

「J言語のクィック・リファレンス」

帝京平成大学 鈴木義一郎

【局所定義と大局定義】

局所定義は イコールピリ(=.) イコールコロンの(=:)で 大局定義
引数の 低次のランクの 全てのものを 「1セル」 「2セル」 などと呼ぶ
演算は 1つ低次の セル相手 これを名づけて 「アイテム」と呼ぶ

【J言語の特徴】

タシット(Tacit)で 定義するのが 醍醐味さ J特有の 面白さ 演算を 右から順に 作動さす ことも可能さ エクスプリシト(Explicit)	(sum=:+/)D=:3 1 2 6 (sum1=:3 :'+/y.')D 6	(mean=:+/%#)D 2 3 :'+(+/y.)%#y.'D 2	
動詞が3つ 並んだときは 左右が先よ 中の動詞は 3番手(フォーク Fork)	+/D=:3 1 2 6	# D 3 (+/%#)D 2	
片側動詞に 両側動詞が 連結すれば カッコでくくり これ「フック(Hook)」	(-mean=:+/%#)D 1 1 0	(]-mean)D 1 1 0	
並んだ動詞は 右からフォーク 残った動詞で またフォーク(フック)	(]-+/%#)D 1 1 0	(-+/%#)D 1 1 0	
動詞との 出会いひたすら 待つ「副詞」 右にこだわる 「接続詞」	av=:/ 4!:0<'av' 1 NB. 副 詞	c=:& 4!:0<'c' 2 NB. 接続 詞	mean=:+/%# 4!:0<'mean' 3 NB. 動詞
計算を マトメテ演算 したければ レベル(L:0)やイーチ(&>)を 使えばよい オープン(>)は 動詞でアンド(&)は 接続詞 イーチ(&> &.>)にすれば 副詞に変身]d=:1 2 3;4 5 1 2 4 5 3 +/L:0 d 6 9	+/&> d 6 9 +/&.> d 6 9	

(+/.*)は 行列同士の掛算を 行う便利な プリミティブ	C +/. * :C=:i.2 3 5 14 14 50
(-/.*)は 片側動詞で 正方向列の	-/. * A=:1+I.2 2 (1*4)-(1*2)

行列式の値を 出力す	2	2
([%1:,.])は 回帰係数を 出力する	v=:3 5 6 [u=:1 2 4	
チョー便利な 定義関数	v ([%1:,.]) u	
	2.5 0.928571	

【 “+ . +:” 】

実数に プラス(+) の片側 そのままで 複素数には 共役複素数	+ 0.4 0 3 0.4 0 3	+ 3j4 3j_4
プラス(+) の両側 フツアの足し算 複素数でも イッツオーケー	1 2 3 + 4 5 6 5 7 9	1j2 + 3j4 4j6
プラスピリ(+.) 複素数値に 適用すれば 実部と虚部の 数値を出力	+ . 3 3 0	+ . 3j4 3 4
プラス・ピリ(+.) の 両側形は 最大公約数 複素数でも(イッツオーケー)	6 +.8 2 1j2 % 0j1 2j_1	1j2 +. _1j2 0j1 _1j2 % 0j1 2j1
]Z1=:j.^:(i.4)1 1 0j1 _1 0j_1 (]+.)Z1 1 0j1 0j1 1]Z11=:j.^:(i.4)1j1 1j1 _1j1 _1j_1 1j_1 (]+.)Z11 1j1 1j1 1j1 1j1]Z12=:j.^:(i.4)1j2 1j2 _2j1 _1j_2 2j_1 (]+.)Z12 0j1 1 1 0j1
/:~ Z1,Z11,Z12 _2j1 _1j_2 _1j_1 _1 _1j1 0j_1 0j1 1j_1 1 1j1 1j2 2j_1		
プラスコロン(+:) 右引数を 倍にする 両側形は 機能無し	+: 2 4	+: 3j4 6j8

【 “- . -:” 】

マイナス(-) の 片側形は 符号の反転 複素数でも 実・虚同時に	- _5 0 2 5 0 _2	- 3j4 3j_4
マイナス(-) の両側 フツアの引き算よ 複素数でも イッツオーケー	4 5 6 - 1 3 2 3 2 4	3j4 - 1j2 2j2
マイナスピリ(-.) の 片側形は 足して1になる 補数を出力	-. 4 0.3 1 3 0.7 0	-. 1j2 0j_2
マイナスピリ(-.) の 両側形は 右引数に 無いモノを出す	(a=:1 2)-.b=:1 3 5 2	(-. a e. b)#a 2
マイナスコロン(-:) の 片側形は 右引数を 半分にする	-.: 2 4 6 1 2 3	-.: 2j4 1j2
マイナスコロン(-:) の 両側形は 形まで含めて 一致か否か	1 2 -: 1 2 1	1 2 = 1 2 1 1

