのどのインレットへのシグナルに対応するかを決定する。1つ目のインレットの番号は1である。

## 出力

**シグナル** poly~オブジェクトに読み込まれたパッチャー内のそれぞれのin~オブジェクトは、poly~オブジェクトの上部のインレットとして表れる。poly~オブジェクトの1つ目のメッセージ・インレットに受け取られたシグナルは、読み込まれたパッチャー内の1つ目のin~オブジェクト（つまり、in~ 1オブジェクト）に送られる。他の場合でも、同じように処理される。poly~オブジェクトの全体のインレットの数は、読み込まれたパッチ内のin~オブジェクトかinオブジェクトのアーギュメントのより大きな値によって決定される（例えば、読み込まれたパッチャーが3つのin~オブジェクトと、2つのinオブジェクトを持っていた場合、そのpoly~オブジェクトは3つのインレットを持ち、2つのインレットがシグナルとメッセージの両方を受け付け、3つ目のインレットがシグナルの入力のみを受け付けることになる）。

## 利用例



poly~ オブジェクトのシグナルのインレットは、読み込まれたパッチ内のin~ オブジェクトに対応する

## 参照

**in** poly~オブジェクトによって読み込まれるパッチのためのメッセージの入力  
**out** poly~オブジェクトによって読み込まれるパッチのためのメッセージの出力  
**out~** poly~オブジェクトによって読み込まれるパッチのためのシグナルの出力  
**poly~** パッチのポリフォニック処理とオーディオ処理を管理する  
**thispoly~** poly~オブジェクトのヴォイス割当とミュートのコントロール

## 入力

- シグナル** 第1インレット：参照名で関連付けた**buffer~**に記録されたサンプル・データをサンプル番号に従って読み出す。
- 整数** 第2インレット：**buffer~**から取り出すデータのチャンネルを指定する。デフォルトは対応する**buffer~**の1つめのチャンネルとなる。
- set** **set**に続いて**buffer~**オブジェクトに付けられた参照名を与えて、読み出す**buffer~**を指定する。
- (マウス) **index~**オブジェクトをダブル・クリックすると、波形編集ウィンドウが開き、参照名で関連付けた**buffer~**オブジェクトの中身を見ることができる。

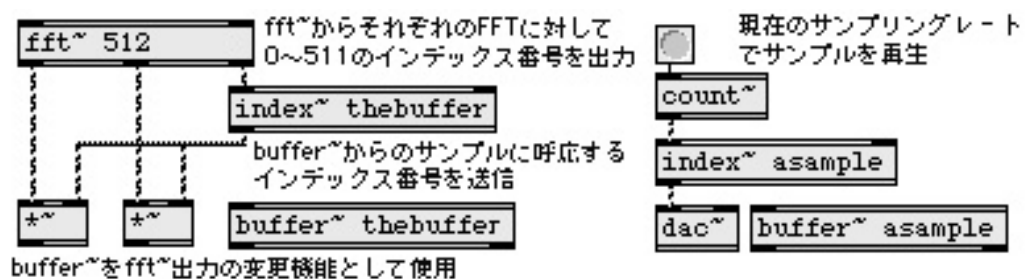
## アーギュメント

- シンボル** 必須。サンプルの保管場所として用いる**buffer~**オブジェクトの参照名を指定する。
- 整数** オプション。**buffer~**参照名に続いて記述することで、その**buffer~**の読み出すチャンネルを指定する。デフォルトはチャンネル1となる。

## 出力

- シグナル** インレットに入力されたシグナルの数値をインデックス番号として、**buffer~**から読み出したサンプルを出力する。入力サンプルが整数でない場合、サンプルの補間を行わない。

## 利用例



buffer~をfft~出力の変更機能として使用

index~を利用したbuffer~内の特定サンプルの参照

## 参照

- 2d.wave~** 2次元のウェーブテーブル
- cycle~** テーブル参照可能なオシレーター
- buffer~** オーディオ・サンプルの格納

**buffir~**

**fft~**

**wave~**

**buffer~**を利用したFIRフィルター

高速フーリエ変換

サイズ可変のウェーブテーブル再生

Tutorial 13

Sampling: Recording and playback (英文PDF参照)

## 入力

**bang** 第1インレット：参照名で関連付けた**buffer~**に格納されたサンプル・データに関する情報を出力する。

(マウス) **info~**オブジェクトをダブル・クリックすると、波形表示ウィンドウが開き、参照名で関連付けた**buffer~**オブジェクトの中身を見ることが出来る。

## アーギュメント

**シンボル** 必須。サンプルの保管場所として用いる**buffer~**オブジェクトの参照名を指定する。

## 出力

**info~**オブジェクトが出力する情報の大半は、関連付けされた**buffer~**で最も最近に読み込まれたオーディオ・ファイルに関する情報である。この情報が得られない場合であっても、サンプリング・レートだけは第1アウトレットから出力される。何も出力されない場合は、サウンド・ファイルの情報を失ったこととなる。

**実数** 第1アウトレット：サンプリング・レートを出力。

第3アウトレット：サステイン・ループの開始位置をミリ秒単位で出力。

第4アウトレット：サステイン・ループの終了位置をミリ秒単位で出力。

第5アウトレット：リリース・ループの開始位置をミリ秒単位で出力。

第6アウトレット：リリース・ループの終了位置をミリ秒単位で出力。

第7アウトレット：関連付けされた**buffer~**オブジェクトの内部データの総時間をミリ秒単位で出力

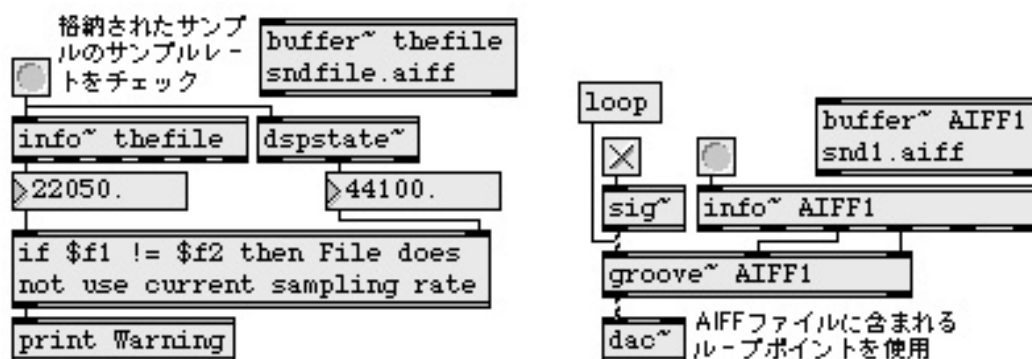
第8アウトレット：最も最近に読み込んだオーディオ・ファイルのファイル名を出力。

**リスト** 第2アウトレット：サンプルに関する次の楽器情報を、リストとして出力。

1. MIDIピッチ情報。

2. MIDIのピッチベンドにおけるベース・ノート・ナンバーからのデチューン量（ピッチベンド量）。
3. サンプルを再生する場合の最低MIDIノート・ナンバー。
4. サンプルを再生する場合の最高MIDIノート・ナンバー。
5. サンプルを再生する場合の最低ベロシティ値。
6. サンプルを再生する場合の最高ベロシティ値。
7. サンプルのゲイン値（0から127）。

## 利用例



サンプル・レートの確認; サンプル・データに含まれる他の情報の利用

## 参照

buffer~  
mstosamps~  
sinfo~

オーディオ・サンプルの格納  
ミリ秒をサンプル数に変換  
オーディオ・ファイルの情報を出力する

Tutorial 14  
Tutorial 20

Sampling: Playback with loops (英文PDF参照)  
MIDI control: Sampler (英文PDF参照)



## 入力

- シグナルまたは実数 第1インレット：現在インデックスで参照されているオシレーターに対して、周波数を設定する。デフォルト値は0である。
- 第2インレット：現在インデックスで参照されているオシレーターに対して、振幅を設定する。
- 第3インレット：**framesync 1**メッセージによってフレーム同期機能が有効である場合、現在インデックスで参照されているオシレーターに対して、0から1.0の値で位相を設定する。
- 第4インレット：参照するオシレーターのインデックスを設定する。
- 実数** 第3インレット：現在参照されているオシレーターに対して、0から1の範囲の実数で位相を設定する。
- clear** すべてのオシレーターに対して周波数を0Hzに、振幅を0に設定する。
- copybuf** 第1インレット：**copybuf**に続いて**buffer~**の参照名のシンボルを指定することで、その**buffer~**から4096サンプルを内部ウェーブテーブルとしてコピーする。オプションとして、参照名に続く整数のアーギュメントを指定することで、読み込み位置をサンプル単位で（オフセットとして）指定できる。
- framesync** **framesync**に続けて0でない数値を指定することで、フレーム同期処理を有効にする。フレーム同期処理が有効である場合、インデックスで参照されているオシレーターの出力は、入力シグナルの値が0の場合（またはN個のサンプルずつのフレームごとに）に変化（または次の値に対するリニアな補間値の計算を開始）する。デフォルトではこの機能はオフである。
- freqsmooth** **freqsmooth**に続いて1つの数字を指定し、周波数変化のスムージング処理を何サンプルで行うか決める。デフォルト値は**1**（スムージングなし）である。
- magsmooth** **magsmooth**に続いて1つの数字を指定し、振幅変化のスムージング処理を何サンプルで行うか決める。デフォルト値は**0**（振幅のスムージングなし）である。
- set** **set**に続いて2つの実数を指定することで、オシレーター・バンク中の1つのオシレーターに対して、周波数と振幅を設定する。 $n$ 対の実数をリストとして入力することで、**ioscbank~**中の最初の $n$ 個のオシレーターに対して設定を行い、残りのオシレーターの振幅を0にすることができる。

**silence** すべてのオシレーターの振幅を0にする。

**size** **size**に続いて1つの数字を指定し、オシレーターの数を設定する。デフォルト値は64である。

## アーギュメント

**整数** オプション。第1アーギュメントはオシレーターの数を指定する。デフォルト値は1個である。

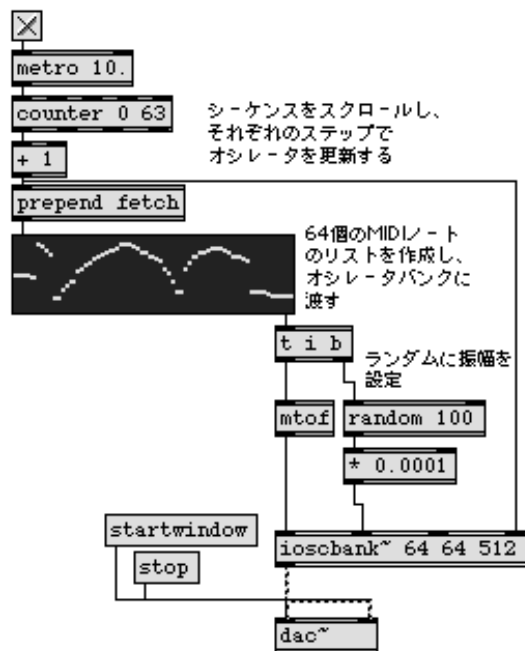
**整数** オプション。第2アーギュメントは周波数変化のスムーシング処理を何サンプルに渡って行うかを指定する。

**整数** オプション。第3アーギュメントは振幅変化のスムーシング処理を何サンプルに渡って行うかを指定する。

## 出力

**シグナル** 周波数および振幅を設定した各オシレーター出力の総和となる波形が出力される。

## 利用例



**ioscbank~**によって、1つのオブジェクトで補完機能付きの複数のオシレーターによるサウンドを扱うことができる

## 参照

**oscbank~** 補間機能なしのオシレーター・バンク

## 入力

**シグナル** 第1インレット：**kink~**への入力は、**phasor~**オブジェクトからの、0から1まで繰り返し変化するノコギリ波の出力であるのが一般的である。

第2インレット：出力（Y）値が0から0.5までに変化する間の出力のスロープに影響を与える乗数。出力が0.5に達すると、波形は入力と同じ周期の時間で0から1まで全体のアウトプットが変化するように、残りの時間で1まで増加する。スロープ乗数1（デフォルト値）はいかなる歪みも生じさせない。1より小さいスロープ乗数では、入力よりも0.5まで到達するのがゆっくりになり、1より大きなスロープ乗数では入力よりも速く0.5まで到達するようになる。

**実数** 第2インレット：シグナルと同じ。シグナルが第2インレットに入力されていれば、実数の入力は無視される。

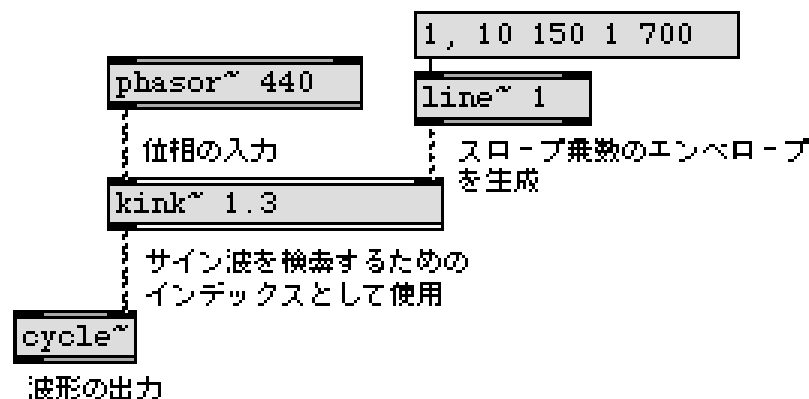
## アーギュメント

**実数** オプション。デフォルトのスロープ乗数を設定する。シグナルが第2インレットに入力されていれば、アーギュメントは無視される。

## 出力

**シグナル** **kink~**の出力は、歪んだサイン波を生成する（フェーズ・ディストーション・シンセシスとして知られるテクニック）ために、**cycle~**（周波数0）の第2インレットに入力することができる。**kink~**の第2インレットのスロープ乗数が1から離れるにつれて、倍音が**cycle~**の波形出力に加わる。**line~**を用いてスロープ乗数が急激に増加した後で減少すると、**cycle~**の出力は楽器音のアタック部分に似たものになるだろう。

## 利用例



典型的な **kink~** の用法で、**phasor~** と **cycle~** の間で使う。

**参照**

**phasor~**  
**triangle~**

ノコギリ波の生成  
三角波のウェーブテーブル

## 入力

**リスト** 最初の数値は目標値を指定する。2番目の数値はその値に到達するためにどれだけの時間がかかるかをミリ秒単位で指定する。指定された時間の中で、**line~**オブジェクトは現在の格納された値から目標値へ勾配を持つシグナルを生成する。**line~**オブジェクトは複合の勾配を生成するために、目標値と到達時間のペアのリストを最高で64個受け取ることができる。(例えば**0 1000 1 1000**というメッセージは現在の値から1秒間かけて0に到達し、次の1秒間で1に到達する。)勾配の中のひとつが目標値に到達すると、次のものがスタートする。新たに第1インレットからリスト、実数、または整数が入力された場合は、まだ生成されていないすべての勾配は消去される。

**実数、または整数** 第1インレット：第2インレットから入力される数値によって指定された時間内に到達する目標値。最後の目標値への時間が指定されない場合は、到達時間は0と考えられて、出力シグナルは即座に目標値となる。

第2インレット：出力シグナルが目標値に到達する時間をミリ秒で指定する。

## アーギュメント

**実数、または整数** オプション。出力シグナルの初期値を設定する。

## 出力

**シグナル** 第1アウトレット：現在の目標値、または現在格納されている値と目標時間に従って目標値に向かって変化する勾配を出力する。

**bang** 第2アウトレット：**line~**オブジェクトがすべての勾配を生成し終えた時、**bang**を出力する。

## 利用例



直線的に変化するシグナル、またいくつかの直線の断片からなる関数

## 参照

### click~ curve~

インパルスの生成  
指数曲線、対数曲線のランプ・ジェネレーター

### Tutorial 2

Fundamentals: Adjustable oscillator (英文PDF参照)

## 入力

**整数または実数** 第1インレット：次式に従った値を計算する。

$$y = be^{-alogc} e^{xlogc}$$

ここで、 $x$ は入力する数値、 $y$ は出力する数値、 $a$ 、 $b$ および $c$ はアーギュメントで指定する数値、 $e$ は自然対数の底（およそ2.718282）である。

出力は2つの項からなるリストで、上式の計算結果である $y$ の値、および第2インレットへ最も最近に入力されたディレイ時間の数値を出力する。

**整数** 第2インレット：出力リストでの上記のスケールリングされた数値に続いて出力するディレイ時間を設定する。このオブジェクトに接続されたline~オブジェクトの出力シグナルは、ここで指定したディレイ時間をかけて、新しい目標値まで徐々に変化する。

## アーギュメント

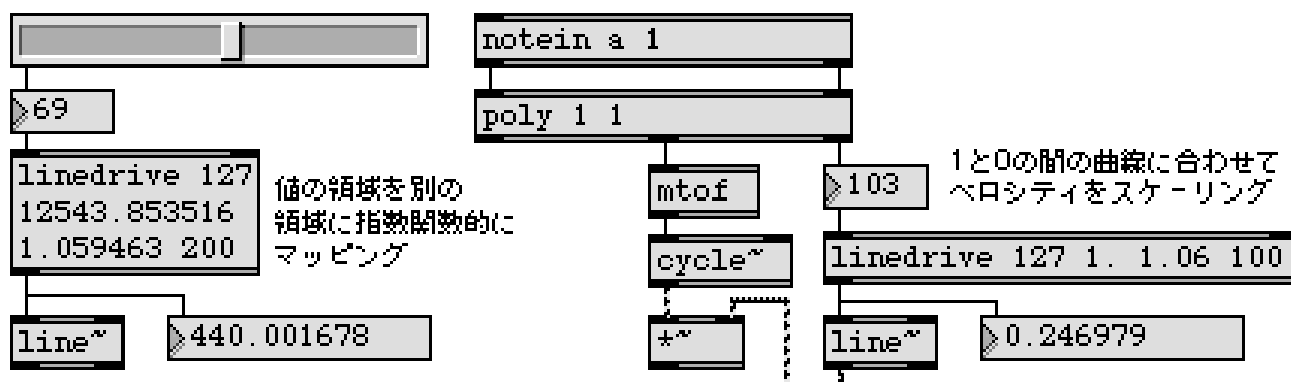
**実数** 必須。第1アーギュメントは入力の最大値を、第2アーギュメントは出力の最大値を指定する。第3アーギュメントは尺度曲線の性質を指定する。第3アーギュメントが小さな値（0から1の間の値）の場合、尺度曲線はより対数的なものとなり、大きな値（1より大きい値）の場合、指数関数的により急峻な変化をする曲線となる。この第3アーギュメントは正の数のみが受け付けられる。第4アーギュメントはミリ秒単位で表したディレイ時間の初期値である。この値は第2インレットへの数値入力に変更できる。

## 出力

**リスト** 整数か実数が第1インレットに入力された時、上式による計算結果とその時点で設定されているディレイ時間をリストとして出力する。



## 利用例



linedriveの利用例

## 参照

**expr**  
**line~**

数式の値を求める  
シグナル・ランプ・ジェネレーター

## 入力

**シグナル** 第1インレット：**log~**はアーギュメント、または最も新しく第2インレットに入力された値を底として、入力シグナルの対数を出力する。入力シグナルの値が0以下場合、**log~**オブジェクトは0.00000001の値を出力する。

**実数、または整数** 第2インレット：対数の底を設定する。デフォルト値は0で、それは自然対数（底がeまたは2.71828182）と同値である。底が1の対数は常に0である。

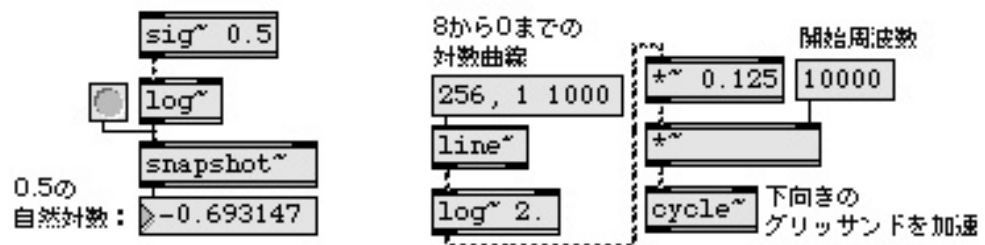
## アーギュメント

**実数、または整数** オプション。対数の底を設定する。デフォルト値は0である。

## 出力

**シグナル** アーギュメント、または最も新しく第2インレットに入力されている値を底として、入力シグナルの対数を出力する。

## 利用例



曲線を作り出すのに、底を指定したシグナルの対数が使われている

## 参照

**pow~** シグナルの累乗計算をする  
**curve~** 指数曲線、対数曲線のランプ・ジェネレーター  
**sqrt~** シグナルの平方根

## 入力

**シグナル** 第1インレット：**buffer~**内部のデータに対し、振幅値へマッピングされるシグナルを入力する。入力シグナルの各サンプルは-1から1の間の値で、現在の**buffer~**内部のサンプル・データに対応するようマッピングされる。-1から0の値は**buffer~**のサンプルオフセット後のデータの前半と、0から1は**buffer~**の後半のデータとマッピングされる。入力シグナルの振幅値が-1から1の範囲を超える場合は、出力は0となる。

第2インレット：第1インレットに入力されるシグナルにマッピングされる**buffer~**内のサンプルに対し、マッピング位置のオフセット値を設定する。設定したオフセットの位置は、第1インレットのシグナルでの-1に対応する。

第3インレット：参照テーブルとして利用する**buffer~**オブジェクトのサンプル数を指定する。第1インレットに入力される-1から1の範囲のシグナルは、ここで設定されたサンプル数に相当する範囲にマッピングされる。このサンプル数のデフォルトは512である。**lookup~**オブジェクトは各シグナルベクターに対して参照処理を行う前にテーブルサイズを変更する。この場合、第3インレットに入力されるシグナル・ベクターの1つ目のサンプルが、テーブル・サイズとして利用される。

**整数または実数** 参照テーブルのオフセットとテーブル・サイズの設定は、第2インレットと第3インレットに数値を入力することで変更することができる。これらのインレットにシグナルのパッチ・コードが接続されている場合、そのインレットへの数値入力は無視される。

**set** **set**に続き、参照テーブルとして関連付ける**buffer~**オブジェクトの参照名シンボルを指定する。

**(マウス)** **lookup~**オブジェクトをダブル・クリックすることで、波形描画ウィンドウが開き、参照テーブルである**buffer~**内のデータを見ることができる。

## アーギュメント

**シンボル** 必須。**lookup~**オブジェクトがテーブルとして参照する**buffer~**の参照名を指定する。

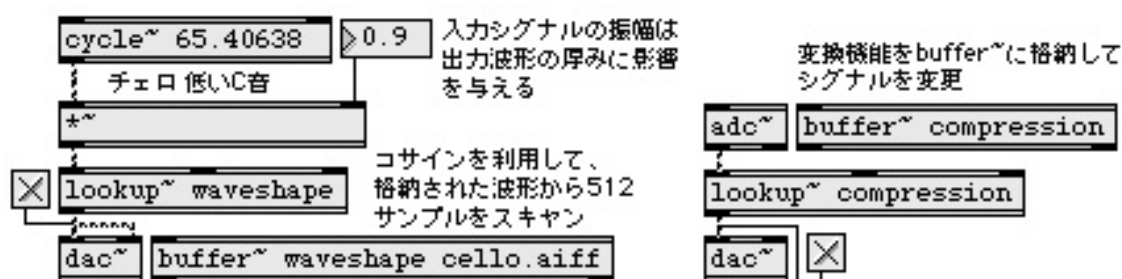
**整数** オプション。**buffer~**の参照名の後に、**buffer~**内でのサンプル・オフセット値を指定する。指定されたオフセットの位置が、テーブル参照時の-1の位置となる。デフォルトのオフセット値は0である。オフセット値の後にはテーブル・サイズの数値を指定することもできる。このテーブル・サイズのデフォルトは512である。**buffer~**に複数のチャン

ネルのデータがある場合、**lookup~**は常に1つめのチャンネルのデータを参照する。

## 出力

**シグナル** 第1インレットに入力された-1から1の範囲のシグナルに対し、関連付けた**buffer~**内の指定した範囲を参照テーブルとして、各サンプル毎に値を読み出して出力する。

## 利用例



## 参照

**buffer~**                    オーディオ・サンプルの格納  
**peek~**                    サンプル値の読み出しと書き込み

Tutorial 12                    Synthesis: Waveshaping (英文PDF参照)

## 入力

- シグナル** 第1インレット：フィルター処理が行われるシグナル。
- 第2インレット：ローパス・フィルターのカットオフ周波数を設定する。
- 第3インレット：0（最小のレゾナンス）から1（最大のレゾナンス）のレゾナンス係数を設定する。1に非常に近い値は、ある特定の種類の入力シグナルではクリッピングを生じるかもしれない。
- 整数または実数** 整数または、実数メッセージはカットオフ周波数、またはレゾナンスを変化させるために第2、または第3インレットに入力される。インレットにシグナルが接続されている場合は、そのインレットに入力される数値は無視される。
- clear** フィルターのメモリをクリアする。**lores~** オブジェクトは再帰的フィルターであるため、このメッセージが暴走状態から回復するために必要になる場合がある。

## アーギュメント

- 整数または実数** オプション。数値はカットオフ周波数とレゾナンスの初期値を設定する。デフォルトは両方とも0である。シグナルが第2インレット、または第3インレットに接続されている場合は、インレットからの入力と一致しているアーギュメントは無視される。

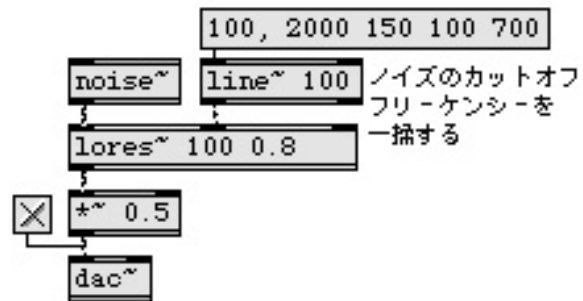
## 出力

- シグナル** フィルター処理が行われたシグナルが出力される。フィルターの式は

$$y_n = scale * x_n - c1 * y_{n-1} + c2 * y_{n-2}$$

である。*scale*と*c1*と*c2*は、カットオフ周波数とレゾナンス係数によって計算されるパラメーターである。

## 利用例



ローパス・フィルターのカットオフ周波数とレゾナンスを指定する

## 参照

|                     |                                           |
|---------------------|-------------------------------------------|
| <b>biquad~</b>      | 2ポール、2ゼロのフィルター                            |
| <b>buffir~</b>      | <b>buffer~</b> を利用したFIRフィルター              |
| <b>comb~</b>        | コム・フィルター                                  |
| <b>filtergraph~</b> | グラフィカル・フィルター・エディター                        |
| <b>onepole~</b>     | 単極のローパス・フィルター                             |
| <b>reson~</b>       | レゾナンス付きバンドパス・フィルター                        |
| <b>teeth~</b>       | フィードフォワードとフィードバックにディレイ・コントロールが付いたコム・フィルター |

**matrix~**オブジェクトはシグナルのコネクターとミキサー（加算器）の配列であり、任意のインレット数およびアウトレット数を持つことができる。各インレットに入力されたシグナルは、可変量によるゲイン調整をされるとともに、1つ以上のアウトレットへ経路接続される。1つのアウトレットが複数のインレットと経路接続されている場合、そのアウトレットの出力は経路接続されたインレットへの入力シグナルの和となる。

**matrix~**オブジェクトは処理に「一対モード」と「非一対モード」の2つのモードを持っている。「一対モード」では、インレットとアウトレットとの間の経路接続が単純なスイッチとして働き、常に単一ゲインで動作する。このモードは動作が速いが、このオブジェクト内の経路接続を切り替えるたびに、出力シグナルには「カチッ」というような音が混入してしまう。「非一対モード」では、経路接続の時点で0でも1でもない値でゲイン調整がされる。また、ゲインの値が変化した場合、ゲインの変化にかかる傾斜の時間も与えられる。これによってシグナルの切り替え時の「カチッ」という音が抑えられる。

## 入力

**シグナル** 任意のインレット：インレットに入力されたシグナルは、そのインレットから経路接続されたアウトレットへ送られ、経路ごとにゲイン調整が行われる。

**リスト** 第1インレット：**matrix~**が一対モードの場合、3つの整数からなるリストはインレットとアウトレットを経路接続する。リストの1つ目の整数はインレットを、2つ目の整数はアウトレットを、3つ目の整数は経路の接続か切断を表すのに用いられる。3つ目の整数が0でない場合、指定したインレットとアウトレットを接続する。0の場合はインレットとアウトレットとの間の接続を切断する。

**matrix~**オブジェクトが非一対モードである場合、2つの整数と1つの実数からなるリストで経路接続を行う。このリスト中の3つ目の実数値は指定経路で調整されるゲインを指定する。注記：各経路接続のゲインを設定するには、**connect**メッセージではなく、3つの要素からなるリストを用いる必要がある。**connect**メッセージによる経路接続は、通常は**matrix~**オブジェクト生成時の第3アークギュメントで指定したゲインの初期値を用いる。しかし、その後にはリストの入力があった場合は、そのリストのゲインが**connect**メッセージで指定した経路に適用される。

**print** 第1インレット：入力した**matrix~**オブジェクト内で設定されているすべての経路接続を、接続ごとのリストとしてMaxウィンドウに表示する。このリストはインレットとアウトレットを表す2つの数値、およびその接続に適用されるゲインの実数からなる。



- dump** 第1インレット：入力した**matrix~**オブジェクト内で設定されているすべての経路接続の現状を、接続ごとのリストとして一番右側のアウトレットから出力する。このリストはインレットとアウトレットを表す2つの数値、およびその接続に適用されるゲインの実数からなる。
- clear** 第1インレット：すべての経路接続を切断する。
- connect** 第1インレット：**connect**に続いて1つ以上の整数の対を指定することで、1つ目の整数が指すインレットと2つ目の整数が指すアウトレットとを経路接続する。複数の経路接続は、そのメッセージに続いて追加で対の整数を指定すれば良い。インレットおよびアウトレットは左から右の順に0から始まる数で表される。例を挙げると、**connect 1 0 1 1**というメッセージは第2インレットから第1および第2アウトレットへの経路接続を行う。
- disconnect** 第1インレット：**disconnect**に続いて1つ以上の整数の対を指定することで、1つ目の整数が指すインレットと2つ目の整数が指すアウトレットの経路を切断する。複数の経路切断は、そのメッセージに続いて追加で対の整数を指定すれば良い。
- ramp** 第1インレット：非一対モードの場合、**ramp**に続いてミリ秒単位の時間を表す数を指定することで、ゲイン調整にかける時間を変更できる。デフォルト値は10ミリ秒となっている。

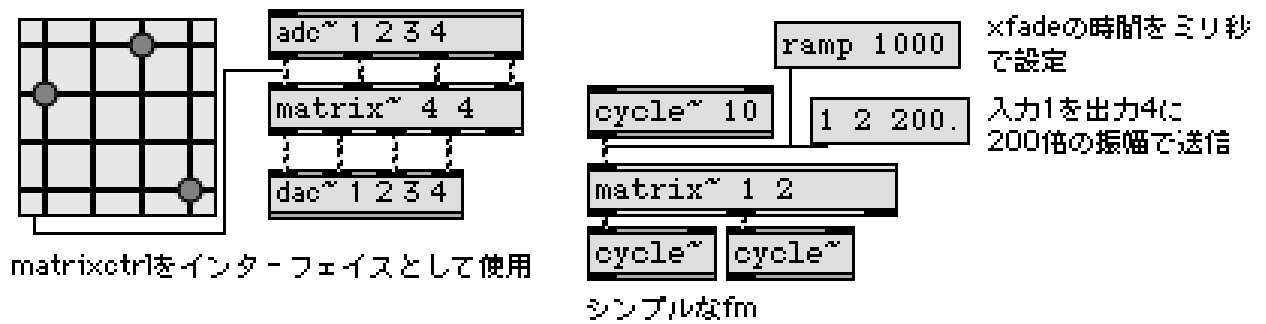
## アーギュメント

- 整数** 必須。第1アーギュメントはインレット数を指定する。
- 整数** 必須。第2アーギュメントはアウトレット数を指定する。
- 実数** オプション。第3アーギュメントとして実数を指定した場合、**matrix~**は非一対モードとして動作する。アーギュメントは接続された経路に対するゲインの値を設定する。

## 出力

- シグナル** 各アウトレットから出力されるシグナルは、各々のアウトレットに経路接続されたインレットへ入力されたシグナルの総和で、ゲイン調整されたものである。

## 利用例



マルチ・チャンネルのオーディオ・シグナルの経路設定

## 参照

- gate~** 複数アウトレットの1つへシグナルを切り替える
- matrixctrl** マトリックス状のスイッチ・コントロール
- receive~** パッチ・コードなしでシグナルを受信する
- selector~** 複数の入力を切り替えて、1つのアウトレットから出力する
- send~** パッチ・コードなしにシグナルを送信する

## 入力

**シグナル** 第1インレット：第2インレットから入力されたシグナル、または定数と比較されるシグナル。2つの値のうち、より大きい方がアウトレットから出力される。

第2インレット：第1インレットに入力されたシグナルとの比較に使われるシグナル。

**実数または整数** 第2インレット：第1インレットに入力されたシグナルと比較される数値。第2インレットにシグナルが入力された場合は、実数や整数は無視される。

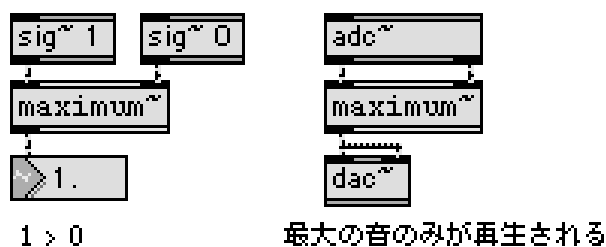
## アーギュメント

**実数または整数** オプション。第1インレットに入力されたシグナルと比較する初期値を設定する。第2インレットにシグナルが入力された場合は、アーギュメントは無視される。

## 出力

**シグナル** 第1インレットと第2インレットに入力されたシグナルを比較し、より大きい方を出力する。

## 利用例



2つのシグナルを比較し、より大きい方を検出する

## 参照

- <=~** 2つのシグナルの比較を行い、小さいか等しければ1を出力する
- >~** 2つのシグナルの比較を行い、大きければ1を出力する
- >=~** 2つのシグナルの比較を行い、大きい等しければ1を出力する
- ==~** 2つのシグナルの比較を行い、等しければ1を出力する
- !=~** 2つのシグナルの比較を行い、等しくなければ1を出力する
- minimum~** 2つのシグナルの比較を行い、小さい方を出力



## 入力

- シグナル** 入力されるシグナルのピーク振幅は、スクリーン上のレベル・メーターのLEDで表示される。
- brgb** 0から255までの数値を伴う**brgb**は、**meter~**オブジェクトの背景色のためのRGB値を設定する。デフォルト値は**brgb 104 104 104**で設定される色である。
- frgb** 0から255までの数値を伴う**frgb**は、**meter~**オブジェクトの低いレベルのLED色のためのRGB値を設定する。デフォルト値は**frgb 0 68 0**で設定される色である。
- interval** 数値を伴う**interval**は、**meter~**オブジェクトの更新時間間隔をミリ秒単位で指定する。最小の更新間隔は10ミリ秒、最大は2秒、デフォルト値は100ミリ秒である。このメッセージは**meter~**がピーク値を出力する時間間隔も同時に設定する。
- rgb2** 0から255までの数値を伴う**rgb2**は、**meter~**オブジェクトの高いレベルのLED色のためのRGB値を設定する。デフォルト値は**rgb2 255 153 0**で設定される色である。
- rgb3** 0から255までの数値を伴う**rgb3**は、**meter~**オブジェクトのレベル・オーバー時のインジケーターのLED色のためのRGB値を設定する。デフォルト値は**rgb3 255 0 0**で設定される色である。
- (マウス) パッチ・ウィンドウがアンロックされている時、リサイズ・エリアをドラッグして形状を変更することで、**meter~**オブジェクトを水平から垂直に置き換えることができる。

## インスペクター

**meter~**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**meter~**オブジェクトを選択することで、**meter~**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**meter~**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**meter~**インスペクターでは、*Interval*ボックスに数値を入力することで、表示の更新時間間隔をミリ秒単位で設定することができる。デフォルト値は100ミリ秒である。



Colorポップアップ・メニューでは、swatchカラー・ピッカーまたはRGB値を用いて**meter~**オブジェクトで使用される色を指定することができる。*Background*は**meter~**オブジェクトの背景色を設定する。デフォルトの背景色は104 104 104である。*Foreground*は**meter~**オブジェクトの低いレベルの色を設定する。デフォルト値は0 168 0である。*Warning*は**meter~**オブジェクトの高いレベルのLEDの色を設定する。デフォルト値は255 153 0である。*Overload*は**meter~**オブジェクトのレベル・オーバーのインジケータのLEDの色を設定する。デフォルト値は255 0 0である。

Revertボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

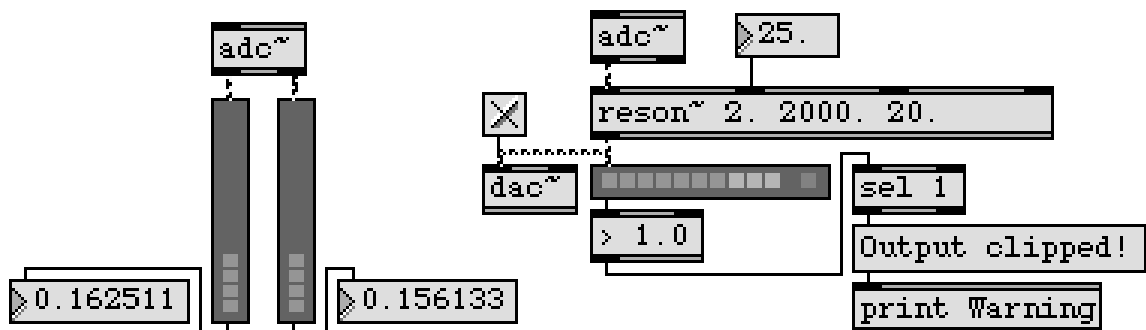
## アーギュメント

なし。

## 出力

**実数** 前の更新間隔の間に受け取ったシグナルのピーク（絶対）値は、オーディオ処理がオンの時に、アウトレットから出力される。

## 利用例



**meter~**はシグナルのピーク振幅を表示し、出力する

## 参照

**average~**  
**scope~**

マルチ・モードのシグナル平均化  
シグナル・オシロスコープ

## 入力

**シグナル** 第1インレット：第2インレットから入力されたシグナル、または定数と比較されるシグナル。2つの値のうち、より小さい方がアウトレットから出力される。

第2インレット：第1インレットに入力されたシグナルとの比較に使われるシグナル。

**実数または整数** 第2インレット：第1インレットに入力されたシグナルと比較される数値。第2インレットにシグナルが入力された場合は、実数や整数は無視される。

## アーギュメント

**実数または整数** オプション。第1インレットに入力されたシグナルと比較する初期値を設定する。第2インレットにシグナルが入力された場合は、アーギュメントは無視される。

## 出力

**シグナル** 第1インレットと第2インレットに入力されたシグナルを比較し、より小さい方を出力する。

## 利用例



## 参照

- <=~** 2つのシグナルの比較を行い、小さいか等しければ1を出力する
- >~** 2つのシグナルの比較を行い、大きければ1を出力する
- >=~** 2つのシグナルの比較を行い、大きいか等しければ1を出力する
- ==~** 2つのシグナルの比較を行い、等しければ1を出力する
- !=~** 2つのシグナルの比較を行い、等しくなければ1を出力する
- maximum~** 2つのシグナルの比較を行い、大きい方を出力

## 入力

- シグナル** 入力されたシグナルの最小値と最大値を計算する。
- bang** シグナルの最小値と最大値に対応する値を、実数として、それぞれ3、4番目のアウトレットから出力する。
- reset** 現在の最小値、最大値をデフォルト値 (0) にリセットする。

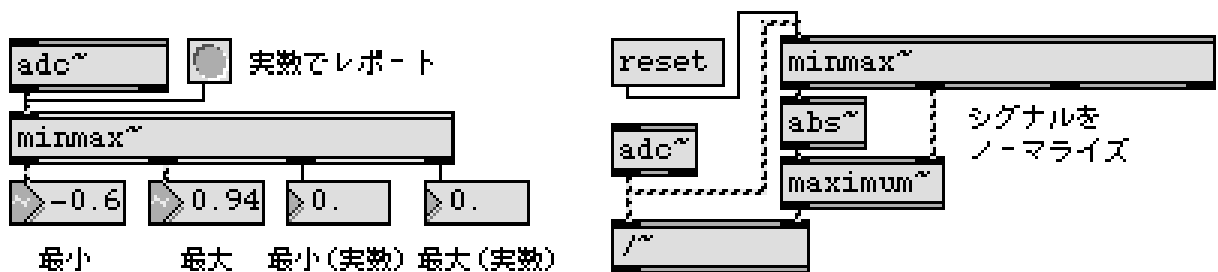
## アーギュメント

なし。

## 出力

- シグナル** 第1アウトレット：処理を開始した時、または最後に**reset**メッセージを受け取った時点から、最も小さい値であるシグナル値。  
第2アウトレット：処理を開始した時、または最後に**reset**メッセージを受け取った時点から、最も大きい値であるシグナル値。
- 実数** 第3アウトレット：**minmax~**オブジェクトに**bang**メッセージを受け取ると、処理を開始した時、または最後に**reset**メッセージを受け取った時点から、最も小さい値である実数値を出力する。  
第4アウトレット：**minmax~**オブジェクトに**bang**メッセージを受け取ると、処理を開始した時、または最後に**reset**メッセージを受け取った時点から、最も大きい値である実数値を出力する。

## 利用例



シグナルの最大／最小のピークを検出する

## 参照

- meter~** 視覚的なピーク・レベル・インジケータ
- peakamp~** 入力シグナルのピーク値を出力する
- snapshot~** シグナルの値を数値に変換する



## 入力

**実数または整数** 入力されたミリ秒数を、現在のサンプリング・レートにもとづきサンプル数に変換し、第2アウトレットから出力する。出力するサンプル数は、小数部分を含むことができる。例えば、サンプリング・レートが44.1kHzの場合、3.2ミリ秒は、141.12サンプルに変換される。

**シグナル** シグナルとして入力されるミリ秒値。現在のサンプリング・レートにもとづきサンプル数に変換し、第1インレットから出力する。出力するサンプル値は、小数部分を含むことがある。

## アーギュメント

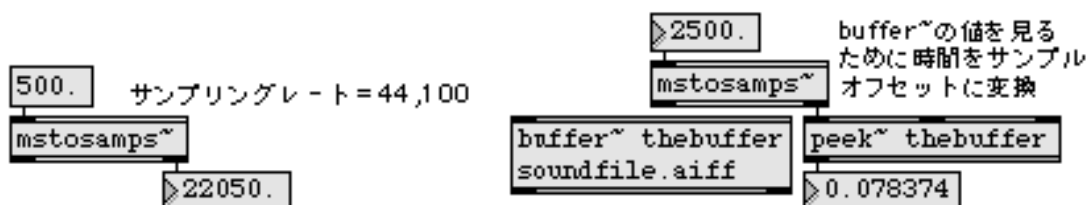
なし。

## 出力

**シグナル** 第1アウトレット：入力されたシグナルのミリ秒値に対応するサンプル数から成るシグナル。

**実数** 第2アウトレット：インレットに実数または整数として受け取ったミリ秒値に対応するサンプル数。

## 利用例



ミリ秒単位で表されている時間をサンプル数として出力する

## 参照

**dspstate~** 現在のDSPの設定を出力する  
**sampstoms~** サンプル数をミリ秒に変換

## 入力

**実数または整数** 0から127のMIDIノート・ナンバー。対応する周波数の値が、アウトレットから出力される。

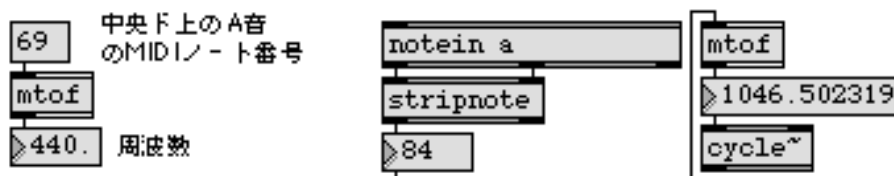
## アーギュメント

なし。

## 出力

**実数** 入力されたMIDIでの音高値に対応する周波数。

## 利用例



MIDIノート・ナンバーを周波数に変換し、オシレータに与える

## 参照

**expr**  
**ftom**

数式の値を求める  
周波数をMIDIノート・ナンバーに変換

## 入力

**整数** 1は、**mute~**オブジェクトのアウトレットに接続されたサブパッチに含まれるすべてのオブジェクトの信号処理を停止させ、0は再び開始する。

## アーギュメント

なし。

## 出力

**mute~**オブジェクトのアウトレットを、コントロールしたいと思うサブパッチの任意のインレットに接続する。**mute~**は好きなだけ多くのサブパッチに接続することができる。しかし、**mute~**は**bpatcher**オブジェクトの内部のパッチには作用しない。

## 利用例



任意のパッチやサブパッチのすべての処理をミュートすることができる

## 参照

**begin~**  
**pass~**

シグナル・ネットワークの切り替え可能な部分を定義する  
ミュートされたサブ・パッチでのノイズを取り除く

Tutorial 5

Fundamentals: Turning signals on & off (英文PDF参照)

## 入力

なし。

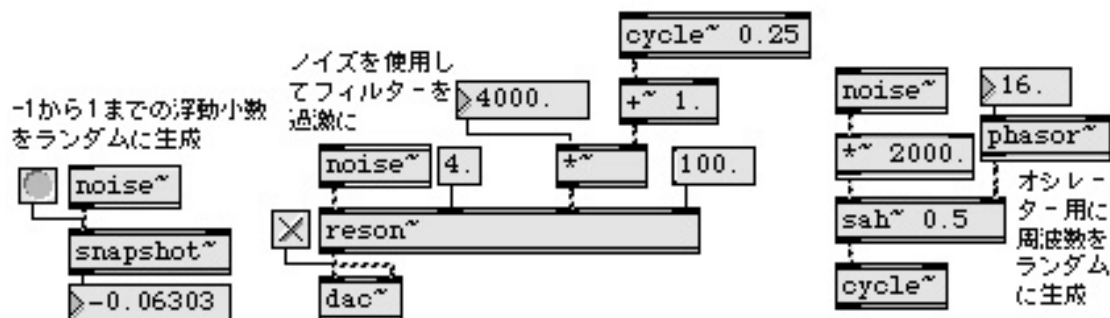
## アーギュメント

なし。

## 出力

シグナル **noise~** オブジェクトは、-1から1の値を一様にランダムに分散した値からなるシグナルを生成する。

## 利用例



ホワイト・ノイズによって作られるランダムな例。様々な方法でフィルター処理を行うことができる

## 参照

**biquad~**

2ポール、2ゼロのフィルター

**pink~**

ピンク・ノイズを発生させるジェネレータ

**reson~**

レゾナンス付きバンドパス・フィルター

Tutorial 3

Analysis: Wavetable oscillator (英文PDF参照)

## 入力

- シグナル** 第1インレット：入力シグナルはノーマライズされる。つまり、ピーク振幅が指定された最大値になるように伸縮される。
- 第2インレット：出力の総合的なスケールリングのための最大出力振幅。
- 実数** 第2インレット：最大出力振幅はシグナルの代わりに実数として送られることもある。シグナルが第2インレットに入力されていると、第2インレットで受け取った実数は無視される。
- reset** 第1インレット：数値を伴う **reset** は最大入力振幅をその数値にリセットする。**reset** が数値を伴わない、または数値が0であれば、最大入力振幅は0.000001に設定される。

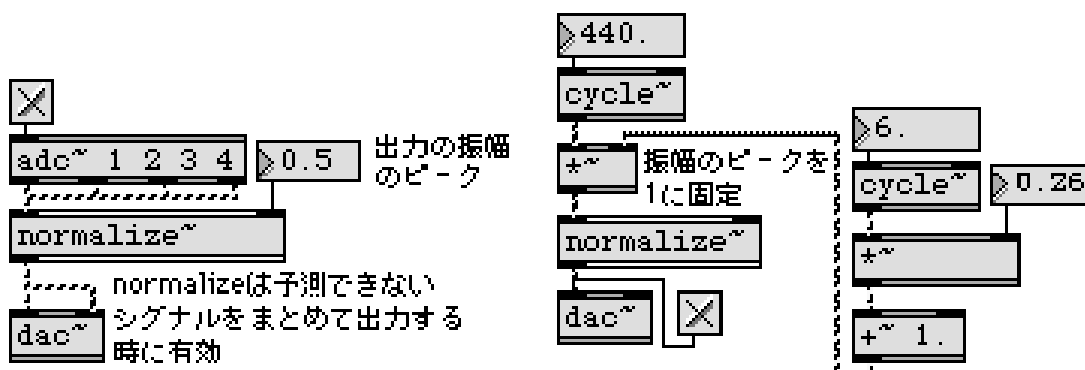
## アーギュメント

- 実数** オプション。最大出力振幅の初期値。デフォルト値は1である。

## 出力

- シグナル** 入力シグナルは最大入力振幅で割られた最大出力振幅によって伸縮される。

## 利用例



正確な倍率に変化する時や分からない時は、**normalize~** がピーク振幅を設定する

## 参照

- \*~ シグナルの乗算



**number~**には異なる2つの表示モードがある。シグナル・モニター・モードでは第1インレットで受け取ったシグナルの値を表示する。シグナル出力モードでは第1インレットで最も最近受け取った、または直接**number~**ボックスに入力した実数または整数の値を表示する（シグナルは第1アウトレットから出力される）。

