

原始動詞の機能一覧

(鈴木義一郎先生のまとめた機能一覧を書籍より)

演算子	片側形	両側形
=	自動分類を行う	等しい (論理演算)
=.	変数や関数の (局所的) 定義	
=:	変数の (大局的) 定義	
>	ボックスを開く	大きい (論理演算)
<	ボックスに入れる	小さい (論理演算)
>.	小数点を切上げた整数を与える	最大値を与える
<.	小数点以下を切捨てた整数	最小値を与える
>:	右側の数値に 1 を加える	大きいか等しい (論理演算)
<:	右側の数値から 1 を引く	小さいか等しい (論理演算)
+	共役複素数を与える	(要素ごとの) 足し算
+.	複素数の実部と虚部を与える	(論理和) を与える
++	2 倍にする 最大公約数	否定論理の和 (論理演算のみ)
-	逆符号を与える	(要素ごとの) 引き算
-.	論理否定 (0 を 1, 1 を 0 に)	右に含まれない要素を出力
-:	2 分の 1 にする	一致していれば 1 を与える
*	符号 (十〇一により 10-1)	(要素ごとの) 掛け算
*.	複素数の極座標を与える	最小公倍数 (論理積) を与える
*:	2 乗する	否定論理の積 (論理演算のみ)
%	逆数を与える	(要素ごとの) 割り算
%.	逆行列 (ベクトル) を与える	行列 (ベクトル) の割り算
%:	平方根をとる	右引数の左引数で与えた累乗根
^	指数関数の値	左引数の数値の右引数の累乗
^.	(底が e の) 自然対数の値	左引数で与えた底の対数
]	右側の内容を表示する	右引数を取り出す (E は左)
{	カタログ	左引数の軸のアイテムの取り出し
{.	先頭の要素の取り出し	指定個数の要素の取り出し
}.	先頭の要素の取り落し	指定個数の要素の取り落し
{:	末尾の要素の取り出し	
}:	末尾の要素の取り落し	
,	リスト化する	アイテムを 0 軸方向に接続する
..	テーブル化する	高いランクの最高軸方向に接続
,:	アレイのランクを 1 つ増やす	高いランクの形に合わせて結合
;	リストにときほぐす	ボックスで囲んで接続する

---

∴	単語ごとにボックスで囲む	
“	実行関数 (数値化する)	右の実行がエラーなら左を実行
“:	文字化する	左引数で指定した書式を与える
\$	ベクトルや行列の形を与える	ベクトルや行列などの形成
#	アレイのアイテムの個数	左引数の個数をコピーする
#.	2 進数の 10 進数化	左引数で指定した底でのベース化
#:	10 進数の 2 進数化	「#.」の逆演算
?	重複を許した整数乱数の生成	重複を許さない整数乱数の生成
~.	重複した要素を排除する	
~:	重複した要素に 0 を与える	等しくない (論理演算)
!	階乗の値を求める	2 項係数を求める
/:	(ベクトル) の昇順の番号	昇順に並べ替える
\:	(ベクトル) の降順の番号	降順に並べ替える
	(実数や複素数の) 絶対値	整数の割算の剰余
.	ベクトルや行列の逆順	ベクトルや行列の回転
:	行列の転置 (行と列の入れ替え)	左で指定した軸に関し転置
e.	アトムとオープンとの包含積	左引数の要素が右にあれば 1
i.	0 からの整数列 (行列等) の生成	右の要素の左引数の位置
j.	90 度回転した複素数を与える	$a j b$ は複素数 $\Gamma a + \text{硝}$
o.	円周率 「 $\pi$ 」 の倍数を与える	左引数の整数で指定した円関数

(原子動詞は他にもいくつかあるが、機能がやや複雑なので省略する)