【 “ , : ” 】

コンマ(,)という 動詞の 片側形は 右引数を リストに変換]t=:i.2 3 0 1 2 3 4 5	,t 0 1 2 3 4 5
コンマにピリ(,)の 片側形は 右引数を テーブル化]a=:i.2 0 1	,. a 0 1
コンマにコロン(:)の 片側形は ランクを1つ 上げたアレイに	,: a 0 1	\$,: a 1 2
コンマ(,)という 動詞の 両側形は ランクを増やさず 左右を接続	1 2 3 , 4 5 1 2 3 4 5	(A=:i.2 3),b=:6 7 8 0 1 2 3 4 5 6 7 8
コンマにピリ(,)の 両側形は 左右の引数を 横に接続 (ランクが増すことも、変わらないこともある)	1 2 3 ,. 4 5 6 1 4 2 5 3 6	(:A),.b 0 3 6 1 4 7 2 5 8
コンマにコロン(:)の 両側形は ランクを上げた アレイになる (ランクは 必ず1つ上がる)	1,:2 1 2 1 2 3 ,: 4 5 6 1 2 3 4 5 6	A,:4 5 0 1 2 3 4 5 4 5 0 0 0 0

【 “ i. i: ” 】

整数を 瞬時に作る アイにピリ(i.) 但し始点は 0にご注意(1ではない!)	i. 3 0 1 2	i._3 2 1 0
マイナスの 整数値まで 出力す	i: 3	i: _3
iにコロン(i:)は 重宝動詞	3 2 1 0 1 2 3	3 2 1 0 1 2 3
(i.)の両側形は 左指定の 数のインデックスを 右引数のリストに 与える	1 2 1 i.1 1 2 1 0 0 1 0	1 2 1 i.1 1 2 4 0 0 1 3
(i:)の両側形は (i.)の結果と ほぼ同じ ただインデックスは 後ろから	1 2 1 i:1 1 2 1 2 2 1 2	1 2 1 i:1 1 2 4 2 2 1 3

【 “* * .*” 】

シグナム(*)は 符号与える 片側形 複素数には 単位円に射影	* _3 0 2 _1 0 1	([,])* 3j4 1 0.6j0.8
スターピリ(*.) 複素数には 大きさと 偏角与える 片側関数	*. _3 2 3 3.14159 2 0	*. _3j0 3 3.14159 *. 3j4 5 0.927295
スター・ピリ(*.)の 両側形は 最小公倍数 複素数でも(イツオーケー)	4 *. 6 12	1j1 *. 3j4 7j1
スターコロンの(*)は 平方値 平方根なら パーセントコロンの(%)	*: 4 16	:%: 4 2

【 “% % .%*” 】

パーセント(%) 片側形なら 逆数で 両側形なら 割算を行う	% 2 4 5 0.5 0.25 0.2	2 4 5 % 2 1 2 2.5
行列の 割算行う パーセントピリ(%) 片側形なら 逆行列 「鶴と亀の頭が14個で、足が40本である。鶴と亀はそれぞれ何匹づついるか？」	14 40 % A=:2 2\$1 1 2 4 8 6]B=:%. A 2 _0.5 1 _0.5	A+/. *B 1 0 0 1
パーセントコロンの(%) 片側形なら 平方根 両側形は 累乗根	:%:2 4 9 1.41421 2 3	3 %: 8 27 2 3

【 “< > ; < . > . < : > :” 】

ボックス(<)で 囲めば全てが アトムに変身 オープン(>)使って 蘇生する ボックスで 囲み連結 セミコロン(;) 片側形なら リストに変身!]B=:<1 2 3 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>2</td></tr><tr><td>3</td><td></td></tr></table>]A=:1 2 ; 3 4 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1 2</td><td>3 4</td></tr></table>	1	2	3		1 2	3 4	> B 1 2 3 ; A 1 2 3 4
1	2							
3								
1 2	3 4							
小にピリ(<.) 片側形は 切り捨てる 大にピリ(>.)なら 切り上げる	<. 3.14 3	>. 3.14 4						
数値から 1をマイナス 小にコロンの(<:) 大にコロンの(>:)は 1を加える	<: 3 3.14 2 2.14	>: 3 3.14 4 4.14						
小にピリ(<.) 両側形は 小さいほう 大にピリ(>.)なら 大きいほう	3 <. 3.14 3	3 >. 3.14 3.14						
小にコロンの(<:)や 大にコロンの(>:)は	3 <: 3.14	3 >: 3.14						