## 入力

**シグナル** 任意のシグナルで、その値は一定の間隔でサンプリングされて、第2アウトレットから出力される。**number~**がシグナル・モニター表示モードの時、シグナルの値が表示される。

**実数** 第1インレット：このインレットに与えた値は、第1アウトレットから一定値のシグナルとして出力される。**number~**がSignal Output表示モードの時は、その値が表示される。現在の変化時間が0でなければ、出力シグナルは前の値と新しく設定された値の間でリニアに変化する。

第2インレット：変化時間をミリ秒単位で設定する。デフォルトの時間は0である。

**整数** 実数に変換される。

**リスト** 最初の数値は第1アウトレットから出力されるシグナルの値を設定し、二番目の数値が変化時間をミリ秒単位で設定する。

**(マウス)** **number~**の左側の三角形の部分をクリックすると、**number~**はシグナル・モニター表示モード（緑色の波形）とシグナル出力モード（黄色または緑色の下向きの矢印）の間でトグルさせる。シグナル出力表示モードでは、数値を表示する部分でクリックすることで**number~**の第1アウトレットから出力されるシグナルの値を変える。また、マウス・クリックによって、**number~**を選択し、キーボードから数値を入力することもできる。

**(キーボード入力)** **number~**がハイライト表示されている時（黄色の下向きの矢印が点灯している時）、数値のキーボード入力はその値を変更する。マウスをクリックしたり、returnキーまたはenterキーを押したりすると、保留中のタイプされた数値を格納し、第1アウトレットから新しい値として出力する。

**allow** 数値を伴う**allow**は、使用することのできる表示モードを設定する。**allow 1**は**number~**をシグナル出力表示モードに制限する。**allow 2**は**number~**を入力モニター表示モードに制限する。**allow 3**は両方のモードを許可し、ユーザーが**number~**の左側の三角形の部分をクリックすることで、2つのモードの間で切り替えることができるようにする。



- brgb** 0から255までの3つの数値を伴う**brgb**は、**number~**ボックスの背景色のためのRGB値を設定する。デフォルト値は白 (**brgb 255 255 255**) である。
- frgb** 0から255までの3つの数値を伴う**frgb**は、**number~**ボックスの数値表示のためのRGB値を設定する。デフォルト値は黒 (**brgb 0 0 0**) である。
- rgb2** 0から255までの3つの数値を伴う**rgb2**は、**number~**ボックスのハイライト中または更新中の数値表示のためのRGB値を設定する。デフォルト値は黒 (**brgb 0 0 0**) である。
- rgb3** 0から255までの3つの数値を伴う**rgb3**は、**number~**ボックスのハイライト中または更新中の背景色のためのRGB値を設定する。デフォルト値は白 (**brgb 255 255 255**) である。
- mode** 数値を伴う**mode**は、現在そのモードが許可されていれば (**allow**メッセージを参照のこと) 現在の表示モードを設定する。**mode 1**はシグナル出力表示モードに設定する。**mode 2**はシグナル入力表示モードに設定する。
- min** オプションの数値を伴う**min**は、**number~**のシグナル出力のための最小値を設定する。フロート・ナンバー・ボックスと異なり、**number~**の最小値は整数値に制限されないことに注意が必要である。**min**が数値を伴わない場合には、任意の最小値は削除される。
- max** オプションの数値を伴う**max**は、**number~**のシグナル出力のための最大値を設定する。フロート・ナンバー・ボックスと異なり、**number~**の最大値は整数値に制限されないことに注意が必要である。**min**が数値を伴わない場合には、任意の最大値は削除される。
- interval** 数値を伴う**interval**は、サンプリング間隔をミリ秒単位で設定する。これにより**number~**が入力モニター表示モードの時には、表示が更新される時間間隔だけでなく、オブジェクトの第2アウトレットから数値が出力される時間間隔も設定される
- flags** 数値を伴う**flags**は**number~**の外観の特性と振る舞いを設定する。特性 (アーギュメントの項で後述する) は、次のように目的のオプションを指定する値を互いに加算することで設定される。4=**ボールド・タイプ**、64=**マウス・アップの時のみ出力する**、128=**マウスでは変更できない**。例えば、**flags196**は、これらのオプションのすべてを設定する。





## インスペクター

**number~**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**number~**オブジェクトを選択することで、**number~**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**number~**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**number~**インスペクターでは次の属性を設定することができる。

格納し、表示し、タイプし、通過させる値を*Range Min.*と*Max.*ボックスに数値を入力することで設定できる。*No Mix.*と*No Max.*のチェックボックスがチェックされていると（デフォルトの状態）、**number~**オブジェクトは最小値と最大値を*None*（なし）に設定する。これらのボックスのチェックを外すと、最小値を最大値は0に設定される。

インスペクターの**Options**セクションでは、**number~**オブジェクトの表示属性を設定することができる。インスペクターにあるオプションは以下の通り。*Bold*（太字で表示する）、*Draw Triangle*（数値を指定する三角形を表示し、それに独特の外観を与える）、*Output Only on Mouse-up*（常にではなくマウスボタンが離された時だけ数値を出力する）、*Can't Change*（マウスまたはコンピュータのキーボードによる変更を許可しない）、*Transparent*（ナンバー・ボックスが**comment**オブジェクトのようになるように、**number~**オブジェクトの中に数値だけを表示してボックスは表示しない）。

*Display Style*ポップアップ・メニューでは、どのように数値を表すのかを設定する。*Decimal*は数値を表示するためのデフォルトの方法である。*Hex*は数値を16進数で表示し、MIDI関連の用途に有効である。*Roland Octal*はいくつかのハードウェア・デバイスで用いられているフォーマットのように、それぞれの桁を1から8までの範囲で表示する。例えば、11は0で88は63である。*Binary*は数値を1と0として表示する。*MIDI Note Names*は60をC3として表示するMIDIピッチの値に対応した数値を表示する。ノート・ネームC4は60をC4として表示するMIDIノート名と同じである。すべての表示モードで、数値はそれが表示されるフォーマットでタイプされなければならない。

*Mode*では、チェックボックスでシグナル・モニター・モードかシグナル出力モードかを選択することができる。デフォルトでは両方のモードがチェックされているが、少なくとも1つのモードはチェックされていなければならない。



*Interval*はサンプリング間隔をミリ秒単位で設定する。これは**number~**が入力モニターモードの時に表示が更新される時間間隔だけでなく、オブジェクトの第2アウトレットから出力される時間間隔も設定する。デフォルト値は250ミリ秒である。

*Color*オプションでは、swatchカラー・ピッカーまたはRGB値で、**number~**ボックスと、その背景の通常時とハイライト時に使用される色を設定できる。*Number*は数値表示のための色（デフォルト値は000）を設定し、*Background*は**number~**ボックス・オブジェクト自身のための色（デフォルト値では221 221 221）を設定し、*Highlighted Number*はナンバー・ボックスが選択されているか、値が更新中の時の数値表示のための色（デフォルト値は000）を設定し、*Highlighted Background*はハイライト中または更新中の**number~**ボックスの色（デフォルト値は221 221 221）を設定する。

**number~**ボックスのフォントの種類とサイズは、Fontメニューで変更することができる。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

なし。

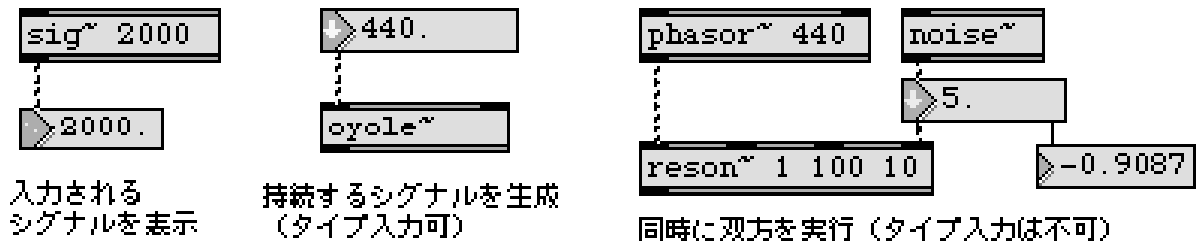
## 出力

**シグナル** 第2アウトレット：オーディオ処理がオンの時、**number~**オブジェクトは第1インレットで最も最近受け取った（またはユーザーによって入力された）数値に等しい一定のシグナルを出力する。そのシグナル入力に関係なくこの値を出力し、それはシグナル出力表示モードであるか否かには関係ない。第2インレットで最も最近受け取った変化時間が0でなければ、出力は前の値と新しい値の間で指定された時間をかけて補間する。

**実数** 第2アウトレット：**interval**メッセージで指定された時間間隔で出力される入力シグナルのサンプル。



## 利用例



number~ オブジェクトのいくつかの使用方法

## 参照

**line~** シグナル・ランプ・ジェネレーター  
**sig~** 一定の値のシグナル  
**snapshot~** シグナルの値を数値に変換する

Tutorial 23 Analysis: Viewing signal data (英文PDF参照)

**onepole~**オブジェクトは、以下のような単純なフィルター式を実装している。

$$output = previous\ input + cf * (input - previous\ input)$$

この式の $cf$ はラジアン単位で表されるフィルターのカットオフ周波数を表している。 $cf$ の値は-1.0~0の範囲にある。これは6dB毎のオクターブ減衰を持つ単極のローパス・フィルターを作り出し、これは耳障りな広域成分（例えばダウンサンプリングによるデジタル的な障害）をなだらかに丸めるのに役立つ。**onepole~**オブジェクトは係数を持つ**biquad~**オブジェクトに等しい。

$$[a0 = 1 + cf, a1 = 0, a2 = 0, b1 = cf, b2 = 0]$$

これらの値を**biquad~**オブジェクトの式に代入すれば、**onepole~**オブジェクトのアルゴリズムが得られる。しかし**biquad~**オブジェクトはその式の中で使用されない部分も計算するので、**onepole~**オブジェクトはそれと比べてかなり早く実行することができる。

## 入力

- シグナル** 第1インレット：フィルター処理が行われるシグナル。
- 第2インレット：シグナルはフィルター処理を行う周波数の値を設定する。実数も同様である。このインレットにシグナルが接続された場合、その値はシグナル・ベクター毎にサンプルされる。
- 実数** 第2インレット：（シグナルが接続されていない場合）フィルター処理を行う周波数を設定する。デフォルトでは、周波数はヘルツ単位で表される。設定する値の許容範囲は0から現在のサンプリングレートの4分の1までである。便宜のために、**onepole~**オブジェクトは追加の2つの入力モードを備えている。それはより慣習的な入力範囲、0~1を使用するものである（詳細は**linear**及び**radians**のメッセージを参照のこと）。
- clear** 各インレット：**onepole~**の内部状態をクリアする。**onepole~**オブジェクトは他のフィルターのタイプに見られる固有の不安定性を持たないので、これは必ずしも必要ではない。
- Hz** 各インレット：周波数の入力モードをヘルツ単位に設定する（デフォルト）。
- linear** 各インレット：周波数の入力モードをリニアに設定する（0~1）。リニア・モードは標準のHzモードの比率を単純に変更したものなので、0~1の範囲以外の値は全周波数範囲を超えることになる。

**radians** 各インレット：周波数の入力モードをラディアン単位に設定する (0~1)。ラディアン・モードは式の中心の周波数を直接設定できる。入力シグナルが同じ範囲 (0~1) を持つ場合は、出力は人間の耳が持つ指数的なピッチ・スケールに近い曲線的な周波数レスポンスを持つ。

## アーギュメント

**実数** オプション。先に述べたようにフィルター処理の中心の周波数を設定する。

**Hz** オプション。周波数の入力モードをヘルツ単位に設定する (デフォルトのモードである。従って、アーギュメントを与えない場合も、同じである)。

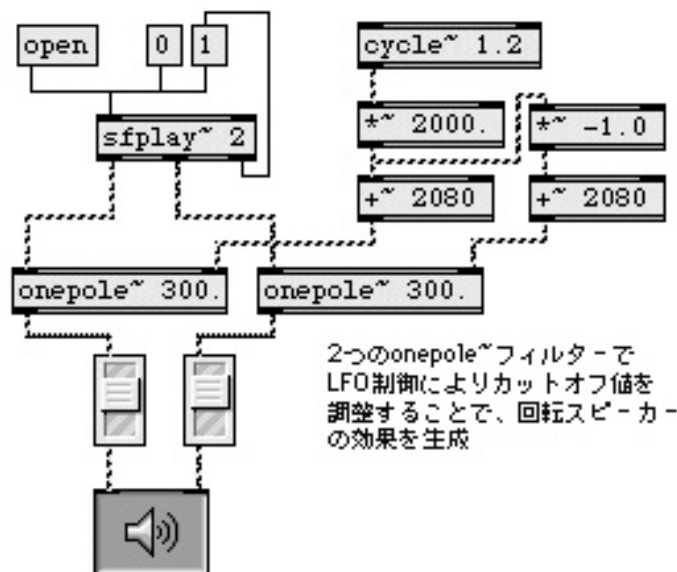
**liner** オプション。周波数の入力モードをリニアに設定する (0~1)。

**radians** オプション。周波数の入力モードをラディアン単位に設定する (0~1)。

## 出力

**シグナル** フィルター処理が行われたシグナル。

## 利用例



**onepole~** オブジェクトは、単純なサンプル・プレイヤーに効果的なフィルター処理を行う

## 参照

**biquad~** 2ポール、2ゼロのフィルター  
**reson~** レゾナンス付きバンドパス・フィルター

## 入力

- シグナルまたは実数** 第1インレット：現在のインデックス番号で参照されているオシレーターに対して、周波数を設定する。デフォルト値は0である。
- 第2インレット：現在のインデックス番号で参照されているオシレーターに対して、振幅を設定する。
- 第3インレット：**framesync 1**メッセージによってフレーム同期機能が有効である場合、現在インデックスで参照されているオシレーターに対して、0から1.0の値で位相を設定する。
- 第4インレット：参照するオシレーターのインデックス番号を設定する。
- 実数** 第3インレット：現在参照されているオシレーターに対して、0から1.0の範囲の実数で位相を設定する。
- clear** すべてのオシレーターに対して周波数を0Hzに、振幅を0に設定する。
- copybuf** 第1インレット：**copybuf**に続いて**buffer~**の参照名のシンボルを指定することで、その**buffer~**から4096サンプルを内部ウェーブテーブルとしてコピーする。オプションとして参照名に続く整数のアーギュメントを指定することで、読み込み位置をサンプル単位で（オフセットとして）指定できる。
- framesync** **framesync**に続けて0でない数値を指定することで、フレーム同期処理を有効にする。フレーム同期処理が有効である場合、インデックスで参照されているオシレーターの出力は、入力シグナルの値が0の場合（またはN個のサンプルずつのフレームごとに）に変化する。デフォルトではこの機能はオフである。
- freqsmooth** **freqsmooth**に続いて1つの整数を指定し、周波数変化のスムージング処理を何サンプルで行うか決める。デフォルト値は1（スムージングなし）である。
- magsmooth** **magsmooth**に続いて1つの整数を指定し、振幅変化のスムージング処理を何サンプルで行うか決める。デフォルト値は0（振幅のスムージングなし）である。



- set** **set**に続いて2つの実数を指定することで、オシレーター・バンク中の1つのオシレーターに対して周波数と振幅を設定する。 $n$ 対の実数をリストとして入力することで、**oscbank~**中の最初の $n$ 個のオシレーターに対して設定を行い、残りのオシレーターの振幅を0にすることができる。
- silence** すべてのオシレーターの振幅を0にする。
- size** **size**に続いて1つの数字を指定し、オシレーターの数を設定する。デフォルト値は64である。
- tabpoints** **tabpoints**に続いて1つの整数を指定し、**oscbank~**内部に持つウェーブテーブルのサンプル数を設定する。デフォルト値は4096である。ウェーブテーブルのサンプル数は2の2乗から16乗の間の、2の累乗の数でなければならない。その他の値の場合は、最も近い2の累乗数に丸められる。

## アーギュメント

- 整数** オプション。第1アーギュメントはオシレーターの数を指定する。デフォルトでは1個である。
- 整数** オプション。第2アーギュメントは周波数変化のスムージング処理を何サンプルに渡って行うかサンプル数を指定する。
- 整数** オプション。第3アーギュメントは振幅変化のスムージング処理を何サンプルに渡って行うかサンプル数を指定する。
- 整数** オプション。第4アーギュメントで内部サイン波の参照テーブルのサンプル・サイズを設定する。デフォルト値は4096である。**oscbank~**のオシレーターは補間処理を行わないため、CPUパワーを多量に消費するような大きなサイズのサイン波テーブルを使うこともできる。

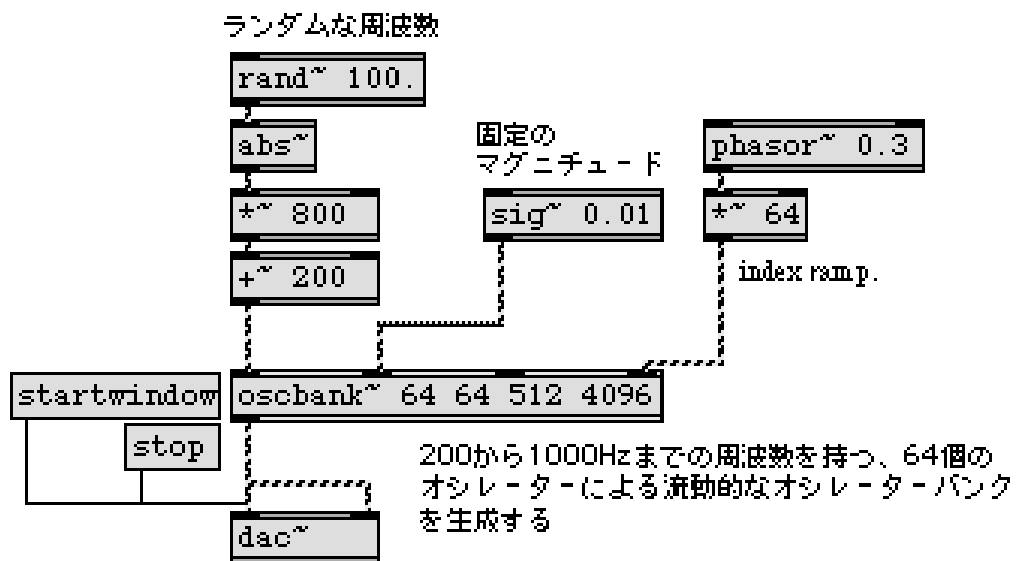
注記：**oscbank~**内には1つだけウェーブテーブルがあり、すべてのオシレーターはそれを参照する。

## 出力

- シグナル** 周波数および振幅を設定した各オシレーター出力の総和となる波形が出力される。



## 利用例



oscbank~ によって、1つのオブジェクトでオシレーター・バンクの制御ができる

## 参照

### ioscbank~

補間機能付きのオシレーター・バンク

## 入力

メッセージ **poly~** オブジェクトに読み込まれたパッチ内のそれぞれの **out** オブジェクトは、**poly~** オブジェクトの下部のアウトレットとして表れる。読み込まれたパッチ内の **out** オブジェクトに受け取られたメッセージは、**poly~** オブジェクトの対応するアウトレットから出力される。**poly~** オブジェクトの同じアウトレットを通じて、パッチのすべてのインスタンスからのメッセージが出力される。

## 出力

なし。

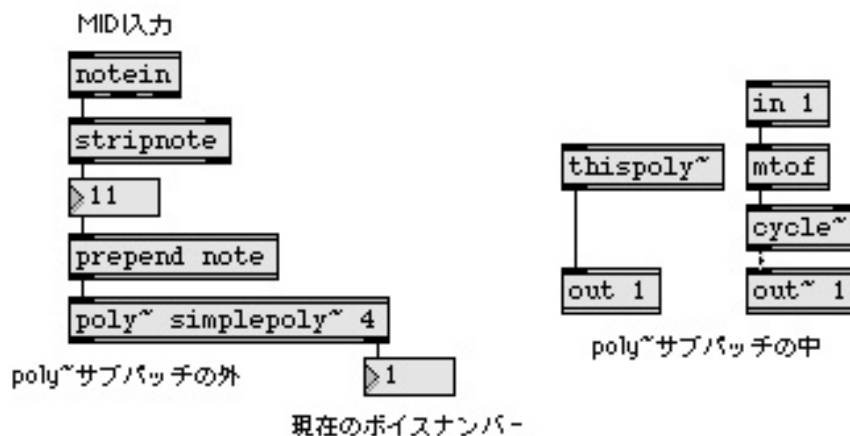
## アーギュメント

整数 必須。それぞれの **out** オブジェクトは、重複しないインデックス番号によって区別され、**poly~** オブジェクトのどのアウトレットへメッセージを出力するかを決定する。1つ目のアウトレットの番号は1である。

## 出力

(パッチ) 読み込まれたパッチ内の **out** オブジェクトに受け取られた任意のメッセージは、その **out** オブジェクトのアーギュメントに対応する **poly~** オブジェクトのアウトレットから出力される。**poly~** オブジェクトの同じアウトレットを通じて、パッチのすべてのインスタンスからのメッセージが出力される。

## 利用例



**poly~** オブジェクトのメッセージのアウトレットは、読み込まれたパッチ内の **out** オブジェクトに対応する

# out

**poly~** オブジェクトによって読み込まれる  
パッチのためのメッセージの出力

---

## 参照

|                  |                                              |
|------------------|----------------------------------------------|
| <b>in</b>        | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのメッセージの入力 |
| <b>in~</b>       | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのシグナルの入力  |
| <b>out~</b>      | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのシグナルの出力  |
| <b>poly~</b>     | パッチのポリフォニック処理とオーディオ処理を管理する                   |
| <b>thispoly~</b> | <b>poly~</b> オブジェクトのヴォイス割当とミュートのコントロール       |

Tutorial 21

MIDI control: Using the **poly~** object. (英文PDF参照)

## 入力

**シグナル** poly~ オブジェクトに読み込まれたパッチ内のそれぞれのout~ オブジェクトは、poly~ オブジェクトの下部のアウトレットとして表れる。読み込まれたパッチ内のout~ オブジェクトに受け取られたシグナルは、poly~ オブジェクトの対応するアウトレットから出力される。poly~ オブジェクトからのシグナルの出力は、パッチのすべてのインスタンスの出力をミックスしたシグナルになる。

## 出力

なし。

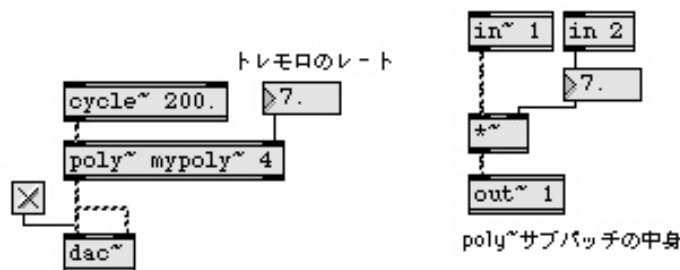
## アーギュメント

**整数** 必須。それぞれのout~ オブジェクトは、重複しないインデックス番号によって区別され、poly~ オブジェクトのどのアウトレットへシグナルを出力するかを決定する。1つ目のアウトレットの番号は1である。

## 出力

(パッチ) 読み込まれたパッチ内のout~ オブジェクトに受け取られた任意のシグナルは、そのout~ オブジェクトのアーギュメントに対応するpoly~ オブジェクトのシグナルのアウトレットから出力される。poly~ オブジェクトからのシグナルの出力は、パッチのすべてのインスタンスの出力をミックスしたシグナルになる。

## 利用例



poly~ オブジェクトのシグナルのアウトレットは、読み込まれたパッチ内のout~ オブジェクトに対応する

## 参照

**in** poly~ オブジェクトによって読み込まれるパッチのためのメッセージの入力  
**in~** poly~ オブジェクトによって読み込まれるパッチのためのシグナルの入力  
**out** poly~ オブジェクトによって読み込まれるパッチのためのメッセージの出力  
**poly~** パッチのポリフォニック処理とオーディオ処理を管理する  
**thispoly~** poly~ オブジェクトのヴォイス割当とミュートのコントロール

**overdrive~**オブジェクトはウェーブ・シェーピング・ファンクションを使用して、オーディオ・シグナルに歪ませる。シグナルを増幅し、シグナルの最大値を±1に制限する。この範囲外の値は、真空管回路によってオーバードライブされるような、ソフト・クリッピングを使ってカットされる。

## 入力

**シグナル** 第1インレット：歪ませるシグナル。

**実数** 第2インレット：**overdrive~**オブジェクトは浮動小数点のドライブ係数を受け取る。ドライブ係数は通常1.0-10.0の範囲内でなくてはならない。係数1.0を使用すると、歪みを引き起こさずにリニアな対応関係となる。1より大きな値は歪みを増やす。負の値を含む1より小さな値は、非常に激しく歪んだシグナルを生成する。この機能は注意して使用したほうがよい。このふるまいは、オブジェクトの開発者が**overdrive~**オブジェクトが持つ特徴で、そのままにするべきだと主張するまでは、元々バグだと考えられていた。

**整数** 実数に変換される。

## アーギュメント

**実数** オプション。ひとつの数値がドライブ係数に与えられる。アーギュメントが与えられない場合は、ドライブ係数は1.0に設定される。

## 出力

**シグナル** 歪ませられたシグナル。

## 利用例

```
cycle~ 440
|
overdrive~ 100.
OUCH!
```

オーバードライブによる増幅のようなシグナルのウェーブ・シェーブ

## 参照

**kink~** ノコギリ波を歪ませる  
**lookup~** を利用したテーブル参照機

## 入力

**シグナル** シグナルを扱うすべてのオブジェクトのアウトレットに**pass~**オブジェクトを使用する。サブ・パッチ内のオーディオが使用可能の時、**pass~**オブジェクトはインプットからアウトプットにシグナルを通過させる。しかし**mute~**オブジェクトを使用しているか、または、**enable 0**メッセージを**pcontrol**に送ることで、サブ・パッチ内のオーディオが使用停止になる時、**pass~**オブジェクトは値が0のシグナルをアウトレットから出力する。

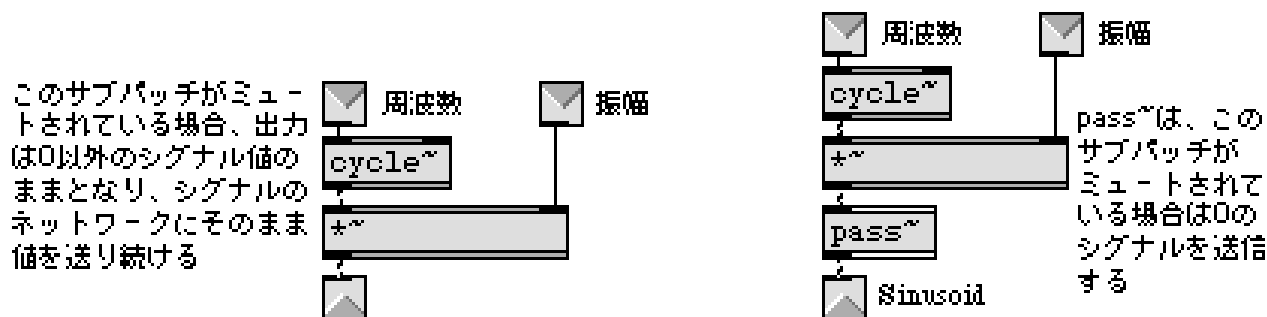
## アーギュメント

なし。

## 出力

**シグナル** **pass~**オブジェクトを含むサブ・パッチ内のオーディオが使用可能の時、出力は入力と同じである。**mute~**オブジェクトを使用しているか、または、**enable 0**メッセージを**pcontrol**に送ることで、サブ・パッチ内のオーディオが使用停止になる時、**pass~**オブジェクトは値が0のシグナルをアウトレットから出力する。

## 利用例



**pass~** オブジェクトは、ミュートされたシグナルを確実に無音にする

## 参照

### **mute~**

サブパッチのオーディオ処理を停止する

### Tutorial 5

Fundamentals: Turning signals on & off (英文PDF参照)

## 入力

- シグナル** 第1インレット：ピークの振幅値を求めるシグナル。
- bang** 第1インレット：前回の出力時から計測して、最も大きなシグナルの振幅値（絶対値）を出力する。
- 整数** 第2インレット：入力シグナルの振幅のピーク値を自動的に出力する場合の内部時計の時間間隔を設定する。時間間隔が0であれば、内部時計は停止する。正の整数であれば、時間間隔はデータを出力する割合を設定する。シグナル・ベクターの時間よりも短い時間間隔が指定された場合は、シグナル・ベクターごとに一度だけ振幅のピーク値が求められる。
- 実数** 第2インレット：整数と同じ。

## アーギュメント

- 整数** オプション。ミリ秒単位で**peakamp~**オブジェクトの内部時計の時間間隔を設定する。時間間隔が0の場合、内部時計は使用されない。よって**peakamp~**は**bang**メッセージを受け取った時だけデータを出力する。時間間隔が0でない場合は、**peakamp~**は入力されるシグナルの振幅のピーク値を時間間隔で設定した時間ごとに出力する。時間間隔のデフォルト値は0である。

## 出力

- 実数** **peakamp~**オブジェクトが**bang**メッセージを受け取った時、または内部時計がオンになっている時、入力シグナルのピークの絶対値をアウトレットから出力する。

## 利用例



## 参照

- meter~**  
**snapshot~** 視覚的なピーク・レベル・インジケータ  
シグナルの値を数値に変換する



**peek~**オブジェクトはオーディオ処理が開始されていなくても機能する。そのため、**peek~**を利用することでシグナル処理ではないアプリケーションで、Maxの**table**オブジェクトと同様に**buffer~**を浮動小数点用のデータ・テーブルとして利用することができる。

## 入力

**整数** 第1インレット：関連付けられた**buffer~**オブジェクトのサンプル・メモリの位置を示すインデックス番号。この値によって**buffer~**のインデックス位置に格納された数値が**peek~**オブジェクトのアウトレットから出力される。しかし、第2インレットにも同時に数値が入力された場合は、**peek~**から数値が出力されるのではなく、第2インレットへ入力した数値が**buffer~**内のインデックス番号の示す箇所に格納される。**buffer~**オブジェクト内で確保された領域に存在しないインデックスを入力した場合には、何も起こらない。

第2インレット：実数に変換される。

第3インレット：関連付けられた**buffer~**が複数チャンネルを持つ場合に、読み込みおよび書き出しを行うチャンネルを1から4で指定する。

**実数** 第1インレット：整数に変換される。

第2インレット：関連付けられた**buffer~**に格納するサンプル値を入力する。第1インレットにサンプル・インデックス番号を受け取った時に、その位置にサンプル値が書き込まれる。

第3インレット：整数に変換される。

**clip** 第1インレット：**clip**に続いて0でない値を指定した場合、-1.0~1.0の範囲でのデータ・クリッピングを有効にする。デフォルトでクリッピングは有効となっている。**clip 0**というメッセージを入力することで無効となる。

**リスト** 第1インレット：関連付けられた**buffer~**に対し、1つ目の値をインデックス番号として、2つ目の数値を格納する。リストデータが3つ目の数値を持つ場合は、マルチ・チャンネルの**buffer~**にデータを格納するべきチャンネルを指定する。それがない場合、最も最近に指定したチャンネルが利用される。

**set** 第1インレット：**set**に続き、関連付ける**buffer~**の参照名を指定する。

(マウス) **peek~**オブジェクトをダブル・クリックすることで、関連付けられた**buffer~**オブジェクトの波形表示ウィンドウが開かれる。

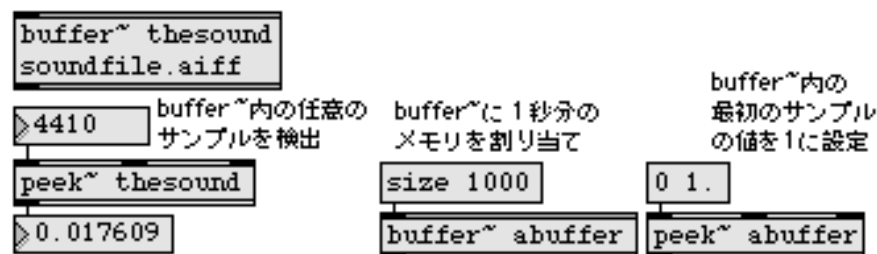
## アーギュメント

- シンボル** 必須。データの読み書きを行う**buffer~**オブジェクトの参照名を指定する。
- 整数** オプション。**buffer~**が複数チャンネルを持っている場合、**buffer~**名に続いて、読み書きするチャンネルを指定できる。デフォルトのチャンネルは1である。
- 整数** オプション。上の**buffer~**名、チャンネル番号に続いて第3アーギュメントを指定した場合、クリッピングを設定する。1を指定した場合、-1.0~1.0のクリッピング設定となる。この設定は**clip**メッセージで変更可能である。

## 出力

- 実数** 第1インレットで入力したインデックス値に対応する**buffer~**内部のサンプル値が**peek~**のアウトレットから出力される。

## 利用例



**buffer~**にあるサンプル値を読み出す例とサンプル値を書き込む例

## 参照

- buffer~** オーディオ・サンプルの格納
- buffir~** **buffer~**を利用したFIRフィルター
- poke~** インデックス番号によるサンプル値の書き込み
- table** 数値の配列をグラフィカルに編集し、格納する

## 入力

- シグナル** **pfft~**内部のサブパッチに含まれる**fftin~**オブジェクトの数によって、シグナル用の入力の数決定される。
- mute** **mute**と、それに続く1か0の数値によって、**pfft~**に対するミュートまたはミュート解除を指定する。これは**pfft~**内で実行されるサブパッチのシグナル処理の停止・実行を指定することになる。
- open** **open 1**によって、**pfft~**に読み込まれているサブ・パッチを表示する。**open 0**の場合は、**pfft~**があるパッチへ戻る。
- wclose** **pfft~**に読み込まれているサブ・パッチが開かれていた場合に、それを閉じる。

## アーギュメント

- シンボル** 必須。第1アーギュメントは**pfft~**に読み込まれ、シグナル処理を行うサブ・パッチの名前を指定する。
- 整数** オプション。高速フーリエ変換におけるフレーム・サイズをサンプル数で指定する。デフォルト値は512である。第2アーギュメントは、**pfft~**オブジェクトが実行するFFTのフレーム・サイズをサンプル数で指定する。これは2の累乗数でなければならず、デフォルト値は512である。
- 整数** オプション。第3アーギュメントは、FFTの窓におけるオーバーラップ数を定義する。FFT処理で実行されるホップ・サイズは、FFTのフレームサイズをオーバーラップ数で割った数に等しい（例：フレームサイズが512、オーバーラップを2とした場合、ホップサイズは256サンプルとなる）。数値は2の累乗数でなければならず、デフォルト値は2である。
- 整数** オプション。第4アーギュメントは、シグナルに対してFFTが実行される開始位置をサンプル数として指定する。現在のシグナル・ベクター・サイズの倍数として指定しなければならない。デフォルト値は0である。

## 出力

- シグナル** FFTベースでシグナル処理を行ったサブパッチの結果が出力される。出力の数は、読み込まれるサブパッチに含まれる**fftout~**オブジェクトの数による。**pfft~**に読み込まれるパッチのアウトレットは、そのパッチ内での**fftout~**オブジェクトによってのみ設定できる。詳しくは**fftout~**の項を参照のこと。

利用例



pfft~は、周波数領域での処理のために、特別に作成されたサブパッチを読み込む

参照

- cartopol** 直交座標を極座標に変換する
- cartopol~** シグナルについて直交座標を極座標に変換する
- fft~** 高速フーリエ変換
- fftin~** pfft~に読み込まれるパッチにおける入力
- fftinfo~** pfft~に読み込まれるパッチに関する情報の取得
- fftout~** pfft~に読み込まれるパッチにおける出力
- frameaccum~** 連続した位相変化データのフレームから位相を計算
- framedelta~** 連続したFFT処理フレーム間の位相変化を計算
- ifft~** 逆高速フーリエ変換
- poltoCAR** 極座標を直交座標に変換する
- poltoCAR~** シグナルについて極座標を直交座標に変換する
- vectral~** ベクター・ベースのエンベロープ・フォロワー

- Tutorial 25 Analysis: Using the FFT (英文PDF参照)
- Tutorial 26 Frequency Domain Signal Processing with pfft~ (英文PDF参照)

## 入力

**シグナル** 位相をシフトするシグナル。

**実数** 第2インレット：180度シフトされるシグナルの周波数を設定する。この周波数以下のシグナルはより小さくシフトされ、この周波数以上のシグナルはより大きくシフトされる。最高360度までシフトされる。

第3インレット：係数Qの値を設定する。または、オブジェクトが位相をシフトする0度から360度までの段階数を設定する。一般的に、Qの有用な範囲は0.から10.である。

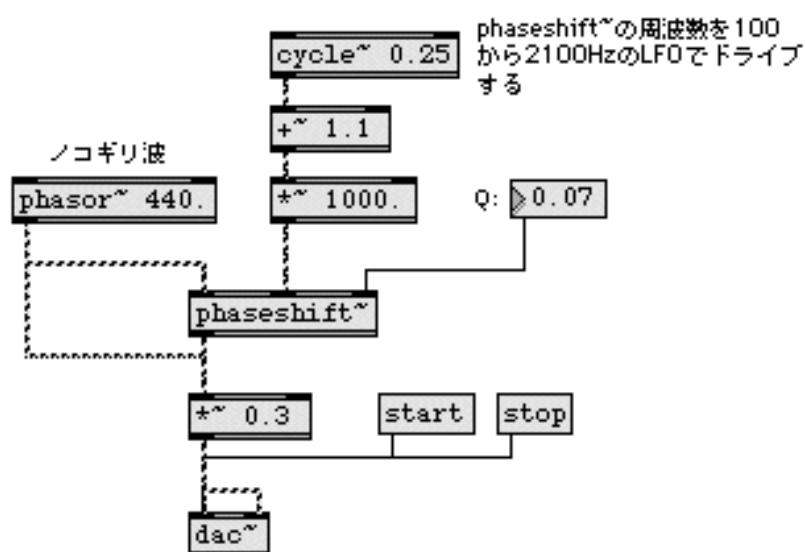
## アーギュメント

**実数** オプション。アーギュメントが1つ与えられた場合は、**phaseshift~**オブジェクトの周波数のパラメーターを設定する。アーギュメントが2つ与えられた場合は、1つ目が周波数のパラメーターを設定し、2つ目が係数Qを設定する。

## 出力

**シグナル** 入力シグナルの周波数の構成要素と倍音成分は0から360度まで位相をシフトされる。そのシフトは、入力シグナルの周波数とオブジェクトで設定される周波数とQの値に依存する。

## 利用例



**phaseshift~** オブジェクトとLFOを使用して、アナログ的なフェーズ・シフターをシュミレートする

## 参照

**allpass~** オールパス・フィルター  
**comb~** コム・フィルター

## 入力

**シグナル** シグナルの範囲が変更される。入力シグナルの値が $\pi$ (3.14159)を超える時、出力シグナルは下の限界値が $-\pi$ (-3.14159)となる範囲に変更される。従って、上昇するシグナルは $-\pi$ と $\pi$ を下限値と上限値とするノコギリ波となる。

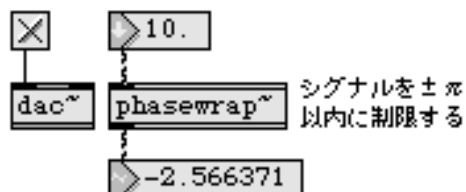
## アーギュメント

なし。

## 出力

**シグナル** 変更された入力シグナルを出力する。

## 利用例



シグナルが $-\pi$ から $+\pi$ までのラディアン値にするために **phaseswap~** を用いる

## 参照

**cartopol~**  
**pfft~**  
**pong~**

シグナルについて直交座標を極座標に変換する  
周波数領域処理を行うパッチの管理  
可変範囲でのシグナルの折り返し処理

## 入力

**シグナル** 第1インレット：ノコギリ波形の周波数を指定する。

**実数または整数** 第1インレット：ノコギリ波形の周波数を指定する。このインレットにシグナルが入力される場合、実数または整数を入力しても無視される。

第2インレット：0から1の範囲の値で波形の位相を指定する。出力シグナルはこの位相での値から続けて出力される。

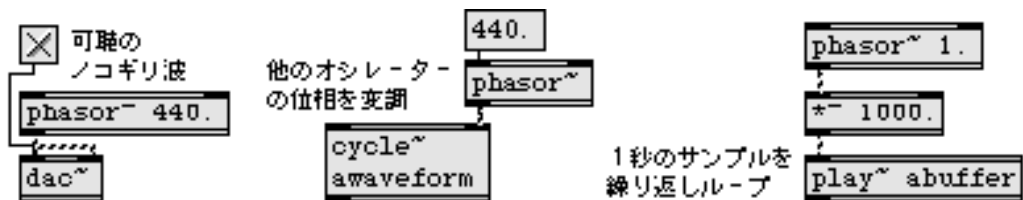
## アーギュメント

**実数または整数** オプション。ノコギリ波形としての周波数の初期値を与える。このインレットにシグナルが入力される場合、このアーギュメントによる設定値は無視される。

## 出力

**シグナル** 指定した周波数で0から1に周期的に増加するノコギリ波形を出力する。

## 利用例



オーディオ処理および周波数処理に傾斜波形の繰り返しは便利である

## 参照

- 2d.wave~** 2次元のウェーブテーブル
- cycle~** テーブル参照可能なオシレーター
- line~** シグナル・ランプ・ジェネレーター
- trapezoid~** 台形波のウェーブテーブル
- triangle~** 三角波のウェーブテーブル
- wave~** サイズ可変のウェーブテーブル再生

Tutorial 3 Analysis: Wavetable oscillator (英文PDF参照)



## 入力

なし。

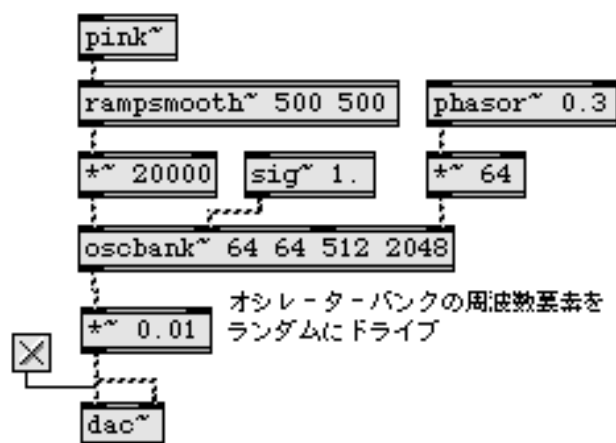
## アーギュメント

なし。

## 出力

**シグナル** **pink~** オブジェクトは1オクターブあたりのエネルギーが均等であるような、-1.0から1.0の範囲にあるランダムな値によるシグナルを生成する。このようなエネルギーの分布を持つノイズはピンク・ノイズとして知られている。**noise~** オブジェクトによって生成されるホワイト・ノイズはすべての周波数帯において均等なエネルギーを持っている。知覚的には、ホワイト・ノイズは明るく荒く聴こえる。一方、ピンクノイズはより均等で自然な音を持つ。

## 利用例



**pink~** オブジェクトは1オクターブあたりのエネルギーが均等となる周波数のランダムな値を生成する

## 参照

**noise~**                      ホワイト・ノイズ・オシレーター

## 入力

**シグナル** 第1インレット：**buffer~**オブジェクトに保持しているサンプル・メモリの再生位置をミリ秒で表した値を持つシグナルを入力する。シグナルが時間に沿って増加する場合、通常方向へサンプルを再生する。シグナルの値が減少する場合は、時間の逆方向へ再生する。シグナルが同じ値で続く場合は、同じサンプルを繰り返し出力する。これはサンプル値のDCオフセットと同じである。

**set** **set**に続いて再生データを持つ**buffer~**オブジェクトの参照名を指定する。

## アーギュメント

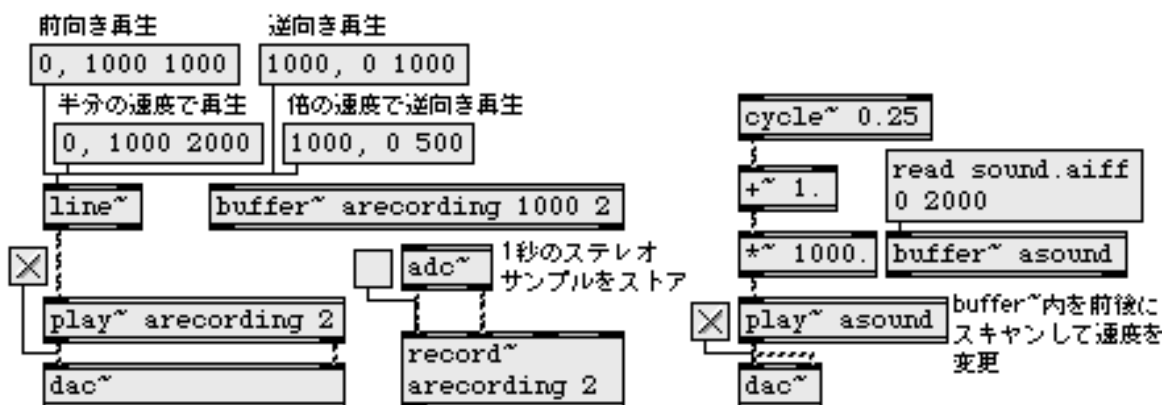
**シンボル** 必須。**play~**がサンプルの再生に用いる**buffer~**の参照名を指定する。

**整数** オプション。参照名のアーギュメントに続いて、出力チャンネル数を1、2、4で指定することができる。デフォルトのチャンネル数は1である。**buffer~**側のチャンネルが**play~**での設定チャンネル数より少ない場合、**play~**側で余分に存在するチャンネルは無音での出力となる。**buffer~**のほうチャンネル数が多い場合は、チャンネルがミックスされて出力される。

## 出力

**シグナル** **buffer~**から読み出したサンプルを出力する。**play~**が2つか4つの出力チャンネルを持つ場合は、第1アウトレットはサンプル・データの左チャンネルを出力し、残りのアウトレットには、その他のチャンネルが割り当てられる。

## 利用例



**play~** は通常は **line~**からの傾斜シグナルによって動作するが、他のシグナルを利用することで変わったエフェクト出力が可能である

## 参照

**2d.wave~**

2次元のウェーブテーブル

**buffer~**

オーディオ・サンプルの格納

**buffir~**

**buffer~**を利用したFIRフィルター

**groove~**

速度可変のループ再生

**record~**

サウンドを**buffer~**のサンプル・メモリへ録音する

Tutorial 13

Sampling: Recording and playback (英文PDF参照)

**plugconfig**オブジェクトは、プラグインの振る舞いをスクリプトを用いて設定する。このスクリプトは、**env**や**menubar**オブジェクトのスクリプトと同じである。**plugconfig**オブジェクトをダブル・クリックすることで、スクリプトを編集することができる。プラグインのパッチについて、**plugconfig**オブジェクトは1つに限定するべきである。なぜなら、1つ以上の**plugconfig**オブジェクトがある場合には、最後に読み込まれた**plugconfig**オブジェクトがランタイム・プラグイン環境で使用される。この場合、どのオブジェクトが使用されるのかを確定させるのは簡単ではないため、1つの**plugconfig**オブジェクトだけを使用するのが良い。

**plugconfig**をダブル・クリックすると、既に適当な位置に短いスクリプトがあることがわかる。これらはデフォルトの設定で、パッチに全く**plugconfig**オブジェクトが含まれていない場合の設定と同じものである。

**plugconfig**がランタイム・プラグイン環境で使われる場合には、読み出しのみのオブジェクトとなる。ランタイム・プラグイン環境は、オブジェクトのスクリプトから設定を読み出し、それに従って設定を行う。メッセージ・ビューとオフセットを送って、パッチを新しい位置にスクロールさせることもできるが、ほとんどのプラグインではプラグイン・インターフェースの上に現れるViewメニューを使って、ユーザーが設定できるようになっている。