## よく使われる副詞

～ 両側形動詞を修飾する場合

- i) 片側だけに入力したときは右引数を左引数にも入力する
- ii) 両側に入力したときは右引数と左引数の値を入れ替える

名詞を修飾する場合

代名詞や代動詞などの内容を呼び出す

/ 片側だけに入力したときは右引数のアイテム間に動詞を挿入する  
両側に入力すれば左右のアイテム間でのクロス演算のテーブル

/.  
対角要素を逐次取り出して演算する

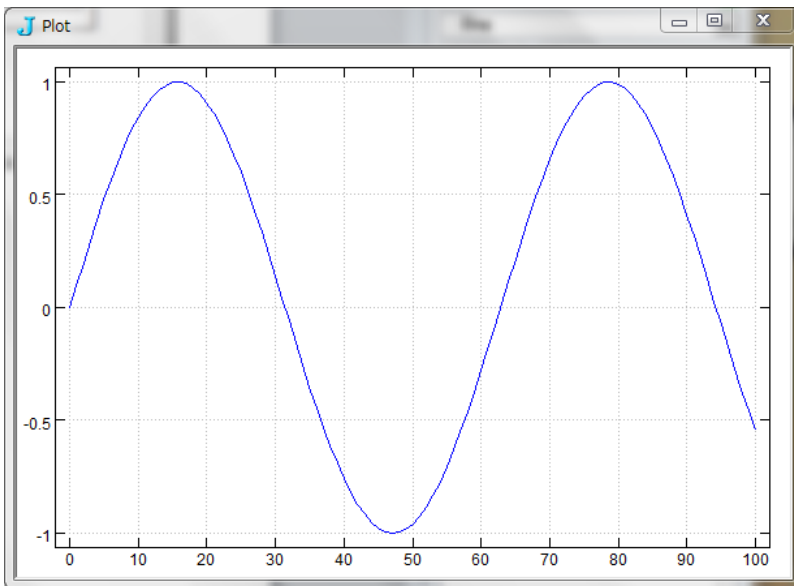
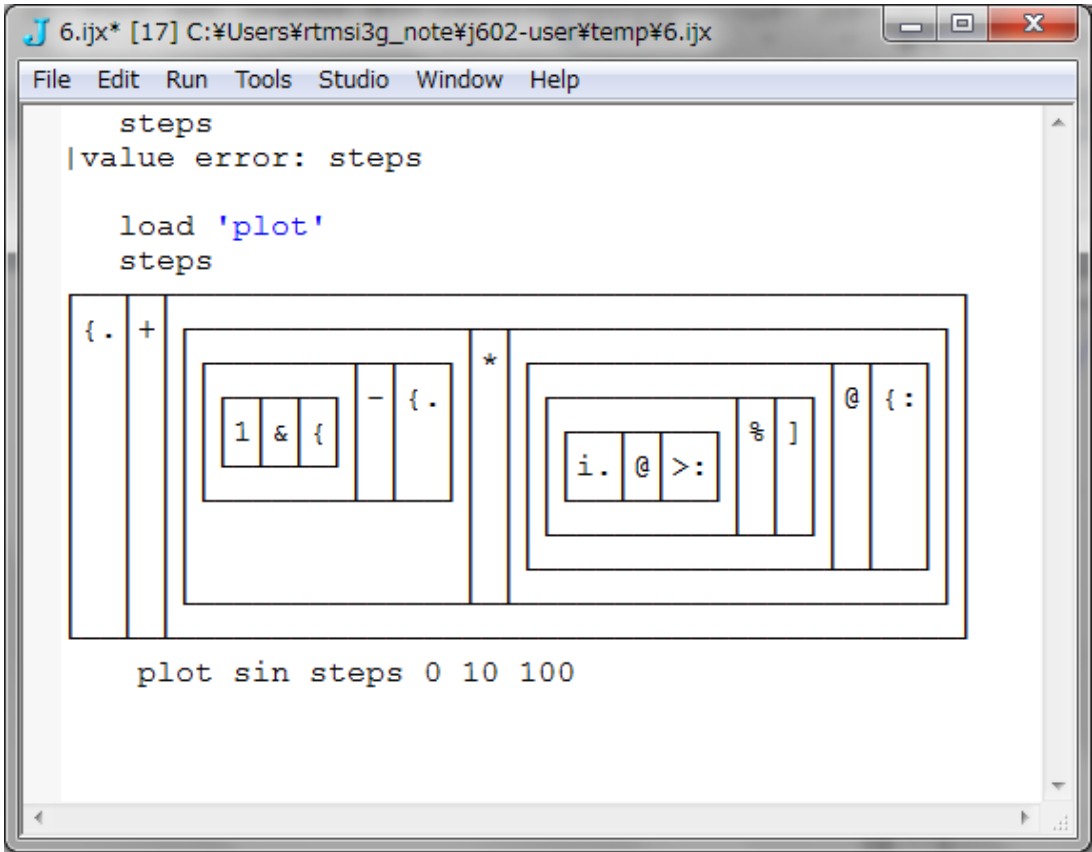
\ 片側だけに入力したときはアイテムを逐次追加して演算する  
左引数に正の整数を入力すればその個数分を重複して取り出す  
左引数に負の整数を入力すればその個数分を重複せず取り出す