イコール含む不等式(\leq, \geq)の 論理演算	1	0
---------------------------------	---	---

【 “^ ^ . ^:” 】

ハット(^)という 動詞の片側形は 指数関数を 出力す	$\hat{\ } i.3$ 1 2.71828 7.38906	$(1x1)^{\ } i.3$ 1 2.71828 7.38906
ハット(^)という 動詞の両側形は 左の数だけ 累乗す	$\hat{\ } 2 3 \hat{\ } 3$ 4 8 81	$\hat{\ } \&2(2 3)$ 4 9 4 9 16
ハットピリ(^.) 片側形は 自然対数 両側形は 左を底の対数值	$\hat{\ } . 1 2$ 0 0.693147	$5 10 \hat{\ } . 125 100$ 3 2
ハットコロンの(^:) 反復演算の 接続詞 マイナス1なら 逆演算	$>:\hat{\ }:2 i.3$ 2 3 4	$>:\hat{\ }:_1 (2 3 4)$ 1 2 3

【 “! ! ^!_0 ^!_1 ^!_2 ^!_1 ^!_2 !:” 】

感嘆符(!) 片側形は 階乗よ 両側形は 2項係数 (“!. ” や “!: ” は 接続詞)	$! 3 4 5$ 6 24 120 2 ! 5 10	$! 0.5 1.5$ 0.886227 1.32934 (bic=:i.@>:!)5 1 5 10 10 5 1
ビックリピリ(!.)は ハット(^)と数を接続し 両側動詞を 生成する(custemize) (どんな目的で、このような関数が必要なのかは不可解！)		
$f0=:\hat{\ }!.0$ $g0=:[:*/[+0:*i.@]$	$f0^0^c=:2 3$ 4 4 27 256	$g0^0^c$ 3 f0 2 9 8
$f1=:\hat{\ }!.1 [g1=:[:*/[+1:*i.@]$ $f2=:\hat{\ }!.2 [g2=:[:*/[+2:*i.@]$	$f1^0^c=:2 3 4$ 6 60 840 8 105 1920	$g1^0^c$ 6 60 840 8 105 1920
$f_1=:\hat{\ }!._1 [g_1=:[:*/[+_1:*i.@]$ $f_2=:\hat{\ }!._2 [g_2=:[:*/[+_2:*i.@]$	$f_1^0^c$ 2 6 24 0 3 0	$g_1^0^c$ 2 6 24 0 3 0

“0! :n”, “1! :n”, …… , “7! :n”, “9! :n”, “11! :n”, “13! :n”, “14! :n”, “15! :n”
 “128! :0”, “128! :1” は、外部接続詞で、いろいろなシステム関数が用意されている。

データメな 数を生み出す	? 5 5 5 5 5	5 ? 5
ハテナキー(?) 両側形は 重複許さず	1 0 4 4 1	0 2 3 1 4
ハテナピリ(?) シード固定の 乱数よ	? . 5 5 5 5 5	5 ? . 5
両側形は 非重複固定乱数	1 0 4 2 4	1 4 0 3 2
	? . 5 5 5 5 5	5 ? . 5
	1 0 4 2 4	1 4 0 3 2

【 “| | :” , “+ / . *” , “- / . *” , “[% . 1 : . .]” 】

割算の 余り求める 棒()一本	3 i.6	1 _2 3 _4									
片側形なら 絶対値	0 1 2 0 1 2	1 2 3 4									
棒にピリ(.) 片側形なら アイテムの 順序をそっくり 逆にする	.d=:1 2 3 4 5										
棒ピリ(.)の 両側形は 左の数だけ 右に回転(rotate) 負なら左へ	1 .d	_1 . d									
	2 3 4 5 1	5 1 2 3 4									
	2 .d	_2 . d									
	3 4 5 1 2	4 5 1 2 3									
棒にコロンの(:) 片側形なら アレイの軸の 順序をソックリ 入れ替える	m; :m=:2 3\$' abcdef'										
	<table border="1"> <tr><td>abc</td><td>Ad</td></tr> <tr><td>def</td><td>be</td></tr> <tr><td></td><td>cf</td></tr> </table>		abc	Ad	def	be		cf			
abc	Ad										
def	be										
	cf										
棒にコロンの(:) 両側形は 左指定の 軸を0軸に 転置(transpose)する 左にボックスの データを入力すれば 対角要素を 出力す	(1 0 :m); (1 :m); (<0 1) :m										
	<table border="1"> <tr><td>ad</td><td>abc</td><td>Ae</td></tr> <tr><td>be</td><td>def</td><td></td></tr> <tr><td>cf</td><td></td><td></td></tr> </table>		ad	abc	Ae	be	def		cf		
ad	abc	Ae									
be	def										
cf											