## 入力

プラグイン内部に組み込まれるプリセットの集合を作るためには、**capture**と**recall**メッセージを用いる。

- capture** プログラム番号（1から始まる）とオプションのシンボルを伴う**capture**は、**plugconfig**を含むパッチとそのサブパッチのすべての**pp**オブジェクトと**plugmultiparam**オブジェクトの現在の設定を格納する。設定は**plugconfig**オブジェクトのスクリプトに追加された**setprogram**メッセージを用いて格納される。**pp**と**plugmultiparam**オブジェクトのパラメーター番号は、**setprogram**メッセージの中の値の順序を決定する。**capture**はランタイム・プラグイン環境では動作しない。
- recall** プログラム番号（1から始まる）を伴う**recall**は、すべての**pp**オブジェクトと**plugmultiparam**オブジェクトを、**plugconfig**オブジェクトのスクリプトの中の**setprogram**メッセージに格納されている値に設定する。**pp**と**plugmultiparam**オブジェクトのパラメーター番号は、**setprogram**メッセージの中身をどのように割り当てられるかを決定する。
- read** オプションのシンボルを伴う**read**は、Cubaseフォーマットで保存されたエフェクト・プログラムを読み込み、**plugconfig**オブジェクトに**setprogram**メッセージとして保存する。あるプラグインのためのエフェクト・プログラムが、**plugconfig**オブジェクトの中の1つと同じ重複しないIDコードを含んでいるか否かを確かめるチェックは行われな

い。また、**plugconfig**パラメーターの相手の数を確保するためのチェックも行われたい。アーギュメントとしてシンボルを指定すると、**plugconfig**はその名前のファイルを探す。そうでない場合には、エフェクト・プログラム・ファイルを選択するために、標準ファイル・ダイアログが表示される。

**view** **plugconfig**オブジェクトのスクリプトで定義されるビューの名前のシンボルを伴う**view**は、そのビューに割り当てられた座標オフセットに、**plugconfig**オブジェクトを含むパッチをスクロールする。

**offset** XとY座標の数値を伴う**offset**は、**plugconfig**オブジェクトを含むパッチを指定された座標にスクロールする。

## スクリプト・メッセージ

### Viewの設定のためのメッセージ

Viewはプラグインのエディット・ウィンドウの特定の環境設定である。**plugconfig**は、どのビューを見たいかをコントロールできるようにし、メニューで選択できる様々なピクセル・オフセットに、プラグイン・パッチのビューを追加する。これらはユーザーに提供するコントロールの「ページ」に相当する。

**usedefault** アーギュメント：なし

このメッセージがスクリプトに現れると、プラグイン・エディット・ウィンドウはなくなる。代わりに、ホスト環境のパラメーター編集機能を使用される。デフォルトでは、**usedefault**はスクリプトにないため、プラグインのエディット・ウィンドウが現れる。

**useviews** アーギュメント：次に説明するように、1または0の組み合わせでビューの表示を指定する。

**useviews**は、どのプラグイン・エディット・ウィンドウをユーザーに見せるかを決定する。ビューは次の順序で指定される：Parameters (卵形スライダー)、Interface (Maxパッチ・ベースのインターフェース)、Messages (プラグイン開発に有用な、Maxウィンドウと同じ出力)、Plug-in Info (自分のプラグインについて自慢できる場所)。エディット・ウィンドウが目に見える場合には、Pluggo Info ビューが常に表示される。

例えば、**useviews 1 0 0 0**は、Parametersビューだけをエディット・ウィンドウのViewメニューに配置する。ユーザーは他のビューに切り替えることはできない。

**defaultview** アーギュメント：名前、 $x$ のオフセット、 $y$ のオフセット、および初期ビューの表示を指定する1または0

**defaultview**は、プラグインのViewメニューのInterfaceアイテムの名前を、アーギュメントで指定した名前に変更し、ビューが表示される時に指定された $x$ と $y$ のオフセットにパッチをスクロールする。**defaultview**の4番目のアーギュメント（オプション）が0以外であれば、そのビューはプラグインのエディット・ウィンドウが開かれた時に表示される初期ビューになる。（**useviews**で指定された）Parametersビューがない場合にも、このことは有効である。

**addview** アーギュメント：名前、 $x$ のオフセット、 $y$ のオフセット

**addview**は、指定された $x$ と $y$ のオフセットとして、プラグインのViewメニューに新しいInterfaceビューを追加する。これにより、インターフェースの別の部分を見せるために、パッチをスクロールすることができるようになる。これはパラメーター・コントロールの「ページ」に相当する。追加したビューの名前をアーギュメントとして**view**メッセージを**plugconfig**に送ると、パッチ・ウィンドウは $x$ と $y$ のオフセットまで、そのビューをスクロールする。これはランタイム・プラグイン環境と同様にMaxでも動作し、これによってインターフェースの設定を試すことができる。

**dragscroll** アーギュメント：許可する（1）、許可しない（0）

このメッセージは、現時点ではまだ実装されていない。

**meter** アーギュメント：1（入力をメーターに表示する、デフォルト）、2（出力をメーターに表示する）、3（オフ）

**meter**メッセージは、プラグイン・エディット・ウィンドウの上部に表示されるレベルメーターの初期モードを設定する。現時点では固定的にメーターを非表示にする方法はないが、幅の狭すぎるエディット・ウィンドウに設定すれば、すべてを表示するだけの十分なスペースがないために、非表示となる。

## ウィンドウ設定のためのメッセージ

**autosize** アーギュメント：なし

**autosize**は、すべてのパラメーターを表示するのに十分な高さや幅になるよう、プラグイン・エディット・ウィンドウのサイズを自動的に設定する。デフォルトでは、この機能は有効である。



**setsize** アーギュメント：幅、高さ  
**setsize**は、プラグイン・エディット・ウィンドウをピクセル単位で指定したサイズにする。Parametersビューにおいて、卵形スライダーを適切に表示するためには狭過ぎるウィンドウ・サイズを指定した場合は、その指定は無視される。Viewメニューとレベル・メーター・パネルの高さを含めるために、パッチ・ウィンドウのサイズに約30ピクセルを加えなければならないことに注意。

**window size** アーギュメント：なし  
**window size**はプラグイン・エディット・ウィンドウのサイズをパッチ・ウィンドウのサイズに設定する。

## プログラム情報のためのメッセージ

**numprograms** アーギュメント：プログラムの数  
**numprograms**は、プラグインに格納されたプログラムの数を設定する。プログラムは**pp**と**plugmultiparam**オブジェクトを用いて設定した、それぞれのパラメーターの値（0から1まで）の集合である。プログラムの数のデフォルト値は64で、最小値は1、最大値は128である。デフォルトでは、すべてのプログラムで、それぞれのパラメーターは0に設定されているが、**setprogram**メッセージを用いて上書きすることができる。

**setprogram** アーギュメント：数値、名前、スタート・インデックス・オフセット、値のリスト…

通常は、**setprogram**メッセージをスクリプトとして入力することはないだろう。なぜなら、**capture**メッセージを送ることで、自動的に生成することができるからである。しかし、最終的にはそれを編集する（例えば、プログラムの名前を変更する）ことになるため、メッセージのフォーマットについて少し知っておくことは有用である。**setprogram**によって特定のプログラムの名前を付けたり、オプションとして、初期値を決めたりすることができる。プログラム番号（1番目のアーギュメント）は1から始まる。名前はシンボルであるため、名前の中にスペースがある場合には、シングル・スマート・クオートで括らなければならない。スタート・インデックス・オフセットは1に加える番号を設定し、これによってメッセージの中に並べられたパラメーターの値の開始パラメーター番号を決定する。このアーギュメントの後に、1つまたは2つのパラメーターが続く。定義されているパラメーターに対して十分な数の値を与えない場合には、残りのパラメーターは0に設定される。そもそも0に設定したい場合には、値を設定する必要がない。しかし、**plugconfig**スクリプトを再度開いた場合には、残りのパラメーターとして0が付加されているだろう。スター



ト・インデックス・オフセットのアーギュメントは、格納されたプログラムが256より多くのパラメーターを持つ場合に用いる。256はMaxメッセージの最大サイズである。

**initialpgm** アーギュメント：プログラム番号

**initialpgm**メッセージは、プラグインが最初に開かれた時に読み込まれるプログラムを指定する。デフォルト値は0で、何もプログラムが読み込まれないことを意味する。この場合には、**loadbang**オブジェクトを用いてプラグインのパラメーターの初期値を設定するかもしれない。しかし、この振る舞いは、読み込まれた時に、プログラム1から設定すべき値を取得する（なぜならプラグインがホスト・アプリケーションのドキュメントの一部として呼び戻されない限り、1は常に初期プログラムであるため）という大多数のプラグインとは一致しない。好みの設定の組み合わせができたなら、それらを**plugconfig**内の最初のプログラムに格納し、**initialpgm 1**メッセージを追加するのが良いだろう。これは、パラメーターを初期化するために用いていた**loadbang**オブジェクトが不要になる以外の恩恵もある。（最大で**numprograms**メッセージで指定したプラグインのプログラム数までの）任意の他のプログラム番号を読み込むことができるが、ホスト・シーケンサーのウィンドウに表示される現在のプログラム番号を、プラグインが変更することはできず、すべてのホスト・シーケンサーが初期状態でプログラム1に設定すると仮定すると、初期状態で他のプログラム番号を読み込むことは、結果としてユーザーを混乱させることになるからである。

## DSP設定のためのメッセージ

**accurate** アーギュメント：なし

**accurate**メッセージは、ランタイム・プラグイン環境において、Maxイベント（またはコントロール）スケジューラーをシグナル・ベクター・サイズと同じサンプル数の時間間隔で実行するように設定する。32サンプルの場合には、このインターバルは1ミリ秒よりも少し短くなるが、スケジューラーを実行すると、プラグインの全体のCPU負荷に、いくらかの影響を与える。デフォルトでは、**accurate**モードは有効になっておらず、スケジューラーはホスト環境のI/Oベクター・サイズと同じ（典型的には512または1024サンプル）時間間隔で実行する。**accurate**が唯一影響するのはプラグインに対するパラメーターの更新で、例えばLFOのコントロール・レートであれば、このモードを用いたくなるかもしれない。**accurate**モードの使用は、コントロール・レートで実行される**plugmod**処理からのパラメーターの更新の頻度も上げる。

**sigvs default** アーギュメント：なし

ランタイム・プラグイン環境において、このメッセージは現時点では無視される。32は現時点で唯一可能なシグナル・ベクター・サイズである。

**oversampling** アーギュメント：コード番号

ランタイム・プラグイン環境において、このメッセージは現時点では無視される。

**preempt** アーギュメント：コントロール・メッセージの優先度を設定する1または0

ランタイム・プラグイン環境において、このメッセージは現時点では無視される。

## 記述情報のためのメッセージ

プラグインの情報ビューの設定をする時、**infotext**でテキストを用いる、**infopict**でピクチャーを用いる、**noinfo**で情報ビューを持たない、の中から選択することができる。

**infotext** アーギュメント：独立した単語と数値としてのテキスト

**infotext**によって、エフェクトとプラグインのInfoビューに表示されるテキストを記述することができる。約256単語までの制限がある。特別なシンボルの<P>は改行を意味する。テキスト中のコンマとセミコロンの前には、バックスラッシュを付けることに注意しよう。そうしないと、保存時に残りのスクリプトを破壊させてしまうこともあり得る。

**infopict** アーギュメント：Maxサーチ・パス中のPICTファイルのファイル名

**infopict**によって、プラグインのInfoビューに表示する画像を含めることができる。**infopict**を使う場合には、プラグインのコレクティブ・スクリプトに（手動で）その画像を含める必要がある。ランタイム・プラグイン環境はコレクティブ中から画像を見つけることができる。

**noinfo** アーギュメント：なし

これはプラグイン情報のデフォルトの設定である。エフェクトに関する情報として、テキストも画像も用意されていない場合には、先述した**useviews**コマンドで有効にしたとしても、ViewメニューにPlug-in Infoアイテムは現れない。**noinfo**と**infopict**または**infotext**の両方がスクリ

プトに含まれている場合には、**noinfo**は無効になり、情報ビューが表示される。

**welcome** アーギュメント：独立した単語と数値としてのテキスト

**welcome**メッセージに続くテキストのアーギュメントは、下部のヒント・エリアに表示される。これは、ユーザが最初にプラグインのエディター・ウィンドウを開いてParametersビューを見る時や、Parametersビューが表示されていて、マウスがウィンドウの上部に移動した時に、表示される。**nohintarea**メッセージがスクリプトに含まれているなら、Parametersビューにはヒント・エリアが含まれなくなるので、**welcome**メッセージは表示されない。

**nohintarea** アーギュメント：なし

**nohintarea**がスクリプトに現れた場合には、ランタイム・プラグイン環境はParametersビューの下の部分のヒント・エリアのためのスペースを用意しない。しかし、**window**sizeまたは**set**sizeで定義されたサイズにおいて、卵形スライダーの数がエディット・ウィンドウを完全に覆い尽くさない場合には、ヒント・エリアは表示される。

**swirl** アーギュメント：なし

**swirl**メッセージは、ヒント・エリアの背景をpluggoのパッケージにインスパイアされた渦巻きに設定する。ヒント・エリアのデフォルトのアピアランスはpainであり、渦巻きの背景ではない。渦巻きの色を設定するには**hintfg**と**hintbg**を用いる。ちなみに、pluggoのパッケージは、クラシックなフランス映画「Les Demoiselles de Rochefort (邦題：ロシュフォールの恋人達)」の宣伝用ポスターにインスパイアされている。

**hintbg** アーギュメント：16ビットの値としての赤、緑、青の成分

ヒント・エリアの黄色の背景が気に入らない場合には、他の色に変えることができる。例として、中間のグレーは**hintbg 40000 40000 40000**で指定され、白の背景は**hintbg 65535 65535 65535**で指定される。

**hintfg** アーギュメント：16ビットの値としての赤、緑、青の成分

ヒント・エリアに**swirl**モードを用いる場合、**hintfg**メッセージで渦巻きの暗い部分の色を指定する。最も良い結果を得るには、**hintfg**は**hintbg**よりも暗くならない。

**uniqueid** アーギュメント：**id1 id2 id3** (0から255まで)

最初に開いた時に、**plugconfig**スクリプトにこのメッセージがあるのを見つけるだろう。アーギュメントはランダムに生成された0から255までの数値で、IPアドレスの4分の3のようなものである。

これらの数値は、プラグインを個別のものとして識別するためのIDコードを生成するために用いられる。このコードは、パッチ間の**plugmod**接続を行うために用いられるのと同時に、pluggoベースのプラグインを識別するために用いられる。

これらの数値には、ランダムに生成された値や、何か計画的な値の両方を用いることができ、約1600万の可能性がある。0 0 0は予約されているので使用することはできない。0の後に2つの0以外の値が続くものはCycling '74と登録プラグイン開発者によって予約されている。このIDコードに関わる必要はないかもしれないが、この一部が**plugmod**オブジェクトによる出力の浮動小数点による「パッチ・コード」の基礎として使われていることは、知っておく必要があるかもしれない。しかし、その浮動小数点の値は、何らかの形で、あなたの選択したIDに似ているというわけではない。

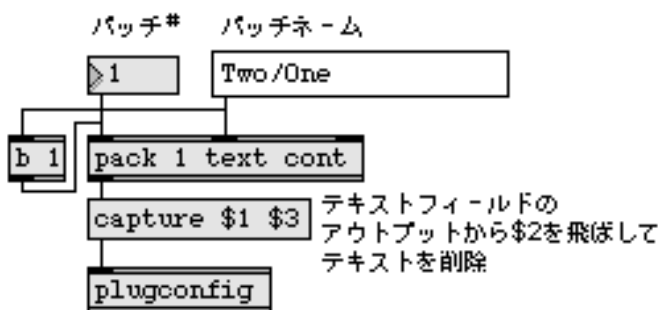
## アーギュメント

なし。

## 出力

なし。

## 利用例



captureメッセージを**plugconfig**に送ってプリセットを生成する

## 参照

### plugmod

プラグインのパラメーター値を変更する

**plugin~**と**plugout~**はプラグインへのシグナルの入力と出力を定義する。**plugin~**にテスト・シグナルを入力し、**plugout~**の出力を**dac~**オブジェクトに接続することで、Maxにおいて単純なスルー・オブジェクトとして使うことができる。しかしながら、**plugin~**と**plugout~**がランタイム環境で実行される場合は、異なる動作をする。**plugin~**は入力を無視して、代わりにホストのミキサーからプラグインへ送られたシグナルを出力する。**plugout~**はいかなるタイプのシグナルもアウトレットから出力せず、代わりにシグナル入力をホストのミキサーへプラグインからのシグナルを出力する。

## 入力

**シグナル** 第1インレットと第2インレット：Max/MSPで使用される時、**plugin~**オブジェクトへの入力を、**plugin~**オブジェクトから、そのまま出力する。ランタイム・プラグイン環境で使用される時は、その入力に送られるシグナルは無視され、代わりにプラグインへのオーディオ入力が、**plugin~**オブジェクトのアウトレットから出力される。

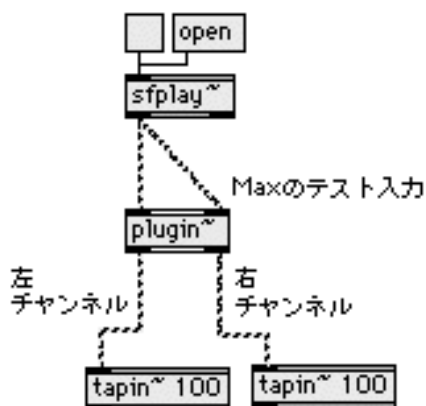
## アーギュメント

なし。**plugin~**は常に2つの入力と2つの出力を持つ。

## 出力

**シグナル** Max/MSPで使用される時、**plugin~**オブジェクトへ入力されたシグナルが、そのまま出力される。ランタイム・プラグイン環境で使用される時は、ホストからプラグインへ入力された左右のチャンネルのオーディオ・シグナルが出力される。プラグインがモノラルとしてインストールされている場合には、左のチャンネルのみオーディオ・シグナルが出力され、右チャンネルは0であるということがあり得る。プラグインへのオーディオ入力の正確な実態は、ホスト・ミキサーによって異なる。

## 利用例



**plugin~**

プラグインのオーディオ  
入力を定義する

---

**参照**

**plugout~**

プラグインのオーディオ出力を定義する

**plugmidiin**はプラグインへ送られた任意のMIDI情報を入力する。Maxの**midin**オブジェクトに類似した機能を持ち、生のMIDIを連続したバイト・ストリームとして出力する。アウトレットに**midiparse**オブジェクトを接続することが多いだろう。MIDI情報は、常に**plugmidiin**によって高い優先度（インタラプト）レベルで出力される。1つのパッチの中に1つ以上の**plugmidiin**オブジェクトを持つこともできる。この場合は、それぞれの**plugmidiin**オブジェクトは同じ情報を入力する。

## 入力

なし。

## アーギュメント

なし。

## 出力

**整数** 連続したMIDIメッセージ・バイト。例えば、チャンネル1のノート番号60でベロシティ64のノート・オン・メッセージは、144の後に60と64が続く出力となる。

## 利用例

plugmidiinは、一部のPluggoプラグインでMSPのテスト時に使用される



ホスト・アプリケーションから受信したMIDIメッセージは、**plugmidiin**オブジェクトによって出力される

## 参照

**midiparse**  
**plugmidiout**

生のMIDIデータを解釈する  
プラグインのホストに対してMIDIを送信する



**plugmidiout**はMIDI情報をホストに送信するが、現在の設定に従って接続先が決めらる。プラグインはそのMIDI出力の接続に関して何の支配力も持たない。**plugmidiout**は**midiout**に類似した機能を持ち、連続した生のMIDIバイトを入力することになる。**midiformat**を用いることで、数値を適切なMIDIメッセージに変換して、**plugmidiout**へ入力することができる。

## 入力

**整数** 連続したMIDIメッセージ・バイト。例えば、チャンネル1のノート番号60でベロシティ64のノート・オン・メッセージは、144の後に60と64が続くことになる。

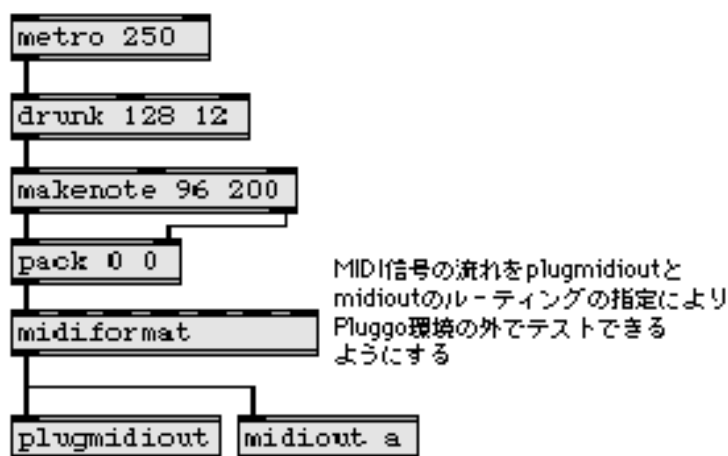
## アーギュメント

なし。

## 出力

なし。

## 利用例



## 参照

**midiformat**  
**plugmidiin**

MIDIメッセージの形式でデータを用意する  
プラグインのホストからMIDIを受信する

**plugmod**によって、プラグインは、他のプラグインのパラメーター値を変更することができる。**plugmod**はすべての現在読み込まれているプラグインの、すべての目に見えるパラメーターをリスト化したポップアップ・メニューを生成する。このメニューの出力はオブジェクトに入力され、**plugmod**が受け取る数値入力によって、どのパラメーターが変更されるのかを知ることができる。追加のインレットとアウトレットは、エフェクトの特定のプラグインとパラメーターに対する**pp**オブジェクトに関連付けられる。これにより**plugmod**は、シーケンサーのドキュメントが再度読み込まれた時に、ターゲットとなるプラグインとパラメーターに再接続することができる。

## 入力

**任意** 第1インレット：パラメーター・インデックスを伴うプラグインの名前は、**plugmod**オブジェクトがその数値入力によって変更するパラメーターを設定する。このプラグインとパラメーターは、オブジェクトのターゲットとして参照される。

**No Connection** 第1インレット：**No Connection**を受信すると、**plugmod**オブジェクトは（もしあれば）現在のターゲットとのあらゆる接続を解除し、ターゲット・パラメーターに影響を与えることを止める。**No Connection**シンボルは、ランタイム環境にプラグインが挿入または削除された時に、**plugmod**オブジェクトの第1アウトレットによって生成されたメニューの中で、常に最初の項目となる。

**整数または実数** 第1インレット：0から1までの間に制約された受信した値は、ターゲットのプラグインとパラメーターに割り当てられる。

第2インレット：受信した値は**plugmod**が変更を行う前のパラメーターの値に加算される

第3インレット：受信した値は**plugmod**が変更を行う前のパラメーターの値に乗算される

**実数** 第4インレット：値はターゲットとしての新しいプラグインに割り当てられるコードとして解釈される。**pp**オブジェクトのアウトレットは、通常このインレットに接続される。

第5インレット：値はターゲットとしての新しいパラメーターに割り当てられるコードとして解釈される。**pp**オブジェクトのアウトレットは、通常このインレットに接続される。

## アーギュメント

なし。

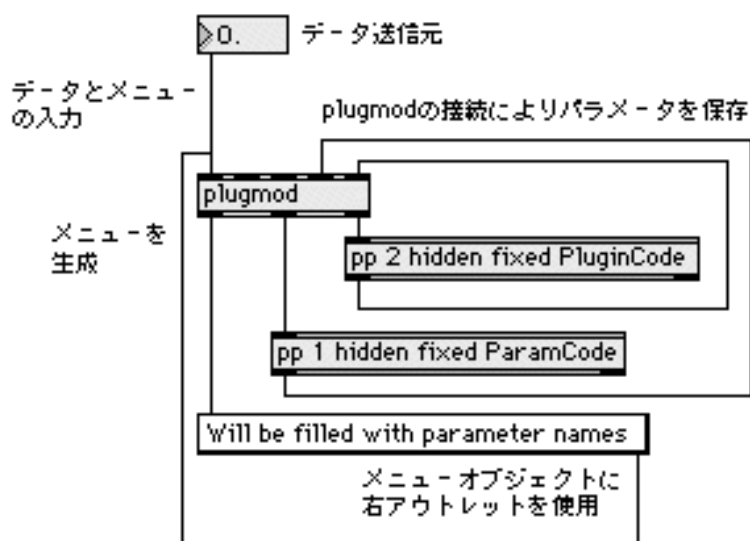
## 出力

**任意** 第1アウトレット：**plugmod**オブジェクトのこのアウトレットからの出力は、新しいプラグインが挿入または削除された時に発生する。メッセージは、このオブジェクトが変更する潜在的なターゲットであるプラグインとパラメーターの新しいリストによるメニュー・オブジェクトを更新する。

**実数** 第2アウトレット：接続されたポップアップ・メニュー・オブジェクトから、オブジェクトの第1インレットにメッセージが送られてオブジェクトのターゲットが変更された時や、新しいプラグイン・コードを第4インレットに受け取った時に、現在のプラグイン・コードを出力する。

第3アウトレット：接続されたポップアップ・メニュー・オブジェクトからオブジェクトの第1インレットにメッセージが送られてオブジェクトのターゲットの変更された時や、新しいプラグイン・コードを第5インレットに受け取った時に、現在のパラメーター・コードを出力する。

## 利用例



## 参照

**umenu**

コマンドを表示し、出力するポップアップ・メニュー

**plugmorph**は、2つまたはそれ以上のエフェクト・プログラムの加重平均を作り出すことにより、他のプラグインのパラメーター値を変更することを可能にする。そのような平均は「モーフ」としても知られるが、それはしばしば（しかし常にではなく）あるエフェクト・プログラムと他のエフェクト・プログラムの間の連続した知覚空間を作り出す。**plugmorph**は現在読み込まれているすべてのプラグインのポップアップ・メニュー・リストを生成する。

このメニューの出力はオブジェクトの入力にフィードバックされ、**plugmorph**が受け取った入力に従って、どのプラグインが変更されるべきかをユーザーが指定することを可能にする。追加のインレットとアウトレットのインターフェースは、**pp**オブジェクトと一緒にあって、特定のプラグインへのオブジェクトの接続を保存する。これによって、シーケンサーのドキュメントが再度読み込まれた時に、**plugmorph**がそのターゲットのプラグインに再接続することができるようになる。

## 入力

**任意** 第1インレット：プラグインの名前は、**plugmorph**オブジェクトがその入力とともに変更を加えるプラグインの名前を設定する。このプラグインはオブジェクトのターゲットとみなされる。

**No Connection** 第1インレット：**No Connection**を受け取ると、**plugmorph**オブジェクトは（もしあれば）現在の接続とともにその接続を切断し、それ以上プラグインのパラメーターに変更を加えなくなる。**No Connection**シンボルは、実行環境でプラグインが挿入または削除された時に、**plugmorph**オブジェクトの第1アウトレットによって生成されるメニューの中で常に最初のアイテムである。

**リスト** 第1インレット：接続されたプラグインのパラメーターのための新しい値を**plugmorph**に計算させる。リストのフォーマットは、エフェクト・プログラムの番号の後に重みの割合が続くというものである。最大128までのプログラム番号を指定することができる。重みの合計が1に満たなければ、そのようになるように正規化される。例えば、**1 0.5 2 0.5**というリストはターゲットのプラグインのパラメーターをエフェクト・プログラム1と2の単純な平均に設定する。**1 0.6 2 0.6 3 0.6 4 0.6**というリストは、それぞれのプログラムが均等に表現されるように最初の4つのエフェクト・プログラムの加重平均を計算する。言い換えれば、それぞれのプログラムのパラメーターの値は、モーフされた値に25%の影響を与えている。ターゲットのプラグインの現在のエフェクト・プログラムが、モーフされるエフェクト・プログラムの中に含まれている場合には、ユーザーが1つ以上のモーフを実行することができるように、そのパラメーターの値を保存しない試みがなされる。生成されたパラメーターの値は、**plugmorph**への**store**メッセージを用いて後で保存することができる。しかしながら、**multiSlider**ベースのプラグインのいくつかは、この保存防止機構が働かないような方法での

パラメーターの変更を行わない。これは、現在のエフェクト・プログラムをモーフに参与しない番号に設定するよう、ユーザに要求することになる。

- morphfixed** 第1インレット：1つの数値を伴う**morphfixed**は、**fixed**としてマークされたパラメーターがモーフに含まれるか否かを決定する。数値が0であれば、**fixed**パラメーターは含まれず、その値は変化しないまま残される。数値が0以外であれば、**fixed**パラメーターは含まれる。**plugmorph**のデフォルトのふるまいは**fixed**パラメーターを含む。
- morphhidden** 第1インレット：1つの数値を伴う**morphhidden**は、**hidden**としてマークされたパラメーターがモーフに含まれるか否かを決定する。数値が0であれば、**hidden**パラメーターは含まれず、その値は変化しないまま残される。数値が0以外であれば、**hidden**パラメーターは含まれる。**plugmorph**のデフォルトのふるまいは**hidden**パラメーターを含む。
- store** 第1インレット：**store**はターゲットとなるプラグインのパラメーターの現在の値を、エフェクト・プログラムに保存する。
- 実数** 第2インレット：与えられた値はターゲットとしての新しいプラグインを割り当てるためのコードとして解釈される。通常、**pp**オブジェクトのアウトレットは、このインレットに接続される。

## アーギュメント

なし。

## 出力

- すべて** 第1アウトレット：**plugmorph**オブジェクトのこのアウトレットからの出力は、新しいプラグインが追加または削除された時に起きる。メッセージは、潜在的なターゲットであるプラグインの新しいリストで結びついている**menu**オブジェクトを更新する。
- 実数** 第2アウトレット：新しいプラグインがターゲットして選択されると、**plugmorph**はエフェクト・プログラムの数をアウトレットから出力する。
- 第2アウトレット：現在のパラメーター・コードは、結びついているポップアップ・メニュー・オブジェクトからのメッセージ経由で、オブジェクトの第1インレットにオブジェクトのプラグインのターゲットが変化した時、または新しいパラメーター・コードを第2インレットで受け取った時に出力される。





**plugmultiparam** オブジェクトを用いることで、単一のオブジェクトによって表示され、変更される3つかそれ以上のパラメーターを設定することができる。しかしながら、これらのパラメーターはプラグイン・ウィンドウのParametersビューからは隠されることになる。これらのパラメーターは、Maxのユーザー・インターフェースを作成することによってのみ変更することができる。もともと、**plugmultiparam**は**multiSlider**オブジェクトと共に使われるように設計された。しかし、**plugstore**や、うまく組み合わせられた1組の**pack**と**unpack**オブジェクトと共に動作することもできる。

## 入力

- 整数** 指定されたパラメーター・インデックスの値は、オブジェクトの第2アウトレットから出力される
- リスト** オブジェクトのパラメーターに割り当てられる1組の値として解釈され、最小の番号を付けられたパラメーターから始まる。オブジェクトによって定義されたパラメーター数よりもリストが長い場合には、余分の要素は無視される。リスト中の値は、オブジェクトの**minimum**と**maximum**アーギュメントの間に制限される。
- bang** 現在格納されている値を、オブジェクトの第1アウトレットから送信する
- setmessage** シンボルを伴う**setmessage**は、変化が生じた時（例えば、格納されたプログラムが変更された時など）に、個々の値を設定するためのメッセージを設定する。デフォルトの**select**メッセージは**multiSlider**オブジェクトと共に使う時に有効である。

## アーギュメント

- 整数** 必須。オブジェクトが扱う開始パラメーター・インデックスを定義する。
- 整数** 必須。オブジェクトが扱うパラメーター・インデックスの数を定義する。
- 実数または整数** オプション。すべてのパラメーターに対する入出力の最小値を設定する。デフォルト値は0である。
- 実数または整数** オプション。すべてのパラメーターに対する入出力の最大値を設定する。デフォルト値は1である。

例：値の幅が1から99までの32個のパラメーターをパラメーター・インデックスの13から順に格納する場合は、**plugmultiparam**オブジェクトのアーギュメントは次のようになる。

```
plugmultiparam 13 32 1 99
```



## 出力

**リスト** 第1アウトレット：第1アウトレットはオブジェクトが**bang**メッセージを受信した時に、現在の値をリストとして出力する

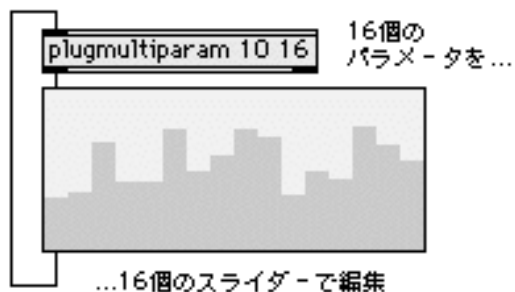
**任意のメッセージ** 第1アウトレット：**plugmultiparam**オブジェクトは、次のフォーマットに従って、個々の値を設定するためのメッセージも出力する。

<メッセージの名前> <インデックス> 値

デフォルトでは、メッセージの名前はselectであり、これは**multiSlider**オブジェクトの中の1つの値を設定するのに適している。上記のように**setmessage**メッセージを用いて、メッセージを他のシンボルに変えることができる。最初のパラメーターに対するインデックスのアーギュメントは0から始まり、それに続くパラメーターごとに1ずつ大きくなる。これは**plugmultiparam**への開始パラメーター・インデックスには影響されない。インデックスのアーギュメントの後には、現在のパラメーターの値が続く。

**実数** 第2アウトレット：整数のメッセージを受け取った時、指定されたパラメーター・インデックスの値が出力となる。

## 利用例



## 参照

**plugstore  
pp**

複数のプラグインのパラメーターの値を格納する  
プラグイン・パラメーターを定義する

**plugin~**と**plugout~**は、プラグインのシグナルの入出力を定義する。**plugin~**にテスト・シグナルを入力し、**plugout~**の出力を**dac~**オブジェクトに接続することで、Maxにおいては単純なスルー・オブジェクトとして使うことができる。しかしながら、**plugin~**と**plugout~**がランタイム環境で実行される場合は、異なる動作をする。**plugin~**は入力を見捨て、代わりにホスト・ミキサーからプラグインへ送られたシグナルを出力する。**plugout~**はいかなるタイプのシグナルもアウトレットから出力せず、代わりにシグナル入力をホスト・ミキサーへプラグインからのシグナルを出力する。

## 入力

**シグナル** 第1と第2インレット：Max/MSPで使われる時、**plugout~**オブジェクトへ入力されたシグナルが、そのまま出力する。ランタイム・プラグイン環境で使われる時は、**plugout~**へ入力されたシグナルが、ホスト・ミキサーへ出力される。

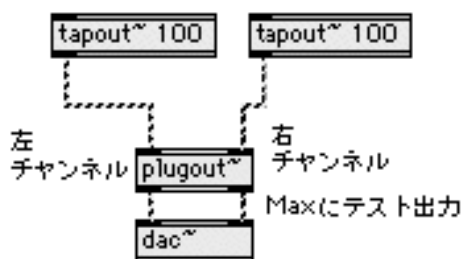
## アーギュメント

**整数** オプション。1つまたは2つの整数のアーギュメントによって、プラグインの出力チャンネルの出力先を指定する。アーギュメントがなければ、**plugout~**はチャンネル1と2にアサインされた2つの出力を持つことになる。

## 出力

**シグナル** Max/MSPで使用される時、**plugout~**へ入力されたシグナルが、そのまま出力される。ランタイム・プラグイン環境で使われる時、アウトレットからシグナルは出力されず、プラグインからホスト・ミキサーへシグナルが出力される。

## 利用例



## 参照

**plugin~** プラグインのオーディオ入力を定義する

**plugphasor~**は、ホスト・シーケンサーの拍にサンプル単位で同期したオーディオ・レート  
のノコギリ波を出力する。この波形は他のオーディオ・オブジェクトに送ることができ、  
それによってホストのオーディオ処理に、プラグインのオーディオ処理を同期させること  
ができる。

## 入力

なし。

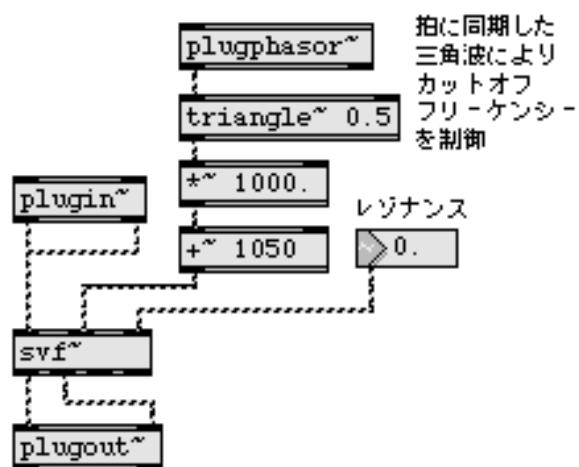
## アーギュメント

なし。

## 出力

シグナル **plugphasor~**の出力は**phasor~**に似ており、1拍の時間を通して、0から  
1.0まで立ち上がる。現在のホスト環境が同期をサポートしていないか、  
またはトランスポートが停止している時には、**plugphasor~**の出力は無  
音になる。

## 利用例



拍に同期した鋸歯状波でオシレーターを駆動する

## 参照

**plugsync~** ホストの同期情報を出力する

**plugreceive~**と**plugsend~**オブジェクトは、あるプラグインから他のプラグインにオーディオ・シグナルを送信するために使用される。これは、**pluggo**に含まれる、多くのプラグインの**PluggoBus**機能の実装としても使用されている。

## 入力

**シグナル** **plugreceive~**オブジェクトへの入力は、関連付けられている**plugsend~**オブジェクトから送信される。最初の状態では、**plugreceive~**オブジェクトのアーギュメントと同じ名前を持つ**plugsend~**になる。

**set** **plugsend~**オブジェクトの名前となるシンボルを伴う**set**は、**plugreceive~**オブジェクトを指定された**plugsend~**オブジェクトに関連付け、**plugsend~**に入力されたシグナルが、**plugreceive~**オブジェクトから出力される。このシンボルと一致する**plugsend~**オブジェクトが存在しない場合は、オーディオ出力はゼロになる。

## アーギュメント

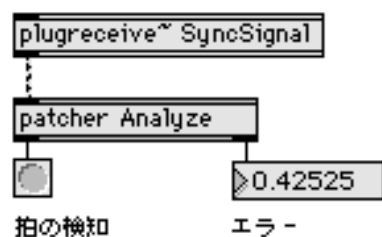
**シンボル** 必須。1つまたはそれ以上の**plugsend~**オブジェクトと関連付けるための、**plugreceive~**オブジェクトの名前を指定する。

## 出力

**シグナル** このオブジェクトに関連付けられた**plugsend~**オブジェクトへ入力されたシグナル。関連付けられる**plugsend~**オブジェクトが存在しない場合は、オーディオ出力はゼロになる。

**plugreceive~**の出力と、**plugreceive~**が置かれているプラグインの入力の間にはホスト・ミキサーの処理 (I/O) ベクター・サイズ分の遅れがあるかもしれない。これは、オーディオ処理の順序として、送信する先である**plugreceive~**の処理よりも後に、**plugsend~**の処理が行われる場合に生じる。

## 利用例



## 参照

**plugsend~** 他のプラグインにオーディオを送信する

# plugsend~

他のプラグインに  
オーディオを送信する

**plugreceive~**と**plugsend~**オブジェクトは、あるプラグインから他のプラグインにオーディオ・シグナルを送信するために使用される。これは、**pluggo**に含まれる、多くのプラグインの**PluggoBus**機能の実装としても使用されている。

## 入力

**シグナル** 同じプラグインの間でも、異なるプラグインの間でも、**plugsend~**オブジェクトへの入力は、他の**plugsend~**オブジェクトへの入力とミックスされ、関連付けられた任意の**plugreceive~**オブジェクトのシグナル・アウトレットに送信される。

## アーギュメント

**シンボル** 必須。他の**plugsend~**と**plugreceive~**オブジェクトを関連づけるために使用される**plugsend~**オブジェクトの名前を与える。

## 出力

なし。

## 利用例



## 参照

**plugreceive~** 他のプラグインからオーディオを受信する

**plugstore** オブジェクトは **plugmultiparam** と共に動作し、パッチ中の複数の場所で **plugmultiparam** に値を格納したり、取り出したりすることを可能にする。

## 入力

- bang** 格納されたリストをオブジェクトのアウトレットから出力する
- リスト** (オブジェクトのサイズまでの) リストの要素を格納し、オブジェクトのアウトレットから出力する。
- select** インデックスと値を伴う **select** は、指定したインデックス（最初の要素が1から始まる）に値を格納し、格納されたリストをオブジェクトのアウトレットから出力する。
- set** インデックス番号と値を伴う **set** は、指定したインデックス（最初の要素が1から始まる）に値を格納するが、格納されたリストは出力しない。

## アーギュメント

- 整数** 必須。 **plugstore** オブジェクトのリストに格納される要素の数を設定する。

## 出力

- リスト** リスト、 **bang**、または **select** メッセージを受け取った時は、常に格納されたリストが出力される。

## 利用例



## 参照

- plugmultiparam** 複数のプラグイン・パラメーターを設定する

**plugsync~** オブジェクトは、ホストの現在の状態についての情報を出力する。サンプル・カウント情報はいかなるホストでも有効である。このオブジェクトによる他の情報の有効性は、ホストがどのような同期に関する処理を実装しているかに依存する。フラグ（第9）アウトレットからの値は、どの情報が有効であるかを告げる。**plugsync~**からの出力は、スケジューラーが実行されている間は途切れない。

## 入力

なし。

## アーギュメント

なし。

## 出力

- 整数** 第1アウトレット：ホストのトランスポートが現在走行中であれば1。停止中または一時停止中であれば0。
- 整数** 第2アウトレット：ホスト・シーケンスの現在の小節数で、最初の小節が1から始まる。ホストが同期をサポートしていない場合には、このアウトレットからは何も出力されない。
- 整数** 第3アウトレット：ホスト・シーケンスの現在の拍数で、最初の拍が1から始まる。ホストが同期をサポートしていない場合には、このアウトレットからは何も出力されない。
- 整数** 第4アウトレット：ホスト・シーケンスの現在の拍の小数部分で、0から1.0までの値をとる。ホストが同期をサポートしていない場合には、出力は0になる。ホストが同期をサポートしていない場合には、このアウトレットからは何も出力されない。
- リスト** 第5アウトレット：現在の拍子を分子の後に分母という形式でリストとして出力する。例えば、3/4拍子は3 4というリストとしての出力される。ホストが拍子情報をサポートしていない場合には、このアウトレットからは何も出力されない。
- 実数** 第6アウトレット：拍あたりのサンプル数で表した現在のテンポ。この数値は次の式で1分間当たりの拍数（BPM）に変換することができる。（サンプリングレート/拍あたりのサンプル数） \* 60。ホストが同期をサポートしていない場合には、このアウトレットからは何も出力されない。



**実数** 第7アウトレット：1 PPQで表した現在の拍数。この数値は拍の間の分数の部分を含む場合がある。ホストが同期をサポートしていない場合には、このアウトレットからは何も出力されない。

**実数** 第8アウトレット：ホストによって定義される現在のサンプル・カウント

**整数** 第9アウトレット：**pluginsync~**からのその他の情報の有効性を示す数値。次の値でマスクすることで**pluginsync~**からの情報が有効であるかどうかを判断する。

| 項目        | マスクの値 | 補足                   |
|-----------|-------|----------------------|
| サンプル・カウント | 1     | 常に真                  |
| 拍数        | 2     | 第2、第3、第4、第7アウトレットが有効 |
| 拍子        | 4     | 第5アウトレットが有効          |
| テンポ       | 8     | 第6アウトレットが有効          |
| トランスポート   | 16    | 第1アウトレットが有効          |

## 参照

**plugphasor~**          ホストに同期したノコギリ波

## 入力

- シグナル** 第1インレット：**buffer~**へ書き込むシグナル。
- 第2インレット：第1インレットに入力するシグナルを書き込む位置を指定するインデックスとなるシグナル。**-1**の値を持つシグナルが入力された場合は、何も書き込まない。
- 実数** **peek~**オブジェクトと同様、**poke~**も実数値を**buffer~**に書き込むことができる。しかし、左側の2つのインレットは、**peek~**オブジェクトと逆になっていることに注意が必要である。
- 第1インレット：**buffer~**に対し指定されたインデックスの位置へ書き込む値。インデックスが**-1**でない場合に書き込まれる。
- 第2インレット：整数に変換される。
- 第3インレット：整数に変換される。
- 整数** 第1インレット：実数に変換される。
- 第2インレット：第1インレットに入力された値を次に書き込む位置を指定するインデックス値。このインレットにシグナルが入力されている場合、実数の入力は無視される。
- 第3インレット：データを書き込む**buffer~**のチャンネルを指定する。数値としては、1が1つ目のチャンネルを表す。
- list** 第1インレット：2つ以上の値によるリストは、2つ目の値が示すインデックスの位置に、1つ目の値を書き込む。3つ目の値が設定されている場合、それは**buffer~**内でのオーディオ・チャンネルを表す。
- set** **set**に続き、関連付ける**buffer~**の参照名を指定する。これにより、**poke~**へ入力されるサンプルの書き込み先を変更することができる。
- (マウス) **poke~**オブジェクトをダブル・クリックすることで、関連付けされた**buffer~**オブジェクトの波形編集ウィンドウが開く。

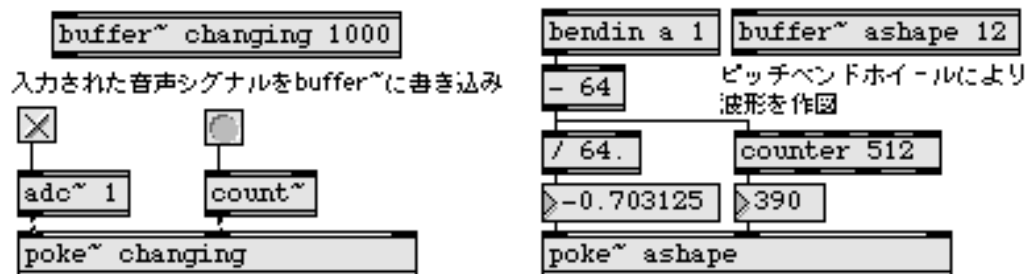
## アーギュメント

- シンボル** 必須。データの読み書きを行う**buffer~**オブジェクトの参照名を指定する。
- 整数** オプション。**buffer~**が複数チャンネルを持っている場合、**buffer~**名に続いて、書き込みを行うチャンネルを指定できる。デフォルトのチャンネルは1である。

## 出力

なし。

## 利用例



シグナルまたは数値によるbuffer~へのデータの書き込み

## 参照

buffer~  
buffir~  
peek~

オーディオ・サンプルの格納  
buffer~を利用したFIRフィルター  
サンプル値の読み出しと書き込み

## 入力

**シグナル** 第1インレット：周波数ビンの大きさ（振幅）は、ペアとなった直交座標（実部/虚部）のシグナルに変換される。

第2インレット：周波数ビンの位相はペアとなった直交座標（実部/虚部）のシグナルに変換される。

## アーギュメント

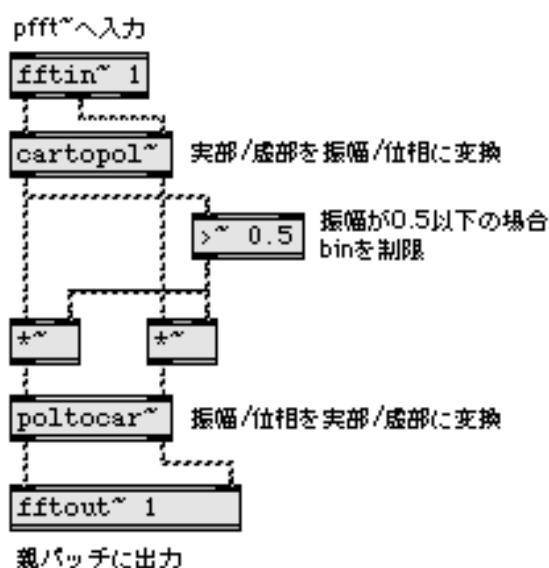
なし。

## 出力

**シグナル** 第1アウトレット：周波数領域でのシグナルの実数部分をシグナルで **ifft~** オブジェクト、または **ffftout~** オブジェクトに入力するのに適した形で出力する。

第2アウトレット：周波数領域でのシグナルの虚数部分シグナルで **ifft~** オブジェクト、または **ffftout~** オブジェクトに入力するのに適した形で出力する。

## 利用例



**poltoocar~** オブジェクトが、振幅/位相のペアを **ffftout~** が使用する極座標に変換する

## 参照

|                    |                                  |
|--------------------|----------------------------------|
| <b>cartopol</b>    | 直交座標を極座標に変換する                    |
| <b>cartopol~</b>   | シグナルについて直交座標を極座標に変換する            |
| <b>fft~</b>        | 高速フーリエ変換                         |
| <b>fftin~</b>      | <b>pfft~</b> に読み込まれるパッチにおける入力    |
| <b>fftinfo~</b>    | <b>pfft~</b> に読み込まれるパッチに関する情報の取得 |
| <b>fftout~</b>     | <b>pfft~</b> に読み込まれるパッチにおける出力    |
| <b>frameaccum~</b> | 連続した位相変化データのフレームから位相を計算          |
| <b>framedelta~</b> | 連続したFFT処理フレーム間の位相変化を計算           |
| <b>ifft~</b>       | 逆高速フーリエ変換                        |
| <b>pfft~</b>       | 周波数領域処理を行うパッチの管理                 |
| <b>poltoacar</b>   | 極座標を直交座標に変換する                    |
| <b>vectral~</b>    | ベクター・ベースのエンベロップ・フォロワー            |