\.  
片側だけに入力したときはアイテムを逆順で追加して演算する  
左引数に正の整数を入力すればその個数分を重複して取り去る  
左引数に負の整数を入力すればその個数分を重複せず取り去る

f. 先に定義された代動詞の内容を固定して定義する

(この他にも J 言語には種々の副詞があるが、その機能がかなり複雑なのでここでは省略する。また接続詞についてもいろいろなものがあり、数行で説明するのは困難なので割愛する。)

J602 で steps 関数は plot Addons を定義すると、使用可能となる。



J803 では、

```

J Term
File Edit View Run Tools Project Help
load 'plot'
plot sin steps 0 10 100
|value error: steps
| plot sin      steps 0 10 100
    
```

Plot を定義しても、steps 関数は使用できない。

そこで、拙作ながら steps 関数を自作してみた。

```

temp/2015_05_16_plotsincurve.ijs *
NB. 002_plot_3DGraph.ijs
NB. Example Sin curve-Graph
NB. By Taiwan Dong Hua Univ.Dr.Guo Pi
NB. Function steps By.R.Toribe
steps=.3 : 0 NB.initial data set
ss=.0}y      NB.Start point
ee=.1}y      NB.End point
tt=.2}y      NB.Times

pp=.(ee-ss)%tt NB.Average pich
dd=.ss+pp*i.tt+1 NB.Point Table
)

load 'plot'

plot sin steps 0 10 100
    
```

Edito 画面

```

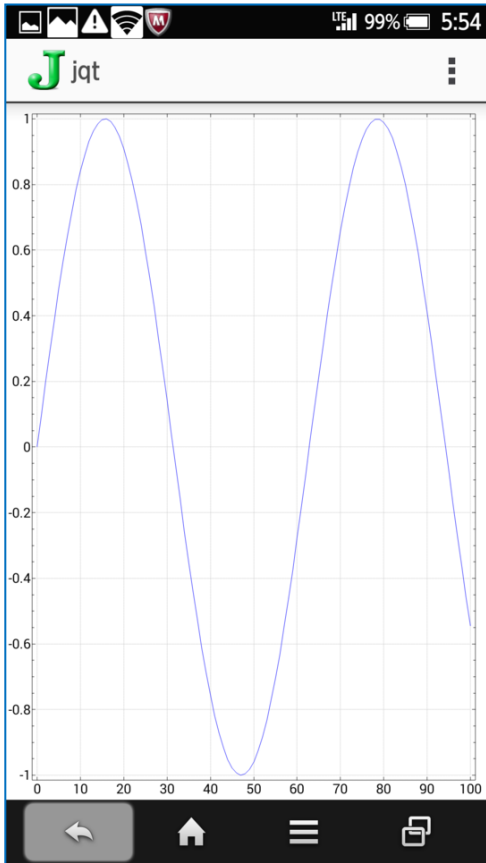
NB. 002_plot_3DGraph.ijs
NB. Example Sin curve-Graph
NB. By Taiwan Dong Hua Univ.Dr.Guo
NB. Function steps By.R.Toribe
  steps=.3 : 0 NB.initial data set
ss=.0}y      NB.Start point
ee=.1}y      NB.End point
tt=.2}y      NB.Times

pp=.(ee-ss)%tt NB.Average pich
dd=.ss+pp*i.tt+1 NB.Point Table
)

load 'plot'

plot sin steps 0 10 100
    
```

Terminal 画面



Plot 画面

NB. 002_plot_3DGraph.ijs
NB. Example Sin curve-Graph
NB. By Taiwan Dong Hua Univ.Dr.Guo Pin Xin(郭平欣)
NB. Function steps By.R.Toribe
steps=.3 : 0 NB.initial data set
ss=.0}y NB.Start point
ee=.1}y NB.End point
tt=.2}y NB.Times
pp=(ee-ss)%tt NB.Average pich
dd=.ss+pp*i.tt+1 NB.Point Table
)
load 'plot'
plot sin steps 0 10 100