セイム(same [,])は 左右のいずれかを 出力させる 便利な動詞	2 3 [4 5	2 3] 4 5
	2 3	4 5

キャップ(:)はなんとも 不思議な動詞 何もしないで フォークを作る	([:*+:)2 16	([:%+:)2 2	%:&+: 2 2
	c=[:>+: c 1 2 3 3 5 7	h=[:>:+* 2 h 1 4	
キャップ([: cap])は 演算結果に 関係せず	abs=: : [: abs _1 2 3 1 2 3	res=:[: : 2 res _1 2 3 1 0 1	

Obverse(.:)は 逆が正しい 定義なら 逆変換(^:_1)で 元に戻る(関数 “f”)	f=:* :. %: f i.5 0 1 4 9 16 f^:_1 f i.5 0 1 2 3 4	g=:* :. +: g i.5 0 1 4 9 16 g^:_1 g i.5 0 2 8 18 32
---	---	---

【アレイの形と変形】

形なき たったひとつは 「アトム」なり アトムが並んで 「リスト」を作る	\$ 2 # 2 1	\$ 2 1 # 2 1 2
テーブルの 「形」を示す ドル(\$)マーク アイテム数は シャープ(# talley)さん]M=:i.2 3 0 1 2 3 4 5	\$ M # M 2 3 2
右で与えた データから 左指定の 個数取り出す 両側コピー(# copy)	0 1 1 # 1 3 5 3 5	0 1 # M 3 4 5
シャープ・ピリ(#) 片側形は 2進数 10進数の 数値に変換	#.1 0 1 5	+/(1 0 1)*2^2 1 0 5
左で与えた 進数で 右の数値を変換す シャープ・ピリ(#)の 両側形	10 #. d=.1+i.4 1234 8 #. d 668	+/d*(10^3 2 1 0) 1234 +/d*(8^3 2 1 0) 668
シャープ・コロンの(:) 片側形は 10進数を 2進の数値に 変換す]b=:#:3 5 7 0 1 1 1 0 1 1 1 1	#.b 3 5 7
シャープ・コロンの(:) 両側形は	t=:24 60 60]tt=(*/¥.}.t),1

シャープ・ピリ(#.)の 両側形の逆変換]s=:t #.2 3 4 7384 t #: s 2 3 4	3600 60 1 +/2 3 4*tt 7384
		self_reference=:1:~(*\$:@<:)@.* self_reference 5 120	! 5 120
		1:~(*<:)@.* 5 20	
		*:&\$.5 25	*:&.(\$.^:_1)5 25
	0\$. 0 1 3 1 1 2 3 \$ 0\$. 0 1 3 3	0\$. i.2 2 0 1 1 1 0 2 1 1 3 \$ 2\$. i.2 2 2	2\$. i.2 2 0 1

【接続詞 右にこだわる 接着剤】

片側の 動詞を順に 結ぶのが アンド(&)やアット(@ @:)の 接続詞	*:&+: 2 16	*:@+: 2 16	*:@+: 2 16
アンダー(&.)で 2つの動詞を 連結すれば 逆演算が 付加される	*:&.+ : 2 8	-:&*:&+: 2 8	
動詞と名詞を アンド(&)で結べば 新たな動詞を 作り出す(“@”は不可)	*&2 a=:2 3 4 6	2&* a 4 6	+ : a 4 6
複数の 動詞を交互に 連結するのは タイ(tie)と呼ばれる 接続詞	+`*/ i.6 29 +`%/ 3 1 4 3.25	0+1*2+3*4+5 29 3 + 1 % 4 3.25	
(u`v`:)は 全ての動詞を 演算し (u`v`:3)は (u`v/)と 同じに機能 (u`v`:6)は (uv)というフック (u`v`w`:6)は (uvw) というフォーク と同じ演算	+`-`:`:0(3 2 4) 6 4 8 1.5 1 2 +`*`:`:3 a=:1 2 3 7 +`*`:`:6(3 2 5) 4 3 6 +`*`-`:`:6(3 2 5) 9 4 25	(+`,`:`,`)3 2 4 6 4 8 1.5 1 2 +`*/ a 7 (+*) 3 2 5 4 3 6 (+*-) 3 2 5 9 4 25	1+2*3 7
Even(..)は 2つの動詞を 接続し 別の動詞を 作り出す	*: .. +: 4 40 +: .. *: 4 20 >: .. <: 5 5.5 <: .. >: 5 4.5	-:@(*:~*:&+:) 4 40 -:@(+:~+:&*:) 4 20 -:@(>:+>:&<:)5 5.5 -:@(<:+<:&>:)5 4.5	
Odd(.:)は 2つの動詞を 接続し 別の動詞を 作り出す	*: .: +: 4 _24 +: .: *: 4 _12 >: .: <: 5 0.5	-:@(*:~*:&+:) 4 _24 -:@(+:~+:&*:) 4 _12 -:@(>:->:&<:)5 0.5	