Tutorial 26

Frequency Domain Signal Processing with **pfft~** (英文PDF参照)

**poly~**オブジェクトは**patcher**オブジェクトと良く似ていて、パッチをオブジェクト・ボックス内にカプセル化する。名前からも想像できるように、**patcher**オブジェクトはカプセル化されたパッチのコピーを1つだけ持つことができるが、**poly~**オブジェクトは読み込まれたパッチのインスタンス（コピー）を1つ以上持つことができる。パッチ・ファイル名といくつのインスタンスを作るかは、**poly~**オブジェクトのアーギュメントとして指定できる。

**poly~**オブジェクトは、パッチのインスタンスの内部にある**in**オブジェクトや**in~**オブジェクトへのインレットに受け取ったシグナルとイベント（メッセージ）を管理する。そのパッチはシグナルかイベントを**poly~**オブジェクトから出力させるために**out**オブジェクトや**out~**オブジェクトを含むことができる。**poly~**オブジェクトへのメッセージは、読み込まれたパッチ・インスタンス内でのオーディオ処理のコントロールや、イベントのルーティングのコントロールを行う。

## 入力

**任意** **poly~**オブジェクトのインレットやアウトレットの数は、読み込まれるパッチによって決定される。**poly~**オブジェクトに読み込まれたパッチのインレットは、シグナルとイベントの入力を両方とも受け取ることができる。

シグナルは**in~**オブジェクトを、イベントは**in**オブジェクトを用いて、読み込まれたパッチの内部で処理される。**poly~**オブジェクトの全体のインレットの数は、読み込まれたパッチの内部の**in~**オブジェクトか**in**オブジェクトの最も大きい数によって決定される（例えば、3のアーギュメントを持った**in~**オブジェクトと、4のアーギュメントを持った**in**オブジェクトがあれば、**poly~**オブジェクトは4つのインレットを持つことになる）。すべてのインレットは、それぞれのインレットに対応する**in~**オブジェクトがなくてもシグナルの接続を受け取ることができる。

シグナルの入力はすべてのインスタンスに送られる。

**任意のメッセージ** 任意のインレット：メッセージは、**poly~**オブジェクトにおいて現在ターゲットとなっているパッチ・インスタンスの中の**in**オブジェクトに送られる。**poly~**オブジェクトの第1インレットに受け取ったメッセージは、**in 1**オブジェクトに送られ、同じように、第2インレットに受け取ったメッセージは**in 2**オブジェクトに送られる。

**シグナル** 任意のインレット：すべてのパッチ・インスタンス内の対応する**in~**オブジェクトにシグナルは送られる。**poly~**オブジェクトの第1インレットに接続されたシグナルは**in~ 1**オブジェクトに受け取られ、同じように第2インレットに接続されたシグナルは**in~ 2**オブジェクトに受け取られる。

- リスト** 任意のインレット：**poly~**オブジェクト自体のコントロールに使われる単語で始まるメッセージを、**poly~**オブジェクトのインスタンスに送りたい場合は、その先頭に**list**という単語を加える。例えば、**list target 2**というメッセージを**poly~**オブジェクトの第1インレットに送ると、現在のターゲットのインスタンスを2番目のパッチに切り替えるのではなく、**in 1**オブジェクトのアウトレットから**target 2**というメッセージを出力する。
- down** 第1インレット：**down**には2の累乗となる数値が続き、現在読み込まれているパッチ上での処理を、その指定された2の累乗数でダウンサンプリングするように設定する。**down 2**メッセージは2の因数によってダウンサンプリングすることを指定する（例えばサンプリング・レート44100Hzの環境では22050Hz）。新しくパッチに設定されたサンプリング・レートは、一連のDSPのネットワークが次にコンパイルされる時から有効になる。**down**メッセージは、一連のDSPのネットワークを強制的に再コンパイルすることはない。
- midinote** 第1インレット：**midinote**には1以上の数値が続き、読み込まれたパッチの1つ目のインスタンスの1つ目の**in**オブジェクトにデータを送る。そのパッチは、対応するノート・オフ・メッセージなしにノート・オン・メッセージを受け取る。**midinote**の後の1つ目の数値はノート・ナンバーとなり、ベロシティが続く。例として、**midinote 60 64**メッセージを2つのインスタンスを持つ**poly~**オブジェクトに送れば、1つ目のインスタンスをビジー状態としてマークする。続く**midinote 67 64**メッセージは2つ目のインスタンスへと向けられる。次に、**midinote 60 0**メッセージが**poly~**オブジェクトに送られると、それは1つ目のインスタンスへと送られる（**poly~**オブジェクトがノート・オン・メッセージを受け取ったインスタンスを記憶している）。同じように、**midinote 67 0**メッセージは2つ目のインスタンスへと向けられる。
- mute** 第1インレット：**mute**には数値と0か1が続き、指定された**poly~**オブジェクトに読み込まれたパッチのインスタンスのシグナル処理をオフにする。2つ目の数値が1の時は、パッチ・インスタンスの処理をオフに（ミュート）する。2つ目の数値が0の時はパッチ・インスタンスの処理をオンにする。**mute 0 1**メッセージはすべてのインスタンスをミュートし、**mute 0 0**メッセージはすべてのパッチのインスタンスの処理をオンにする。
- note** 第1インレット：**note**には1つのメッセージが続き、パッチ・インスタンスの内部の**thispoly~**オブジェクトに1を送ることで、ビジー状態としてマークされていない最初のインスタンスの1つ目の**in**オブジェクトにデータを送る。
- open** 第1インレット：**open**には1つの数値が続き、指定されたパッチのインスタンスを開く。ボイス（インスタンスと同義）の数（**voices**メッセ



ージか**poly~**オブジェクトのアーギュメントによって設定される)まで、任意のパッチのインスタンスの動作を見ることが出来る。このメッセージを用いて、動作しているパッチ個々の様子を監視することが出来る。アーギュメントを指定しなかった場合には、**open**メッセージは現在、ターゲットになっているインスタンス（以下の**target**メッセージを参照のこと）を開く。

- steal** 第1インレット：**steal**には0か1が続き、ボイス・スティーリングを切り替える。ボイス・スティーリングは**steal 1**メッセージを用いて設定でき、**poly~**オブジェクトはビジイー状態としてマークされているインスタンスへ、**note**メッセージや**midinote**メッセージによってデータを送る。インスタンスでの割り込み処理の仕方によって異なるが、この操作はクリック音を生じる場合がある。
- target** 第1インレット：**target**には1つの数値か、数値によって構成されたリストが続き、**poly~**オブジェクトに読み込まれたパッチへ、続く行動を切り替える。**target**メッセージは、**poly~**オブジェクトのインレットに受け取ったメッセージが、どのパッチ・インスタンスに受け取られるかを設定する。あるイベントを複数のターゲットを指定して送る場合は、**target**メッセージにリストのアーギュメントを与えれば良い。
- voices** 第1インレット：**voices**には1つの数値が続き、読み込まれたパッチのインスタンス（コピー）の数を指定する。パッチのインスタンスは必要に応じて読み込まれたり、削除されたりする。
- up** 第1インレット：**up**には2の累乗となる数値が続き、現在読み込まれているパッチ上での処理を、その指定された2の累乗数でアップサンプリングするように設定する。**up 2**メッセージは2の因数によってアップサンプリングすることを指定する（例えばサンプリング・レート44100Hzの環境では88100Hz）。新しくパッチに設定されたサンプリング・レートは、一連のDSPのネットワークが次にコンパイルされる時から有効になる。**up**メッセージは、一連のDSPのネットワークを強制的に再コンパイルしない。
- wclose** 第1インレット：**wclose**には1つの数値が続き、読み込まれたパッチの、その数値をインデックスとして識別されるインスタンスのウィンドウを閉じる。これは、**open**メッセージに対応するものである。
- vs** 第1インレット：**vs**には2から2048までの範囲の2の累乗の数値が続き、**poly~**オブジェクトに読み込まれたパッチ内でのシグナル・ベクター・サイズを指定する。新しくパッチに設定されたシグナル・ベクター・サイズは、一連のDSPのネットワークが次にコンパイルされる時から有効になる。**vs**メッセージは、一連のDSPのネットワークを強制的に再コンパイルしない。**vs 0**メッセージは、ベクター・サイズを指定し

ない。デフォルト値は現在設定されているシグナル・ベクター・サイズである。

## アーギュメント

**symbol** 必須：1つ目のアーギュメントは、必ずパッチの名前でなくてはならない。

注記：**patcher**オブジェクトとは異なり、**poly~**オブジェクトにパッチ名のアーギュメントを指定した時に、編集のためにサブ・パッチ・ウィンドウを自動的に開くことはない。このオブジェクトに含まれるパッチは、既に存在しているか、Max/MSPのサーチ・パス内で見つからなくてはならない。

**整数** オプション。パッチ名のアーギュメントの後に、読み込まれたパッチのインスタンスの数（**voices**メッセージで指定された数値と対応する）を指定する。デフォルト値は1である。**voices**メッセージを用いることで、有効なボイス（インスタンスと同義）の数は動的に変えることができる。

**local** オプション。**local**には0か1が続き、**poly~**オブジェクトに読み込まれたパッチのローカル・スケジューリングを切り替える。ローカル・スケジューリングとは、Maxのスケジューラーではなく、オーディオ処理が行われている間に動いている**poly~**オブジェクトが保持している、それ自身のスケジューラーということである。これは、複数のパッチで生成されるイベントの精度を向上させる。しかし、ローカルであってもなくても、オーディオ処理をオフにすればスケジューリングは発生しない。デフォルト値は0（**local 0**）である。ローカル・スケジューリングは、**poly~**オブジェクトにメッセージを送ることで変えることができない。ローカルなスケジューラーは、どんなパッチを読み込もうとも不変である。

**up** オプション。**up**には2の累乗となる数値が続き、現在読み込まれているパッチ上での処理をその指定された2の累乗数でアップサンプリングするように設定する。**up 2**メッセージは2の因数によってアップサンプリングすることを指定する（例えばサンプリング・レート44100Hzの環境では88100Hz）。**poly~**オブジェクトには**up**アーギュメントと**down**アーギュメントの両方を指定することが許されるが、**down**アーギュメントの方が**up**アーギュメントより高い優先順位を持つ。

**down** オプション。**down**には2の累乗となる数値が続き、現在読み込まれているパッチ上での処理を、その指定された2の累乗数でダウンサンプリングするように設定する。**down 2**メッセージは2の因数によってダウンサンプリングすることを指定する（例えばサンプリング・レート44100Hzの環境では22050Hz）。**poly~**オブジェクトには**up**アーギュメン

トとdownアーギュメントの両方を指定することが許されるが、downアーギュメントの方がupアーギュメントより高い優先順位を持つ。

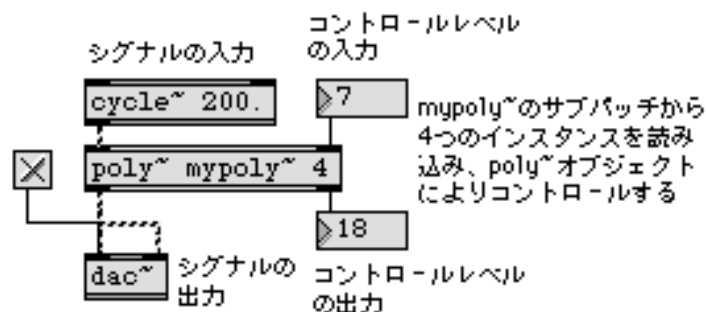
**args** オプション。**args**は、読み込まれたパッチの初期設定の#アーギュメント（例えば#1のような）として使用できる。これを使用する場合、**args**アーギュメントは使用されるアーギュメントの最後になければならない。**args**の後にあるすべてのものが、アーギュメントの値として扱われる。

## 出力

**任意** **poly~**オブジェクトのアウトレットの数は、読み込まれたパッチの内部の**out**オブジェクトと**out~**オブジェクトの、最も大きいアーギュメントの数を加算した値によって決定される。例えば、3のアーギュメントを持った**out**オブジェクトと2のアーギュメントを持った**out~**オブジェクトがあるとしたら、**poly~**オブジェクトは5つのアウトレットを持つことになる。**out~**オブジェクトに対応するシグナルのアウトレットは**poly~**オブジェクトの最も左に位置され、**out**オブジェクトに対応するイベントのアウトレットが続く。

シグナルは、それぞれのパッチ・インスタンス内の**out~**オブジェクトのインレットへと送られ、1つ以上のインスタンスがあればミックスされ、対応する**poly~**オブジェクトのアウトレットから出力される。

## 利用例



**poly~**オブジェクトは、複数のサブパッチのインスタンスを管理する

## 参照

|                  |                                              |
|------------------|----------------------------------------------|
| <b>in</b>        | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのメッセージの入力 |
| <b>in~</b>       | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのシグナルの入力  |
| <b>out</b>       | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのメッセージの出力 |
| <b>out~</b>      | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのシグナルの出力  |
| <b>patcher</b>   | パッチの中でサブ・パッチを作る                              |
| <b>thispoly~</b> | <b>poly~</b> オブジェクトのボイス割当てとミュートのコントロール       |

Tutorial 20

MIDI control : Sampler (英文PDF参照)

Tutorial 21

MIDI control : Using the **poly~** object (英文PDF参照)

## 入力

**シグナルまたは実数** 第1インレット：アーギュメントで指定した数値の範囲を越えるシグナルか実数を入力した場合、その範囲内に値が収まるよう折り返した値となる（例として、1より大きな値は1を越えた量だけ1より小さな値となり折り返したような波形となる。負の値についても同様な処理が行われる）。または、**pong~**オブジェクトの動作モード（**mode**メッセージの欄参照）によっては、範囲を越えた場合は範囲の逆側の境界から範囲内に出てくるような巻き取り処理を行う（例としては、1より大きな値は2を引いた形となり、-1より小さな値は2を加えた値となる）。

第2および第3インレット：折り返し処理を行う範囲の下側および上側境界値を指定する。アーギュメントがないか、または1つ指定している場合、**pong~**オブジェクトは2つのインレットを持つ。第2インレットでは折り返す範囲の上側境界値を設定する。この場合、下側境界値は0となる。

2つのアーギュメントを指定した場合、**pong~**オブジェクトは3つのインレットを持つ。第2インレットは下側境界値を設定し、第3インレットは上側境界値を設定する。アーギュメントがない場合、デフォルトのオブジェクト設定として、第3インレットは上側境界値を設定することになる。

低いほうの境界値が高いほうの値より大きな場合、それらの値は入れ替えたものとして動作する。

**mode** **mode**に続いて**0**か**1**を指定することで、**pong~**オブジェクトの動作モードを設定する。

**mode 0**はオブジェクトをシグナルの折り返し処理の動作に設定する。1より大きな値は1を越えた量だけ1小さい値となり、負の数についても同様の動作となる。これはデフォルトの動作モードである。

**mode 1**はオブジェクトをシグナルの巻き取り処理の動作に設定する。

## アーギュメント

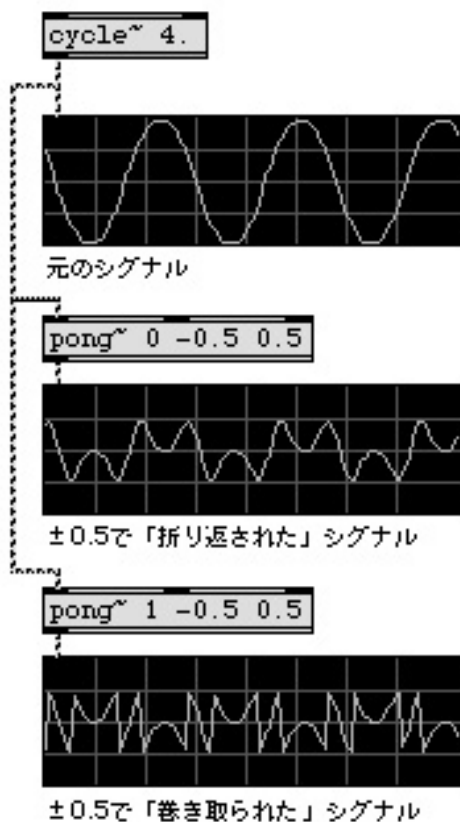
**整数** オプション。アーギュメントで**pong~**の動作モードを設定することができる。**0**はシグナルの折り返し処理（デフォルト）のモードで、**1**は巻き取り処理のモードとなる（上記**mode**メッセージの欄を参照のこと）。

**実数** オプション。pong~の処理を行うための下側および上側境界値をアーギュメントとして設定することができる。1つの実数しか与えられない場合、それは折り返しの上側境界値を指定することになり、下側境界値は0.0に設定される。アーギュメントで実数を2つ与えた場合、1つめの実数は下側境界値を、2つめは上側境界値を指定することになる。

## 出力

**シグナル** 折り返し処理されたシグナルまたは実数値が出力される。

## 利用例



pong~は、設定された上側および下側境界値によって、折り返し処理、または巻き取り処理を行い、シグナルを歪ませる

## 参照

### phaserwrap~

シグナルを $-\pi(-3.14159)$ から $\pi(3.14159)$ に変更する



**pow~**オブジェクトは、底（第2インレットに設定する）を累指数（第1インレットに設定する）で累乗計算を行う。どちらのインレットも、シグナル、実数、整数を受け取ることができる。

## 入力

**シグナル** 第1インレット：累指数を設定する。

第2インレット：底を設定する。

**実数または整数** 第1インレット：累指数を設定する。第1インレットにシグナルが接続されている時は、第1インレットからの数値は無視される。

第2インレット：底を設定する。第2インレットにシグナルが接続されている時は、第2インレットからの数値は無視される。

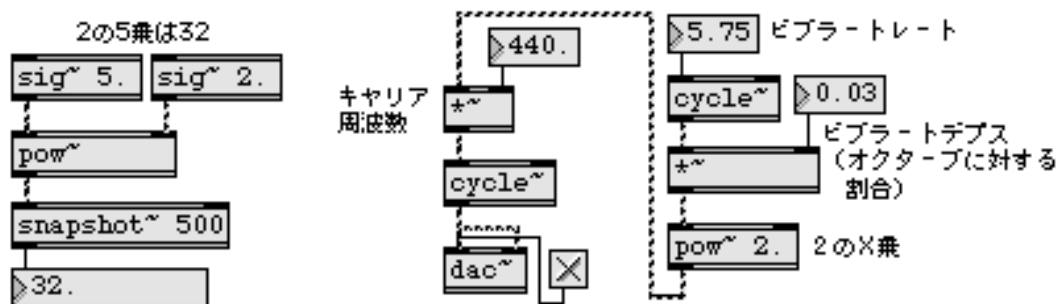
## アーギュメント

**実数または整数** オプション。底を設定する。デフォルト値は0である。第1インレットにシグナルが接続されている時は、アーギュメントは無視される。

## 出力

**シグナル** 底（第2インレットからの入力）を累指数（第1インレットからの入力）で累乗した値が出力される。

## 利用例



数式を計算してxy座標を対数的、または指数的な比率に変換する

## 参照

**log~**  
**curve~**

シグナルの対数計算をする  
指数曲線、対数曲線のランプ・ジェネレーター



**pp**オブジェクト (plug-in parameterの略語) はプラグイン・パラメーターを定義する。いくつかのオプションのアーギュメントを持ち、パラメーターの最小値と最大値を設定する、表示からパラメーターを隠す、対応する卵形スライダーの色を設定するなどができる。**pp**オブジェクトの出力は、格納したパラメーターに従ってコントロールするオブジェクトに入力する。プラグインがMaxパッチ・インターフェースを使用するのであれば、パラメーターの値を変更するインターフェース・オブジェクトを、**pp**オブジェクトのインレットに接続する必要がある。**pp**オブジェクトは、様々な場合に新しいパラメーターの値を出力する。例えば、卵形スライダーを動かした時、新しいエフェクト・プログラムに切り替えた時、ホスト・ミキサーがプラグインのパラメーターの変更をオートメーションで操作している時などである。

内部的には、**pp**オブジェクトとランタイム・プラグイン環境は0から1.0の間で値を記憶する。**pp**オブジェクトに最小値と最大値のためのオプションのアーギュメントを指定すると、任意の範囲の値を記憶し、出力することができ、**pp**オブジェクトは希望する値と内部表現の値とを変換するようになる。何らかの理由で内部の0-1.0表現について知りたい場合には、オブジェクトの第2アウトレットから取得することができる。内部の0-1.0表現に基づく値を出力したい場合には、**rawfloat**メッセージを使う。

## 入力

**bang** 0から1.0の間で内部 (スケーリングされていない) 形式で、パラメーターの現在の値をオブジェクトの第2アウトレットから出力し、次にオブジェクトの第1アウトレットからオブジェクトの最小値と最大値でスケーリングされた値を出力する。

**実数または整数** 第1インレット: パラメーターの現在の値を設定し、次に新しい値を、**bang**メッセージで説明したように、第2アウトレットと第1アウトレットから出力する。入力される数値は、オブジェクトの最小値と最大値の間に制限される。

**実数または整数** 第2インレット: パラメーターの現在の値を出力することなく設定する。入力される数値は、オブジェクトの最小値と最大値の間に制限される。

**open** Objectメニューから**Get Info...**を選んだ場合と同じ働きをする。

**text** 1つのシンボルを伴う**text**は、ユーザーがパラメーターに対応する卵形スライダーの上にマウスを動かした時に、プラグイン・エディット・ウィンドウのParametersビューに表示されるテキストを設定することができる。

**rawfloat** 0から1.0までの数値を伴う**rawfloat**は、オブジェクトの最小値と最大値によってスケーリングすることなく、現在のパラメーターの値をその数値に設定する。その値は、**bang**メッセージで説明したように、オブジェクトの第2と第1アウトレットから出力される。

(Get Info...) Objectメニューから**Get Info...**を選ぶと、ユーザーがパラメーターに対応する卵形スライダーの上にマウスを動かした時に、プラグイン・エディット・ウィンドウのParametersビューに表示されるパラメーターの説明文を編集するためのインスペクターを開く。

## インスペクター

**pp**オブジェクトの振る舞いは、インスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている場合は、任意の**pp**オブジェクトを選択することで、フローティング・ウィンドウに**pp**のインスペクターを表示することができる。**pp**オブジェクトを選択して、Objectメニューから**Get Info...**を選ぶことでも、インスペクターを表示することができる。

パラメーターの説明を*Describe Parameter*テキスト・エリアの中に入力する。

**Revert**ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

**pp**オブジェクトはいくつかのアーギュメントを指定することができる。これらをアーギュメントに現れるべき順序で説明する。

**整数** 必須。第1のアーギュメントはパラメーター番号を設定する。最初のパラメーター番号は1である。パラメーター番号は連続しているはずであり（しかしそうである必要はない）、2つの**pp**オブジェクトは同じパラメーター番号を持つことはないはずである。同じパラメーター番号を持つ**pp**オブジェクトが存在すると、ランタイム・プラグイン環境のMessagesビューにエラーが表示される。

**hidden** オプション。**hidden**をアーギュメントとして指定すると、そのパラメーターはプラグイン・エディット・ウィンドウで卵形スライダーとして表示されることはなく、**plugmod**オブジェクトによって生成されるポップアップ・メニューにも現れない。

**fixed** オプション。**fixed**をアーギュメントとして指定すると、パラメーターはRandomizeとEvolveコマンドに影響されなくなる。これらのコマンドは、プラグイン・エディット・ウィンドウの中でユーザーが

commandキーを押しながらインターフェースの中をクリックした時に現れるポップアップ・メニューにある。このオプションは、ランダムイズが望ましい結果をもたらさないゲイン・パラメーターに適している。

**c2-c5** オプション。**c2**、**c3**、**c4**、**c5**をアーギュメントとして指定すると、卵形スライダーのカラーは通常のパープルとは異なる色に設定される。現在の**c2**はワイルド・チェリー、**c3**はターコイズ、**c4**はハーベスト・ゴールド、そして**c5**はピースフル・オレンジである。

**シンボル** オプション。任意のオプションのキーワードの後のシンボルは、パラメーターの名前を設定する。この名前はParametersビューのName欄と、**plugmod**オブジェクトによって生成されるポップアップ・メニューに現れる。

**実数または整数** オプション。パラメーターの名前の後に続く1つの数値は、パラメーターの最小値を設定する。最小値と最大値は**pp**オブジェクトに入出力する値の範囲を設定する。最小値のタイプはオブジェクトが受け入れて出力するパラメーターの値のタイプを決定することになる。最小値が整数であれば、パラメーターは整数として解釈されて出力される。最小値が実数であれば、パラメーターは実数として解釈されて出力される。

**実数または整数** オプション。最小値の後に続く1つの数値は、パラメーターの最大値を設定する。最小値と最大値は**pp**オブジェクトへの入出力だけでなく、Parametersビューで表示される値の範囲も決定する。

**シンボル** オプション。最小値と最大値に続くシンボルは、パラメーターの単位を表示するために用いられるラベルを設定する。例えば、周波数ならHz、振幅ならdB、そしてミリ秒ならms、などを指定する。

**choices** オプション。最小値と最大値の後に**choices**が指定されると、それに続くシンボルのアーギュメントは、オブジェクトを個別に設定するリストとして取り扱われ、そのシンボルそのものがParametersビューなどで表示される。例えば**pp 1 Mode 0 3 choices Thin Medium Fat**はパラメーターの範囲を3つに分割し、0（0.33以下のすべて）はThinに対応し、0.5（および0.33から0.67の間も）はMediumに対応し、1（および0.67から1.0の間も）はFatに対応する。パラメーターの実際の値ではなく、該当する名前だけがParametersビューに表示される。

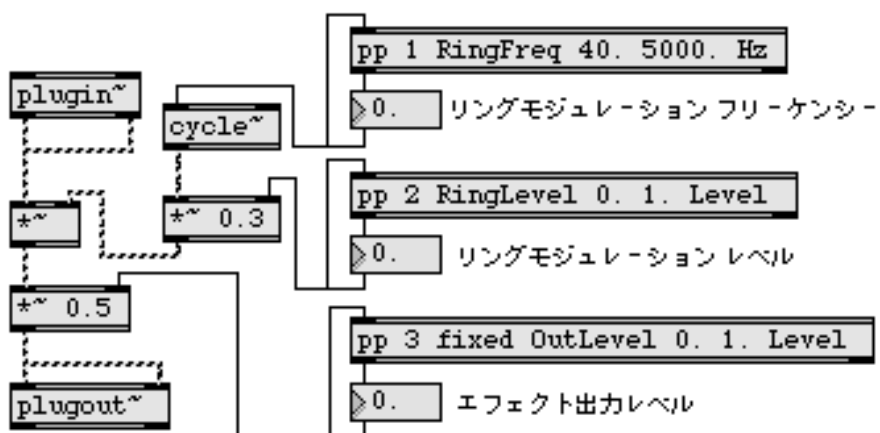
**dB** オプション。**choices**をアーギュメントとして指定しなければ、dBというシンボルによって、1.0が0dBで0.0がマイナス無限大dBに対応するデシベル値として、パラメーターの値を表示することを指定する。

## 出力

**整数または実数** 第1アウトレット：ランタイム環境の中で変化が起きた時、または **bang**、**整数**、**実数**、**rawfloat**メッセージをオブジェクトのインレットで受け取った時に、パラメーターのスケーリングされた値が出力される。パラメーターの値はランタイム環境の中で、次のような方法で変更することができる。すなわち、ユーザーが卵形スライダーを動かす、パラメーターがホスト・ミキサーによってオートメーション操作される、ユーザーがホスト・ミキサーでプラグインのために新しいエフェクト・プログラムを選択する、といった場合である。

**実数** 第2アウトレット：ランタイム環境がパラメーターの値を変えた時、または **bang**、**整数**、**実数**、**rawfloat**メッセージをオブジェクトのインレットで受け取った時に、スケーリングされていないパラメーターの値が出力される。ユーザーに対して表示する値ではなく、プラグインの計算法として違う値を使いたいのであれば、この値を使うことになるかもしれない。

## 利用例



## 参照

**plugmultiparam**  
**plugstore**

複数のプラグイン・パラメーターを設定する  
複数のプラグインのパラメーターの値を格納する

## 入力

**bang** モード・パラメーター（0から3まで）の現在の値をオブジェクトの第2アウトレットから出力し、テンポ・パラメーターの現在の値をオブジェクトの第1アウトレットから出力する。

**整数** 第1インレット：テンポ・パラメーターの現在の値を設定し、新しい値を第1アウトレットから出力する。入ってくる数値はオブジェクトの最小値と最大値の間に制約される。

第2インレット：モード・パラメーターの現在の値を設定し、新しい値を第2アウトレットから出力する。数値は0から3までの間に制限される。モードの値は次の通り。

**値**      **説明**

0      フリー・モード。このパラメーターに関連付けられた卵形スライダー表示があれば、無効にされる。その他のパラメーターはミリ秒またはヘルツ単位の“テンポ”を設定すると判断される。

1      ホスト・モード。このパラメーターに関連付けられた卵形スライダー表示があれば、それは有効になるが変更することはできない。代わりに、そのテンポはホストによって設定されスライダーによって、ただ単に表示される。パッチは何らかの方法（恐らく **plugsync~** または **plugphasor~** オブジェクト）でホストへの同期を有効にしなければならない。

2      PluggoSyncモード。このモードはホスト・モードと似たような働きをし、卵形スライダーは有効だが、ユーザーが変更することはできない。代わりに、テンポはホストによって設定されスライダーによって、ただ単に表示される。パッチは何らかの方法でPluggoSyncへの同期を有効にしなければならない。

3      ユーザー定義テンポ（UDT）モード。このモードでは同期はなく、ユーザーはテンポ・スライダーを任意の求められる値に変えることができる。パッチはこの値をある種の時間ベースのふるまいを計算するのに用いなければならない。

**set** 第2インレット：数値を伴う **set** は、同期・モード・パラメーターをその数値に設定するが、同期モードとテンポを出力しない。

**rawfloat** 第1インレット：0から1までの数値を伴う **rawfloat** は、その数値を最小値から最大値までの間にスケーリングした値にテンポを設定する。例



例えば、最小テンポが100で最大テンポが200だとすると、**rawfloat 0.5**というメッセージは、テンポを150に設定する。

第2インレット：0から1までの数値を伴う**rawfloat**は、同期モード・パラメーターをその数値に3をかけて切り捨てた値に基づく値に設定する。0.33よりも小さい数値は同期モードを0（フリー）に、0.33から0.66までの数値はホストに、0.67以上で1より小さい値はPluggoSyncに、1に等しい数値はユーザー定義テンポに設定する。

**rawlist** 2つの数値を伴う**rawlist**は、最初の数値付きの**rawfloat**メッセージを第1インレットに、2番目の数値付きの**rawfloat**メッセージを第2インレットに入力したことと同じである。

**(Get Info...)** Objectメニューから**Get Info...**を選択すると、ユーザーがカーソルを卵形スライダーの上に動かした時に、そのパラメーターに対応するプラグイン・エディット・ウィンドウの**Parameters**ビューに表示されるパラメーターの説明を編集するためのインスペクターを開く。

## インスペクター

**pptempo**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**pptempo**オブジェクトを選択することで、**pptempo**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**pptempo**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでもインスペクターを表示することができる。

*Describe Parameter*（パラメーターの説明）テキスト・エリアでテキスト入力することで、パラメーターの説明を指定する。

**Revert**ボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

**整数** 必須。1以上の数値はテンポ・パラメーターのパラメーター・インデックスを設定する。

**整数** 必須。1以上の数値は同期モード・パラメーターのパラメーター・インデックスを設定する。

- hidden** オプション。**hidden**がアーギュメントとして現れると、パラメーターがプラグイン・エディット・ウィンドウで卵形スライダーとして現れることはなくなり、**plugmod**オブジェクトによって生成されたポップアップ・メニューにも現れなくなる。
- fixed** オプション。**fixed**がアーギュメントとして現れると、パラメーターはユーザーがインターフェースの中でcommandキーを押しながらクリックした時に、プラグイン・エディット・ウィンドウに現れるパラメーター・ポップアップ・メニューのRandomizeとEvolveコマンドに影響されなくなる。
- c2からc5** **c2**、**c3**、**c4**または**c5**がアーギュメントとして現れると、卵形スライダーの色は、通常のパープル以外の何か他のものに設定される。現在は**c2**はワイルド・チェリー、**c3**はターコイズ、**c4**はハーベスト・ゴールド、そして**c5**はピースフル・オレンジである。
- シンボル** オプション。任意のオプションのキーワードの後に続くシンボルは、テンポ・パラメーターを名付ける。この名前はParametersビューのNameカラムと**plugmod**オブジェクトによって生成されたポップアップ・メニューに現れる。同期モード・パラメーターの名前は、**mode**を伴うテンポ・パラメーターの名前になる。デフォルトのパラメーター名はParam*N*で、*N*は**pptempo**への最初のアーギュメントによってテンポ・パラメーターに割り当てられたインデックスである。
- 実数または整数** オプション。パラメーター名に続く数値はパラメーターの最小値を設定する。最小値と最大値は**pptempo**オブジェクトの第1インレットとアウトレットから入出力される値と同様に、プラグイン・エディット・ウィンドウのParametersビューに表示される値の範囲を決定する。最小値のタイプはオブジェクトが受け入れたり、出力したりするパラメーターの値のタイプを決定する。最小値が整数であれば、パラメーターが整数として解釈され整数として出力される。最小値が実数であれば、パラメーターは実数として解釈され出力される。
- 実数または整数** オプション。最小値の後の数値はパラメーターの最大値を設定する。最小値と最大値は**pptempo**オブジェクトの第1インレットと第1アウトレットに入出力される値と同様に、プラグイン・エディット・ウィンドウのParametersビューに表示される値の範囲を決定する。
- (Get Info...)** Objectメニューから**Get Info...**を選択すると、ユーザーがカーソルを卵形スライダーの上に動かした時に、そのパラメーターに対応するプラグイン・エディット・ウィンドウのParametersビューに表示されたパラメーターの説明を編集するためにインスペクターを開く。

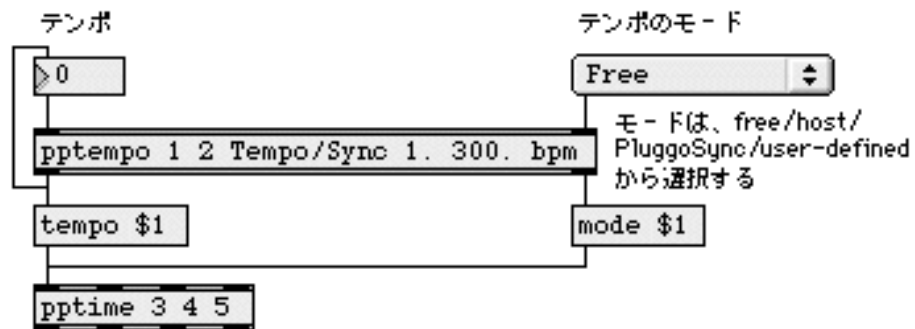


## 出力

**実数または整数** 第1アウトレット：実行環境の中で変化した時や、**bang**、**整数**、**実数**または**rawfloat**をオブジェクトのインレットで受け取った時に、テンポ・パラメーターのスケールされた値を出力する。パラメーターの値は次のような方法で実行環境で変更される。これは、ユーザーが卵形スライダーを動かした、ホスト・ミキサーによってパラメーターがオートメーションされた、またはユーザーがホスト・ミキサーでそのプラグインのために新しいエフェクト・プログラムを選択する場合である。

第2アウトレット：パラメーターが実行環境で変化した時や、**整数**、**実数**または**rawfloat**メッセージをオブジェクトの第2インレットで受け取った時、または**bang**メッセージをオブジェクトのインレットで受け取った時に、0から3までの同期モード・パラメーターの値を出力する。モードは入力の項で説明されている。

## 利用例



pptempo は pptime にテンポや同期の情報を提供する

## 参照

pp  
pptime

プラグイン・パラメーターを定義する  
時間ベースのプラグイン・パラメーターを設定する

**pptime**オブジェクトは、ホスト・シーケンサー・アプリケーションとの同期を提供するプラグインで使用するための、時間ベースのプラグイン・パラメーターを設定する。**pp**オブジェクトのように、**pptime**は多数のオプションのアーギュメントによってパラメーターを設定し、一般的なプラグイン・インターフェースを用いる場合のアピランスのコントロールを可能にする。

**pptime**はホスト同期の4つのモードをサポートする。オブジェクトの機能はオペレーションのモードによって変化する。Free（フリー）モードでは、ms/Hzパラメーターのために**pptime**は**pp**のように動作し、第1インレットとアウトレットを使う。Host Sync（ホスト同期）モードとPluggo Sync（Pluggo同期）モードでは、卵形スライダー表示は、より小さなスライダーと単位ポップアップ・メニューに変化する。スライダーかメニューに変化が起きると、ビート値の出力（最も右のアウトレット）は、**rate~**オブジェクトに与えることができるような値を生成する。User-Defined Tempo（ユーザー定義テンポ）モードは、**tempo**メッセージ（このために**pptempo**を用いることができる）で、**pptime**に与えられるテンポの値を要求する。次に**pptime**は現在のテンポ、単位乗数と単位に基づいてms/Hzでの値を計算し、その値を第1アウトレットから出力する。

## 入力

**実数または整数** 第1インレット：ms/Hzの値のためのパラメーター・インデックスを設定する。

第2インレット：単位乗数の値を設定する。値の範囲は0.0から15.0までである。

第3インレット：単位インデックスを設定する。単位インデックスは0から18までの実数または整数で表現される、それぞれの数値は音楽的な再分割の単位を表す。

単位インデックスは以下のように定義されている。

| 単位インデックス | 音符    | 値           |
|----------|-------|-------------|
| 0        | 1     |             |
| 1        | 1/2   |             |
| 2        | 1/2.  | （付点二分音符）    |
| 3        | 1/2t  | （二分音符の三連符）  |
| 4        | 1/4   |             |
| 5        | 1/4.  | （付点四分音符）    |
| 6        | 1/4t  | （四分音符の三連符）  |
| 7        | 1/8   |             |
| 8        | 1/8.  | （付点八分音符）    |
| 9        | 1/8t  | （八分音符の三連符）  |
| 10       | 1/16  |             |
| 11       | 1/16. | （付点十六分音符）   |
| 12       | 1/16t | （十六分音符の三連符） |

|    |                    |
|----|--------------------|
| 13 | 1/32               |
| 14 | 1/32. (付点三十二分音符)   |
| 15 | 1/32 (三十二分音符の三連符)  |
| 16 | 1/64               |
| 17 | 1/64. (付点六十四分音符)   |
| 18 | 1/64t (六十四分音符の三連符) |

第4インレット：単位の値の入力を設定する。

- bang** パラメーターの現在の値をオブジェクトの第1アウトレットから出力する。
- mode** 第1インレット：0から3までの数値を伴う**mode**は、ホスト同期モードを指定する。ホスト同期モードは次のように定義されている。0=フリー、1=ホスト同期、2=Pluggo同期、3=ユーザー定義テンポ。デフォルト値は1 (フリー・モード) である。
- open** Object メニューから**Get Info...**を選択するのと同じ。
- rawfloat** 0から1.0までの数値を伴う**rawfloat**は、オブジェクトの最小値と最大値によってスケールリングすることなく、現在のパラメーターの値をその数値に設定する。値はオブジェクトの第1および第2アウトレットから、**bang**メッセージのところで前述したように、出力される。
- timesig** 第1インレット：2つの数値を伴う**timesig**は、拍子記号を指定するのに用いられる。(分子と分母からなる) 拍子記号は、同期モードで拍数を計算するために、そしてユーザー定義テンポ・モードのms/Hzの値を計算するために用いられる。このリストは**plugsync~**オブジェクトの出力から与えることもできる。デフォルトは4/4 (**timesig 4 4**) である。
- tempo** 第1インレット：**pptime**オブジェクトがユーザー定義テンポ・モードの時、1つの数値を伴う**tempo**は現在のテンポを指定し、そのテンポに関連したms/Hzでの値が第1アウトレットから出力される。
- (Get Info...)** Objectメニューから**Get Info...**を選択すると、対応するパラメーターの上にユーザーがマウス・カーソルを動かした時に、プラグイン・エディット・ウィンドウのParametersビューに表示されるパラメーターの説明を編集するためのインスペクター・ウィンドウを開く。このコマンドは実行プラグイン環境では使用できない。

## インスペクター

**pptime**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**pptime**オブジェクトを選択することで、**pptime**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができ

る。**pptime**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでもインスペクターを表示することができる。

*Describe Parameter* (パラメーターの説明) テキスト・エリアでテキスト入力することで、パラメーターの説明を指定する。

*Revert*ボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Editメニューの**Undo Inspector Changes**を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

**pptime**オブジェクトには3つの必須アーギュメントと、数多くのオプションのアーギュメントがある。以下は、それらが出現すべき順序で並べている。

**実数** 必須。3つの必須な実数アーギュメントはms/Hzの値、単位乗数、そして単位インデックスのためのパラメーター・インデックスである。

**hidden** オプション。**hidden**がアーギュメントとして現れると、そのパラメーターがプラグイン・エディット・ウィンドウで卵形スライダーを与えられず、**plugmod**オブジェクトによって生成されるポップアップ・メニューに現れることもなくなる。

**fixed** オプション。**fixed**がアーギュメントとして現れると、そのパラメーターはユーザーがcommandキーを押しながらインターフェースをクリックした時に表示されるパラメーター・ポップアップ・メニューの中のRandomizeとEvolveに影響されなくなる。これはランダム化によって困った結果を起こしがちなゲイン・パラメーターに適している。

**c2-c4** オプション。**c2**、**c3**または**c4**がアーギュメントとして現れると、卵形スライダーの色は通常のパープル以外の何か別の色に設定される。現在は**c2**はワイルド・チェリー、**c3**はターコイズ、そして**c4**はハーベスト・ゴールドである。

**シンボル** オプション。任意のオプションのキーワードに続くシンボルはパラメーターの名前を付ける。その名前はParametersビューのNameカラムと、**plugmod**オブジェクトによって生成されるポップアップ・メニューに現れる。

**実数または整数** オプション。パラメーターの名前の後の数値はパラメーターの最小値を設定する。最小値と最大値は、**pptime**オブジェクトに入出力されると同様に、Parametersビューに表示される値の範囲を決定する。最小値のタイプは、オブジェクトが受け入れて出力するパラメーター値のタイプを決定する。最小値が整数であれば、パラメーターは整数とし

て解釈され出力される。最小値が実数であれば、パラメーターは実数として解釈され出力される。

**実数または整数** オプション。最小値に続く数値はパラメーターの最大値を設定する。最小値と最大値は、**pptime**オブジェクトに入出力されると同様に、Parametersビューに表示される値の範囲を決定する。

**シンボル** オプション。最小値と最大値に続くシンボルはパラメーターの単位を表示するラベルを設定する。例えばHzを周波数に、dBを振幅に、msをミリ秒にする。

## 出力

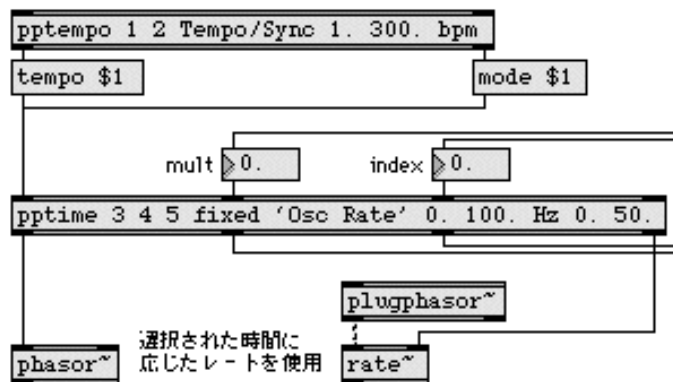
**整数または実数** 第1アウトレット：パラメーターのスケーリングされた値は、実行環境で変化した時、または**bang**、**整数**、**実数**または**rawfloat**メッセージをオブジェクトのインレットで受け取った時に出力される。実行環境ではパラメーター値は次のような方法で変えることができる。すなわち、ユーザーが卵形スライダーを動かす、ホスト・ミキサーによってパラメーターが自動制御される、またはユーザーがホスト・ミキサーでプラグインのために新しいエフェクト・プログラムを選択する、といった方法である。

第2アウトレット：単位乗数の値。値は0.0から15.0の範囲である。

第3アウトレット：単位インデックス。単位インデックスは0から18までの実数または整数で表される。

第4アウトレット：拍数の値の出力。

## 利用例



pptimeを用いて、拍同期や時間同期のパラメーターをコントロールする

## 参照

pp  
pptempo

プラグイン・パラメーターを定義する  
プラグインのテンポと同期パラメーターを設定する

## 入力

**シグナルまたは実数** スムース処理を施すシグナルか実数を入力する。新たな値が入力されると、その値に向かって指定したサンプル数の時間をかけて、リニアに徐々に値を変化させることで、スムーズ処理が行われる。

**ramp** 第1インレット：**ramp**に続いて数値を指定する。この数値はスムーズ処理にかけるサンプル数を表す。オブジェクトは入力値が変化する毎に、サンプル数で指定した時間をかけて、入力値へ向かってリニアに値を変化させる。デフォルト値は0である。

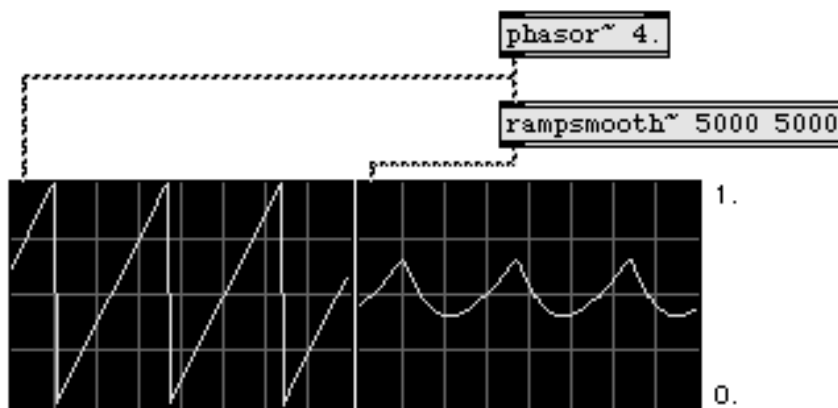
**rampdown** 第1インレット：**rampdown**に続いて数値を入力する。この数値はスムーズ処理にかけるサンプル数を表すが、新しい入力値が、その時点の値よりも小さい場合にスムーズ処理を行う。

**rampup** 第1インレット：**rampup**の文字に続いて数値を入力する。この数値はスムーズ処理にかけるサンプル数を表すが、新たな入力値が、その時点の値よりも大きい場合にスムーズ処理を行う。

## アーギュメント

**整数** オプション。入力値の増加方向と減少方向の、それぞれのスムーズ処理にかけるサンプル数を、2つの整数の組として指定する。

## 利用例



シグナルが5000サンプルに渡ってリニアに円滑化

**rampsmooth~**は、入力シグナルに対するリニアなスムーズ処理を行う

## 参照

**slide~**

シグナルへの対数的なフィルター処理



# rand~

バンド幅が制限された  
ランダムなシグナルを発生する

## 入力

**シグナル** -1から1までの新しいランダムな数値で周波数が生成される。**rand~**オブジェクトは指定された割合で、ランダムな値を選択してリニアに補間する。

**実数または整数** シグナルと同じ。シグナルがインレットに入力されている場合は、実数または整数メッセージは無視される。

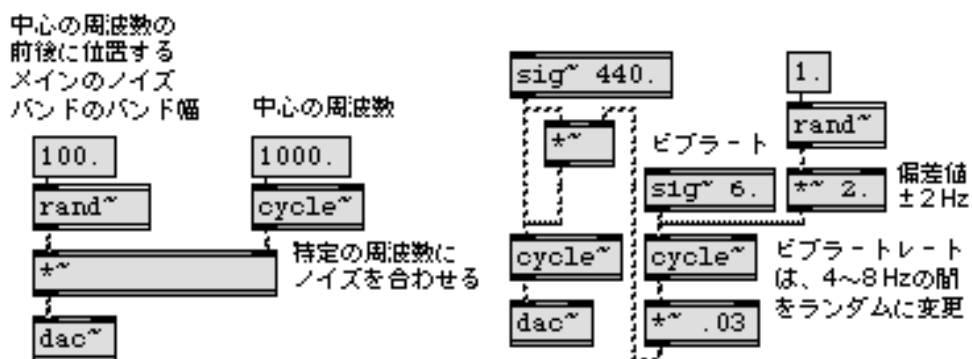
## アーギュメント

**実数、または整数** 周波数の初期値を設定する。デフォルト値は0である。シグナルがインレットに入力されている場合は、アーギュメントは無視される。

## 出力

**シグナル** -1から1の範囲にあるランダムな値である線分から成るシグナル。ランダムな値は、入力によって指定される周波数に従って生じる。

## 利用例



**rand~** オブジェクトを粗いバンド・リミテッド・ノイズを生成するのに使用するまたは、ランダムなバリエーションを生成するコントロール・シグナルとして使用する

## 参照

**line~** シグナル・ランプ・ジェネレーター  
**noise~** ホワイト・ノイズ・オシレーター  
**pink~** ピンク・ノイズを発生させるジェネレータ



## 入力

**シグナル** 第1インレット：**phasor~**オブジェクトからの入力シグナル。**rate~**オブジェクトは乗数の値によって、**phasor~**オブジェクトからの入力シグナルをタイム・スケールする。乗数はアーギュメントか**rate~**オブジェクトの第2インレットに入力される実数によって指定される。