	$\langle; .: \rangle: 5$ 0.5	$-\:@(\langle;-\langle; \& \rangle:)5$ 0.5																								
セミコロン(;)の 片側形は 右のレイを リストにほぐす	$; i.2 3$ 0 1 2 3 4 5 $\langle 0 1 2 \rangle, \langle 3 4 \rangle$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td><td>3</td><td>4</td></tr> <tr><td>2</td><td></td><td></td><td></td></tr> </table>	0	1	3	4	2				$;/ i.2 3$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td><td>3</td><td>4</td></tr> <tr><td>2</td><td></td><td>5</td><td></td></tr> </table> 0 1 2 ; 3 4 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td><td>3</td><td>4</td></tr> <tr><td>2</td><td></td><td></td><td></td></tr> </table>	0	1	3	4	2		5		0	1	3	4	2			
0	1	3	4																							
2																										
0	1	3	4																							
2		5																								
0	1	3	4																							
2																										
$(];.0)$ は 全ての軸を 逆順に	$];.0 i.2 2$ 3 2 1 0	$ \."1 . i.2 2$ 3 2 1 0																								
	$2 2];.0 i.3 2$ 0 1 2 3	$2 1];.0 i.3 2$ 0 2																								
$\langle; .1)$ は先頭 $\langle; .2)$ は末尾 フレット(0 1)の表れた 位置で切る	$\langle; .1 (3 2\$i.4)$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>0 1</td></tr> <tr><td>1</td><td></td></tr> <tr><td>2</td><td></td></tr> <tr><td>3</td><td></td></tr> </table>	0	0 1	1		2		3		$\langle; .2 (3 2\$i.4)$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>2 3</td></tr> <tr><td>1</td><td>0 1</td></tr> </table>	0	2 3	1	0 1												
0	0 1																									
1																										
2																										
3																										
0	2 3																									
1	0 1																									
$\langle; .1)$ や $\langle; .2)$ の 両側形は 先頭や末尾から 1が現れると 区切る	$1 0 1 \langle; .1 i.3$ 2 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>4 5</td></tr> <tr><td>1</td><td></td></tr> <tr><td>2</td><td></td></tr> <tr><td>3</td><td></td></tr> </table>	0	4 5	1		2		3		$0 1 0 \langle; .2 i.3$ 2 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td></td></tr> <tr><td>1</td><td></td></tr> <tr><td>2</td><td></td></tr> <tr><td>3</td><td></td></tr> </table>	0		1		2		3									
0	4 5																									
1																										
2																										
3																										
0																										
1																										
2																										
3																										
$\langle; ._1)$ は先頭 $\langle; ._2)$ は末尾 フレットを除いて 区切りを入れる	$\langle; ._1 (3 2\$i.4)$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>2</td><td></td></tr> <tr><td>3</td><td></td></tr> </table>	2		3		$\langle; ._2 (3 2\$i.4)$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td></td><td>2 3</td></tr> </table>		2 3																		
2																										
3																										
	2 3																									
$\langle; ._1)$ や $\langle; ._2)$ の 両側形は $\langle; .1)$ や $\langle; .2)$ の 先頭や末尾を 削除する	$1 0 1 \langle; ._1 i.3$ 2 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>2</td><td></td></tr> <tr><td>3</td><td></td></tr> </table>	2		3		$0 1 0 \langle; ._2 i.3$ 2 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td></td></tr> <tr><td>1</td><td></td></tr> </table>	0		1																	
2																										
3																										
0																										
1																										
$();.3)$ の 片側形は	$\langle; .3 i.3$	$3 \langle; .3 i.3$																								