**実数** 第1インレット：**rate~**オブジェクトから出力されるシグナルの位相の値を設定する。

第2インレット：シグナルの乗数値は**phasor~**オブジェクトのシグナルの入力をスケールするのに使用される。1.0よりも小さい実数値は、いくつかの勾配をフェーズ・サイクルごとに生成する。1.0よりも大きな数値は、より少ない勾配を生成する。これは単独で**phasor~**オブジェクトを参照する複数のプロセスを同期させるのに役立ち、それらの比率の関係性を保つことができる。

**goto** 第1インレット：実数が伴う**goto**によって、**rate~**オブジェクトは直ちに指定した値に設定される。オプションの2番目のアーギュメントは、設定した値に変化する時間を指定する（例えば**goto 1.0 .5**は、**phasor~**オブジェクトの入力シグナルの勾配の中間地点で1.0の値を出力する）。

**reset** 第1インレット：**reset**は、次のリセット時に入力に出力を一致させる。**goto 0. 0.**というメッセージと等しい。

**sync** 第1インレット：0から2の数値、または**cycle**、**lock**、**off**が伴う**sync**は、**rate~**オブジェクトのシンク・モードを設定する。シンク・モードは**rate~**オブジェクトが入力シグナルによる位相に従うか否かと、同期のために使われる方法を決定する。**rate~**オブジェクトの出力が“in phase”の時、入力シグナルと出力シグナルはそれらの周期の最小公倍数において整列する（すなわち、周期的に同時に0を通過し、新しいサイクルを開始する）。シグナルが同じ位相で、新たな乗数値が入力される場合は、**rate~**オブジェクトは出力の勾配の周波数をそれに応じて変化させる。しかし、乗数値による変化は2つのシグナルが同位相ではなくなるかもしれない。**rate~**オブジェクトはシンク・モードに依存する3つの方法の中から1つの状況を取り扱う。シンク・モードは以下のようなになる。

**モード**                      **説明**

**cycle**                      **sync 0**、または**sync cycle**というメッセージは、**rate~**オブジェクトを**cycle**モードに設定する（デフォルトのモード）。**cycle**モードの時、**rate~**オブジェクトは現在の周期の終わりまで出力の位相を変化させな

い。入力された勾配がそのピークに到着し、0から始まる時、**rate~**オブジェクトは出力される勾配を直ちにリスタートし、出力シグナルは不連続になるが、ただちに位相は同期される。**lock sync sync 1**または**sync lock**というメッセージは、**rate~**オブジェクトをlockモードに設定する。lockモードの時、**rate~**オブジェクトは新たな乗数を受け取る時は、いつでも同期する。**rate~**オブジェクトは直ちに新たな乗数値に従って“in phase”になるよう勾配の位置を計算して、その位置にジャンプする。

**off**

**sync 2**または**sync off**というメッセージは、**rate~**オブジェクトのシンク・モードを停止する。このモードの時、**rate~**オブジェクトは位相の違いにまったく対応しない。新しい乗数が入力された時、**rate~**オブジェクトは出力の勾配のスピードを調節しするだけで、中断することなく処理を続ける。このモードは勾配シグナルの不連続なジャンプを引き起こさないなので、位相が重要でない場合は役に立つかもしれない。

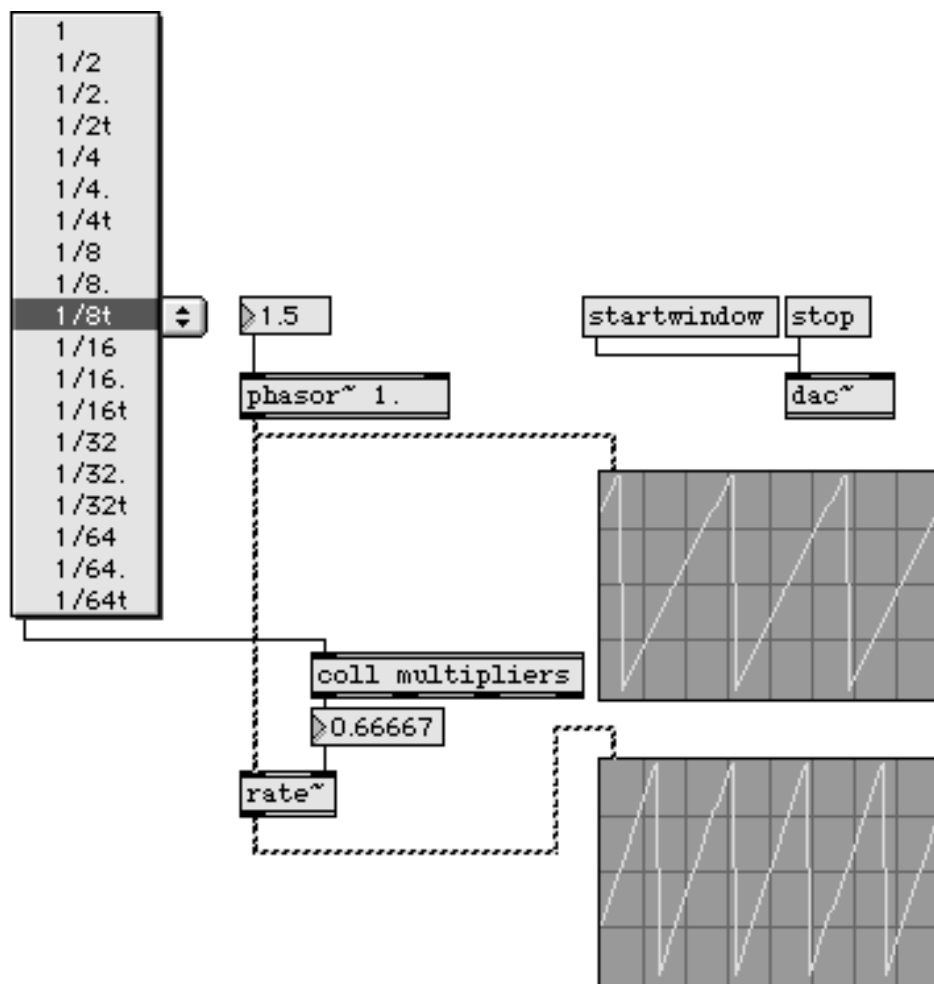
## アーギュメント

**実数** オプション。乗数値は出力シグナルをスケールするのに使われる。

## 出力

**シグナル** タイム・スケールされたフェーズ・シグナル。

## 利用例



rate~ オブジェクトを使用して同期するウェーブ・フォーム、またはコントロール・ソースを生成する

## 参照

- buffer~ オーディオ・サンプルの格納
- phasor~ ノコギリ波の生成
- record~ サウンドをbuffer~のサンプル・メモリへ録音する

## 入力

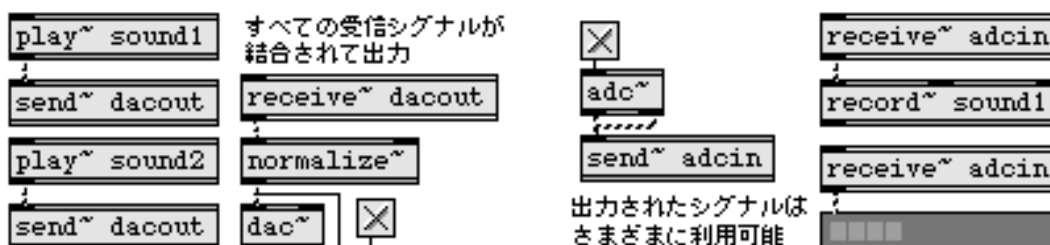
**シグナル** **receive~**オブジェクトは同じ名前を持つすべての**send~**オブジェクトからのシグナルを受信する。すべてのシグナルを加えた合計がアウトレットから出力される。現在の名前と同じものを持つ**send~**オブジェクトが存在しない場合は、**receive~**オブジェクトの出力は**0**である。**send~**オブジェクトは対応する**receive~**オブジェクトと同じパッチの中に存在する必要はない。

**set** シンボルが後に続く**set**メッセージは、**receive~**オブジェクトの名前を変更し、そのシンボルを持つ違う**send~**オブジェクトと接続することになる。同じ名前を持つ**send~**オブジェクトが存在しない場合、**receive~**オブジェクトの出力は**0**である。アーギュメントシンボル必須。**receive~**オブジェクトの名前を設定する。

## 出力

**シグナル** **receive~**オブジェクトと同じ名前を持つすべての**send~**オブジェクトに入力された、すべてのシグナルを合成したシグナル。

## 利用例



シグナルはすべてのロードされているパッチからパッチ・コードなしで受信される

## 参照

**send~** パッチ・コードなしにシグナルを送信する

Tutorial 4 Fundamentals : Routing signals (英文PDF参照)

## 入力

**シグナル** 第1インレット：録音がオンの時、入力シグナルは現在のサンプリング・レートで**buffer~**のサンプル・メモリに録音される。

第2インレット以降のインレット：複数の入力チャンネルを持っている場合、追加チャンネルとして**buffer~**へ録音する。

**整数** 第1インレット：非ゼロ値を入力することで録音を開始し、0で録音を停止する。録音データは追記モードがオンでないと記録開始点（下記参照）から記録される。

**整数または実数** 右端のインレットの1つ左側のインレット：録音データを**buffer~**へ書き込む場合の記録の開始点をミリ秒単位で指定する。デフォルトでは開始点は0（**buffer~**が確保したメモリの先頭）となる。

右端のインレット：録音の記録終了点を指定する。デフォルトでは終了点は**buffer~**オブジェクトが確保したメモリの終わりの地点となる。

**append** **append**に続いて非ゼロ値を指定することで録音の追記モードをオンにする。このモードでは、録音が始まると前回の録音で終了した記録点の続きから録音データの記録を行う。**append 0**は追記モードをオフにし、この場合は録音データの記録は常に記録開始点から行われる。追記モードはデフォルトでオフである。

**loop** **loop**に続いて非ゼロ値を設定することで、ループ録音が可能となる。ループモードがオンの場合、録音データの書き込みがバッファの記録終了点（上記参照）まで来ると記録開始点から書き込みを続けることでループ録音が行われる。**loop 0**はループ録音モードをオフにし、この場合は録音データの書き込みが記録終了点まで来ると録音停止となる。デフォルトではループ録音モードはオフである。

**set** **set**に続いて**record~**が録音データの記録に用いる**buffer~**の参照名を指定する。

(マウス) **record~**オブジェクトをダブルクリックすることで、波形表示ウィンドウが開き、関連付けられた**buffer~**内部のデータを見ることができる。

## アーギュメント

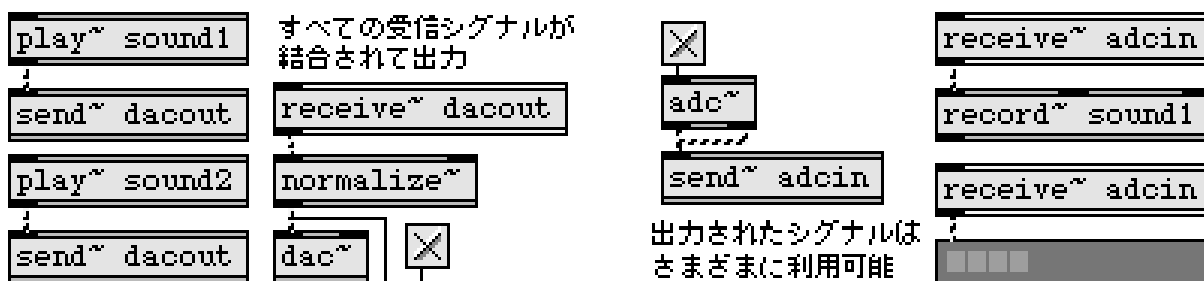
**シンボル** 必須。**record~**が録音データを書き込む**buffer~**の参照名を指定する。

**整数** オプション。**buffer~**参照名に続いて入力チャンネル数（1、2、4のいずれか）を設定することができる。これは**record~**オブジェクトのインレット数を決めるものとなる。右側2つのインレットは常に録音位置の開始点と終了点を指定するために使われる。

## 出力

**シグナル** 同期シグナルが出力される。録音中はこのアウトレットから録音開始を0、録音終了を1とする0から1までの間の値を持つシグナルが出力される。録音中でない時は0のシグナル出力となる。

## 利用例



後で利用するために一部のシグナルを記録する

## 参照

- 2d.wave~** 2次元のウェーブテーブル
- buffer~** オーディオ・サンプルの格納
- groove~** 速度可変のループ再生
- play~** ポジション・ベースのサンプル再生

Tutorial 13      Sampling: Recording and playback (英文PDF参照)

## 入力

- シグナル** 第1インレット：フィルター処理を行う任意のシグナル。
- 第2インレット：バンドパス・フィルターのゲインを設定する。この値は一般的には1以下でなければならない。
- 第3インレット：バンドパス・フィルターの中心の周波数をHz単位で設定する。
- 第4インレット：バンドパス・フィルターのQを設定する。それはおよそそのフィルターの鋭さを表し、Qの位置はフィルター・バンド幅によって分けられる中心の周波数で定義される。役立つQの値は一般的には0.01から500の間である。
- 整数または実数** 整数または実数はフィルターのゲイン、中心の周波数、そしてQの値、これら3つの値を変化させるために3つのインレットに入力される。これらのインレットにシグナルが接続されている場合、そのインレットに入力される数値は無視される。
- リスト** 最初の数値はフィルターのゲインを設定する。
- 2番目の数値はフィルターの中心の周波数を設定する。
- 3番目の数値はフィルターのQの値を設定する。これらのパラメーターに対応するインレットのいずれかにシグナルが入力されている場合、リストの中の対応している値は無視される。
- clear** フィルターのメモリーを消去する。**reson~**オブジェクトは再帰的フィルターであるため、このメッセージはブローアップ（爆発的な発振状態）から回復する場合に必要なになる。

## アーギュメント

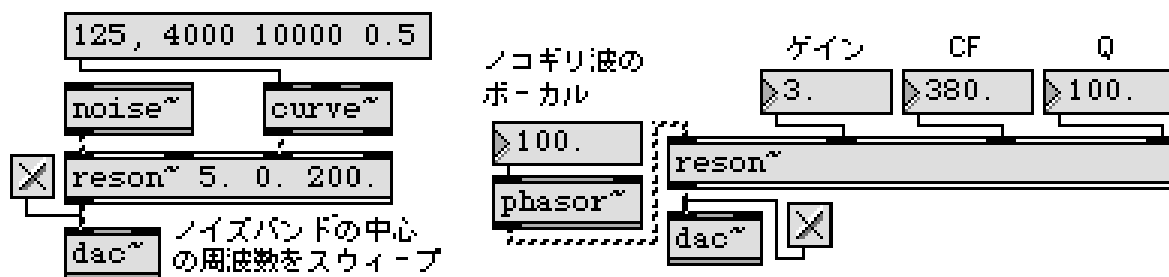
- 整数、または実数** オプション。数値はゲイン、中心の周波数、そしてQの初期値を設定する。デフォルト値のゲインは**0**、中心の周波数は**0**、そしてQは**0.01**である。

## 出力

- シグナル** フィルター処理が行われたシグナルが出力される。フィルターの式は
$$y_n = gain * (x_n - r * x_{n-2}) + c1 * y_{n-1} + c2 * y_{n-2}$$
ここで使用される割合、r、c1そしてc2は中心の周波数とQによって計算される。



## 利用例



バンドパス・フィルターのゲイン、中心の周波数、そしてQをコントロールして、豊かなシグナルに変化させる

## 参照

- biquad~**                    2ポール、2ゼロのフィルター
- comb~**                    コム・フィルター

ReWireシステムはオーディオ・アプリケーションをお互いに接続し、オーディオを生成するプログラム（クライアント）の出力をオーディオを再生するプログラム（ミキサー）へ送り込むことができるようにする。システムは機能拡張フォルダーにReWire共有ライブラリを必要とする。Max/MSPフォルダーの中のaudio stuffフォルダーの中にMSPと互換性のあるReWire機能拡張があるはずだ。より新しい機能拡張（例えばシーケンサーによってインストールされたもの）があれば、Max/MSPインストーラーが提供したのではなく、それを使う方が良い。ReWire機能拡張に加えて、それぞれのReWireクライアントごとに共有ライブラリが必要である。MSPはReWireクライアントまたはReWireミキサーとしてふるまうことができるが、同時に両方になることはできない。**rewire~**オブジェクトはMSPをReWireミキサーに変える。したがって、一度に1つのミキサーしか存在し得ない。

**reiwire~**オブジェクトとMSPのReWireオーディオ・ドライバー・オブジェクトを同時に使おうとすると、お互いのオブジェクトが他方を待つという膠着状態を作り出すことになり、サウンドは出力されないだろう。**rewire~**オブジェクトは、現在使用されているオーディオ・ドライバー・オブジェクトがReWireであれば警告を表示する

複数の**rewire~**オブジェクトを使うこともできる。この時、それぞれのオブジェクトは1つのReWireクライアントに関連付けられる。

**rewire~**は他のReWire対応ソフトウェア・シンセサイザーとともに用いることを目的としている。対応アプリケーションのリストはPropellerheadsのウェブ・サイト (<http://www.propellerheads.se>) から得ることができる。

ReWireはPropellerhead Software ASの商標である。

## 入力

- |                   |                                                                              |
|-------------------|------------------------------------------------------------------------------|
| <b>bang</b>       | 第1インレット：ReWireデバイスが読み込まれると、 <b>bang</b> は出力チャンネルの名前のリストを右から2番目のアウトレットから出力する。 |
| <b>整数</b>         | 第1インレット：1はReWireのトランスポートをスタートし、0はストップする。トランスポートをスタートしない限り音は何も出ない。            |
| <b>play</b>       | 第1インレット：ReWireのトランスポートをスタートする                                                |
| <b>stop</b>       | 第1インレット：ReWireのトランスポートをストップする                                                |
| <b>openpanel</b>  | 第1インレット：現在のデバイスがユーザー・インターフェース・パネルを持つのであれば、 <b>openpanel</b> はそれを開く。          |
| <b>closepanel</b> | 第1インレット：現在のデバイスのユーザー・インターフェース・パネルが開かれていれば、それを閉じる。                            |

- device** 第1インレット：数値を伴う**device**は、その数値インデックスで指定されるReWireデバイスに切り替える。インデックスは、右から2番目のアウトレットに接続したポップアップ・メニュー・オブジェクトに表示されている順序として得ることができる。
- 任意のシンボル第1インレット：シンボルはReWireデバイスの名前として解釈される。名前が有効であれば、**rewire~**はそのデバイスに切り替えようとする。
- tempo** 第1インレット：数値を伴う**tempo**は、テンポを1分間あたりの拍数で設定する。ReWireは整数テンポだけを扱うことができ、テンポが更新されるのはクライアントからオーディオ・サンプルを返すための次の呼び出しがあった時である。
- position** 第1インレット：数値を伴う**position**は、現在の再生位置を（サンプル数で）設定する。
- loop** 第1インレット：3つの数値を伴う**loop**は、現在のループ位置とモードを設定する。1番目の数値はループ開始位置をサンプル数で設定する。2番目の数値はループ終了位置をサンプル数で設定する。3番目の数値が1であれば、ループはオンになる。3番目の数値が0であれば、ループはオフになる。しかしながら、ReWireクライアントはトランスポートまたはタイム・ベースの出力を生成しない場合には、ループを無視するかもしれないことに注意せよ。例えば、MIDIノート・コマンドだけに反応するソフトウェア・シンセサイザーは、恐らくループには影響されないだろう。
- midi** 第1インレット：4つまたは5つの数値を伴う**midi**は、MIDIイベントをReWireデバイスに送信する。1番目の数値はタイム・スタンプの値で、現状では無視される（言い換えれば、イベントはただちに送信される）。2番目の数値はMIDIバスのインデックスである。ReWire 2は0から255までのインデックスを付けられた256のMIDIバスを持つ。3番目の数値はMIDIメッセージ・ステータス・バイトで、4番目と5番目の数値はMIDIメッセージ・データ・バイトである。
- map** 2つの数値を伴う**map**は、ReWireデバイスの出力チャンネルを、**rewire~**オブジェクトのアウトレットにマッピングする。ReWireチャンネルは1始まりで最大256である。**rewire~**オブジェクトのアウトレットは第1アウトレットを1として指定され、0はReWireチャンネルをオフにする。例えば、**map 3 2**はReWireデバイスのオーディオ・アウトプット・チャンネル3を、**rewire~**オブジェクトの左から2番目のアウトレットにする。**rewire~**オブジェクトがReWireデバイスに接続された後で、**bang**メッセージによってReWireオーディオ・アウトプット・チャンネルの名前を見つけることができる。デフォルトでは、オーディオ・アウトレットはReWireデバイスの最初のチャンネルにマッピング

されている。言い換えれば、最も左のシグナル・アウトレットは、デバイスの最初のチャンネルを出力する。

## アーギュメント

**シンボル** オプション。ReWireデバイス名を指定することができる。**rewire~**はオブジェクトが初期化された時に、デバイスを開こうとする。

**整数** オプション。**rewire~**オブジェクトが持つことになるオーディオ出力の数を指定する。アーギュメントがない場合には、1つのオーディオ・アウトレットが生成される。アウトレットの最大値は256である。

## 出力

**シグナル** (左から始まる) オーディオ・アウトレット：ReWireデバイスからのオーディオ・シグナル出力は、**rewire~**オブジェクトのアウトレットから出力される。デフォルトでは、最も左のアウトレットがデバイスの最初のチャンネルを出力する。しかし、このマッピングは**map**メッセージで変更することができる。

**シンボル** 右から4番目のアウトレット：ReWireデバイスのトランスポートの状態を表すメッセージ。**position**メッセージに整数のアーギュメントを付けると、トランスポート位置を15360 PPQとして出力する。**play**メッセージと**stop**メッセージは、トランスポートのスタートやストップを表す。

**MIDI** 右から3番目のアウトレット：ReWireデバイスから受信したMIDIイベントは、このアウトレットから**midi**という単語に続けて出力される。1番目のアーギュメントは常に0（これはタイム・スタンプ）で、2番目のアーギュメントはReWireのMIDIバスのインデックスである。3番目のアーギュメントはMIDIステータス・バイトであり、4番目と（オプションの）5番目のアーギュメントはMIDIデータ・バイトである。

**シンボル** 右から2番目のアウトレット：**bang**メッセージを受け取ると、現在利用可能なReWireデバイスのリストが出力される。

**シンボル** 右のアウトレット：現在使用されているReWireデバイスの現在利用可能なデバイス出力の名前のリスト（チャンネル順）が出力される。

## 利用例



**rewire~**はReWire対応デバイスへのMIDIコミュニケーションと、そのデバイスからのシグナルの出力を可能にする

## 参照

**vst~**

VSTプラグインをホストする

# round~

入力されたシグナル値の  
端数の切り捨て／切り上げを行う

## 入力

**シグナル** 第1インレット：入力されたシグナル値の端数を切り捨てる、または切り上げる。

第2インレット：第1インレットから入力されたシグナル値の端数を、入力されたシグナルの値を単位として切り捨てる、または切り上げる。第1インレットに受け取ったシグナル値は、このインレットに受け取った値の整数倍の値に絶対値として、もっとも近い値に切り捨てられるか、切り上げられる。または、このインレットに受け取った値と0との間での整数倍に、もっとも近い値に切り捨てられる（後者については**nearest**メッセージを参照のこと）。

**nearest** 第1インレット：**nearest**と0以外の値をメッセージとして入力すると、**round~**オブジェクトは、第1インレット入力されたシグナル値を、第2インレットに入力された値の整数倍値で最も近い値にする。デフォルトではこのモードになっている。**nearest 0**を入力した場合は、入力されたシグナル値を、第2インレットに入力された値の整数倍値か0に切り捨てる。

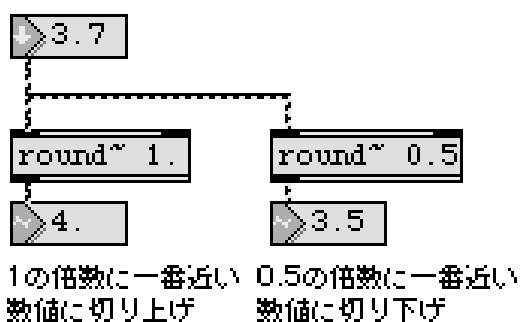
## アーギュメント

**整数または実数** オプション。入力されたシグナルが切り捨て、または切り上げられる値を設定する。

## 出力

**シグナル** 入力されたシグナルを切り捨て、または切り上げたもの。

## 利用例



**round~** オブジェクトを用いて、特定の値を単位としてシグナルの小数部分を切り捨て、または切り上げる

## 参照

**rampsmooth~** 入力シグナルにスムーズ処理を行う  
**slide~** シグナルへの対数的なフィルター処理

## 入力

**シグナル** 第1インレット：サンプルするシグナルを入力。第2インレットに入力するコントロール・シグナルが、現在のトリガー値と同じ値かそれより小さい値から、トリガー値より大きくなった時に、第1インレットのシグナルがサンプルされ、そのサンプル値が一定のシグナルとしてアウトレットから出力される。

第2インレット：サンプル&ホールドのコントロール・シグナルを入力する。**sah~**の出力を変化させるためには、コントロール・シグナルが、トリガー値と同じ値またはそれより小さい値から、トリガー値より大きな値になる必要がある。この変化が起こった時に第1インレットに入力されたシグナルがサンプルされ、新しい出力シグナルの値となる。

**整数または実数** 第1インレット：トリガー値を設定する。

## アーギュメント

**整数または実数** オプション。トリガー値を設定する。デフォルト値は0である。

## 出力

**シグナル** 第2インレットに入力されたコントロール・シグナルが、トリガー値と同じかそれより小さい値から、トリガー値より大きな値へと変化した時に、出力シグナルは第1インレットに入力されているシグナルの値に変化する。このシグナル値は次にコントロール・シグナルがトリガー値を越えるまで出力し続ける。

## 利用例



次のトリガーまで一定のシグナル値を保持する

## 参照

**phasor~** ノコギリ波の生成



## 入力

**実数または整数** 入力されたサンプル数を、現在のサンプリング・レートにもとづきミリ秒に変換し、第2インレットから出力する。入力するサンプル値は、小数部分を含んでも構わない。例えば、サンプリング・レートが44.1kHzの場合、322.45サンプルは、7.31ミリ秒に変換される。(オーディオ処理がオフになっている時でも、実数または整数を受け取った時に出力される。)

**シグナル** サンプル数を表すシグナル値。現在のサンプリング・レートにもとづきミリ秒に変換し、第1インレットから出力する。入力するサンプル値は、小数部分を含んでも構わない。

## アーギュメント

なし。

## 出力

**シグナル** 第1インレット：サンプル数として入力されたシグナルの値に対応するミリ秒の値としてのシグナル。

**実数** 第2インレット：インレットに受け取ったサンプル数に対応するミリ秒の値。

## 利用例



あるオブジェクトはサンプル単位で、あるオブジェクトはミリ秒単位で時間を扱う

## 参照

**dspstate~** 現在のDSPの設定を出力する  
**mstosamps~** ミリ秒をサンプル数に変換



## 入力

**シグナル** 第1インレット：入力シグナルをオシロスコープのX軸に沿って表示する。

第2インレット：入力シグナルをオシロスコープのY軸に沿って表示する。

シグナル・オブジェクトが第1インレットと第2インレットの両方に繋がれている場合は、第1インレット (X)に入るシグナルの値を水平位置とし、第2インレット (Y)に入るシグナルの値を垂直位置とする点を描画するX-Yモードとして動作する。2つのシグナルが同じであり、位相がずれていない場合は、左から右に上昇する直線が表示される。2つのシグナルが同じであり、位相が180度ずれている場合は、左から右へ下降する直線が表示される。他の組み合わせでは円、楕円、リサーチ図形が表示される。

**整数** 第1インレット：表示用バッファ内の値を集約するサンプル数を指定する。小さな数値であれば、表示は拡大され、素早くスクロールされる。この最小値は2、最大値は8092であり、デフォルト値は256である。XモードまたはYモードでは、この周期における最大値または最小値が使われる。X-Yモードでは、この周期の代表値が使われる。

第2インレット：表示用バッファのサイズを設定する。これは、**scope~**が表示のスケールと描画を更新する速度を決めることになる。バッファ・サイズが大きければ、シグナルのイメージは長い時間画面に表示され、視覚的には縮小される。バッファ・サイズが小さければ、シグナルのイメージが更新されるまで短い時間画面に表示され、視覚的に拡大される。

表示用バッファを集約するサンプル数と、表示用バッファのサイズは、同じような働きをするように見えるが、それらは僅かに異なる効果がある。最適な表示を行うパラメータを決めるためには、それらをいずれも試してみる必要がある。

**brgb** **brgb**に続く0から255までの3つの数値によって、**scope~**オブジェクトが表示する背景色をRGB値で設定する。デフォルト値は135 135 135である。

**bufsize** **bufsize**に続く数値によって、**scope~**オブジェクトが使用するバッファのサイズをサンプル数として設定する。

**frgb** **frgb**に続く0から255までの3つの数値によって、**scope~**オブジェクトが表示する波形の色をRGB値で設定する。デフォルト値は102 255 51である。



- range** **range**に続く実数か整数の2つの数値によって、シグナルの振幅表示の最小値と最大値を設定する。デフォルト値は-1と1である。
- delay** **delay**に続く数値によって、**scope~**が数値を集め始める前に待機するディレイ時間をミリ秒単位で設定する。0ではないディレイ時間の後に、**scope~**はトリガー条件を待つ状態になる。トリガー条件とは、入力シグナルがトリガー状態 (**trigger**メッセージで設定する) やトリガー・レベル (**triglevel**メッセージで設定する) を満たす条件のことである。デフォルトでは、ディレイ時間は0である。
- trigger** トリガー・モードを設定する。0ではないディレイ時間 (**delay**メッセージで設定する) の後に、**scope~**は、入力シグナルが指定した状態になってから、サンプルを集め始める。**trigger 1**は、シグナルがトリガー・レベル (デフォルト値は0) 以下から、同じかそれ以上になる増加方向のトリガーを設定する。**trigger 2**は、シグナルがトリガー・レベル (デフォルト値は0) 以上から、同じかそれ以下になる減少方向のトリガーを設定する。デフォルトのトリガー・モードは0で、0ではないディレイ時間の後、すぐにサンプルを集め始める。これは、ライン・トリガー・モードとも呼ばれる。
- triglevel** **triglevel**に続く数値によって、トリガー・モード1とトリガー・モード2で使われるトリガー・レベルを設定する。デフォルトのトリガー・レベルは0である。波形表示において、トリガー・レベルを僅かに変化させることによって、波形を左右に動かすことができる。表示を変えないために、トリガー・レベルを設定することもできる。
- (マウス) **scope~**をクリックすれば、マウス・ボタンを押し続けている間、表示が静止する。

## インスペクター

**scope~**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**scope~**オブジェクトを選択することで、**scope~**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**scope~**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**scope~**のインスペクターは、以下の属性を設定することができる。

*Buffer per Pixel*は、**scope~**が表示する1ピクセルあたりのバッファ数指定する。デフォルト値は25である。*Buffer Size*は**scope~**が使用する1バッファあたりのサンプル数を指定する。デフォルト値は128である。*Range*の数値は、表示する最小値と最大値を設定する。デフォ



ルトの *Min.* は -1.0、*Max.* は 1.0 である。*Delay* の数値は、**scope~** が数値を集め始める前のデレイ時間をミリ秒単位で設定する。*Trigger Mode* のチェック・ボックスでは、*Line Up* (デフォルト) か、*Line Down* のトリガー・モードを指定する (先の **trigger** メッセージを参照のこと)。*Trigger Level* は、トリガー・モード1とトリガー・モード2で用いるトリガー・レベルを設定する (先の **triglevel** メッセージを参照のこと)。デフォルトのトリガー・レベルは 0 である。

*Colors* のポップアップ・メニューでは、描画色か背景色を選択し、スウォッチ形式のカラー・ピッカーかRGB値によって、その色を指定する。*Phosphor* を選べば、波形の色を設定できるようになる。デフォルトの描画色は 102 255 51 である。*Background* なら、背景の色を設定する。デフォルトの背景色は 135 135 135 である。

*Revert* ボタンは、このオブジェクトに設定したすべての変更点を、インスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、EditメニューのUndo Inspector Changesを選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

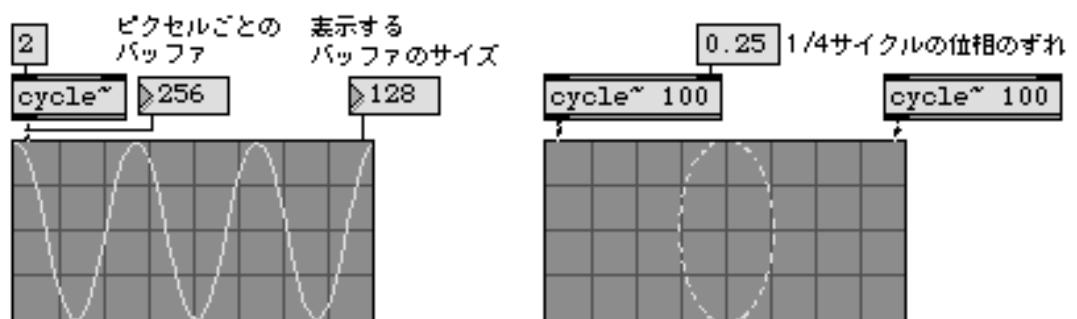
## アーギュメント

なし

## 出力

なし

## 利用例



シグナルを表示するか、X-Yモードで二つのシグナルを描画する

## 参照

### meter~

視覚的なピーク・レベル・インジケータ

### Tutorial 24

Analysis: Oscilloscope (英文PDF参照)

## 入力

### 整数または実数

第1インレット：第1インレットにシグナルが入力されていない場合、整数または実数を入力することで、他のインレットに入力されたシグナルのどれをアウトレットから出力するかを決めることができる。入力値が0か負の数である場合、すべてのインレットが閉じられ、無音のシグナルが出力される。入力値が1以上2未満の場合、第2インレットへの入力シグナルがアウトレットから出力される。入力値が2以上3未満の場合、第3インレットへの入力シグナルがアウトレットから出力される。それ以降の入力値の場合も同様の動作となる。

### シグナル

第1インレット：第1インレットにシグナルが入力されている場合、そのシグナルの値によって他のどのインレットに入力されたシグナルをアウトレットから出力するかを決めることができる。シグナルの値が0か負の数である場合、他のインレットは閉じられてアウトレットからは無音のシグナルが出力される。入力値が1以上2未満の場合、第2インレットへの入力シグナルがアウトレットから出力される。入力値が2以上3未満の場合は第3インレットへの入力シグナルがアウトレットから出力される。それ以降の入力値の場合も同様の動作となる。

第2以降のインレット：任意のシグナルを入力する。どのインレットの入力シグナルがアウトレットから出力されるかは、第1インレットへ入力した整数か実数、またはシグナルの値によって決められる。そのインレットの指定する値については、第2インレットが1、第3インレットが2...という対応となっている。

第1インレットにはシグナルが接続されておらず、**begin~**で始まるシグナル・ネットワークがそれ以外のインレットに接続されている場合、そのインレットが出力対象となっていなければ、**begin~**と**selector~**間の処理がオフになる。

## アーギュメント

### 整数

オプション。第1アーギュメントは入力するシグナルの数を指定する。そのデフォルト値は1である。第2アーギュメントは出力するシグナルの対応インレット初期値を指定する。そのデフォルト値は0であり、すべてのインレットが閉じられて、無音シグナルが出力されることになる。シグナルが第1インレットに入力されている場合、第2アーギュメントは無効となる。

## 出力

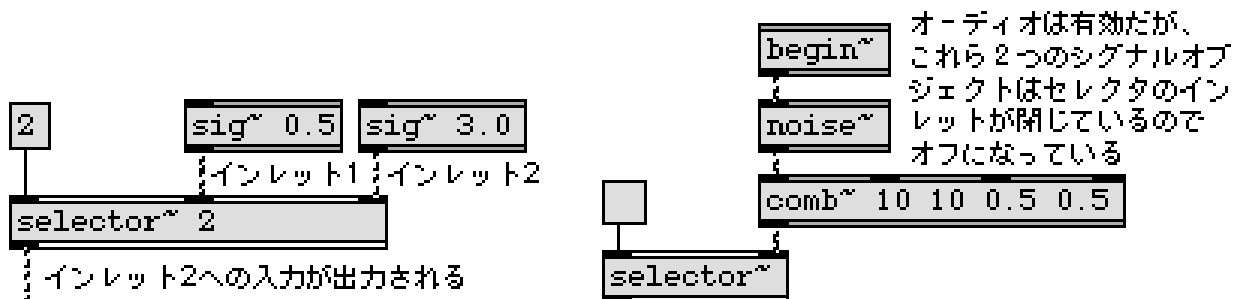
### シグナル

第1インレットに入力された数値かシグナルで指定された、開かれているインレットへ入力したシグナルが出力される。すべてのインレットが閉ざされている場合は、出力は無音となる。

# selector~

複数の入力を切り替えて、  
1つのアウトレットから出力する

## 利用例



複数のシグナルから通過させるものを1つだけ選択：  
必要がないシグナル処理を任意にオフにすることが可能である

## 参照

**gate~**  
**begin~**

複数アウトレットの1つへシグナルを切り替える  
シグナル・ネットワークの切り替え可能な部分を定義する

Tutorial 5

Fundamentals: Turning signals on & off (英文PDF参照)



# send~

パッチ・コードなしに  
シグナルを送信する

## 入力

**シグナル** **send~**オブジェクトは入力シグナルを同じ名前を持つすべての**receive~**オブジェクトに送信する。**send~**オブジェクトは対応する**receive~**オブジェクトと同じパッチの中に存在する必要はない。

**set** シンボルが後に続く**set**メッセージは、そのシンボルを持つ異なる**receive~**オブジェクトと接続するために**send~**オブジェクトの名前を変更する。(同じ名前を持つ**receive~**オブジェクトが存在しない場合は**send~**オブジェクトは何も行わない。)

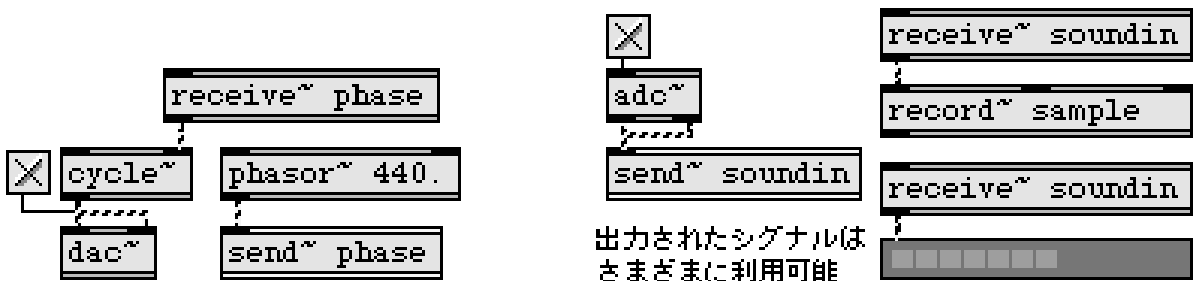
## アーギュメント

**シンボル** 必須。**send~**オブジェクトの名前を設定する。

## 出力

なし。

## 利用例



シグナルを **send~** オブジェクトへ入れ、同じ名前を持つすべての **receive~** オブジェクトから出す

## 参照

**receive** パッチ・コードなしでメッセージを受信する

Tutorial4 Fundamentals : Routing signals (英文PDF参照)



## 入力

**シグナル** 値が0.から1.0の間にある入力シグナル（通常は**phasor~**の出力）を、イベント・シーケンサーを動作させるために用いる。

**任意のメッセージ** **seq~**オブジェクトはメッセージを記録したり、再生することができる。入力シグナルの現在値に従って、インレットに受け取ったすべてのイベントを記録する。任意のメッセージは、**seq~**オブジェクト自体へのコマンドを除いて、記録することができる。後に示す利用例で、この制限を解決する簡単な方法を示している。

注記：MIDI入力ストリームをMIDIイベントのリストに変換すれば、**seq~**はMIDIデータを記録することができる。この変換は、壊れたMIDIストリームを出力しないようにするために必要である。これは、MIDIストリームのままの整数メッセージを個々に記録した場合、**seq~**が単純に正方向にリニアに再生しなければ、MIDIストリームとして正常ではなくなるからである。

**bang** **seq~**オブジェクトの現在のシーケンス・ナンバー、実行モード（録音、オーバーダブ、再生）、現在のイベントの合計数についての情報を、Maxウィンドウに表示する。

**add** **add**に続く整数と実数とメッセージは、整数によって指定するシーケンス・ナンバーに対して、実数が指定する時刻に、そのメッセージによって指定するMaxイベントを追加する（例えば、**add 2 0.5 honk**なら、シーケンス 2の中間時刻に再生されるように**honk**メッセージを追加する）。

**dump** 格納しているすべてのイベントのシーケンスの内容を、第2アウトレットから出力する。**dump**の後に数値が続く場合は、その数値によって指定されるシーケンスを出力する。

**erase** すべてのシーケンスを消去する。

**overdub** **overdub**の後に1が続けば、**seqnum**メッセージによって設定されるシーケンスにおいて、オーバーダブ・モードでMaxイベントの記録を開始する。この記録はループの現在地点から始まり、入力シグナルが1に達すると最初に戻るように行われる。**overdub 0**メッセージは、オーバーダブ・モードを停止する。

**play** **play**の後に1が続けば、入力シグナルの現在の値によって指定されるループ地点から、**seqnum**メッセージによって指定された現在のシーケンスのMaxイベントの再生を開始する。**play 0**は再生を停止する。デフォルトでは、再生は停止している。

- read** **seq~**オブジェクトの**write**メッセージによって作られたMaxイベントのシーケンスを含むテキスト・ファイルを開き、**seq~**のメモリーに読み込む。**read**の後にシンボルのアーギュメントがなければ、標準ファイル・ダイアログを開き、読み込むことができるテキスト・ファイルを表示する。**read**にシンボルが続く場合は、標準ファイル・ダイアログを開かずに、そのシンボルに一致する名前のファイルを開き、その内容を**seq~**のメモリーに読み込む。
- record** **record**の後に**1**が続けば、現入力シグナルの現在の値によって指定されるループ地点から、**seqnum**メッセージによって指定された現在のシーケンスへの、Maxイベントの記録を開始する。**record 0**は記録を停止する。デフォルトでは記録は停止している。
- seqnum** **seqnum**に続く0から15までの数値によって、録音や再生を行う現在のMaxイベントのシーケンス・ナンバーを指定する。
- write** 標準ファイル保存ダイアログが開き、ファイルの名前を指定すれば、現在のMaxイベントのシーケンスのすべての内容を、テキスト・ファイルに保存する。**write**の後にシンボルが続けば、**seq~**オブジェクトを含むパッチと同じフォルダに、そのシンボルをファイル名としてファイルを保存する。パッチが保存されていない場合は、**seq~**はMaxアプリケーションと同じフォルダにファイルを保存する。

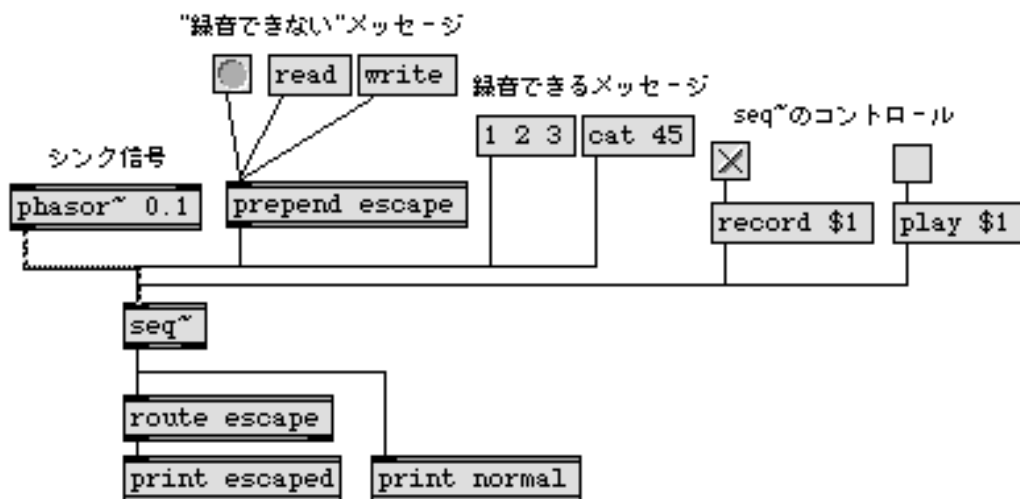
## アーギュメント

なし。

## 出力

- 任意のメッセージ 第1アウトレット：**play 1**メッセージによって再生を行っている時、**seq~**オブジェクトは、入力シグナルによって指定される時刻に記録されたすべてのイベントを出力する。
- リスト 第2アウトレット：**dump**メッセージによって、**seq~**は指定されたシーケンスの内容をリストとして出力する。このリストには、シーケンス・ナンバーを指定する整数、時刻を指定するシグナルの値である実数、そして、その時刻に出力する整数、実数、シンボル、リストのいずれかが含まれる。

利用例



参照

phasor~

ノコギリ波の生成

## 入力

- open** オーディオ・ファイル名を伴う**open**は、その名前のファイルがMaxのサーチ・パスの中に存在する場合に、そのファイルを開く。ファイル名がない場合は、標準ファイル・ダイアログを開き、ファイルを選択させる。ファイルが開かれた後、**sfinfo~**オブジェクトはファイルに問い合わせ、チャンネル数、サンプル・サイズ、サンプリング・レート、ミリ秒単位でのファイルの長さ、サンプル・タイプ、そしてファイル名を出力する。
- bang** **open**メッセージまたは**sfinfo~**オブジェクトのアーギュメントによって指定されて、既にファイルが開かれている場合、**bang**はそのファイルのチャンネル数、サンプル・サイズ、サンプリング・レート、ミリ秒単位でのファイルの長さを**sfinfo~**オブジェクトのアウトレットから出力する。
- getnamed** 第1インレット：**sfplay~**オブジェクトの名前を指定するシンボルを伴う**getnamed**は、指定された名前の**sfplay~**オブジェクトに問い合わせ、チャンネル数、サンプル・サイズ、サンプリング・レート、ミリ秒単位でのファイルの長さ、サンプル・タイプ、そしてファイル名をアウトレットから出力する。

## アーギュメント

- シンボル** オプション。**sfinfo~**がその後に**bang**メッセージを受け取った時、**sfinfo~**が出力するファイルの名前を付ける。このファイルはMaxのサーチ・パスの中に存在しなくてはならない。

## 出力

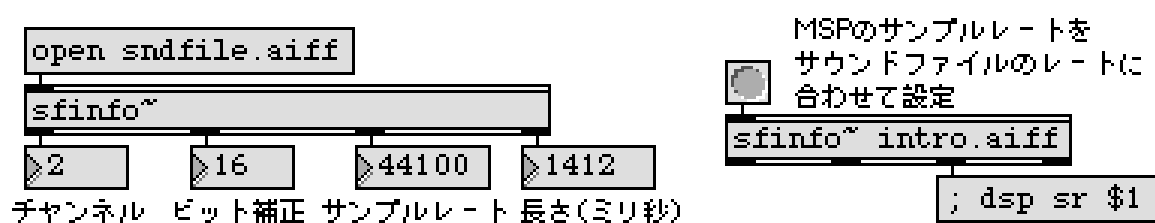
- 整数** 第1アウトレット：オーディオ・ファイルのチャンネル数。
- 第2アウトレット：ビット単位でのオーディオ・ファイルのサンプル・サイズ（一般的には16）。
- 実数** 第3アウトレット：オーディオ・ファイルのサンプリング・レート。
- 第4アウトレット：ミリ秒単位でのオーディオ・ファイルの長さ。
- シンボル** 第5アウトレット：オーディオ・ファイルのサンプル・タイプ。
- 以下のサンプル・データの方がサポートされる。

|               |                |
|---------------|----------------|
| <b>int8</b>   | 8-bit integer  |
| <b>rint16</b> | 16-bit integer |

|                |                           |
|----------------|---------------------------|
| <b>int24</b>   | 24-bit integer            |
| <b>int32</b>   | 32-bit integer            |
| <b>float32</b> | 32-bit floating point     |
| <b>float64</b> | 64-bit floating point     |
| <b>mulaw</b>   | 8-bit $\mu$ -law encoding |
| <b>alaw</b>    | 8-bit a-law encoding      |

第6アウトレット：オーディオ・ファイルのファイル名。

## 利用例



指定したオーディオ・ファイルの情報を入力する

## 参照

**info~**  
**sflist~**  
**sfplay~**

**buffer~**内のサンプルデータに関する情報の出力  
オーディオ・ファイルのキューを格納する  
ディスクからオーディオ・ファイルを再生する

Tutorial 16

Sampling: Record and play audio files (英文PDF参照)