	<table border="1"> <tr><td>0</td><td>1</td><td>1 2</td><td>2</td></tr> <tr><td>2</td><td></td><td></td><td></td></tr> </table>	0	1	1 2	2	2				<table border="1"> <tr><td>0</td><td>1</td><td>1 2</td><td>2</td></tr> <tr><td>2</td><td></td><td></td><td></td></tr> </table>	0	1	1 2	2	2																			
0	1	1 2	2																															
2																																		
0	1	1 2	2																															
2																																		
	<p>2 2 <;.3 i.3 2</p> <table border="1"> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>3</td></tr> <tr><td>2</td><td></td></tr> <tr><td>3</td><td></td></tr> <tr><td>2</td><td>3</td></tr> <tr><td>3</td><td>4</td></tr> <tr><td>4</td><td></td></tr> <tr><td>5</td><td></td></tr> <tr><td>4</td><td>5</td></tr> <tr><td>5</td><td></td></tr> </table>	0	1	1	3	2		3		2	3	3	4	4		5		4	5	5		<p>1 2 <;.3 i.3 2</p> <table border="1"> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td></td></tr> <tr><td>2</td><td>2</td></tr> <tr><td>3</td><td></td></tr> <tr><td>4</td><td>5</td></tr> <tr><td>5</td><td></td></tr> </table>	0	1	1		2	2	3		4	5	5	
0	1																																	
1	3																																	
2																																		
3																																		
2	3																																	
3	4																																	
4																																		
5																																		
4	5																																	
5																																		
0	1																																	
1																																		
2	2																																	
3																																		
4	5																																	
5																																		

ボックスで 与えた要素の 組合せ 片側動詞の カタログ({ catalogue)なり (アトムに対しては“ボックス(<)”と同じ)	<p>0 1 ; 2 3</p> <table border="1"> <tr><td>0</td><td>2 3</td></tr> <tr><td>1</td><td></td></tr> </table> <p>{3}</p> <table border="1"> <tr><td>3</td></tr> </table>	0	2 3	1		3	<p>{ 0 1 ; 2 3</p> <table border="1"> <tr><td>0</td><td>0 3</td></tr> <tr><td>2</td><td></td></tr> <tr><td>1</td><td>1 3</td></tr> <tr><td>2</td><td></td></tr> </table>	0	0 3	2		1	1 3	2	
0	2 3														
1															
3															
0	0 3														
2															
1	1 3														
2															
中カッコ({} 左で与えた インデクスの アイテムを取る 両側関数	<p>1 { 1 2 3</p> <p>2</p>	<p>0 { i.2 3</p> <p>0 1 2</p>													
カッコ閉じ()} 左で与えた インデクスの アイテム修正 両側関数	<p>1 1 0 } i.2 3</p> <p>3 4 2</p>														
修正値と インデクスを左に 入力すれば 右引数の値を 修正す() amend)	<p>3 4(0 1)}i.3</p> <p>3 4 2</p>														
ヘッド({.}で先頭 テール({:})で末尾 要素取り出す 片側動詞	<p>{. K=:1+i.3</p> <p>1</p>	<p>{: K</p> <p>3</p>													
左で与えた個数分 take({.}は取りで drop().)は除く 便利な両側 動詞なり	<p>2 {. K</p> <p>1 2</p>	<p>2 }. K</p> <p>3</p>													

【 “: . :.” 】

単項と 2項の動詞を コロン(:)で結べば 同時に定義 できますよ	<p>log=:10&^. : ^.</p> <p>log 10 100</p> <p>1 2</p>	<p>8 log 10 100</p> <p>1.10731 2.21462</p>
Obverse(:.)は 逆が正しい 定義なら	<p>f=:* :. %:</p>	<p>g=:* :. +:</p>

<p>逆変換(\wedge:_1)で 元に戻る(関数“f”) (“g”の定義関数では“:.”の右の関数が 左の関数の逆関数でないので右の関数の演算結果が表示される)</p>	<pre>f i.5 0 1 4 9 16 f^:_1 f i.5 0 1 2 3 4</pre>	<pre>g i.5 0 1 4 9 16 g^:_1 g i.5 0 2 8 18 32</pre>
<p>「u :: v」は エラーが無ければ “u” エラーがあれば “v” の演算結果を出力 (“::”は「Adverse」という接続詞)</p>	<pre>p=:3 1 0 2 q=:3 1 1 0 test=:A. ::(!@#) test p 20</pre>	<pre>test q 24 A. q index error A.q</pre>