## 入力

- open** **open**に続いてAIFF形式やSound Designer II形式（Macintoshのみ）、WAV形式、NeXT/Sun形式のオーディオ・ファイル名を指定することで、Maxのサーチ・パス内にその名前のファイルがあれば読み込む。ファイル名を指定せずに**open**メッセージのみを入力した場合、標準ファイル・ダイアログが開き、ファイルを選択する。ファイルを開くと、次に別のファイルが開かれるまではメモリ内にデータが読み込まれ、そのオーディオ・データの先頭がキュー1として定義される。その後のキューについては、ファイル名の記述なしで**preload**メッセージを使うことで、既に読み込んだファイルに対して指定することができる。**open**メッセージが新たに入力された場合、これまで読み込んでいたファイルに対してであっても、**preload**メッセージで以前のファイル名を指定することで、そのファイルに対するキューを定義することができる。以前に読み込んだファイルで定義したキューは、別のファイルが現在のファイルになったとしても、その定義は有効である。
- clear** **clear**に続くアーギュメントがない場合、すべてのキューの定義をクリアする。**clear**メッセージが入力されても、1番のキューだけは再生できる（既に一度はファイルが開かれた場合）。**clear**に続いて1つ以上の数字が指定された場合は、その**sflist~**オブジェクトが格納している指定番号のキューを削除する。
- embed** **embed**に続いて0でない整数が指定された場合、パッチを保存する時に、**sflist~**は格納しているすべてのキューの定義と現在の読み込みファイル名を保存する。**embed 0**というメッセージは、パッチの保存時に**sflist~**の内部情報を保存しないという指定となる。デフォルトの設定では、パッチ保存時に現在のファイル名およびキューの情報は保存されない。**sflist~**オブジェクトが格納しているキューとともに保存された場合、その保存されたパッチが読み込まれると、その時点で保存されたキューに対応するオーディオ・ファイルなども読み込まれ、キューの情報が設定される。
- fclose** **fclose**に続いて既に開かれたファイル名を指定することで、そのファイルを閉じるとともに、そのファイルに対するキューの情報すべてを消去する。**fclose**のみを指定した場合は、現在のファイルを閉じるだけとなる。
- openraw** **openraw**メッセージは**open**と全く同じ機能を持つが、任意のファイルを開いて現在のファイルとすることができる。指定したファイルは44100Hzのサンプリング・レートを持つ16ビットのステレオで、ヘッダ情報がない（ファイル先頭から実際のサンプル・データまでのオフセットが0）ファイルとみなして読み込まれる。**samptype**、**offset**、**srates**や**srchans**のメッセージによってファイル・タイプが明示的に指定されている場合は、それらの情報に従って読み込まれる。



**preload** 2以上の整数とともに指定することで、ファイル中の指定領域に対するキューを定義する。キュー番号が**sflist~**で定義された後、**sfplay~**オブジェクトにそのキュー番号を指定すると、対応するファイルの領域が再生される。キュー番号1は常に現在のファイル（最も最近**open**メッセージで開いたファイル）の先頭であり、**preload**メッセージで変更することはできない。

**preload**メッセージには多くの入力形式がある。**preload**に続いて2から32767のキュー番号を指定することは必須である。そのキュー番号の後ろに再生するファイル名を指定する。そのファイルは現在既に行っているファイルか、またはMaxのサーチ・パス上にあるファイルを指定し、キュー定義後はそのキュー番号にそのファイルが対応付けられる。ここで、キューで使うために**open**メッセージによって明示的に開かれたファイルについては、**preload**メッセージでのファイル指定は必要ないことに注意が必要である。その場合、ファイル名の指定がなければ、現在開かれているファイルが使われることになる。

オプションとしてファイル名を指定した後は、そのファイルの再生開始位置をミリ秒単位で指定することができる。再生開始位置が指定されない場合、キューの再生位置はファイルの先頭に設定される。再生開始位置の後には、再生終了位置をミリ秒単位で指定することができる。再生終了位置を指定しない場合、または0を指定した場合、キューはファイルの最後まで再生する。再生終了位置が開始位置より前の時刻を指定している場合、キューは定義できても再生されない。将来的にはそれで逆再生のキューが指定できるようになるかもしれない。

開始・終了位置のアーギュメントの後には定義したキューを逆再生できるように双方向バッファ・フラグをオプションとして指定することができる。このフラグを1にすると逆再生が可能なキューとなる。デフォルトの設定では0である（双方向バッファリングはオフ）。

最後のオプションであるアーギュメントは再生速度を設定するのに使われる。**sfplay~**オブジェクトでの**speed**メッセージで設定されたオブジェクト全体の再生速度に対する相対速度を、1つの実数で設定する。デフォルト値は1である。

定義した各キューには、**sfplay~**でのチャンネルごとにおおよそ40Kバイトのメモリが必要で、双方向バッファリングがオフの時にはデフォルトのバッファ・サイズ（40,320バイト）である。双方向バッファリングがオンの場合は、各キューのメモリ量は倍になる。

**print** 現時点で定義されたすべてのキューのリストを表示する。

**samptype** **samptype**に続いて、サンプル・タイプを表すシンボルを指定する。このサンプル・タイプはオーディオ・ファイルのサンプル・データを解



積する時に使われる（実際のサンプル・タイプを解釈する上で非常に重要な値である）。これは時には“header munging”（ヘッダー情報の書き換え）と呼ばれる。**openraw**メッセージによってファイルが読み込まれる場合、特に指定しないと16ビット整数のサンプル・タイプとして読み込もうとする。**samptype**メッセージによるサンプル・タイプの指定はディスク上のファイルを修正するわけではない。

サンプル・タイプとしては次に示すものがサポートされている。

|                |                       |
|----------------|-----------------------|
| <b>int8</b>    | 8ビット整数                |
| <b>int16</b>   | 16ビット整数               |
| <b>int24</b>   | 24ビット整数               |
| <b>int32</b>   | 32ビット整数               |
| <b>float32</b> | 32ビット浮動小数点            |
| <b>float64</b> | 64ビット浮動小数点            |
| <b>mulaw</b>   | 8ビット $\mu$ -law エンコード |
| <b>alaw</b>    | 8ビット a-law エンコード      |

**srcchans** **srcchans**に続いてチャンネル数の数値を指定する。この数値はオーディオ・ファイル中のサンプル・データを解釈する時に使われる（実際のチャンネル数を解釈する上で非常に重要な値である）。これは時には“header munging”（ヘッダー情報の書き換え）と呼ばれる。**openraw**メッセージによってファイルが読み込まれる場合、特に指定しないとチャンネル数は2として読み込もうとする。**srcchans**メッセージによるチャンネル数の指定はディスク上のファイルを修正するわけではない。

## アーギュメント

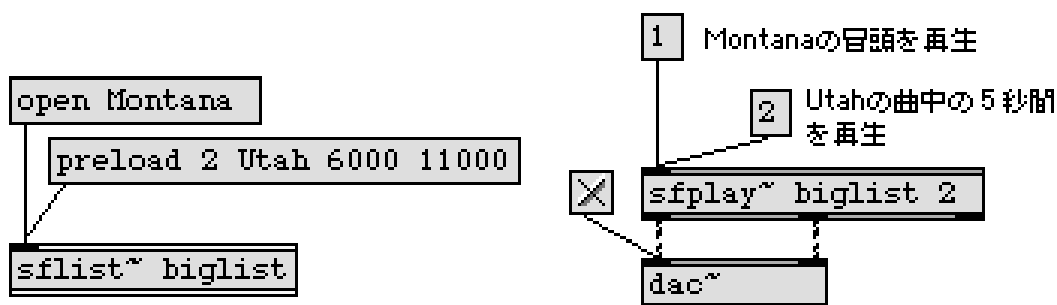
**シンボル** 必須。**sflist~**の参照名を設定する。**sfplay~**オブジェクトはこの名前を使って**sflist~**に格納されたキューを参照する。

**整数** オプション。あらかじめ読み込んでおくオーディオ・ファイルに用いるバッファ・サイズを設定する。そのデフォルトのサイズは16384バイトで、最小サイズでもある。このあらかじめ用意しておくバッファは、オーディオ・ファイルの各チャンネルごとにバッファ・サイズの4倍のサイズとなる。

## 出力

なし。

## 利用例



**sfplay~** オブジェクトが使うキュー全体のリストを格納する

## 参照

|                  |                        |
|------------------|------------------------|
| <b>buffer~</b>   | オーディオ・サンプルの格納          |
| <b>groove~</b>   | 速度可変のループ再生             |
| <b>play~</b>     | ポジション・ベースのサンプル再生       |
| <b>sfinfo~</b>   | オーディオ・ファイルの情報を出力する     |
| <b>sfplay~</b>   | ディスクからオーディオ・ファイルを再生する  |
| <b>sfrecord~</b> | ディスクへオーディオ・ファイルとして録音する |

Tutorial 16

Sampling: Record and play audio files (英文PDF参照)

## 入力

**実数** 第2インレット：オーディオ・ファイルの再生速度を定義する。1.0はオーディオ・ファイルを通常の方法での再生を表す。-1は通常の方法での逆再生を表す。2は通常の方法の2倍の方法で、0.5は通常の方法の半分の方法での再生を表す。

**シグナル** 第1インレット：入力シグナルはあらかじめ格納されたキューに対し、正確なトリガーに基づく再生を行うのに使うことができる。シグナルが第1インレットに入力されると、そのシグナルの整数部分がチェックされる。そしてその整数部分の値がキューのインデックス番号と同じ値になった時、その該当キューをトリガー再生する。負の数は無視される。

第2インレット：オーディオ・ファイルの再生速度も、シグナルによって定義できる。これによって、再生速度を時間とともに変化させてビブラート効果を作り出したり、その他の再生速度に基づくエフェクトをかけることができる。実数値を用いて行えるような再生速度の設定もシグナルの値によって可能である。

**整数** 第1インレット：**open**メッセージによってファイルが開かれている場合、**1**は（最も最近に開かれたファイルの）再生、**0**はその停止を表す。1より大きな値については、**preload**メッセージであらかじめ設定されたキューに対してトリガーをかける。または、**sfplay~**オブジェクトの保存状態を定義することになる。ファイルが再生されている時は、そのファイル中のオーディオ・データはオブジェクトが持つチャンネルの数に対応したアウトレットから出力される。キューの再生が完了、または**0**を入力して再生を停止した時、一番右側のアウトレットからは**bang**が出力される。**sfplay~**オブジェクトが**sflist~**を参照するように設定されている場合（**set**メッセージかアーギュメントで設定）、入力される整数は、**sfplay~**内部ではなく**sflist~**に格納されたキューに対するトリガーとして利用される。**sfplay~**内部に持つキューをリセットするには、アーギュメントのない**set**メッセージを入力する。

**anything** 第1インレット：**sflist~**の参照名と、それに続く数字が**sfplay~**に入力されると、指定した**sflist~**にキューが存在するならば、そのキューを再生する。

**clear** 第1インレット：アーギュメントのない**clear**メッセージは定義されたすべてのキューをクリアする。**clear**メッセージが入力された後でも、1の定義だけは再生できる（該当するオーディオ・ファイルを開くことができる場合）。**clear**に続いて1つ以上の数字が指定された場合は、その数字に該当する**sfplay~**内のキューが削除される。

**embed** 第1インレット：**embed**に続いて0でない任意の整数が入力されると、**sfplay~**はパッチ保存時に、保持しているキューの定義と現在開いているファイル名を保存するようになる。**embed 0**というメッセージは、パッチ保存時に**sfplay~**の内部情報を保存しないという指定となる。デフォルトの設定では、パッチ保存時に現在のファイル名とキューの情報

は保存されない。**sfplay~**オブジェクトが格納しているキューとともに保存された場合、その保存されたパッチが読み込まれると、その時点で保存されたキューに対応するオーディオ・ファイルなども読み込まれ、キューの情報が設定される。

- fclose** 第1インレット：**fclose**に続いて既に開かれたファイル名を指定することで、そのファイルを閉じるとともに、そのファイルに対するキューの情報すべてを消去する。**fclose**のみを指定した場合は、現在のファイルを閉じるだけとなる。
- リスト** 第1インレット：**sfplay~**に対して再生するキューの順序列をリストとして与えると、その順に再生する。次のキューは現在再生中のキューが終了してから再生される。キューのリストの最大数は128である。リスト中のキュー番号が定義されていなかった場合は、その番号は無視して、その次のキューの再生を試みる。**sfplay~**が**sflist~**オブジェクトを参照する設定となっている場合は、リストは参照先の**sflist~**内に格納されたキューを再生する。参照先が指定されていない場合は、**sfplay~**内に格納されたキューを再生する。
- loop** 第1インレット：**loop**に続いて1を指定するとループ再生をオンにし、**loop 0**はオフにする。デフォルト設定ではループ再生はオフである。
- name** **name**に続いて**sfInfo~**などのオブジェクトが**sfplay~**を参照する際に用いる参照名のシンボルを指定する。参照名を変更した場合、古い参照名を指定していたオブジェクトからの参照は断たれる。各**sfplay~**の参照名はひとつに限るべきであり、既に他の**sfplay~**オブジェクトが持つ参照名を指定した場合、その名前を持っていた元々の**sfplay~**はその名前では参照されなくなる。
- offset** 第1インレット：**offset**に続いて数値を1つ指定することで、サンプル単位での再生開始点を設定する。デフォルト値は0である。この値はサンプルの整列に有効であるとともに、ファイルのヘッダー情報にあたる部分をサンプル・データとして再生することを回避するのに有効である。
- open** 第1インレット：AIFF、Sound Designer II (Macintoshのみ)、WAV、NeXT/Sun、rawフォーマットのオーディオ・ファイルの名前を指定することで、再生するファイルとして開き、現在のファイルとする。ファイル名は**open**の後に続けて指定し、Maxのサーチ・パス内に存在すれば、そのファイルを開く。ファイル名を指定せずに**open**メッセージが入力された場合は、標準ファイル・ダイアログが開くので、ファイルを選択する。あるファイルが開かれた場合、別のファイルが開かれるまで、その内容がメモリ内に読み込まれ、**sfplay~**に1を入力することで、データの先頭から再生することができる。**open**メッセージが入力されると、それまで読み込まれていたファイルは開かれたままであり、**preload**メッセージでキューを指定する時に、ファイル名前を指定して参照することができる。どのキューも直前のファイルに対して定義されたものである場合、新しいファイルが開かれた後でも、それらキューの定義は有効である。

- openraw** 第1インレット：**openraw**メッセージは**open**と全く同じ機能を持つが、任意のファイルを開いて、現在のファイルとすることができる。指定したファイルは44100Hzのサンプリング・レートを持つ16ビットのステレオで、ヘッダ情報がない（ファイル先頭から実際のサンプル・データまでのオフセットが0）ファイルとみなして読み込まれる。**samptype**、**offset**、**srate**や**srchans**のメッセージによってファイル・タイプが明示的に指定されている場合は、それらの情報に従って読み込まれる。
- pause** 第1インレット：**pause**メッセージは、オーディオ・ファイルの再生をその時点の再生位置で一時停止にする。再生の再開は**resume**メッセージで行える。
- preload** 第1インレット：2以上の整数とともに指定することで、ファイル中の指定領域に対するキューを定義する。キュー番号が定義された後、そのキュー番号を指定すると、対応するファイルの領域が再生される。キュー番号1は常に現在のファイル（最も最近**open**メッセージで開いたファイル）の先頭であり、**preload**メッセージで変更することはできない。

**preload**メッセージには多くの入力形式がある。**preload**に続いて2から32767のキュー番号を指定することは必須である。そのキュー番号の後ろに再生するファイル名を指定する。そのファイルは現在既に関いているファイルか、またはMaxのサーチ・パス上にあるファイルを指定し、キュー定義後はそのキュー番号にそのファイルが対応付けられる。ここで、キューで使うために**open**メッセージによって明示的に開かれたファイルについては**preload**メッセージでのファイルの指定は必要ないことに注意が必要である。その場合、ファイル名の指定がなければ、現在開かれているファイルが使われることになる。

オプションとしてファイル名を指定した後は、そのファイルの再生開始位置をミリ秒単位で指定することができる。再生開始位置が指定されない場合、キューの再生位置はファイルの先頭に設定される。再生開始位置の後には、再生終了位置をミリ秒単位で指定することができる。再生終了位置を指定しない場合、または0を指定した場合、キューはファイルの最後まで再生する。再生終了位置が開始位置より前の時刻を指定している場合、キューは定義できても再生されない。将来的にはそれで逆再生のキューが指定できるようになるかもしれない。

開始・終了位置のアーギュメントの後には定義したキューを逆再生できるように双方向バッファ・フラグをオプションとして指定することができる。このフラグを1にすると逆再生が可能なキューとなる。デフォルトの設定では0である（双方向バッファリングはオフ）。

最後のオプションであるアーギュメントは再生速度を設定するのに使われる。オブジェクト全体に設定された再生速度に対し、相対速度を1つの実数で設定する。デフォルト値は1である。

定義した各キューには、**sfplay~**でのチャンネルごとにおおよそ40Kバ



イトのメモリが必要で、双方向バッファリングがオフの時にはデフォルトのバッファ・サイズ（40,320バイト）である。双方向バッファリングがオンの場合は、各キューのメモリ量は倍になる。

**preload**メッセージは、常に低い優先順位として実行される。**pause**や**resume**、整数のメッセージはそうではない。オーバードライブ・モードでこれらのメッセージを利用しようとして問題がある場合（最も最近のキューが事前に設定される前に再生される場合など）は、**defer**オブジェクトを使うと良い。

**print** オブジェクトの状態を表示すると同時に、現時点で定義されたすべてのキューのリストを表示する。

**resume** 再生が一時停止状態にある場合、一時停止位置から再生を再開する。

**samptype** 第1インレット：**samptype**に続けてサンプル・タイプを表すシンボルを指定する。このサンプル・タイプはオーディオ・ファイルのサンプル・データを解釈する時に使われる（実際のサンプル・タイプを解釈する上で非常に重要な値である）。これは時には“header munging”（ヘッダー情報の書き換え）と呼ばれる。**openraw**メッセージによってファイルが読み込まれる場合、特に指定しないと16ビット整数のサンプル・タイプとして読み込もうとする。**samptype**メッセージによるサンプル・タイプの指定はディスク上のファイルを修正するわけではない。

サンプル・タイプとしては次に示すものがサポートされている。

|                |                      |
|----------------|----------------------|
| <b>int8</b>    | 8ビット整数               |
| <b>int16</b>   | 16ビット整数              |
| <b>int24</b>   | 24ビット整数              |
| <b>int32</b>   | 32ビット整数              |
| <b>float32</b> | 32ビット浮動小数点           |
| <b>float64</b> | 64ビット浮動小数点           |
| <b>mulaw</b>   | 8ビット $\mu$ -lawエンコード |
| <b>alaw</b>    | 8ビットa-lawエンコード       |

**seek** 第1インレット：**seek**に続けて再生開始位置をミリ秒単位で指定することで、現在のファイル上で指定された地点から再生が始められる。再生開始位置の後にオプションとして再生終了位置も指定して、再生を停止する位置を設定することができる。**seek**メッセージはキューの開始および終了位置を確認したり、正確な位置合わせをすることに使われたりする。

注記：**seek**メッセージは常に低い優先順位として実行される。オーバードライブ・モードでこのメッセージを利用しようとして問題がある場合（新たな地点を探し終える前に他の処理が始まる場合など）は、**defer**オブジェクトを使うと良い。

- set** 第1インレット：**set**に続けて**sflist~**オブジェクトの参照名を指定することで、**sfplay~**オブジェクトに整数またはリストを入力した場合に、参照先の**sflist~**に格納されたキューを再生するようにする。参照名の記述なしで**set**メッセージを**sfplay~**に入力した場合は、それまで設定されていた**sflist~**への参照をリセットし、その**sfplay~**オブジェクトで定義されたキューを利用して再生処理が行われるようになる。
- speed** 第1インレット：**speed**に続けて1つの数値を指定し、オブジェクト内のすべてのキューに対して再生速度の乗数を設定する。値が1.0（デフォルト値）の場合はすべて通常速度で再生し、-1は通常速度での逆再生を意味する。2は本来の倍の速度での再生を表し、0.5は本来の半分の速度ということになる。例えば、あるキューのその時点での再生速度が2だった場合、次に**speed**メッセージで3を設定すると、そのキューは通常速度の6倍の速度で再生することになる。
- srate** **srate**に続けてサンプリング・レート（Hz:ヘルツ）の数値を設定する。この数値はこの数値はオーディオ・ファイル中のサンプル・データを解釈する時に使われる（実際のサンプリング・レートを解釈する上で非常に重要な値である）。これは時には“header munging”（ヘッダー情報の書き換え）と呼ばれる。**openraw**メッセージでファイルを読み込む場合、特に指定しない限りサンプリング・レートは44,100Hzとして読み込もうとする。**srate**メッセージによるサンプリング・レートの設定は、ディスク上のファイルを修正するわけではない。
- srcchans** **srcchans**に続いてチャンネル数の数値を指定する。この数値はオーディオ・ファイル中のサンプル・データを解釈する時に使われる（実際のチャンネル数を解釈する上で非常に重要な値である）。これは時には“header munging”（ヘッダー情報の書き換え）と呼ばれる。**openraw**メッセージによってファイルが読み込まれる場合、特に指定しないとチャンネル数は2として読み込もうとする。**srcchans**メッセージによるチャンネル数の指定はディスク上のファイルを修正するわけではない。

## アーギュメント

- シンボル** オプション。第1アーギュメントがシンボルである場合、**sflist~**の参照名と解釈し、**sfplay~**は対応する**sflist~**で設定されたキューを再生するようになる。シンボルのアーギュメントがない場合は、**sfplay~**自身の内部で定義されたキューを再生することとなる。
- 整数** オプション。出力チャンネル数を設定する。この値は**sfplay~**オブジェクトが持つシグナルのアウトレット数を定義することになる。チャンネルの最大数は16で、デフォルト値は1となる。読み込んだオーディオ・ファイルが設定したよりも多くのチャンネル数を持っている場合、設定以上のチャンネルについては再生されない。オーディオ・ファイル側のほうがチャンネル数が少ない場合、余ったアウトレットからは0のシグナルが出力される。



追加オプションのアーギュメントとして、サンプル数で表すディスク・バッファ・サイズを指定することができる。このアーギュメントの値が0であると、デフォルトのディスク・バッファ・サイズが利用される。

更に追加オプションのアーギュメントとして、生成されるアウトレットの表示順番の情報を指定することができる。最後のアーギュメントに1を指定すると、1つのシグナル用アウトレットが1番右側から1つ目の位置に生成される。1番右のアウトレットは再生終了または停止した場合に**bang**を出力するためのアウトレットであり、現在の再生位置をミリ秒単位の数値で出力するものでもある。

MSP上のすべてのオーディオ・シグナルと同様、この再生位置の数値は32ビット単精度浮動小数点で表されるシグナルである。より高い精度の再生位置の数値を必要とするならば、最後のアーギュメントに2を指定することで、2つ目のアウトレットは単精度の丸め込み誤差を含んだ32ビット単精度浮動小数点のシグナルを出力するアウトレットを生成する。これら2つのシグナルを一緒に使うことで、倍精度浮動小数点に近い精度の数値を得ることができる（注記：数分後には単精度浮動小数点の値は実際のサンプルとは合わなくなる）。これら2つのシグナルを、**ht.+~**オブジェクトのように、Max/MSPは未サポートだが高精度なシグナル処理オブジェクトとともに利用すると、正確なファイルの位置によるサンプル位置を計算できる可能性がある。

**シンボル** オプション。最後のアーギュメントがシンボルである場合、その名前は他のオブジェクトが**sfplay~**オブジェクトを参照してアクセスするための名前として設定される。

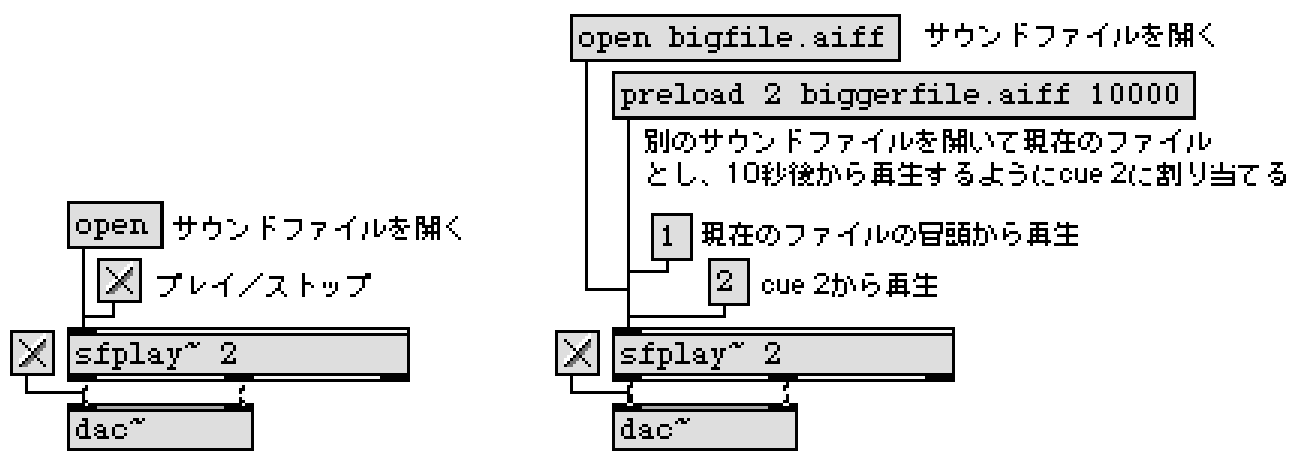
## 出力

**シグナル** インレットにキュー番号が入力されると、オーディオ・ファイルの各チャンネルのデータが出力される。アウトレットは、出力チャンネルとして指定された数（**sfplay~**自身またはアーギュメントとして設定される数）に個々に対応したものが（第1アウトレットがチャンネル1という具合に）生成されたものである。

出力位置に関するオプション・アーギュメントを指定する場合、シグナル出力の表示位置の情報が1つか2つ記述されることになる。そのアーギュメントが1である場合、1つのシグナル用アウトレットが1番右側から1つ目の位置に生成される。1番右のアウトレットは、再生終了または停止した場合に**bang**を出力するためのアウトレットであり、現在の再生位置をミリ秒単位の数値で出力するものでもある。最後のアーギュメントを2とした場合、2つ目のアウトレットが生成され、単精度の丸め込み誤差を含んだ32ビット単精度浮動小数点のシグナルを出力するために使われる（詳細は**sfplay~**のアーギュメントのアウトレットの位置に関する欄を参照のこと）。

**bang** 一番右側のアウトレット：ファイルの再生が終了、または0メッセージによって停止された場合、**bang**が出力される。

## 利用例



オーディオ・ファイルはそのすべてが読み込まれるわけではなく、ハード・ディスクから再生される

## 参照

|                  |                        |
|------------------|------------------------|
| <b>buffer~</b>   | オーディオ・サンプルの格納          |
| <b>groove~</b>   | 速度可変のループ再生             |
| <b>play~</b>     | ポジション・ベースのサンプル再生       |
| <b>sfinfo~</b>   | オーディオ・ファイルの情報を出力する     |
| <b>sflist~</b>   | オーディオ・ファイルのキューを格納する    |
| <b>sfrecord~</b> | ディスクへオーディオ・ファイルとして録音する |

Tutorial 16

Sampling: Record and play audio files (英文PDF参照)

## 入力

- open** 第1インレット：録音先となるファイルを開く。デフォルトではファイル・タイプはAIFFであるが、Sound Designer II (Macintoshのみ) や NeXT/Sun、WAV形式などもサポートしている。ファイル名のアーギュメントなしで**open**メッセージが入力された場合は、標準ファイル保存ダイアログが開くので、そこでファイル名を設定する。オプションのシンボルとして**aiff**、**au**、**raw**、**sd2** (Macintoshのみ)、**wave**を指定することで、各ファイル形式が設定される (標準ファイル保存ダイアログでも、ポップアップ・メニューによってファイル形式を指定できる)。**open**に続いて、それらファイル形式以外のシンボルが指定された場合は、現在のデフォルト・ボリュームにファイルを作成する。同じ名前のファイルが既に存在していた場合は、上書きする。ファイル形式のシンボル (**sd2**など) を、ファイル名のアーギュメントの後に指定することもできる。
- 整数** 第1インレット：**open**または**opensd2**メッセージでファイルが既に開かれている場合は、0でない値を入力することで、録音を開始する。そして、0で録音を停止してファイルを閉じる。0を入力した後に、再び録音するには、もう一度**open**か**opensd2**メッセージでファイルを指定する必要がある。
- 録音中にハードディスク容量がなくなった場合など、何らかのエラーが起これると、録音を自動的に終了することもある。
- loop** 第1インレット：**loop**に続き1を指定することで、ループ処理をオンにする。**loop 0**はオフにする。デフォルトではループ処理はオフである。
- nchans** **nchans**に続いて1から16までの整数を指定し、録音されるオーディオ・ファイルのチャンネル数を設定する。デフォルト値は1である。
- print** 録音の進行状況に関する内部情報をMaxウィンドウに表示する。
- record** 第1インレット：**open**または**opensd2**メッセージでファイルが開いている場合、**record**に続けてミリ秒単位の時間を指定することで、録音を開始して指定時間後に録音を終了する。その指定時間の録音が終了する前に**sfrecord~**の第1インレットに0を入力することで録音を停止することができる。
- resample** **resample**に続き1つの実数を指定することで、ファイルのアップサンプルまたはダウンサンプルを行う。変換後のサンプル・レートは浮動小数点の値で指定する。1.0は現在のサンプル・レートであり、0.5は現在の半分、2.0は2倍、というように指定する。
- samptype** 第1インレット：**samptype**に続いてシンボルを指定し、録音するオーデ

ィオ・ファイルのサンプル形式を設定する（オーディオ・ファイルへ実際に記録するサンプル形式として非常に重要な値である）。これは時には“header munging”（ヘッダー情報の書き換え）と呼ばれる。**openraw**メッセージによってファイルが開かれる場合、特に指定しないと16ビット整数のサンプル・タイプとして録音する。

次に示すサンプル・タイプがサポートされている。

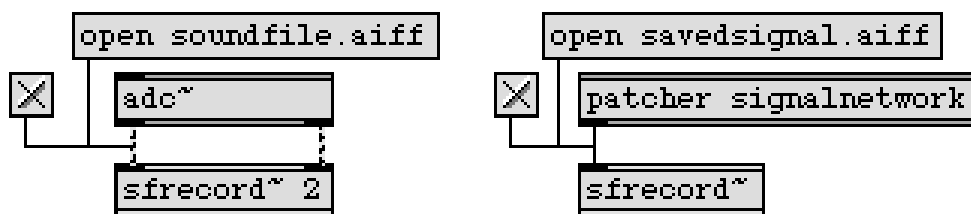
|                |                      |
|----------------|----------------------|
| <b>int8</b>    | 8ビット整数               |
| <b>int16</b>   | 16ビット整数              |
| <b>int24</b>   | 24ビット整数              |
| <b>int32</b>   | 32ビット整数              |
| <b>float32</b> | 32ビット浮動小数点           |
| <b>float64</b> | 64ビット浮動小数点           |
| <b>mulaw</b>   | 8ビット $\mu$ -lawエンコード |
| <b>alaw</b>    | 8ビットa-lawエンコード       |

**シグナル** 各インレットはオーディオ・ファイルの各チャンネルに対応しており、録音が始まるとシグナルの入力を受け付ける。

## アーギュメント

**整数** オプション。インレットの数である入力チャンネルの数を設定する。チャンネルの最大数は16で、デフォルト値は1である。作成されるオーディオ・ファイルは、ここで指定した数と同じチャンネルを持つことになる。実際に何チャンネルまで同時録音可能かは、CPUおよびハードディスクの速度によって異なる。

## 利用例



入力されるサウンドやMSPで作られたサウンドを、オーディオ・ファイルへ保存する

## 参照

### sfplay~

ディスクからオーディオ・ファイルを再生する

### Tutorial 16

Sampling: Record and play audio files（英文PDF参照）

## 入力

**整数または実数** 一定の値のシグナルとして出力する数値

**シグナル** 一般のシグナルの入力は受け付けない。唯一 **begin~** の出力だけを接続することができ、**selector~** を使ってシグナルの切り替えを行う時に、**begin~** によって切り替える各シグナル処理の開始位置を宣言することができる。

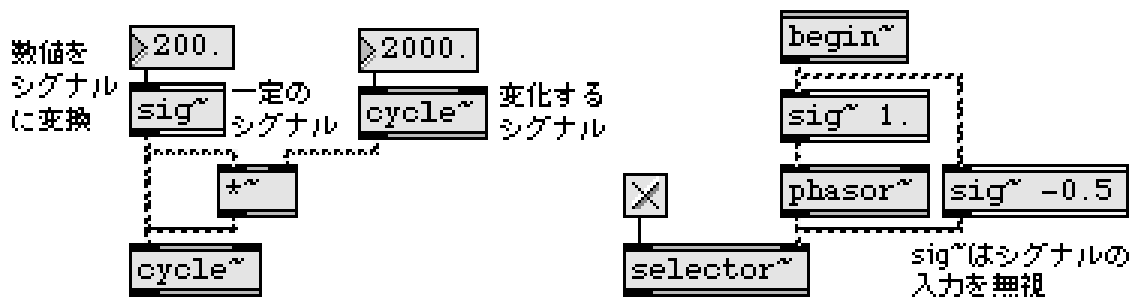
## アーギュメント

**整数または実数** オプション。出力シグナルの初期値を指定する。

## 出力

**シグナル** アーギュメントで指定した数値、または最も最近にインレットへ入力した整数か実数による一定の値のシグナルが出力される。

## 利用例



sig~ を利用することで一定の値のシグナルが扱える

## 参照

**+~** シグナルの加算  
**begin~** シグナル・ネットワークの切り替え可能な部分を定義する  
**line~** シグナル・ランプ・ジェネレーター

Tutorial 4 Fundamentals: Routing signals (英文PDF参照)

## 入力

シグナル 双曲線サイン関数への入力。

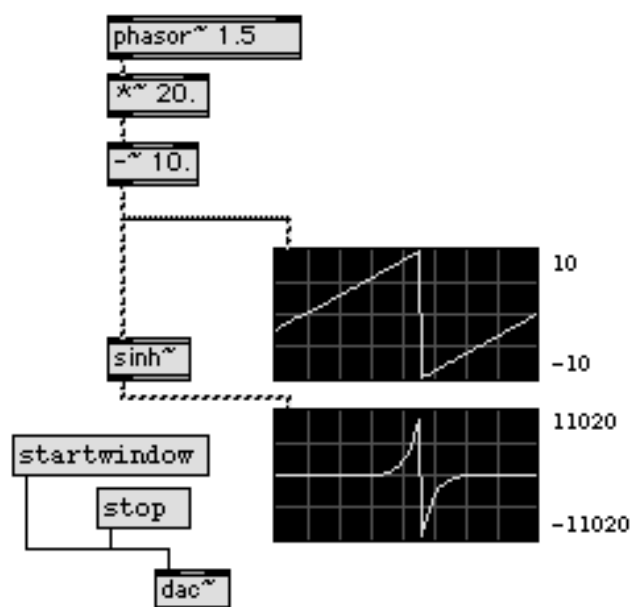
## アーギュメント

なし。

## 出力

シグナル 入力の双曲線サイン値。

## 利用例



sinh~ を用いて、オシレーターに同期した面白いオーディオ・コントロール・シグナルを作る

## 参照

|               |                         |
|---------------|-------------------------|
| <b>acos~</b>  | シグナルのアーク・コサイン関数         |
| <b>asin~</b>  | シグナルのアーク・サイン関数          |
| <b>atan~</b>  | シグナルのアーク・タンジェント関数       |
| <b>atan2~</b> | シグナルのアーク・タンジェント関数 (2変数) |
| <b>cos~</b>   | シグナル (0-1の範囲) のコサイン関数   |
| <b>cosh~</b>  | シグナルの双曲線コサイン関数          |
| <b>cosx~</b>  | シグナルのコサイン関数             |
| <b>sinh~</b>  | シグナルの双曲線サイン関数           |
| <b>sinx~</b>  | シグナルのサイン値               |
| <b>tanh~</b>  | シグナルの双曲線タンジェント関数        |
| <b>tanx~</b>  | シグナルのタンジェント関数           |

## 入力

シグナル サイン関数への入力。

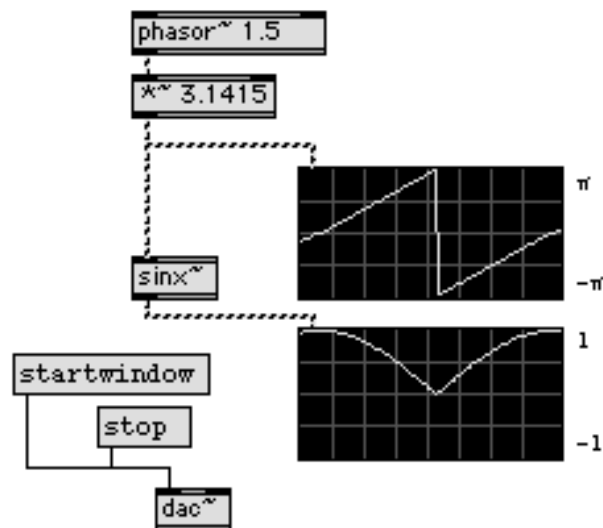
## アーギュメント

なし。

## 出力

シグナル 入力のサイン値。

## 利用例



sinx~ を用いて、オーディオ・コントロール・シグナルとしてのサイクロイドを作る

## 参照

|               |                         |
|---------------|-------------------------|
| <b>acos~</b>  | シグナルのアーク・コサイン関数         |
| <b>asin~</b>  | シグナルのアーク・サイン関数          |
| <b>atan~</b>  | シグナルのアーク・タンジェント関数       |
| <b>atan2~</b> | シグナルのアーク・タンジェント関数 (2変数) |
| <b>cos~</b>   | シグナル (0-1の範囲) のコサイン関数   |
| <b>cosh~</b>  | シグナルの双曲線コサイン関数          |
| <b>cosx~</b>  | シグナルのコサイン関数             |
| <b>sinh~</b>  | シグナルの双曲線サイン関数           |
| <b>tanh~</b>  | シグナルの双曲線タンジェント関数        |
| <b>tanx~</b>  | シグナルのタンジェント関数           |



## 入力

**シグナル** フィルター処理されるシグナル。新しい値が入力されると、**slide~**はその入力シグナルに対して、次の式で表される処理を行って対数的なフィルター処理を行う。

$$y(n) = y(n-1) + ((x(n) - y(n-1))/slide)$$

**slide~**から出力されるサンプル値は、入力シグナル値とその直前の出力サンプル値の差分をスライド値 (Slide) で割った値を、直前の出力サンプル値に加えたものとなる。スライド値が**1**の場合、出力は入力と同じとなる。スライド値が**10**の場合、出力シグナルは入力の変化に対して速くとも1/10の変化しか表れない。この処理は特にローパス・フィルターやエンベロープ・フォロワーの処理として有効となり得る。

**実数** 第2インレット：入力値が直前の値より大きい場合に、スライド値として使われる値 (スライド・アップ値) を指定する。

第3インレット：入力値が直前の値より小さい場合に、スライド値として使われる値 (スライド・ダウン値) を指定する。

## アーギュメント

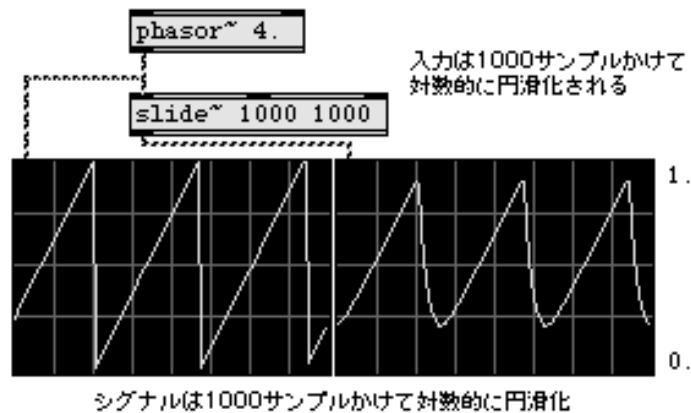
**実数** オプション。スライド・アップ値を設定する。デフォルト値は1である。

**実数** オプション。第2アーギュメントはスライド・ダウン値を設定する。デフォルト値は1である。

## 出力

**シグナル** フィルター処理されたシグナル。

## 利用例



**slide~**は入力シグナルに対して対数的なスムーズ処理を施す

## 参照

**rampsmooth~** 入力シグナルにスムーズ処理を行う

## 入力

- シグナル** 第1インレット：シグナルの値がサンプリングされて、アウトレットから出力される。
- 整数、または実数** 第1インレット：すべての0でない数値はオブジェクトの内部時計をオンにし、0はオフにする。正の値として内部時計の時間間隔が与えられている時、内部時計の初期状態はオンである。
- 第2インレット：入力シグナルからの値を自動的に出力するトリガーの内部時計の時間間隔をミリ秒単位で設定する。時間間隔が0の場合、内部時計は停止する。時間間隔が正の整数の場合、データを出力する割合を変化させる。
- bang** 現在のシグナルの値を出力する。
- offset** 数値を伴う**offset**は、**snapshot~**オブジェクトがアウトレットから出力する時に、出力されるシグナル・ベクターの中の数値を設定する。数値は0（デフォルト）から現在のシグナル・ベクター・サイズの間の値に強制的に整えられると。

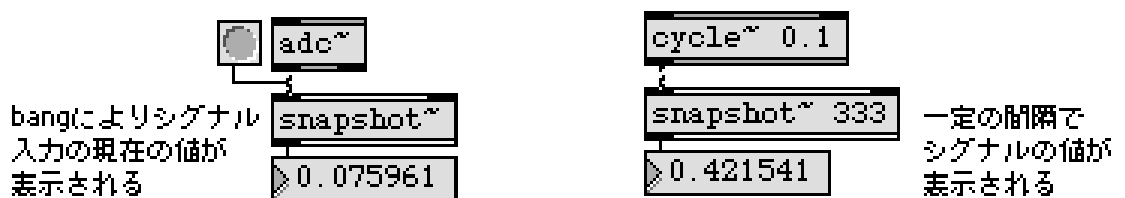
## アーギュメント

- 整数** オプション。最初のアーギュメントは内部時計の時間間隔を設定する。0の場合は、内部時計は使用されない。この場合、**snapshot~**オブジェクトは**bang**メッセージを受け取った時にのみデータを出力する。デフォルトの時間間隔は0である。2番目のアーギュメントは出力されるシグナル・ベクター内のサンプル番号を設定する。

## 出力

- 実数** **snapshot~**が**bang**を受け取った時、あるいは内部時計がオンの時は、入力シグナルのサンプル値がアウトレットから出力される。

## 利用例



ある瞬間のシグナルのサンプル値を見る

## 参照

### **capture sig~**

数値を格納し、表示・編集する  
一定の値のシグナル

Tutorial 23

Analysys : Viewing signal data (英文PDF参照)

## 入力

**シグナル** 第1インレット：解析するシグナルを入力する。**spike~**オブジェクトは、入力シグナルを解析してゼロである値からゼロでない値への変化を検出して、その時間間隔をミリ秒単位で出力する。頻繁に変化を検出するような場合には、出力を行わない時間を設定することができ、そのように設定した場合は、一定時間の後に時間間隔が出力される。

**整数または実数** 第2インレット：出力を行わない時間をミリ秒単位で指定する。これによって、シグナルの変化が検出された後、指定した時間の間はシグナルの変化を出力しない。この時間が過ぎると、次にゼロである値からゼロでない値へ変化した時に、その時間間隔を出力する。出力を行わない時間のデフォルト値は0である。

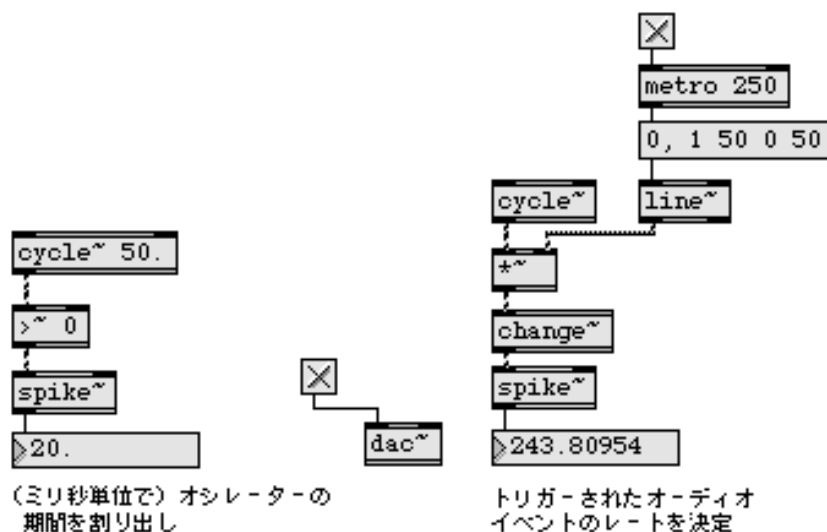
## アーギュメント

**整数または実数** オプション。出力を行わない時間を設定できる（上記参照）。

## 出力

**実数** ゼロである値からゼロでない値へとシグナルの値が変化した時に、その時間間隔をミリ秒単位で出力する。（出力を行わない時間が設定されている場合はその時間も含めた値となる）

## 利用例



**spike~** は、入力シグナルに対してどれだけ頻繁にゼロからゼロでない値へと変化しているかを出力する

## 参照

**change~** シグナル値の変化方向を出力する  
**edge~** シグナルの論理的变化を検出する  
**zerox~** ゼロ地点の通過を検出する

## 入力

シグナル **sqrt~** は、入力されたシグナルの平方根であるシグナルを出力する。入力が負の場合、解は実在しないので、0が出力される。

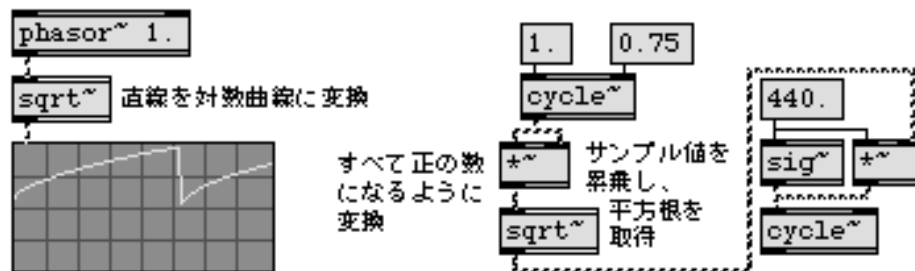
## アーギュメント

なし。

## 出力

シグナル 入力の平方根。

## 利用例



入力されたシグナルの平方根を、シグナルとして出力

## 参照

**curve~**  
**log~**  
**pow~**

指数曲線、対数曲線のランプ・ジェネレーター  
シグナルの対数計算をする  
シグナルの累乗計算をする

## 入力

- シグナル** 第1インレット：第1インレットから入力されるシグナルは録音用バッファに格納される。これは再生用バッファにコピーされ、再生用の音源として使用することができる。
- 第2インレット：正または負の数で指定されるトリガーとなるシグナルを受け取る。シグナルが正しい方向で極性を変化させる場合、第1インレットから録音されたサンプルが再生バッファにコピーされる。
- 第3インレットとその後に続くインレット：それぞれのインレットに0から1までの範囲で入力される位相としてのシグナルは、そのインレットに対応する再生バッファの出力スピードをコントロールする。
- bang** 第1インレット：**bang**によって、録音されたサンプルの最後のバッファを再生バッファにコピーする。
- 第2インレットのトリガーとなるシグナルの代わりに、またはそれと共に、**bang**を使用することができる。
- 整数** 第1インレット：再生バッファのサイズをサンプル単位で指定する。これは**stutter~**オブジェクトの最初のアーギュメントによって決定される最大のメモリーまでの数値として指定できる。
- polarity** 0か1を伴う**polarity**は**stuffer~**のトリガーの極性を、負と正とのそれぞれに変換する。
- ampvar** 実数を伴う**ampvar**は、出力シグナルのランダムな振幅変化を指定する。デフォルト値は0である（変化なし）。
- dropout** 実数を伴う**dropout**は、再生シグナルがドロップアウト（断続的に再生するか、再生していないこと）する可能性の割合を決定する。デフォルト値は0である（ドロップアウトは起こらない）。
- repeat** 実数を伴う**repeat**は、以前の再生バッファをリピートさせるために、入力バッファが再生バッファにコピーされない可能性の割合を決定する。デフォルト値は0である（リピートされない）。
- setbuf** バッファ名のアーギュメント、サンプル・オフセット、チャンネルを伴う**setbuf**は、サンプルを指定した名前を持つ**buffer~**オブジェクトにコピーする。

注記：**stutter~**オブジェクトは、常に再生バッファとして内部バッファを使用する。違う方法で使用するためには必要に応じて、コピーし

たサンプルを、名前を付けた**buffer~**オブジェクトに送ることもできる。指定したメモリー量からバッファにコピーするために必要となる時間は $n/m$ である。ここでの $n$ はコピーするサンプルの数で、 $m$ は**stutter~**オブジェクトの4つ目のアーギュメントである。