【副詞は後から 役割果す 動詞の活躍 拡大す】

両側動詞に ウェーブ(~)つけりゃ 右引数を 左にも	*/~ a=:1+i.3 1 2 3 2 4 6 3 6 9	a */ a 1 2 3 2 4 6 3 6 9
左右に数値が ある場合には 左右の引数を 交換す	5 %~ i.5 0 0.2 0.4 0.6 0.8	(i.5) % 5 0 0.2 0.4 0.6 0.8
ウェーブ・ピリ(~.nub)の 片側形は 重複要素を 排除する ウェーブ・コロン(~:)の 片側形は ダブりの位置に “0” を与える	a =: 1 2 1 3 3 2 1 ~. a 1 2 3	~:a 1 1 0 1 0 0 0 1 2 3
ウェーブ・コロン(~:)の 両側形は 各要素毎の 不一致に “1” マイナス・コロン(-:)の 両側形は 引数マトメテ 一致に “1” (match)	(1 2)~:2 1 1 1 (1 2)~:1 2 0 0 (1 2)~:2 1 3 length error	(1 2)-:2 1 0 (1 2)-:1 2 1 (1 2)-:2 1 3 0

アンダー・バー(_)は 負数を示す 名詞 無限大(∞)も示す 両刀使い	- 3.14 _3.14	- _3.14 3.14
“_” は 無限大(∞)を表す 名詞なり	2 % _ 0	2 % __ 0
“_.” は 不定形(indeterminate)の名詞	_ - _	3 + _. 0
“_:” は 無限大(∞)を出す 動詞なり	_: ''	([]%_:)2 0

“” _” は 数値につけて、“動詞化” す	([]+1”_)i.3 1 2 3	([]+1:)i.3 1 2 3	>: i.3 1 2 3
---------------------------	----------------------	---------------------	-----------------

データに プラス・スラッシュ(+/)	+ / 3 1 2													
合計算	6													
スラッシュ・ピリ(/.) 右引数の テーブルを 逐一斜めの 対角要素(oblique)	</. i.3 4													
	<table border="1"> <tr> <td>0</td> <td>1 4</td> <td>2 5</td> <td>3 6</td> <td>7 10</td> <td>11</td> </tr> <tr> <td></td> <td></td> <td>8</td> <td>9</td> <td></td> <td></td> </tr> </table>		0	1 4	2 5	3 6	7 10	11			8	9		
0	1 4	2 5	3 6	7 10	11									
		8	9											
イコール(=)記号の 片側形は 分類・集計 に チョー便利	a=:1 2 3 1 3 2 1													
]b:= a	(~.a)=a	+/"1 b												
1 0 0 1 0 0 1	1 0 0 1 0 0 1	3 2 2												
0 1 0 0 0 1 0	0 1 0 0 0 1 0													
0 0 1 0 1 0 0	0 0 1 0 1 0 0													
		(~.a), :+/"1 b												
		1 2 3												
		3 2 2												
b <@# A=: 'abcdefg'	a </. A													
<table border="1"> <tr> <td>adg</td> <td>bf</td> <td>ce</td> </tr> </table>	adg	bf	ce	<table border="1"> <tr> <td>adg</td> <td>bf</td> <td>ce</td> </tr> </table>		adg	bf	ce						
adg	bf	ce												
adg	bf	ce												
データを 昇順にする グレードアップ(/:~)	/:~ 3 1 2	¥:~ 3 1 2												
グレードダウン(¥:~)は 降順に	1 2 3	3 2 1												

【 “ ” “:” 】

ダブルクォート(")は ランク指定の 接続詞 (右の例では、 “ ” 2” は無くともよ い)	X=:7 8 9 [Y=:i.2	X, "2 Y
	3	7 8 9
	X, "1 Y	0 1 2
+/i.2 3	+/"1 i.2 3	3 4 5
3 5 7	3 12	7 8 9 3 4 5
ダブルクォート(") ピリ(.)で数値化	1+":2	1+ ". ":2
コロン(:)で文字化 書式も与える スグレモノ (数値と文字の足し算は、エラーになる!)	domain error	3
	1 +":2	
ダブルクォートピリ(".) 文字で書かれた 内容を 瞬時に実行(do) してくれる	a=: '1+2+3'	". a
		6
ダブルクォートピリ(".)の 両側形は 隙間を左の 数値で埋める]b=: '1 2 3', '4 5', ':'	8 ". b
	1 2 3	1 2 3
	4 5	4 5 8
		8 8 8