## アーギュメント

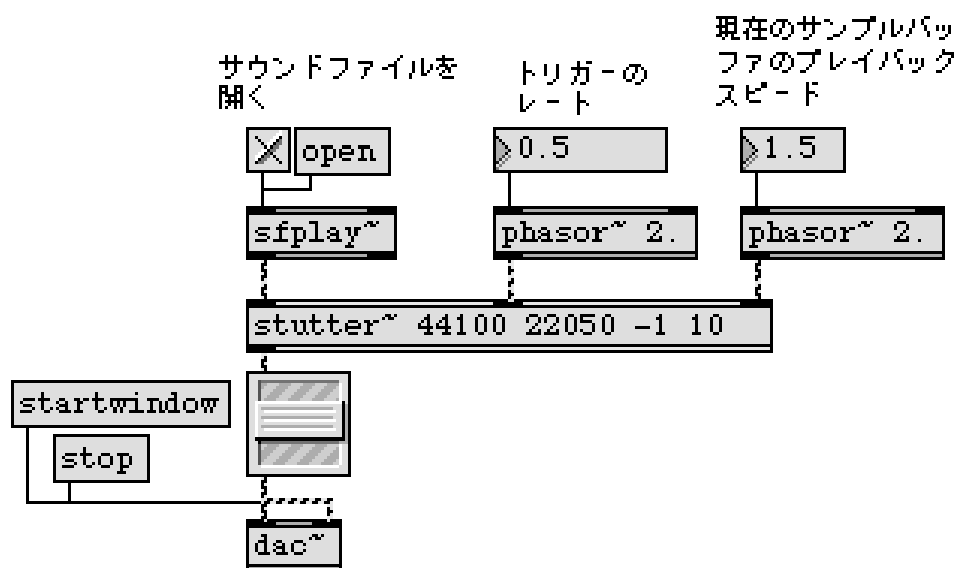
- 整数** 必須。サンプル単位でのバッファの長さの最大値。これは、入力バッファのメモリー・サイズを決定する。オブジェクトがトリガーされた時に、入力バッファの一部が再生バッファにコピーされる。
- 整数** 必須。トリガーを受けて、録音したものから再生バッファにコピーするためのサンプル単位でのバッファ・サイズの初期値。
- 整数** 必須。第2インレットに入力されるトリガー・シグナルに反応するために用いる極性。アーギュメントが0以上の場合、**stutter~**は正のトリガーを受け付ける。そうでない場合、**stutter~**は負のトリガーを受け付ける。
- 整数** 必須。オーディオ・シグナルの繰り返しを実行する場合に、入力バッファから再生バッファにコピーされるサンプル数（シグナル・ベクター・サイズ）。値が大きくなるほど、**stutter~**オブジェクトのメモリーの負荷は減少するが、CPUの負荷は増加する。整数 オプション。5番目のアーギュメントは、再生バッファから再生する時に、**stutter~**オブジェクトが独立した複数のシグナルとして出力する数を設定する。デフォルト値は1で、最大値は30である。**stutter~**オブジェクトに入力する位相シグナルの数値も、このアーギュメントによって決まる。

## 出力

- シグナル** すべてのアウトレット：**stutter~**オブジェクトのアウトレットは再生バッファからのシグナルを出力する。再生位置とスピードは再生用アウトレットに対応する位相入力によって決定される。アウトレットの数は、**stutter~**オブジェクトの5番目のアーギュメントによって決定される。



## 利用例



**stutter~**はトリガーを受け取った時には常に、入力している新しい音の断片をオシレーターとして用いるバッファに格納する

## 参照

**buffer~**  
**phasor~**  
**record~**

オーディオ・サンプルの格納  
ノコギリ波の生成  
サウンドを**buffer~**のサンプル・メモリへ録音する

**svf~**オブジェクトはHal Camberlinの書籍「Musical Applications of Microprocessors」の中で定義されている状況可変フィルターのアルゴリズムを実装したものである。このフィルター・オブジェクトのユニークな特徴はローパス、ハイパス、バンドパス、そしてバンドリジェクト（ノッチ）を同時に4つすべて出力することができる点である。

## 入力

- シグナル** 第1インレット：フィルター処理が行われるシグナル。
- 第2インレット：フィルター処理を行う中心の周波数をヘルツ単位で設定する。
- 第3インレット：バンドパス・フィルターの“Q”（おおよそのフィルターの鋭さ）を設定する。Qは中心の周波数によって分けられるフィルター・バンド幅を定義する。利用価値が高いQの値は一般的に0.01から500の間にある。
- 実数** 第2、第3インレット：実数はフィルターの中心の周波数とQの値を変化させるために入力される。デフォルトでは中心の周波数はヘルツ単位で表される。設定する値の許容範囲は0から現在のサンプリングレートの4分の1までである。便宜のために、**svf~**オブジェクトは追加の2つの入力モードを備えている。それはより慣習的な入力範囲である0~1を使用するものである（詳細は**linear**及び**radians**のメッセージを参照のこと）。シグナルがインレットの内一つに接続されている場合は、そのインレットに入力される数値は無視される。このインレットにシグナルが接続される場合、その値はシグナル・ベクター毎にサンプルされる。
- Hz** 各インレット：周波数の入力モードをヘルツ単位に設定する（デフォルト）
- linear** 各インレット：周波数の入力モードをリニアに設定する（0~1）。リニア・モードは標準のHzモードの比率を単純に変更したものである。0~1の範囲以外の値は全周波数範囲を超えることになる。
- radians** 各インレット：周波数の入力モードをラジアン単位に設定する（0~1）。ラジアン・モードは式の中心の周波数を直接設定できる。入力シグナルが同じ範囲（0~1）を持つ場合は、出力は人間の耳が持つ指数的なピッチ・スケールに近い曲線的な周波数レスポンスを持つ。

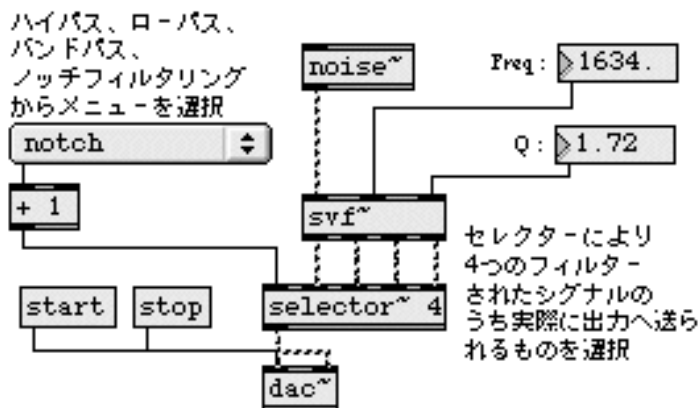
## アーギュメント

- 実数** オプション。数値は、初期のゲイン、中心の周波数、そしてQの値を設定する。デフォルト値はそれぞれ、ゲインは0、中心の周波数は0、そしてQは0.01である。
- Hz** オプション。周波数の入力モードをヘルツ単位に設定する（デフォルトのモードである。従って、アーギュメントを与えない場合も、同じである）。
- liner** オプション。周波数の入力モードをリニアに設定する（0~1）。
- radians** オプション。周波数の入力モードをラディアン単位に設定する（0~1）。

## 出力

- シグナル** フィルター処理が行われたシグナル。

## 利用例



4つのフィルターの出力を同時にsvf~オブジェクトから出力することができる

## 参照

- biquad~** 2ポール、2ゼロのフィルター
- onepole~** 単極のローパス・フィルター

## 入力

シグナル 双曲線タンジェント関数への入力。

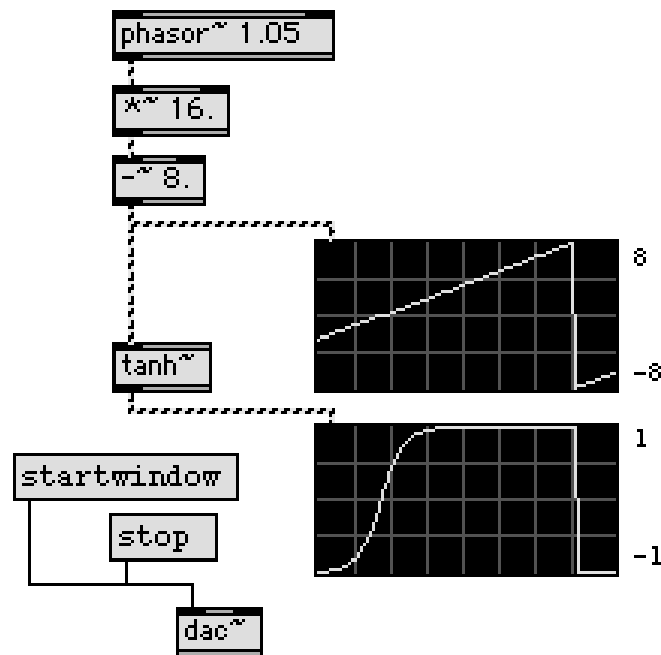
## アーギュメント

なし。

## 出力

シグナル 入力の双曲線タンジェント値。

## 利用例



tanh~を用いて、周期的なコントロール・シグナルを生成する

## 参照

- acos~ シグナルのアーク・コサイン関数
- asin~ シグナルのアーク・サイン関数
- atan~ シグナルのアーク・タンジェント関数
- atan2~ シグナルのアーク・タンジェント関数 (2変数)
- cos~ シグナル (0-1の範囲) のコサイン関数
- cosh~ シグナルの双曲線コサイン関数
- cosx~ シグナルのコサイン関数
- sinh~ シグナルの双曲線サイン関数
- sinx~ シグナルのサイン関数
- tanx~ シグナルのタンジェント関数

## 入力

シグナル タンジェント関数への入力。

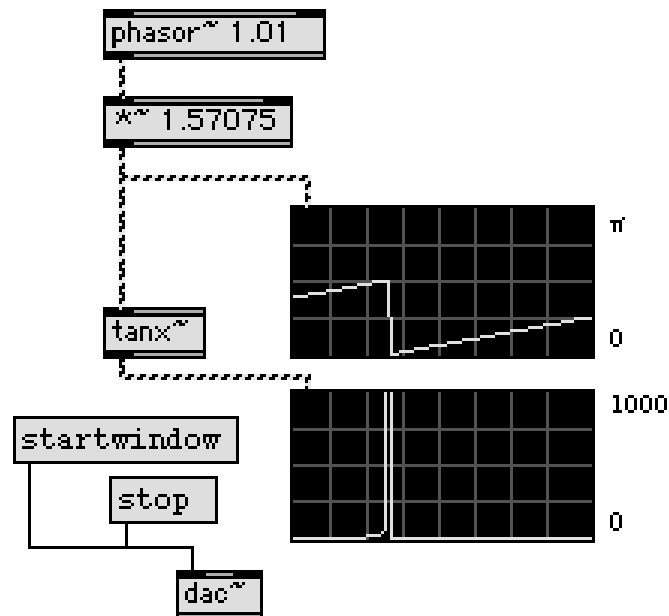
## アーギュメント

なし。

## 出力

シグナル 入力のタンジェント値。

## 利用例



tanx~ を用いて、スパイクを生成する（入力が $\pi/2$ に近付いた時に、タンジェント値は急激に増加する）

## 参照

|        |                        |
|--------|------------------------|
| acos~  | シグナルのアーク・コサイン関数        |
| asin~  | シグナルのアーク・サイン関数         |
| atan~  | シグナルのアーク・タンジェント関数      |
| atan2~ | シグナルのアーク・タンジェント関数（2変数） |
| cos~   | シグナル（0-1の範囲）のコサイン関数    |
| cosh~  | シグナルの双曲線コサイン関数         |
| cosx~  | シグナルのコサイン関数            |
| sinh~  | シグナルの双曲線サイン関数          |
| sinx~  | シグナルのサイン関数             |
| tanh~  | シグナルの双曲線タンジェント関数       |

## 入力

- シグナル** シグナルは**tapout~**オブジェクトによって読み出されるディレイ・ラインに書き込まれる
- clear** ディレイ・ラインのメモリーをクリアするが、これによって出力にクリック・ノイズが生じる場合もある。

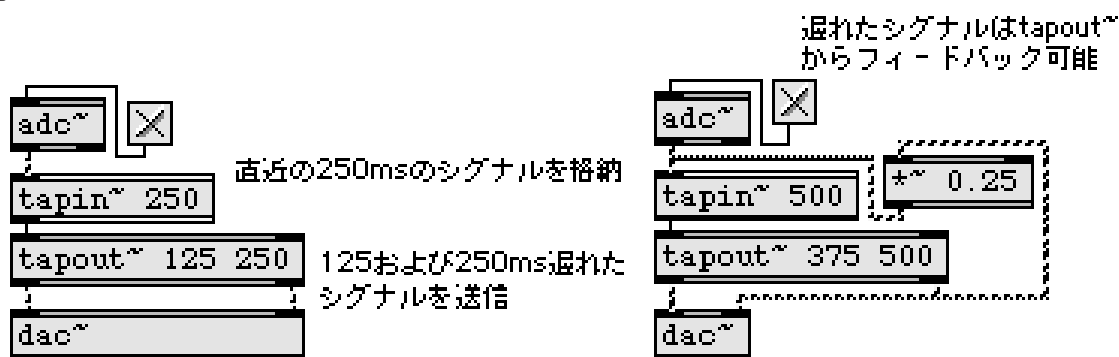
## アーギュメント

- 実数または整数** オプション。ミリ秒単位の最大ディレイ時間で、ディレイ・ラインのメモリー・サイズを決定する。オブジェクトが生成された後にサンプリング・レートが高くなった場合には、**tapin~**はディレイ・ラインをリサイズしようとする。アーギュメントがない場合には、デフォルトの最大ディレイ時間は100ミリ秒となる。

## 出力

- tap** ディレイ・ラインを機能させるためには、**tapin~**のアウトレットが**taoput~**の第1インレットに接続されていなければならない。**tapin~**のアウトレットは、他のいかなるオブジェクトにも接続することはできない。

## 利用例



**tapin~**はディレイ・バッファーを生成し、そこからディレイをかけたシグナルを取り出す

## 参照

- delay~** サンプル単位のディレイ・ライン
- tapout~** ディレイ・ラインからの出力

Tutorial 27      Processing: Delay lines (英文PDF参照)

ディレイ・ラインを機能させるため、**tapin~**オブジェクトのアウトレットは**tapout~**オブジェクトの第1インレットに接続されていなければならない。

**tapout~**は1つまたはそれ以上のインレットを持ち、1つまたはそれ以上のアウトレットを持つ。シグナルまたは数値としてインレットに入力されたディレイ時間は、インレットのすぐ下のアウトレットから出力される出力シグナルに影響する。

## 入力

**シグナル** シグナルが**tapout~**のインレットのいずれかに接続されていると、その下のアウトレットからのシグナルは、連続ディレイ・アルゴリズムを使用する。入力されるシグナルの値は、ミリ秒単位でのディレイ時間を表す。シグナルが十分ゆっくりと増加した場合には、出力のピッチは低くなり、反対にシグナルがゆっくり減少すれば、出力のピッチは高くなる。連続ディレイ・アルゴリズムは、**tapout~**のインレットにシグナルが接続されていない時に用いられる固定ディレイ・アルゴリズムよりも、計算負荷が高くなる。

**実数または整数** シグナルが**tapout~**のインレットに接続されていなければ、固定ディレイ・アルゴリズムが使用され、インレットで受け取った実数または整数によって、対応するアウトレットから出力されるシグナルのディレイ時間を設定する。ディレイ時間が変化した時に出力に、クリック・ノイズが生じる場合もある。固定ディレイはディレイ時間が大幅には変化しない残響などの多くの用途に適しており、連続ディレイ・アルゴリズムよりも計算負荷が低い。

**リスト** 第1インレット：複数の固定ディレイ時間を同時に変更することができる。リストの中の最初の数値は最初のアウトレットのディレイ時間を設定し、その後も同様である。あるインレットにシグナルが入力されていれば、対応するリストの値は無視される。

## アーギュメント

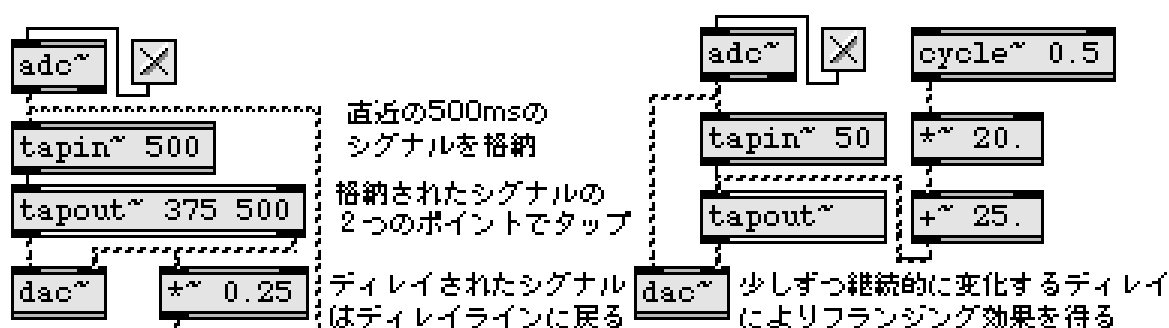
**実数または整数** オプション。ミリ秒単位の1つまたはそれ以上の初期ディレイ時間で、それぞれがディレイ（タップ）インレットとアウトレットのペアの1つ1つに対応する。例えば、**50 100 300**というアーギュメントは、3つのインレットと3つのアウトレットに対応する独立したタップを持つ**tapout~**オブジェクトを生成する。シグナルがインレットに入力されていると、対応するインレットとアウトレットのペアの初期ディレイ時間は無視される。



## 出力

シグナル **tapout~**のそれぞれのアウトレットは、**tapin~**オブジェクトによって書き込まれたディレイ・ラインの、独立してコントロールされるタップに対応する。**tapout~**のアウトレットからの出力シグナルは、アウトレットのすぐ上のインレットで受け取った数値またはシグナル・コントロールで指定されたミリ秒だけ、**tapin~**への入力を遅らせたものである。

## 利用例



**tapout~**は**tapin~**に入力されたシグナルを、一定時間だけ遅らせて出力する

## 参照

**delay~** サンプル単位のディレイ・ライン  
**tapin~** ディレイ・ラインへの入力

Tutorial 27

Processing: Delay lines (英文PDF参照)

## 入力

**シグナル** 第1インレット：フィルター処理を行うシグナル。**teeth~**オブジェクトは**comb~**の変形の1つであり、現在の入力サンプルを、以前の入力と出力の両方、またはいずれかのサンプルとミックスし、入力信号を一定の間をおいた周波数間隔で強調または減衰させるコム・フィルターとなる。**comb~**オブジェクトと異なり、**teeth~**は極端なエフェクトを加えるフィードフォワードとフィードバックを追加する。

第2インレット：フィードフォワードとして、現在の入力に入力の過去のサンプルが加算されるまでのディレイ時間をミリ秒単位で設定する。

第3インレット：フィードバックとして、現在の入力に出力の過去のサンプルが加算されるまでのディレイ時間をミリ秒単位で設定する。

第4インレット：出力に送られる入力サンプルの量をスケールリングするためのゲイン係数を決める。

第5インレット：出力に送られるフィードフォワードの量をスケールリングするためのゲイン係数を決める。

第6インレット：出力に送られるフィードバックの量をスケールリングするためのゲイン係数を決める。

**実数または整数** 第2から第6インレットのフィルター。パラメーターはシグナルではなく、実数でも指定することができる。シグナルがインレットに入力されている場合には、実数での指定は無視される。

**リスト** 5個のパラメーターを、第1インレットへリストとして入力することもできる。第1の数値はフィードフォワード・ディレイ時間、次の数値はフィードバック・ディレイ時間、第3の数値は入力サンプルのためのゲイン係数、第4の数値はフィードフォワード・ゲイン係数、第5の数値はフィードバック・ゲイン係数である。シグナルが対応するインレットに入力されている場合には、リストでそのインレットに入力されたパラメーターは無視される。

**clear** **teeth~**オブジェクトの直前の出力のメモリーを消去して、0にリセットする。

## アーギュメント

**実数** オプション。5個までの数値でフィードフォワードとフィードバックのディレイ時間、ゲイン係数、フィードフォワードとフィードバック

# teeth~

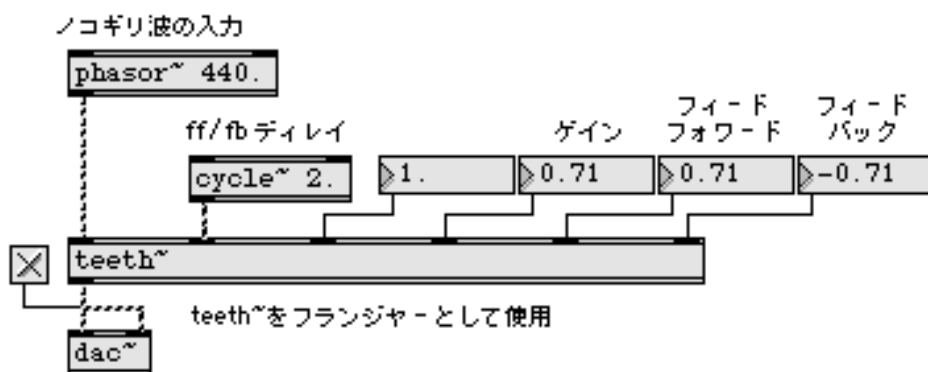
フィードフォワードとフィードバックに  
ディレイ・コントロールが付いたコム・フィルター

のゲイン係数を設定する。シグナルがあるインレットに入力されている場合には、そのインレットにアーギュメントとして設定された係数は無視される。アーギュメントがない場合には、最大ディレイ・タイムはデフォルト値の10ミリ秒に、他のすべての値はデフォルトの0になる。

## 出力

シグナル フィルター処理を行ったシグナル。

## 利用例



**teeth~** は可変のフィードフォワードとフィードバックディレイで、入力シグナルにコム・フィルター処理を行う

## 参照

**allpass~**  
**comb~**  
**delay~**  
**reson~**

オールパス・フィルター  
コム・フィルター  
サンプル単位のディレイ・ライン  
レゾナンス付きバンドパス・フィルター

**thispoly~**オブジェクトは**poly~**オブジェクトに読み込まれるパッチの内部に置かれる。このオブジェクトを読み込んでいる**poly~**オブジェクトからメッセージの入出力を行う。

### 入力

**bang** パッチのインスタンスの番号を出力する。1つ目のインスタンスの番号は1である。

**シグナル** シグナルの入力は、パッチのインスタンスのビジー状態を設定する。入力されるシグナルが0でなければ、パッチのインスタンスのビジー状態は1に設定される。シグナルの入力がなければ、ビジー状態は0に設定される。

**整数** 0か1の値は、パッチのインスタンスのビジー状態を切り替える。ビジー状態の（つまり、1に設定されている）オブジェクトは、読み込んでいる**poly~**オブジェクトの第1インレットに受け取られた**note**メッセージや**midinote**メッセージによって作り出されたメッセージを受け取らない。

**mute** **mute**メッセージは読み込まれたパッチのインスタンスにおけるDSP処理のオフ（0）とオン（1）を切り替える。このメッセージは、パッチのビジー状態を切り替える整数メッセージとともに用いて、音を鳴らす間だけ処理されるパッチのボイスを作り出すことができる。

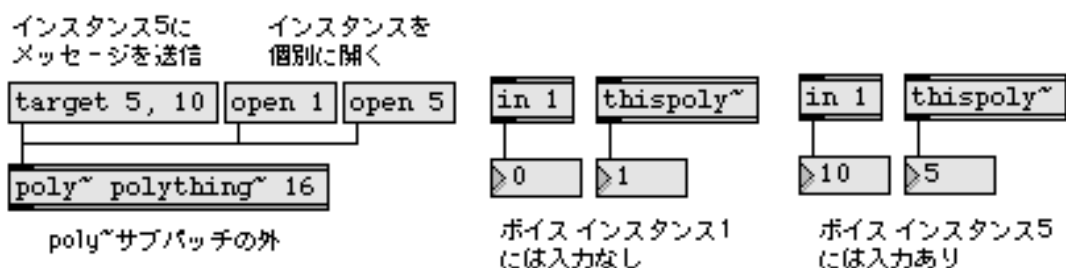
### アーギュメント

なし。

### 出力

**整数** **thispoly~**オブジェクトが**bang**を受け取った時に、1から始まるインスタンスの番号を出力する。**thispoly~**オブジェクトを含んでいるパッチが、**poly~**オブジェクトによって読み込まれていなければ、0を出力する。

### 利用例



**thispoly~**オブジェクトは、その**poly~**オブジェクトのサブ・パッチのインスタンス番号を出力する

### 参照

|              |                                              |
|--------------|----------------------------------------------|
| <b>in</b>    | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのメッセージの入力 |
| <b>in~</b>   | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのシグナルの入力  |
| <b>out</b>   | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのメッセージの出力 |
| <b>out~</b>  | <b>poly~</b> オブジェクトによって読み込まれるパッチのためのシグナルの出力  |
| <b>poly~</b> | パッチのポリフォニック処理とオーディオ処理を管理する                   |

Tutorial 21

MIDI control: Using the **poly~** object (英文PDF参照)

## 入力

- シグナル** 第1インレット：検出したいレベルを設定するシグナル。
- 実数** 第2インレット：入力シグナルに対するロー（リセット）スレッシュホールド・レベルを設定する。入力シグナルのサンプルがハイ（セット）スレッシュホールド・レベル以上の時、**thresh~**は入力シグナルのサンプルが、このリセット・レベル以下になるまで、値が1であるシグナルを出力する。
- 第3インレット：入力シグナルに対するハイ（セット）スレッシュホールド・レベルを設定する。入力シグナルがこのレベル以上の時、**thresh~**は値が1であるシグナルを出力する。

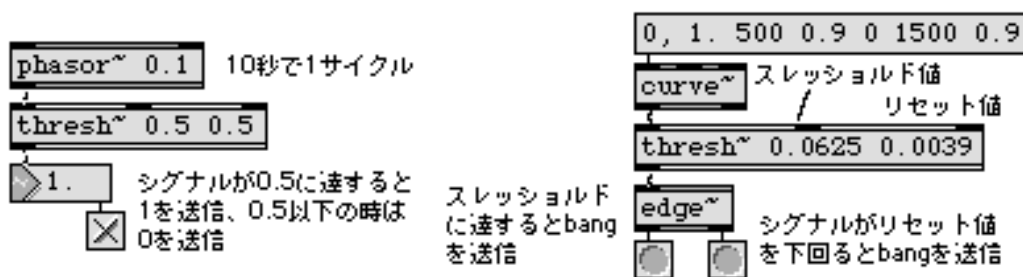
## アーギュメント

- 実数** 第1のアーギュメントはロー（リセット）スレッシュホールド・レベルを指定する。アーギュメントがない場合には、リセット・レベルは0になる。第2のアーギュメントはハイ（セット）スレッシュホールド・レベルを指定する。アーギュメントがない場合には、セット・レベルは0になる。アーギュメントが1つしかない場合には、そのアーギュメントはロー（リセット）スレッシュホールド・レベルを意味し、セット・レベルは0になる。

## 出力

- シグナル** 入力シグナルのサンプルがハイ（セット）スレッシュホールド・レベル以上の時、出力は1になる。出力は入力シグナルのサンプルがロー（リセット）スレッシュホールド・レベル以下になるまで1が続く。ハイ（セット）スレッシュホールド・レベルとロー（リセット）スレッシュホールド・レベルが同じであれば、入力シグナルのサンプルがロー（リセット）スレッシュホールド・レベル以下になるまで出力は1になる。

## 利用例



シグナルがあるレベルを超えるのを検出する

## 参照

- >~** 2つのシグナルの比較を行い、大きければ1を出力する
- change~** シグナル値の変化方向を出力する
- edge~** シグナルの論理的变化を検出する

## 入力

**シグナル** 第1インレット：ミリ秒単位で第1アウトレットから出力されるパルス波の周期（パルス・サイクルの時間間隔）を指定する。

第2インレット：パルス幅またはデューティー・サイクルをコントロールする。パルスのオンの部分（シグナルの値が1）を出し続けるパルス・インターバルの割合を表すシグナル値。この値が0であれば、最小時間だけオンであるパルス・サイズ（通常は1サンプル）で、この値が1であれば最大時間（通常はインターバル全体から1サンプルだけを除いた）のパルス・サイズとなる。この値が.5であれば、半分の時間が1で、残り半分の時間が0であるパルスを作る。

第3インレット：パルスのオンの部分の始まりの位相を設定する。この値が0であれば、インターバルの開始部分がオンとなり、他の値（1まで、1なら0と同じ）なら、全体のパルス間のインターバルの割合としてオンの部分を遅らせる。

**実数または整数** 周期、パルス幅、位相をコントロールするシグナル・オブジェクトの代わりに使用される数値。シグナルがインレットに入力されている場合には、実数または整数メッセージは無視される。

## アーギュメント

**実数または整数** オプション。パルス間のインターバルのミリ秒単位の初期値（デフォルト値は1000）、パルス幅（デフォルト値は0.5）、そして位相（デフォルト値は0）を指定できる。シグナル・オブジェクトがtrain~オブジェクトのインレットのどれかに入力されている場合には、対応する初期アーギュメントの値は無視される。

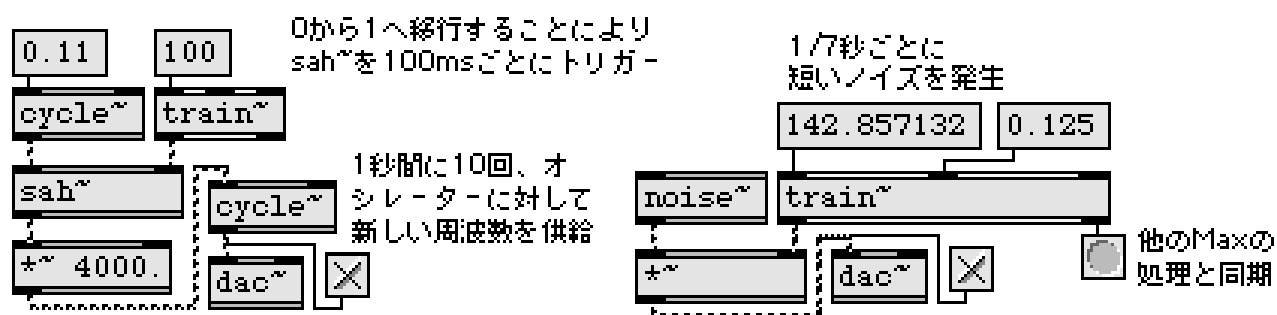
## 出力

**シグナル** 第1アウトレット：指定されたインターバル、幅、位相を持つパルス（矩形）波のシグナル。

**bang** 第2アウトレット：パルスのオン部分が始めると、**bang**が第2アウトレットから出力される。このアウトレットを用いると、浮動小数点（またはシグナル）値として指定できるインターバルを持ったシグナル同期のメトロノームとして、**train~**を使用することができる。しかしながら、パルスのオン部分と実際の**bang**メッセージの出力の間には予測不可能な遅れがある。スケジューラーのインターバルを1ミリ秒に設定すると、この遅れは1ミリ秒以内であることが保証される。



## 利用例



シグナルのリズミックな変化のための正確なパルスを出力する

## 参照

- <~ 2つのシグナルの比較を行い、小さければ1を出力する
- >~ 2つのシグナルの比較を行い、小さいか等しければ1を出力する
- clip~ シグナルの振幅に制限をかける
- phasor~ ノコギリ波の生成

## 入力

**シグナルまたは実数** 第1インレット：任意のシグナルや実数、0から1に増加するシグナルなどを入力する。これらの値は**trapezoid~**オブジェクトが持つウェーブテーブルを走査するのに使われる。**trapezoid~**は繰り返し波形のウェーブテーブルによるオシレーターとして働き、**phasor~**オブジェクトの出力や、その他のオーディオ・シグナルでも、そのコントロールとして利用することができる。

第2インレット：台形波の立ち上がり終了位置を、一波形内での位置として0から1.0の値を入力する。デフォルト値は0.1である。

第3インレット：台形波の立ち下がり開始位置を、一波形内での位置として0から1.0の値を入力する。デフォルト値は0.9である。

**lo** 第1インレット：**lo**に続き、オプションの数値として**trapezoid~**が出力するシグナルの最小値を指定する。デフォルト値は0である。

**hi** 第1インレット：**hi**に続き、オプションの数値として**trapezoid~**が出力するシグナルの最大値を指定する。デフォルト値は1.0である。

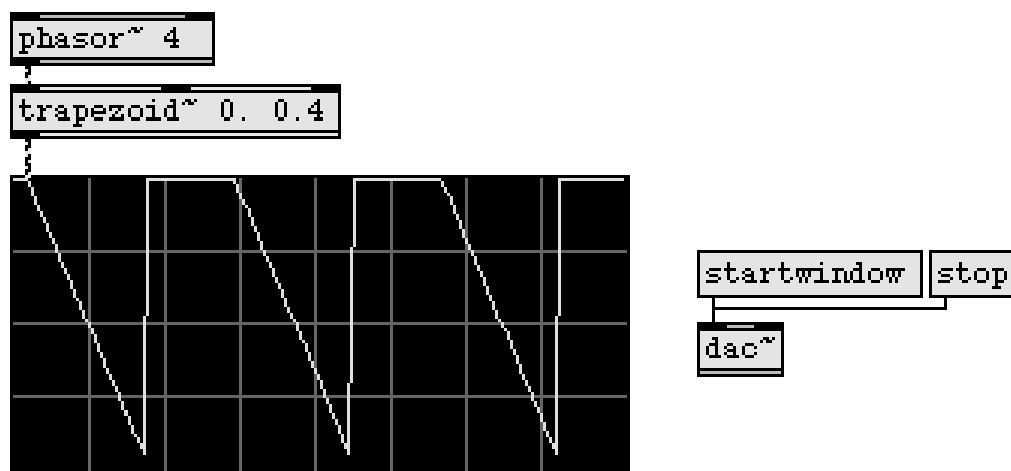
## アーギュメント

**実数** オプション。2つの実数値の組で、立ち上がる終了位置と立ち下がる開始位置を指定する。アーギュメントが **0. 0.** の場合はノコギリ波を生成し、**0.5 0.5**の場合は三角波を生成する。

## 出力

**シグナル** **trapezoid~**に入力したシグナルに対応したシグナルを出力する。**phasor~**オブジェクトの出力か、その他オーディオ・シグナルが**trapezoid~**の走査に使われた場合、その出力は周期的な波形となる。

## 利用例



**trapezoid~**は波形の変化点を設定し、台形波形を生成する

## 参照

**buffer~**

オーディオ・サンプルの格納

**cos~**

シグナル（0-1の範囲）のコサイン関数

**phasor~**

ノコギリ波の生成

**wave~**

サイズ可変のウェーブテーブル再生

Tutorial 2

Fundamentals: Adjustable oscillator（英文PDF参照）

Tutorial 3

Fundamentals: Wavetable oscillator（英文PDF参照）

## 入力

**シグナルまたは実数** 第1インレット：任意のシグナルや実数、0から1に増加するシグナルなどを入力する。これらの値は**triangle~**オブジェクトが持つウェーブテーブルを走査するのに使われる。**triangle~**は繰り返し波形のウェーブテーブルによるオシレーターとして働き、**phasor~**オブジェクトの出力やその他のオーディオシグナルを、そのコントロールに利用することができる。

第2インレット：三角波のピークを一波形内での位置として0から1.0の値で位相オフセット値を入力する。デフォルト値は0.5（この場合は波形の中心にピークがある三角波となる）。**phasor~**オブジェクトの出力を用いて、位相オフセットを0にした場合、**triangle~**の出力波形は傾斜波形となり、位相オフセットを1.0にするとノコギリ波を出力する。

**lo** 第1インレット：**lo**に続き、オプションの数値として**triangle~**が出力するシグナルの最小値を指定する。デフォルト値は-1.0である。

**hi** 第1インレット：**hi**に続き、オプションの数値として**triangle~**が出力するシグナルの最大値を指定する。デフォルト値は1.0である。

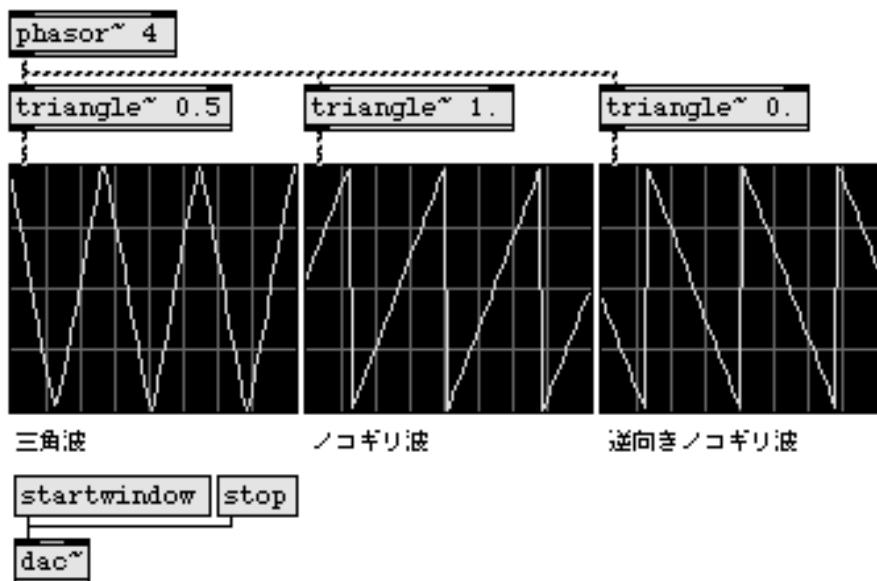
## アーギュメント

**実数** オプション。第2インレットと同じ0から1の値で位相オフセット値を指定する。デフォルト値は0.5である。**phasor~**のシグナルで**triangle~**を動作させる場合、この値が**0**の時は傾斜波形を出力し、**1**の時はノコギリ波を出力する。

## 出力

**シグナル** **triangle~**に入力したシグナルに対応したシグナルを出力する。**phasor~**オブジェクトの出力かその他オーディオシグナルが**triangle~**の走査に使われた場合、その出力は周期的な波形となる。

## 利用例



`triangle~` は異なる折り返し点を設定して異なる傾斜波形を生成する

## 参照

`buffer~`

オーディオ・サンプルの格納

`cos~`

シグナル (0-1の範囲) のコサイン関数

`phasor~`

ノコギリ波の生成

`trapezoid~`

台形波のウェーブテーブル

`wave~`

サイズ可変のウェーブテーブル再生

Tutorial 2

Fundamentals: Adjustable oscillator (英文PDF参照)

Tutorial 3

Fundamentals: Wavetable oscillator (英文PDF参照)

## 入力

**シグナル** 入力されたシグナルの小数部分を切り捨てる。**trunc~**オブジェクトは、小数部分を持つシグナルを、もっとも近く、小さい整数値に変換する。例えば、1.75は1.0に、-1.75は-1.0に切り捨てられる。このオブジェクトの機能は単純だが、コンピュータには負荷のかかる処理である。

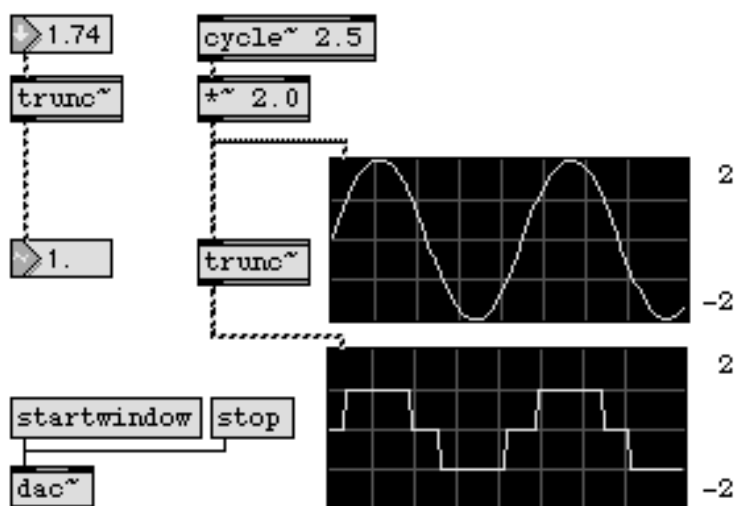
## アーギュメント

なし。

## 出力

**シグナル** 入力されたシグナルを切り捨てたもの。

## 利用例



trunc~に小数部分をもったシグナルを入力し、小数部分を切り捨てる

## 参照

**clip~**  
**round~**

シグナルの振幅に制限をかける  
入力されたシグナル値の端数の切り捨て／切り上げを行う

## 入力

**シグナル** 第1インレット：出力シグナル用のインデックスとなる同期シグナルを入力する。一般的には、このインデックスの数値の範囲は、0から  $n-1$  ( $n$ はシグナル・ベクター・サイズ) となる。

第2インレット：このインレットに入力される同期シグナルは、処理されるベクターの入力時のインデックスとして用いられる。一般的に、この同期シグナルの数値の範囲は、0から  $n-1$  ( $n$ はシグナル・ベクター・サイズ) となる。この同期シグナルの範囲と、第1インレットに入力した出力データのインデックスの範囲が違う場合は、2つの同期シグナル間でインデックスが対応する位置に各データが移動する。

第3インレット：フィルター処理されるシグナル・データ。通常は、このインレットには **fft~** や **fftin~** オブジェクトの出力など周波数領域の情報が入力される。

**rampsmooth** 第1インレット：**rampsmooth** とそれに続く2つの整数を指定する。これらは、連続したフレームのベクター・データを、どのようにリニアにスムーズ処理を行うかを指定する。

**slide** 第1インレット：**slide** とそれに続く2つの実数を指定する。これらは、時間領域処理を行う **slide~** オブジェクトと同様の処理を行い、連続するベクター間の対数的な補間を行う。

**deltaclip** 第1インレット：**deltaclip** とそれに続く2つの実数を指定する。これらは、連続するベクターにおいて対応するデータの変化量に制限をかけるものである。これは時間領域で処理を行う **deltaclip~** オブジェクトと同じ機能である。

## アーギュメント

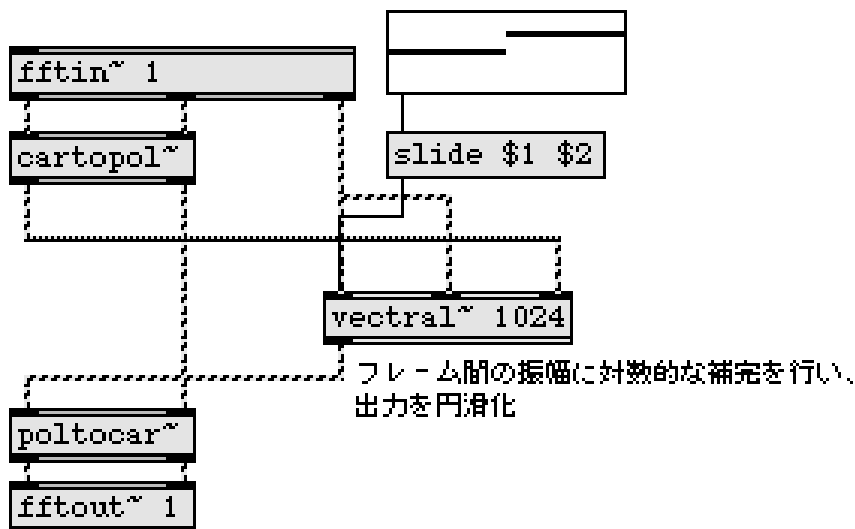
**整数** オプション。処理されるベクター・サイズをアーギュメントとして記述する。デフォルト値は512だが、**vectral~** オブジェクトに入力するシグナル・ベクター・サイズに適切に合わせるべきである。

## 出力

**シグナル** 第3インレットに入力されたシグナルに対してスムーズ処理が施されたシグナルを出力する。その処理結果は、オブジェクトに指定したパラメーターに従う。



## 利用例



**vectral~**は、ベクター・データ（例えばFFTシグナル）のフレーム間に対する、異なるタイプのスムーズ処理を行う

## 参照

|                    |                                  |
|--------------------|----------------------------------|
| <b>cartopol</b>    | 直交座標を極座標に変換する                    |
| <b>cartopol~</b>   | シグナルについて直交座標を極座標に変換する            |
| <b>deltacclip~</b> | シグナルの振幅変化量に制限をかける                |
| <b>fft~</b>        | 高速フーリエ変換                         |
| <b>fftin~</b>      | <b>pfft~</b> に読み込まれるパッチにおける入力    |
| <b>fftinfo~</b>    | <b>pfft~</b> に読み込まれるパッチに関する情報の取得 |
| <b>fftout~</b>     | <b>pfft~</b> に読み込まれるパッチにおける出力    |
| <b>frameaccum~</b> | 連続した位相変化データのフレームから位相を計算          |
| <b>framedelta~</b> | 連続したFFT処理フレーム間の位相変化を計算           |
| <b>ifft~</b>       | 逆高速フーリエ変換                        |
| <b>pfft~</b>       | 周波数領域処理を行うパッチの管理                 |
| <b>poltocar</b>    | 極座標を直交座標に変換する                    |
| <b>poltocar~</b>   | シグナルについて極座標を直交座標に変換する            |
| <b>rampsmooth~</b> | 入力シグナルにスムーズ処理を行う                 |
| <b>slide~</b>      | シグナルへの対数的なフィルター処理                |

注記： **vst~** オブジェクトはMax/MSPで作成されたVSTプラグインを利用できない。

## 入力

- シグナル** プラグインで処理する入力。プラグインがインストゥルメンツ・プラグインであれば、入力は無視される。
- 整数** 第1インレット：現在読み込まれているプラグインのエフェクト・プログラムを変更する。最初のプログラム番号は1である。
- 実数** 整数に変換される。
- リスト** 第1インレット：現在読み込まれているプラグインのパラメーター値を変更する。リストの1番目の要素はパラメーター番号（1で始まる）で、2番目の要素はパラメーター値である。2番目の数値は0から1までの実数であり、0がパラメーターの最小値で、1が最大値である。
- 任意のシンボル** プラグインのパラメーターを指名するシンボルに、0から1までの実数が続き、パラメーターの値を設定する。
- bypass** 0ではないアーギュメントを伴う**bypass**は、現在読み込まれているプラグインの処理をストップし、オブジェクトのシグナル入力をシグナル出力にコピーするようにする。**bypass 0**はプラグインの処理を有効にする。
- disable** 0ではないアーギュメントを伴う**disable**は、現在ロードされているプラグインの処理をストップし、無音を出力する。**disable 0**はプラグインの処理を有効にする。
- get** 数値アーギュメントを伴う**get**は、プラグインの情報をプラグインの右から3番目のアウトレットから出力する。数値アーギュメントが、1から現在読み込まれているプラグインのパラメーターの数の間であれば、**get**メッセージは、番号で指定されたパラメーターの値を（0から1までの数値で）出力する。アーギュメントが0またはマイナスであれば、**get**メッセージは右から2番目のアウトレットから、以下の情報を出力する。
1. プラグインの入力の数
  2. プラグインの出力の数
  3. プラグインのプログラムの数
  4. プラグインのパラメーターの数
  5. プラグインのcanMonoフラグが設定されているか否か。これはプラグインがステレオでもモノラルでも使用できることを表す。
  6. プラグインが独自のエディット・ウィンドウを持っているか1、そうでなければ0。

- midievent** 2個から4個までの数値を伴う**midievent**は、プラグインにMIDIイベントを送信する。最初の3つの数値アーギュメントはMIDIメッセージのバイトである。オプションの4番目のアーギュメントはMIDIノート・メッセージのディチューン・パラメーターとして使用される。この値の範囲は-64セントから64セントで、デフォルト値は0である。
- mix** 第1インレット：**mix 1**はミックス・モードをオンにするが、このモードではプラグインの出力に入力が加算される。**mix 0**はミックス・モードをオフにする。ミックス・モードがオフの時、プラグインの出力に入力は加算されない。つまり、プラグインの出力だけが**vst~**オブジェクトのシグナル・アウトレットから出力される。
- open** プラグインのエディット・ウィンドウを開く
- params** **params**はプラグインのパラメーターのリストを、右から4番目のアウトレットから出力する
- pgmnames** **pgmnames**はプラグインの現在のプログラム名を、右のアウトレットから出力する
- plug** **plug**にアーギュメントを付けない場合は、ホストに新しいプラグインを選択できるように標準ファイル・ダイアログを開く。シンボル・アーギュメントを伴う**plug**は、Maxサーチ・パスだけでなく、Maxアプリケーション・フォルダーの中のVstPlugInsというフォルダーからも、指定された名前を持つVSTプラグインを検索する。新しいプラグインが見つかり、開かると、それ以前に読み込まれたプラグイン（もしあれば）は廃棄されて、新しいプラグインが読み込まれる。
- read** アーギュメントを付けない場合は、**read**はバンクまたは独立したプログラムのフォーマットのエフェクト・プログラム・ファイルを開くために、標準ファイル・ダイアログを開く。**read**はオプションでシンボル・アーギュメントを続けることができ、Maxサーチ・パスからバンクまたはエフェクト・プログラム・ファイルを探索する。
- set** シンボルを伴う**set**は、エフェクトの現在のプログラムの名前をシンボルに変更する。
- wclose** プラグインのエディット・ウィンドウを閉じる
- write** アーギュメントを付けない場合は、**write**はエフェクト・プログラム・ファイルの名前とタイプ（シングル・プログラムまたはバンク）を指定するために、標準ファイル保存ダイアログを開く。**write**はオプションで、保存するファイルのフル・パス名か部分的なパス名を指定するシンボルを、アーギュメントとして続けることができる。この場合には独立したプログラム・ファイルとして保存される。

- writebank** アーギュメントを場合は、**writebank**はエフェクト・プログラム・バンク・ファイルの名前を指定するために、標準ファイル保存ダイアログを開く。**writebank**はオプションで、保存するファイルのフル・パス名か部分的なパス名を指定するシンボルを、アーギュメントとして続けることができる。
- writepgm** アーギュメントを付けない場合は、**writepgm**は独立したエフェクト・プログラムのファイル名を指定するために、標準ファイル保存ダイアログを開く。**writepgm**はオプションで、保存するファイルのフル・パス名か部分的なパス名を指定するシンボルを、アーギュメントとして続けることができる。

## アーギュメント

- 整数** オプション。1番目または1番目と2番目のアーギュメントが数値であれば、オーディオ入出力の数を設定する。1つの数値しかなければ、その数値はアウトレットの数を設定する。2つの数値があれば、1番目がインレットの数を設定し、2番目はアウトレットの数を設定する。
- シンボル** オプション。オブジェクトが生成された時に読み込まれるVSTプラグ・インファイルの名前を設定する。**plug**メッセージによってオブジェクトを生成した後で、プラグインを読み込む（または、現在使用中のプラグインを置き換える）こともできる。
- シンボル** オプション。プラグインの名前の後で、プラグインのためのプリセット・エフェクトを含むファイル名を指定することができる。このファイルが見つかると、プラグインが読み込まれた後に、プリセット・エフェクトが読み込まれる。

## 出力

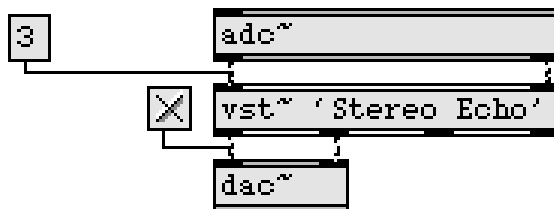
- シグナル** 第1アウトレットとアウトプット・アーギュメントの数で定義されたシグナル・アウトレット：プラグインからのオーディオ出力。第1アウトレットが左チャンネル（またはチャンネル1）。
- シンボル** 右から4番目のアウトレット：**params**メッセージを受け取ると、プラグインのパラメーターが一連のシンボルとして出力される。
- 整数または実数** 注記：一部のプラグイン、特に独自のエディターを持つものでは、パラメーターの名前を得られないことがある。  
右から3番目のアウトレット：**get**メッセージを受け取ると、パラメーター値またはプラグインの情報に関する値が出力される。

**整数** 右から2番目のアウトレット：プラグインによって受信された（ただし、**midievent**メッセージを使って受信したMIDIメッセージを除く）生のMIDIバイト

**シンボル** 右のアウトレット：**pgmnames**メッセージを受け取ると、一連のシンボルが出力される。プログラム名が無ければ**pgmnames: Default**というメッセージが出力される。

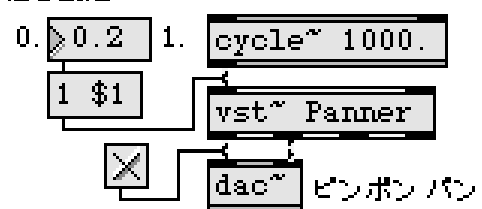
## 利用例

格納されたパラメータ  
設定の「プログラム」を選択



VSTプラグインでオーディオ・シグナルを処理する

最初のパラメータの  
値を設定



## 参照

### rewire~

ReWireデバイスをホストする

## 入力

**シグナル** 第1インレット：連続的に0から1へ変化する入力シグナルは、**buffer~**オブジェクト内のサンプル・データの特定範囲を読み出す目的で使われる。**phasor~**オブジェクトの出力は、**wave~**をオシレーターとして制御したり、**buffer~**からウェーブテーブルとして繰り返し読み出すサンプル範囲を調整することなどに使える。しかし、**wave~**の第1インレットに接続している**phasor~**の周波数を変更した時は注意が必要で、複数の波形や繰り返し波形をウェーブテーブルとして利用していると**wave~**から出力されるシグナルのピッチが、**phasor~**の設定周波数に完全に同期しているとは限らない。**phasor~**の出力の正負逆のシグナルを**wave~**に入力することで波形の逆再生が行える。

第2インレット：**buffer~**オブジェクトのサンプル・メモリに対し、読み出す開始位置をミリ秒単位で波形の先頭からのオフセット値として指定する。

第3インレット：**buffer~**オブジェクトのサンプル・メモリに対し、読み出す終了位置をミリ秒単位で波形の先頭からのオフセット値として指定する。

**実数または整数** 第2および第3インレット：シグナルの代わりに波形読み出しの開始および終了位置を数値で指定することができる。ただし、シグナルのパッチ・コードをこれらのインレットに接続していない場合に限り、数値入力が有効となる。

**enable** 第1インレット：**enable 0** というメッセージは、その後のシグナル入力を無視して、オブジェクトの働きを無効にする。**enable**に続いて0以外の数値を指定すれば、処理を再開する。

**interp** 第1インレット：**interp**に続き、0以外の数値を指定すれば、ウェーブテーブルの補間処理を行うようになる。これは**wave~**オブジェクトのデフォルトの動作である。**interp 0** というメッセージで補間機能を無効にすることができる。

**set** 第1インレット：**set**に続き**wave~**がウェーブテーブルとして利用する**buffer~**の参照名を、シンボルで指定する。この参照名のシンボルに続き、読み出し開始および終了位置の数値をオプションとして指定することができる。これらの値が指定されていない場合、デフォルトの開始および終了位置（すなわち、**buffer~**内の最初と最後のサンプル）が使われる。第2および第3インレットにシグナルが入力されている場合は、**set**メッセージでのこれらの数値は無視される。



## アーギュメント

**シンボル** 必須。ウェーブテーブルとして利用する **buffer~** の参照名を指定する。**wave~** は **buffer~** 内のデータのコピーしたものを出力するわけではないため、**buffer~** 内のデータが変更されると、**wave~** の出力も変化することに注意が必要である。**buffer~** 内部に複数チャンネルのデータがある場合でも、**wave~** は常に1つめのチャンネルのデータを読み出す。

**実数または整数** オプション。**buffer~** の参照名に続き、波形の読み出し開始および終了位置を指定することができる。どちらの値も、**buffer~** オブジェクト内のサンプル・メモリの先頭からミリ秒単位のオフセット値として指定する。デフォルトでは開始位置は0、終了位置はサンプル・データの最後の位置となる。開始位置を0でない値とし、終了位置をサンプル・データの最後のサンプルとする場合、アーギュメントは **buffer~** の後ろに1つの数値だけを指定する。第2インレットにシグナルが入力されている場合は、アーギュメントでの **buffer~** 参照名の後の開始位置の指定は無視される。同じく、第3インレットにシグナルが入力されている場合は、終了位置のアーギュメントは無視される。追加オプションとして、これらの数値の後に **buffer~** 内のチャンネル数を指定することができる。

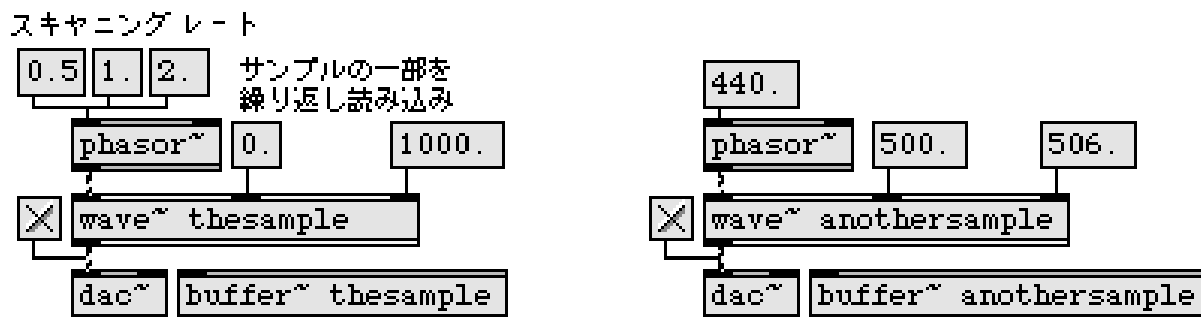
**整数** オプション。出力チャンネル数を指定する。これは **wave~** オブジェクトのアウトレット数となる。この最大数は4である。**wave~** オブジェクトが参照する **buffer~** オブジェクトが **wave~** オブジェクトのアウトレット数より多くのチャンネルを持っている場合、その余分なチャンネルは再生されない。**buffer~** オブジェクトの保持するチャンネルのほうが少ない場合は、**wave~** オブジェクトの余分なアウトレットからは0が出力される。

## 出力

**シグナル** **wave~** オブジェクトで設定された読み出し開始位置と終了位置に相当する **buffer~** 内の一部のデータを、**wave~** の第1インレットに入力された0から1の範囲のシグナルに応じて読み出す。そして、その **buffer~** 内の相当するデータを、**wave~** のアウトレットから出力する。**wave~** への入力シグナルが、**phasor~** オブジェクトが出力するノコギリ波のような繰り返し波形である場合、それに対する出力波形は入力シグナルに対応する周波数で繰り返すものとなる。(出力される1波形は、**buffer~** 内の一部のデータである)



## 利用例



サンプルの一部を利用することで、可変サイズのウェーブテーブルとして繰り返し波形を出力する

## 参照

|                 |                              |
|-----------------|------------------------------|
| <b>2d.wave~</b> | 2次元のウェーブテーブル                 |
| <b>buffer~</b>  | オーディオ・サンプルの格納                |
| <b>buffir~</b>  | <b>buffer~</b> を利用したFIRフィルター |
| <b>groove~</b>  | 速度可変のループ再生                   |
| <b>phasor~</b>  | ノコギリ波の生成                     |
| <b>play~</b>    | ポジション・ベースのサンプル再生             |

Tutorial 15

Sampling: Variable-length wavetable (英文PDF参照)



## 入力

**実数** 第1インレット：表示開始時間をミリ秒単位で設定する。この値を変えると、**buffer~**のサンプル・データの要求された時間が表示の左端に揃うように、ビューをオフセットやズームする。デフォルト値は0（ターゲットの**buffer~**の始まりから表示が始まる）である。

第2インレット：表示の長さをミリ秒単位で設定する。デフォルト値は**buffer~**の長さである。

第3インレット：選択範囲の開始時間をミリ秒単位で設定する。

第4インレット：選択範囲の終了時間をミリ秒単位で設定する。

**リスト** 第5インレット：第5インレットは、任意の数の**waveform~**オブジェクトが開始、長さ、選択開始、選択終了の値を共有できるようにリンク入力を提供する。これらの値の中の任意の値が変化した時はいつでも、**waveform~**はそれらすべてを第2アウトレットからリストとして出力する。このアウトレットが他の**waveform~**オブジェクトのリンク入力に接続されていれば、そのリストを受け取った時に更新される。

回路を完結するには、2番目の**waveform~**オブジェクトのリスト出力を第1のオブジェクトのリンク入力に接続する。このようにすると、どちらか一方の（マウスクリックなどによる）変化は他方にも反映される。これは主に**waveform~**オブジェクトが同じ**buffer~**の異なるチャンネルを表示している時に有効である。任意の数の**waveform~**オブジェクトはこのようなやり方でリンクされ、1つの長い円状のチェーンを形成することができる。この場合には**waveform~**はフィードバックが発生するのを防止する。

**bpm** 1つまたは2つの数値を伴う**bpm**は、**waveform~**の表示で用いられるリファレンスのテンポと小節当たりの拍数を設定する。最初のアーギュメントはテンポを分あたりの拍数で設定する。デフォルト値は120である。2番目のアーギュメントはオプションで、小節あたりの拍数を指定する。デフォルト値は4である。**bpm**メッセージは、あたかもユーザーが**unit bpm**メッセージを送ったかのように、表示時間単位を自動的に**bpm**に設定する。時間の値は浮動小数点で表示される拍の再分割付きの小節と拍で表示される。**offset**メッセージはターゲットの**buffer~**の内容についての測定基準情報をそろえるのに有用である。**waveform~**は**setbpm**メッセージを用いて現在の選択に基づいてテンポを算出することもできる。

**brgb** RGB形式の3つの数値を伴う**brgb**は、残りの表示内容がその上に描画される前に、オブジェクトの矩形全体を塗りつぶすのに使われる背景色を設定する。



- clipdraw** 1を伴う**clipdraw**は、描画モードで変更された値を（**vzoom**メッセージによって決定される）表示の範囲にクリップされるようにする。**clipdraw 0**はクリッピングを無効にし、ウィンドウの範囲を超えて自由に値をスケールリングできるようにする。デフォルト値は0で、クリッピングは行われない。
- constrain** **constrain**メッセージは**waveform~**インターフェースの交代のふるまいをトグルするための、1つのアーギュメントを伴う。その効果は（**mode**メッセージによって決定される）現在のモードによって変化するが、一般的にshiftキーを押しながらマウス操作中を行った時に期待されるのと同じ振る舞いを起こす。例えば、Selectモードでのクリックは追加選択（選択範囲の最も近い終了ポイントだけを設定する）として解釈される。Moveモードでの**buffer~**ナビゲーションが水平方向のパンニングだけに制限されているのであれば、ズームはない。Loopモードでの選択範囲の長さは垂直方向のマウス移動に関係なく保持される。明らかにこのメッセージは、適切なところにこの振る舞いを実装することを意図している。任意の0でない整数アーギュメントによって、制限されたインターフェースの動きが有効になる。0またはまったくアーギュメントがなければ、制限を無効にしてデフォルトの振る舞いに戻す。
- crop** **crop**メッセージはターゲットの**buffer~**のオーディオ・データを現在の選択範囲にトリムする。それは**buffer~**を選択範囲の長さにもリサイズし、選択されたサンプルをそこにコピーし、その結果をデフォルトの設定で表示する。これは破壊編集で、元に戻すことはできない。
- frgb** RGB形式で3つの数値を伴う**frgb**は、波形グラフとして**buffer~**のデータを描画する時に使われる前景色を設定する。
- grid** 1つの整数または実数を伴う**grid**は、現在の時間測定単位に関連して、垂直グリッド・ラインの間隔を指定する。例えば、**waveform~**が表示にミリ秒を用いている場合、メッセージ**grid 1000**は、**waveform~**の表示に1000ミリ秒間隔でグリッド・ラインを表示する。ラベルが有効であれば、グリッド・ラインの間に表示される。0のアーギュメントまたはアーギュメントなしは、グリッド・ラインを無効にする。
- labels** 1つの整数を伴う**labels**は、表示の上部の全域で時間計測の数値ラベルを、有効（1の場合）または無効（0の場合）にする。任意の0以外の整数は数値ラベルを表示する。0のアーギュメント、またはアーギュメントなしの場合には、数値ラベルを表示しない。
- mode** シンボルのアーギュメントを伴う**mode**は、**waveform~**がマウスの動きに対してどのように応答するかを決定する。有効なシンボル・アーギュメントは**none**、**select**、**loop**、**move**と**draw**である。



- none** **waveform~**を、クリックもドラッグも効果を持たない、表示のみモードにする。便宜上、そしてカスタムのインターフェースのふるまいを追加するために、マウスの動きは**mouseoutput**モードに従って送られる。アーギュメントなしの**mode**メッセージは、**mode none**と同じ働きを持つ。
- select** **waveform~**オブジェクトのデフォルトの表示モードを設定する。Selectモードでは、カーソルは**waveform~**表示エリアの中にIビームとして表示される。クリック&ドラッグして値の範囲を選択することができる。マウスの動きは**mouseoutput**設定に従って、**waveform~**に**update**メッセージを生成させる。
- loop** 別のループ選択スタイルに設定し、ここでは垂直方向のマウスの動きが選択範囲の長さを拡大または縮小し、水平方向の動きがポジションに割り当てられる。これは **waveform~.help** ファイルに見られるように、**groove~** オブジェクトをコントロールするのに効果的である。loopモードが選択されている時、表示エリアの中でカーソルを動かすと、そのアピアランスをダブルIビームに変える。
- move** **waveform~**オブジェクトのmoveモードに設定する。このモードは**waveform~**ビューをナビゲートできるようにする。垂直方向のマウスの動きは、ズーム・インまたはズーム・アウトできるようにし、水平方向の動きは、x軸の時間の範囲を通してスクロールできる。グラフの中のポイントをクリックすると、そのポイントが残りのマウス・イベントの中心リファレンス・ポイントになる（マウス・ボタンが離されるまで）。これによって絶えず表示のセンターを合わせ直す必要がなくなり、あるスポットをグラブしてズーム・インまたはズーム・アウトすることができるようになる。
- draw** **waveform~**オブジェクトのdrawモードに設定する。このモードでは、ペンシル・ツールでターゲットの**buffer~**の値を編集することができる。drawモードでクリック&ドラッグすると、**buffer~**のサンプルを直接変更し、取り消すことはできない。サンプル単位を見るにはズーム・アウトし過ぎている場合にも、サンプル値はドラッグするに従って、結果が連続的に変化するようにリニア補間される。
- mouseoutput** シンボル・アーギュメントを伴う**mouseoutput**は、マウスの動きに対応して、いつ選択範囲の開始と終了の値が出力されるかを決定する。選択範囲の開始と終了（第3アウ



トレットと第4アウトレット)だけが影響される。**mouseoutput**モードに関係なく、マウスの情報は常に第5アウトレットから出力される。有効なシンボルは**none**、**down**、**up**、**downup**と**continuous**である。

- none** 選択範囲の開始と終了は、マウスの動きに対応して出力されない。アーギュメントなしの**mouseoutput**メッセージはこのシンボル (**none**) と同じ効果を持つ。
- down** **waveform~**の中でクリックした時のみ、現在の選択範囲の開始と終了の値を (第3アウトレットと第4アウトレットから) 出力する。
- up** **waveform~**の中でクリックした後でマウスのボタンを離れた時のみ、選択範囲の開始と終了を出力する。
- downup** **waveform~**の中でクリックした時と、マウスボタンが離された時の両方で、選択範囲の開始と終了を出力する。
- continuous** クリックした時や離れた時、またドラッグ操作の途中で値が変化した時はいつでも、選択範囲の開始と終了を出力する。
- normalize** 1つの実数を伴う**normalize**は、アーギュメントによって与えられた値に最も高いピークがマッチするように、ターゲットの**buffer~**のサンプル値をスケールリングする。これはオーディオを増幅または減衰させるが、いずれの場合にも、すべての**buffer~**の値はスケールされ、この働きは取り消すことができない。
- offset** 1つの実数を伴う**offset**は、すべてのラベルと時間計測のマーキングを、指定された数値のミリ秒分だけシフトする。スナップの動き具合も同様にシフトする。**offset**は、メッセージ**offset 0**またはアーギュメントなしの**offset**メッセージを送ることで、取り除くことができる。
- rgb2** RGB形式の3つの数値を伴う**rgb2**は、選択範囲を識別する選択矩形の色を設定する。
- rgb3** RGB形式の3つの数値を伴う**rgb3**は、オブジェクトの矩形とラベルのエリアの周りの1ピクセルのフレームの描画に用いられるフレームの色を設定する。
- rgb4** RGB形式の3つの数値を伴う**rgb4**は、ラベル・テキストの色を設定する。
- rgb5** RGB形式の3つの数値を伴う**rgb5**は、ラベルの背景色を設定する。





- rgb6** RGB形式の3つの数値を伴う**rgb6**は、(有効であれば) チェックマークと計測ラインの色を設定する。
- rgb7** RGB形式の3つの数値を伴う**rgb7**は、選択矩形の**OpColor**を設定する。上で指定したように、選択矩形は**rgb2**を前景色として描画される。しかし、このオペレーションの間の転送モードは、**rgb7**を**OpColor**としてブレンドして設定される。**rgb2**と**rgb7**の異なる組み合わせを試みると、それらが色と不透明度に影響するかを見ることができる。グレーのシェードはここでは有用である。
- set** **buffer~**オブジェクトの名前の1つのシンボルまたは整数を伴う**set**は、**waveform~**を**buffer~**にリンクし、デフォルトの表示値で描画する。オプションの整数アーギュメントは、マルチ・チャンネルの**buffer~**オブジェクトを表示するためのチャンネル・オフセットを設定する。リンクされた**buffer~**の名前はMaxパッチに保存されないため、必要であれば他に格納する必要がある。
- setbpm** アーギュメントなしの**setbpm**は、**waveform~**に現在の選択範囲の基づいてテンポを計算させる。それはメッセージ**unit bpm**を送信したかのように、自動的に表示時間の単位を**bpm**に設定する。選択範囲が小節の論理にかなった倍数または分割を含むようにテンポが選択され、現在の小節当たりの拍数を保つように、基準を満足する現在のテンポに最も近い値を探そうとする。適切なテンポが選択された時、**offset**パラメーターは選択範囲の開始時間が、小節線ちょうどに収まるように調整される。
- それゆえに、選択エリアは対応するテンポによって精密に枠にはめられるようになってしまう。この手法の1つの用途は、オーディオの一部に時間のラベルとティック・マークを瞬時に確立させる場合である。できるだけ正確に1小節を選択した後、**setbpm**メッセージを送信して**snap to label**をオンにすると、選択範囲を瞬時に測定基準値に量子化することができるようになる。
- ターゲットの**buffer~**が、既に拍または小節の論理的な数値に切り落とされたオーディオ・セグメントを含む場合に、最も良い方法は(選択開始と選択終了インレットへのメッセージで) **buffer~**のすべての部分を選択し、続いて**setbpm**メッセージを送ることである。**buffer~**が正確に切り落とされている場合には、結果として生じるテンポ・オーバーレイは極めて正確で、測定基準情報と共にテンポを明らかにする。
- 新しいテンポが計算されると、任意のリンクされた**waveform~**オブジェクトを更新するため、そして取り囲んでいるパッチに、要求されたすべての方法で利用されるように、最も右のアウトレット(リンク・アウトレット)から出力される。



- snap** シンボル・アーギュメントを伴う**snap**は、**waveform~**の選択範囲のスナップ・モードを設定する。**snap**は選択の開始ポイントと終了ポイントをスナップ・モードで定義された**buffer~**の特定の部分に自動的に移動する。可能なアーギュメントは**none**、**grid**、**zero**である。
- none** スナップを無効にして自由な選択をできるようにする。これはデフォルトである。アーギュメントなしの**snap**メッセージも、同じ効果を持つ。
- grid** **grid**メッセージによって設定された垂直グリッド・ラインに、選択開始と終了ポイントがスナップするように指定する。グリッド・ラインの間隔は現在の時間計測単位と（オフセットが指定されている場合には）オフセット値によって影響されるため、グリッドへのスナップは、これらのパラメーターにも同様に影響される。
- tick** 選択の開始と終了を、**ticks**メッセージによって指定されるティック単位にスナップする。
- zero** 選択を一定のグリッドにスナップさせる代わりに、このモードでは**buffer~**データのゼロクロスを検索する。これらは正のサンプルの後に負のサンプルが続く、またはその逆のポイントとして定義される。これはループや編集ポイントを探すのに有効である。
- ticks** 1つの数値を伴う**ticks**は、それぞれのグリッド・ラインの間に描画されるティックスの数を指定する。デフォルトは8である。**0**のアーギュメント、またはアーギュメントなしで、ティック・マークを無効にする。
- undo** このモードは**waveform~**の選択に対してだけ作用する。それは選択の開始と終了ポイントを直前の値に戻す。これはマウスで微調整をしている時に、誤って間違った場所をクリックしてしまった場合や、そうでなくてもうっかり選択を変えてしまった時に役立つ。**undo**コマンドを繰り返すと最後の2つの選択状態の間でトグルする。
- unit** 1つのシンボル・アーギュメントを伴う**unit**は、表示に用いられる時間計測単位を設定する。有効なアーギュメントは**ms**、**samples**、**phase**そして**bpm**である。
- ms** 表示単位をミリ秒に設定する。これがデフォルトである。
- samples** ターゲットの**buffer~**のサンプル位置として現在の位置が表示される。**offset**メッセージによってシフトされない限り、最初のサンプルは0として番号付けられる。





- phase** 0が最初のサンプルを参照し、1が最後のサンプルを参照するように正規化された**buffer~**内の位相に基づいて、時間を表示するようにする。このタイプの計測単位は、**phasor~**や**wave~**といった0-1シグナル同期を用いるオブジェクトと共に使う場合に特に関連性がある。
- bpm** そのいずれも**bpm**メッセージで設定できるマスター・テンポと、小節あたりの拍数に比例して時間リファレンス単位として分あたりの拍数を指定する。**waveform~**は**setbpm**メッセージで現在の選択にフィットするテンポを計算することもできる。
- vlabels** 1つの整数を伴う**vlabels**は、**waveform~**表示の右端に沿う垂直軸のラベルを有効または無効にする。任意の0でない数値は、ラベルを表示する。0のアーギュメント、またはアーギュメントなしは、ラベルを表示しない。
- voffset** 1つの実数を伴う**voffset**は、**waveform~**表示の垂直オフセットを設定する。0の値はx軸を中心に配置し、これはデフォルトである。
- vticks** 1つの整数を伴う**vticks**は、**waveform~**表示の左右単に沿う垂直軸のティック・マークを有効または無効にする。任意の0でない数値は、ティック・マークを表示する。0のアーギュメント、またはアーギュメントなしは、ティック・マークを表示しない。
- vzoom** 1つの実数を伴う**vzoom**は、**waveform~**表示の垂直スケーリングを設定する。

## インスペクター

**waveform~**オブジェクトの振る舞いはインスペクターを使用して、表示、編集することができる。Windowsメニューから**Show Floating Inspector**を選択して、フローティング・インスペクターを有効にしている時、任意の**waveform~**オブジェクトを選択することで、**waveform~**オブジェクトのインスペクターをフローティング・ウィンドウに表示することができる。**waveform~**オブジェクトを選び、Objectメニューから**Get Info...**を選択することでも、インスペクターを表示することができる。

**waveform~**のインスペクターでは、次のアトリビュートを設定することができる。

**Unit** (単位) ポップアップ・メニューは、波形表示で用いられる時間計測単位を設定する。これは先述した**unit**メッセージに対応する。



*Label* (ラベル) ポップアップ・メニューは、波形表示で用いられるラベルの種類を設定する。オプションは *none* (なし)、*milliseconds* (ms、デフォルト)、*number of samples* (サンプル数)、*phase units* (位相単位)、または *bpm* (分あたりの拍数) である。

*Mouse Output Mode* (マウス・アウトプット・モード) は、いつマウスの動きが表示をトリガーして選択を出力するかを決定する (先述した **output** メッセージを参照のこと)。デフォルトのモードは *continuous* である。

*Edit Mode* (エディット・モード) ポップアップ・メニューは、**waveform~** オブジェクトの表示モードを設定するのに用いられる。デフォルトは *select* モードである (先述した **mode** メッセージを参照のこと)。

*Color* (カラー) ポップアップ・メニューは、swatch または RGB 値を用いて **waveform~** オブジェクトによる表示で用いる色を指定する。

*Revert* ボタンは、このオブジェクトに設定したすべての変更点をインスペクターを開いた直後の状態に戻す。インスペクターが開いている状態で、Edit メニューの **Undo Inspector Changes** を選択することでも、オブジェクトの状態をそのインスペクターを開く前の状態に戻すことができる。

## アーギュメント

なし。

## 出力

**実数** 第1アウトレット：ミリ秒単位での波形の表示開始時間

第2アウトレット：ミリ秒単位での表示の長さ

第3アウトレット：ミリ秒単位での選択範囲の開始時間

第4アウトレット：ミリ秒単位での選択範囲の終了時間

**リスト** 第5アウトレット：これはマウス・アウトレットであり、**waveform~** オブジェクトの中でクリックすることによって始まるマウスのクリック、ドラッグ、リリースというサイクルに関する情報を出力する。リストは3つの数値を含む。

最初の数値は、**waveform~** オブジェクトに相対する0-1スケール単位のマウスの水平 ( $x$ ) 位置を指定する実数である。**waveform~** の左端では  $x$  は常に0であり、右端では常に1.である。

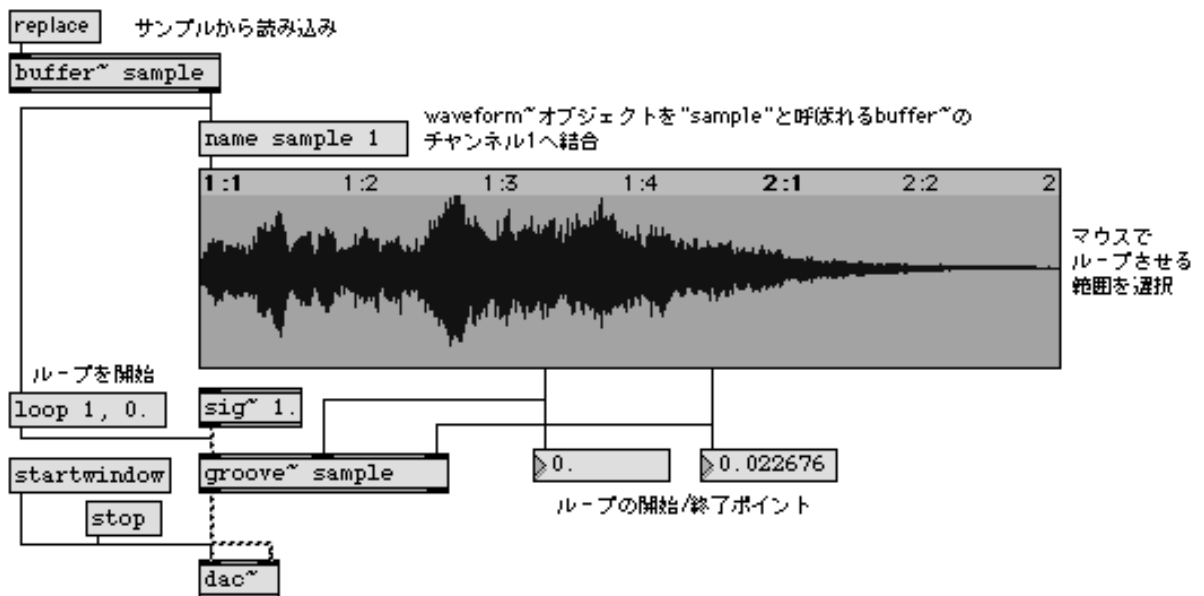


リストの2番目の数値はマウスの浮動小数点y値であり、**buffer~**の値にマッチするようにスケーリングされる。デフォルトのvzoom=1.とvoffset=0.では、**waveform~**の上端は1、下端は-1になる。

最後に、リストの3番目の数値は、現在行われているマウス・イベントの部分を示す整数である。マウス・ダウンまたはクリックの場合には、この値は1となる。ドラッグ中は2で、マウス・アップの時は3となる。これらはマウス・クリックに対する独自の反応を作るのに役立つ。マウスが動いたか否かに関わらず、ドラッグ期間が始まったことを示すために、ドラッグ (2) メッセージはマウス・ダウン (1) メッセージに続いて送信されることに注意が必要である。

第6アウトレット：**waveform~**は表示開始時間、表示区間の長さ、選択開始時間、選択終了時間を含むリストを、これらの値のどれか1つが（マウスの動作や実数入力などによって）変化した時には、いつでも出力が行われる。詳しくは先述したリンク入力情報を参照のこと。

## 利用例



**waveform~**は、**buffer~** オブジェクトからのサンプル・データを表示、選択、編集できる

## 参照

**buffer~**  
**groove~**

オーディオ・サンプルの格納  
速度可変のループ再生

## 入力

**シグナル** 第1インレット：分析するシグナル

**set** 第1インレット：**set**に続く、0.0から1.0までの実数は、第2アウトレットから出力するクリック（インパルス）の音量を設定する。

## アーギュメント

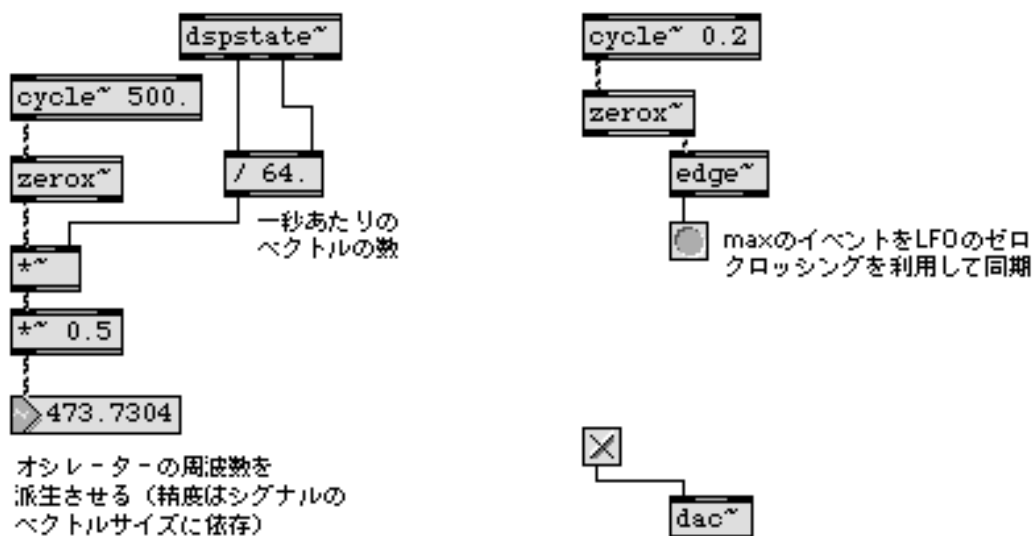
**実数** オプション。第2アウトレットから出力するクリックの音量を設定する。音量の数値は0.0から1.0までの範囲である。デフォルト値では1.0である。

## 出力

**シグナル** 第1アウトレット：シグナル・ベクターごとのゼロ地点の通過回数を示すシグナル。これは直前のシグナル・ベクターにおいて検出した回数である。

第2アウトレット：ゼロ地点を通過したことを検出した時に、第2アウトレットからクリック（インパルス）を出力する。クリックの音量はアーギュメントや**set**メッセージで設定する。

## 利用例



入力シグナルにおいて、ゼロ地点を通過した回数を数えるために**zerox~**を用いる

## 参照

**change~**  
**edge~**  
**spike~**

シグナル値の変化方向を出力する  
シグナルの論理的変化を検出する  
ゼロ値と非ゼロ値の間の時間間隔を計る

**zigzag~**オブジェクトは、**line~**に類似している。**line~**オブジェクトのスタックベースの実装が、出力後に情報を保持しないのに対して、**zigzag~**はリンク・リストの実装を用いる。単に現在のラインを記憶するのに加えて、**zigzag~**オブジェクトはポイントを挿入、削除、追加することで、リストを変更することができる。

**zigzag~**オブジェクトのリンク・リストのそれぞれの要素は値 ( $y$ ) と、ある値から他の値への遷移が起こる時間を指定する遷移時間の値 (**deltax**) を持つ。**zigzag~**オブジェクトがリストを含む時、リストをトリガーする（開始ポイントと終了ポイントを設定したり変更したりすることができる）、異なる速度で前や後ろに移動する、ループするといったことができる。リスト中の現在の位置はジャンプしたり、保持したりすることができる。

## 入力

**mode** 0から3までの数値を伴う**mode**は、**zigzag~**オブジェクトがメッセージやシグナルに反応する方法を指定する。オペレーションのモードをまとめると、以下の通りである。

モード0はデフォルトのオペレーションのモードである。**zigzag~**オブジェクトが**bang**を受け取ると、開始ポイント（または方向がネガティブであれば終了ポイント）にジャンプし、そこから値の出力をはじめ。このジャンプについての時間の値は、**bangdelta**メッセージで設定される長さを持つ。**bangdelta**のデフォルトの値は0である。このモードで、**zigzag~**オブジェクトの第1インレットにシグナルが入力されていれば、リストの現在のインデックスは、そのシグナルによって決定される。以前に設定された**speed**、**loopmode**、**start**、**end**メッセージは無視される。

**zigzag~**のモード1の振る舞いは、**bang**の効果に関してはモード0とまったく同じである。モード1では、シグナル入力はそれとは違って扱われる。モード1で**zigzag~**オブジェクトの第1インレットにシグナルが入力されていれば、入力シグナルはトリガー・シグナルとして機能する。つまり、入力シグナルの傾きが負でない所から負に変化すると、オブジェクトは**bang**を受け取った時と同様にリトリガーされる。

モード2は**zigzag~**オブジェクトをリスト中の次のインデックス（または現在の方向が負であれば直前のインデックス）にジャンプするように設定し、そこから値を出力しはじめる。このジャンプに関連付けられた時間の値は、**bangdelta**メッセージによって設定された長さを持つ。**bangdelta**のデフォルト値は0である。モード2で**zigzag~**オブジェクトの第1インレットにシグナルが入力されていれば、入力シグナルはトリガー・シグナルとして機能する。つまり、入力シグナルの傾きが負でない所から負に変化すると、オブジェクトは**bang**を受け取った時と同様にリトリガーされる。



**bang** 第1インレット：**zigzag~**オブジェクトは、**mode**メッセージを用いて設定される振る舞いのモードに従って、**bang**メッセージに反応する。

**zigzag~**オブジェクトがモード0またはモード1に設定されていれば、**bang**メッセージは**zigzag~**オブジェクトに開始ポイント（方向が負であれば終了ポイントに）に移動させ、そこから値を出力しはじめる。

**zigzag~**オブジェクトがモード2に設定されていれば、**bang**メッセージは**zigzag~**オブジェクトをリスト中の次のインデックス（または現在の方向が負であれば直前のインデックス）にジャンプするように設定し、そこから値を出力しはじめる。

**シグナル** 第1インレット：**zigzag~**オブジェクトは、**mode**メッセージによって設定される振る舞いのモードに応じてシグナルに反応する。

**zigzag~**オブジェクトがモード0に設定されていれば、リストの現在のインデックスは入力シグナルの値によって決定される。以前に設定された**speed**、**loopmode**、**start**、**end**メッセージは無視される。

モード1で**zigzag~**オブジェクトの第1インレットにシグナルが入力されていれば、入力シグナルはトリガー・シグナルとして機能する。つまり、入力シグナルの傾きが負でない所から負に変化すると、オブジェクトは**bang**を受け取った時と同様にリトリガーされる。

モード2で**zigzag~**オブジェクトの第1インレットにシグナルが入力されていれば、入力シグナルはトリガー・シグナルとして機能する。つまり、入力シグナルの傾きが負でない所から負に変化すると、オブジェクトは**bang**を受け取った時と同様にリトリガーされる。

**シグナル** 第2インレット：シグナルの値は、値と時間のペアが出力されるレートを指定する。1.0という値はリストを通常で移動する。再生レート-1はリストを後ろ向き（つまり逆方向に）に移動する。.5という値はリンク・リストを通常で移動する（ディレイ時間の値を効果的に倍にする）。入力シグナルの値は、入力ベクターごとに1回ずつサンプルされる。それゆえに、現在のサンプルレート/(2\*ベクター・サイズ)よりも大きな任意の間隔での周期的な周波数変調では、エイリアス効果が生じる。

**実数** 第1インレット：**zigzag~**オブジェクトのリンク・リストのそれぞれの要素は、ターゲットの値（y）の後に、ミリ秒単位でトータルの時間を指定する2番目の数値（デルタx）からなるペアである。その時間の間に、現在のインデックスの値からターゲットの値までの直線として、定期的に数値が出力される。リスト**0 0 3.5 500 10 1000**は、直線が時間で0の値から始まり、0.5秒後に3.5の値まで増加し、1秒で10の値まで増加することを意味している。

- 整数** 第1インレット：実数に変換される。
- 整数または実数 第2インレット：値と時間のペアが出力されるレートを指定する。1.0という値はリストを通常で移動する。再生レート-1はリストを後ろ向き（つまり逆方向に）に移動する。0.5という値はリンク・リストを通常で移動する（ディレイ時間の値を効果的に倍にする）。
- append** 第1インレット：ポジションを指定する1つの整数（0が最初の要素）を伴う**append**は、新しいイベントのペアを指定したインデックスの後に挿入する。メッセージ**append 0 5 500**は、新しい5という値と500ミリ秒の時間を持つ2番目のエンタリーをリンク・リスト（の0インデックス）に生成する。
- bangdelta** 第1インレット：1つの実数または整数を伴う**bangdelta**は、**zigzag~**オブジェクトが**bang**を受け取った時に、値の間で起きる遷移の時間を指定する。デフォルトは0（すなわち即時の遷移）である。
- bound** 第1インレット：2つの数値を伴う**bound**は、開始と終了のインデックスを指定し（0が最初の要素）、**zigzag~**オブジェクトのリンクリストの開始ポイントと終了ポイントを設定する。
- delete** 第1インレット：ポジションを指定する1つの整数（0が最初の要素）を伴う**delete**は、そのインデックスに関連付けられた値と時間のペアをリストから削除する。リストから複数のイベント・ペアを削除したい場合には、**delete**メッセージの後にリストを続けることもできる。メッセージ**delete 0**は、現在の最初の値と時間のペアをリストから削除する。つまり、2番目の値と時間のペア（すなわちインデックス1の値と時間のペア）はリストの最初の値になる。
- dump** 第1インレット：**dump**は現在格納されている値と時間のペアからなるリストを、**zigzag~**オブジェクトの第3アウトレットから出力する。
- end** 第1インレット：ポジションを指定する1つの整数（0が最初の要素）を伴う**end**は、**bang**によってトリガーされた時に、**zigzag~**オブジェクトがその出力を中止するポイントを設定する。
- insert** 第1インレット：ポジションを指定する1つの整数（0が最初の要素）とリストを伴う**insert**は、指定されたインデックスの前に新しいイベントのペアを挿入する。メッセージ**insert 0 5 500**は値5と時間500ミリ秒の新しい最初のエンタリーを（インデックス0に）生成する。
- jump** 第1インレット：ポジションを指定する1つの数値（0が最初の要素）を伴う**jump**は、リンク・リストのそのポイントまでスキップし、そのポイントから値と時間のペアを出力し始める。オプションの整数は、



次の値までの遷移が起きる時間を、ミリ秒単位で時間を指定するのに用いることができる（デフォルト値は0）。

- jumpend** **jumpend**は**zigzag~**オブジェクトにリンク・リストの最後の値（y）にただちにジャンプさせる。
- jumpstart** **jumpstart**は**zigzag~**オブジェクトにリンク・リストの最初の値（y）にただちにジャンプさせ、現在選択されているリストまたはリストの選択部分を出力する。
- loopmode** 1を伴う**loopmode**は、ルーピングをオンにする。**loopmode 0**はルーピングをオフにする。デフォルトでは、ルーピングはオフである。**loopmode 2**はルーピングを“振り子”モードにし、値と時間のペアは交代に順方向と逆方向に移動する。デフォルトではルーピングはオフである。
- next** 第1インレット：**next**はリンク・リスト中の次の値と時間のペアをスキップする。オプションの整数は、次の値への遷移が起きる間の時間を指定するのに用いられる（デフォルト値は0）。
- prev** 第1インレット：**prev**はリンク・リスト中の前の値と時間のペアをスキップする。オプションの整数は、前の値への遷移が起きる間の時間を指定するのに用いられる（デフォルト値は0）。
- printf** 第1インレット：**print**は**zigzag~**オブジェクトの現在の状態と内容を、Maxウィンドウに表示させる。出力は現在のモード、ループ・モード、開始、終了、現在のリストのループ長、振り子の状態、オブジェクトの変化しつつある値に続いて、yとデルタxの値と一緒にリンク・リストのそれぞれのインデックスのリストである。
- ramptime** 1つの数値を伴う**ramptime**は、出力シグナルがターゲットの値に到達するまでのミリ秒単位の変化時間を設定する。
- setindex** ポジションを指定する1つの整数（0が最初の要素）と1組の実数を伴う**setindex**は、リストの指定されたポジションのターゲットの値（y）と遷移時間（デルタx）を設定する。
- skip** 第1インレット：1つの正または負の数値を伴う**skip**は、**zigzag~**オブジェクトのリンク・リストの指定されたインデックスの値をスキップする。正の数値は準方向にスキップし、負の数値は逆方向にスキップする。オプションの整数を用いて、次または前の値への遷移が起こるまでの時間を指定することができる（デフォルト値は0）。
- speed** 第1インレット：正または負の浮動小数点数値を伴う**speed**は数値を時間のペアが出力されるレートを指定する。メッセージ**speed 1.0**はり

ストを準方向に通常のスPEEDで移動し、**speed -1**はリストを逆方向に移動し、**speed .5**は通常のスPEEDの半分でリンク・リストを移動する(効果的にディレイ時間の値を倍にする)。

**start** ポジションを指定する1つの整数を(0が最初の要素) 伴う**start**は、**bang**によってトリガーされた時に、**zigzag~**オブジェクトが出力をはじめのポイントを設定する。

## アーギュメント

整数または実数 オプション。**zigzag~**オブジェクトの初期状態のターゲット値を設定する。

## 出力

**シグナル** 第1アウトレット：現在のターゲットの値、または現在格納されている値とターゲット時間によってターゲットの値までリニアに移動する。

第2アウトレット：現在のデルタ $x$ の値。

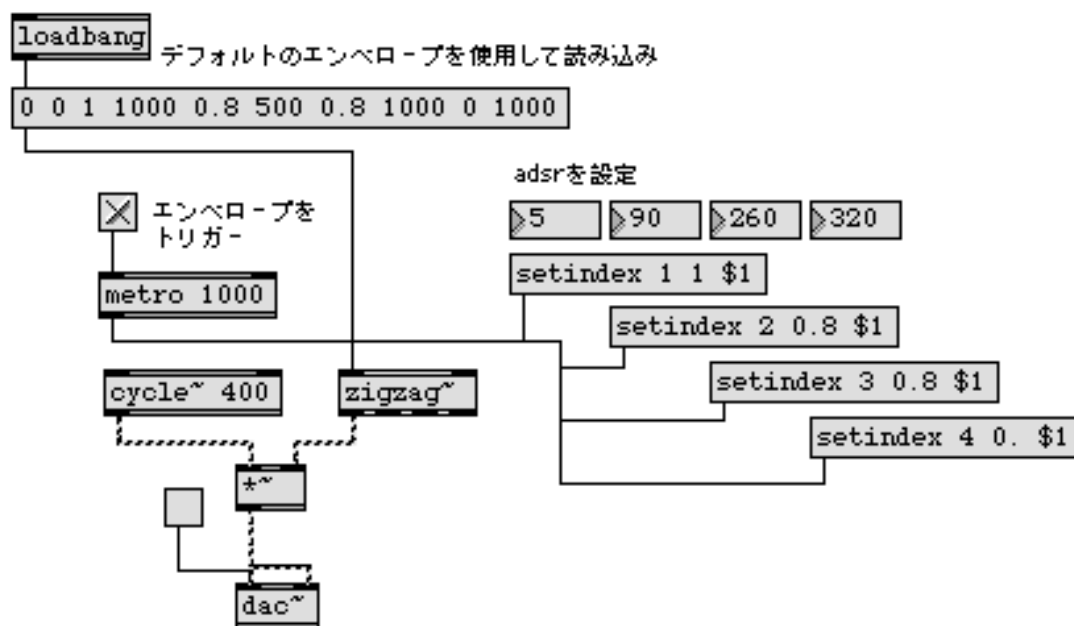
第3アウトレット：**dump**メッセージに反応して、すべての現在格納されている値と時間のペアを、

インデックス    ターゲットの値 ( $y$ )    デルタ $x$

という形式で出力する。

**bang** 第2アウトレット：ルーピングしている場合、ループ(リトリガー)ポイントに到達した時に、**bang**メッセージが出力される。**bang**は**zigzag~**がその変化のすべてを生成し終えた時にも出力される。

## 利用例



zigzag~ は多目的のエディット可能なランプ・ジェネレーターとして使うことができる

## 参照

curve~  
kink~  
line~

指数曲線、対数曲線のランプ・ジェネレーター  
ノコギリ波を歪ませる  
シグナル・ランプ・ジェネレーター

---

|            |                       |
|------------|-----------------------|
| 監修         | Supervising Editor    |
| 赤松正行       | Masayuki Akamatsu     |
| 翻訳 (五十音順)  | Translators           |
| 赤松正行       | Masayuki Akamatsu     |
| 伊藤隆之       | Takayuki Ito          |
| 川村武子 (tn8) | Takeko Kawamura (tn8) |
| 後藤英        | Suguru Goto           |
| 小林茂        | Shigeru Kobayashi     |
| 平井重行       | Shigeyuki Hirai       |
| 福島諭        | Satoshi Fukushima     |
| 古舘健        | Ken Furudate          |
| 真鍋大度       | Daito Manabe          |
| 安野大郎       | Taro Yasuno           |
| 由雄正恒       | Masatsune Yoshio      |