<pre>]s=:a.i.'aA'</pre> <pre>97 65</pre> <pre>(s+/i.26){a.</pre> <pre>abcdefghijklmnopqrstvwxyz</pre> <pre>ABCDEFGHIJKLMNPOQRSTUVWXYZ</pre> <pre>a:</pre> <pre>++</pre> <pre> </pre> <pre>++</pre>	<pre>]r=(i.6)A.012</pre> <pre>012</pre> <pre>021</pre> <pre>102</pre> <pre>120</pre> <pre>201</pre> <pre>210</pre> <pre>A.r</pre> <pre>012345</pre>	<pre>(i.6)A.'abc'</pre> <pre>abc</pre> <pre>acb</pre> <pre>bac</pre> <pre>bca</pre> <pre>cab</pre> <pre>cba</pre>																
「nbm」は「m」の「n進数」の10進数の値	8b22	(2*8)+2																
“u b.0”は動詞(u)のランクを表示	*:b.0	18																
“u b.1”は動詞(u)の「原始定義」	*:b.1	(^b.1)i.33																
ブール代数	<pre>\$&1@{.,@\$)</pre> <pre><"2 :(78114)b./~10</pre> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>1</td><td>0 0</td><td>1 0</td><td>0 1</td></tr> <tr><td>1</td><td>0 1</td><td>0 0</td><td>1 1</td></tr> <tr><td>1</td><td></td><td></td><td></td></tr> <tr><td>0</td><td></td><td></td><td></td></tr> </table>	1	0 0	1 0	0 1	1	0 1	0 0	1 1	1				0				10 10
1	0 0	1 0	0 1															
1	0 1	0 0	1 1															
1																		
0																		
“b._1”は逆関数を出力	*:b._1	+:b._1																
	%:	:-																
	<pre>]p=:5?5</pre> <pre>02431</pre> <pre>/:p</pre> <pre>04132</pre>	<pre>C.p</pre> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>3</td><td>4</td><td>1</td></tr> <tr><td></td><td></td><td>2</td><td></td></tr> </table>	0	3	4	1			2									
0	3	4	1															
		2																
	<pre>pC.56789</pre> <pre>57986</pre>	<pre>p{56789</pre> <pre>57986</pre>																
“u d.n”は関数(u)のn階数値微分 (ランクは0に固定)	*:d.1	^.d.1																
	+:	%																
	(^.**:)d.1	(1:+_3&***:)d.1																
	(%* **:) + ^.* +:	3 2x&p.																
“u D.n”も関数(u)の n階数値微分	4	(cube=:^&3"0)t=:23	cube D.1 t	cube D.2 t														
		12 27 48		12 18 24														

	8 27 64		
“D:1” は 平均変化率	1 cube D:1 t		
	19 37 61		

“x. E. y.”は yの xを含む位置に “1”を与える (menber)	'co' E. 'cocoa' 1 0 1 0 0	8 7 6 7 9 e. 7 8 9 1 1 0 1 1
“j.” の片側形は 複素平面上での 90度の回転 (実数には純虚数)	j. 1 2 3 0j1 0j2 0j3	j. 1j1 3j4 1j1 4j3
“j.” の両側形は 「x. j. y.」 = 「x. + j.y.」	3 j. 4 3j4 3 j. 1j1 2j1	3 + j.4 3j4 3 + j.1j1 2j1

]pai=:o.1	lp1	o.2	2p1	2*pai	lp2	*: pai
3.14159	3.14159	6.28319	6.28319	6.28319	9.8696	9.8696
1 o.o.0.5 1	sin(90°)と		2 o.o.0.5 1	cos(90°)と		
1 1.22461e_16	sin(180°)の値		6.12303e_17 1	coss(180°)の値		
“p.” は 左で与えた 係数の 多項式の値を 出力する			c=:1 5 4 [t=:_1 0 2 c p. t 0 1 27	1 5 4&p. t 0 1 27		
“p:” は 右引数で与えた 番号の 素数の値を 出力する			p: 0 1 2 3 4 5 2 3 5 7 11 13			
“q:” は 右引数の整数の 素因数分解			q:6 2 3	q: 24 2 2 2 3		
“r” は 分数表示			0.75=3r4 1	15 %x: 20 3r4		
“r.” は 単位複素数の生成]z=:r. 2 _0.416147j0.909297 2 r. 2 0.832294j1.81859	z 1 +: z 0.832294j1.81859		
「u T.n」 は “u” という関数を			^T.30 i.3		^ i.3	

n項までテイラー展開した近似値	1 2.71828 7.38906	1 2.71828 7.38906
「(u t.)y。」は “u” という関数の テイラー展開での y. の係数	1&o. t. i.5 0 1 0 _0.166667 0	2&o. t. i.5 1 0 _0.5 0 0.0416667
「u t: y。」は「(!y.)*u t. y。」と同じ	1&o. t: i.8	2&o. t: i.8
「T。」は接続詞で「t。」と「t:。」は副詞	0 1 0 _1 0 1 0 _1	1 0 _1 0 1 0 _1 0
「1x1 = 2.71828」はオイラーの定数	2x3 40.1711	2*1x1^3 40.1711