

APL e programmazione ad array

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

https://ap29600.github.io/corso_apl_poul.pdf

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

*APL is a mistake, carried through to perfection.
It is the language of the future for the programming
techniques of the past:
it creates a new generation of coding bums.
– Edsger W. Dijkstra*

Bingo!

	A	P	L	
I know all sorts of languages but I give up now	Dijkstra knew unicode was a mistake	how come I've never heard of it	how will I know what it does if it isn't a word?	need a Ph.D. in linear algebra to use it
unusable outside a small set of problems	hieroglyphics	RtL evaluation is purposefully obscure	looks like BF	I don't think anyone understands APL
I used APL when I was new to computers, my code was bad	I'm not smart enough to remember all these symbols	FREE SPACE: write-only	line noise	wow a language for code golfers
have you heard of MATLAB	I'm not buying a keyboard for this	let me explain why regex is bad	I rewrote this with words, see, now it can be understood!	it's only for showing off
ergonomics isn't about character count	just define the functions in a better language	$1 \times 2 + 3$ is wrong	why would you ever learn a new writing system for this?	there's a reason why APL is a dead language

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma point-free

Game of Life again

Glossario

Storia

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

- ▶ 1957 - Kenneth E. Iverson: notazione per la descrizione di algoritmi numerici
- ▶ 1962 - A Programming Language
- ▶ 1964 - APL linguaggio di specifica per IBM SYSTEM/360
- ▶ 1965-66 - Prime implementazioni
- ▶ 1966-05 - IP Sharp Associates
- ▶ 1976 - Fondata Dyadic Systems (ora Dyalog)
- ▶ 1979 - Ken Iverson riceve Turing Award
- ▶ 1984 - IBM rilascia APL2
- ▶ 1989 - Prima stesura di un interprete J
- ▶ 1993 - Kx systems rilascia K

[1]: APL Chronology

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Giusto un paio di versioni

IVSYS
APL\360
APL\1130
APL*PLUS
* SHARP APL
APL\700
4004*APL
Xerox APL
EMPL
TIS APL
APLUM
* NARS
VSAPL
APL.68000

* Dyalog APL
APLX
Rationalized APL
* APL2
SAX
A+
I-APL
* J
* K
APL2000
SIGAPL
NARS2000
APEX
GNU APL

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Linguaggi influenzati da APL

APL e
programmazione
ad array

FP

programmazione point-free

Common Lisp

la funzione `reduce`

Wolfram Language, R,
MATLAB, Julia...

array multidimensionali,
mapping implicito degli
operatori

C++

`std::iota`, alcune funzioni
nel modulo `algorithm`

Python

alcune funzioni di libreria di
`itertools`, `numpy`

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Pro e contro di APL oggi

APL e
programmazione
ad array

Pro

- ▶ feedback loop rapido
- ▶ funzioni primitive versatili
- ▶ trasformazione algebrica di programmi
- ▶ programmazione point-free
- ▶ sviluppo di algoritmi paralleli
- ▶ meme

Con

- ▶ unusable outside a small set of problems*
- ▶ looks like BF*
- ▶ scarso supporto per analisi statica
- ▶ prestazioni C-like: spesso non le ha.
- ▶ parallelismo su GPU: supporto sperimentale

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Prime impressioni

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

One-liners e Code Golf

Game of Life

```
{>1 w v.^ 3 4 = +/+/- 1 0 1 °.θ 1 0 1 °.φ <w}  
{ { +./ (1 ,: y) *. 3 4 =/ +/+/ ...  
  _1 0 1 (|. "2"0 _) _1 0 1 (|. "1"0 _) y } }
```

famosa l'implementazione di John Scholes (2009), ma già negli anni '70 era citato come esempio dell'espressività di APL

L'espressività non è golf:

nel programma sono facilmente riconoscibili gli elementi del problema e come ciascuno viene risolto

```
g=function(f,c,g,d,e,b,h){g=[];e=[c+1,c,c-1,1];  
for(b=c*c;b--;g[b]=3==d||f[b]&&2==d,d=0)  
for(h in e)d+=f[b+e[h]]+f[b-e[h]];return g}
```

questo invece è golf.

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Linguaggio J

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

- Questa presentazione è pensata per essere seguita con una REPL aperta!

<https://jsoftware.github.io/j-playground/bin/html2>

- Per esplorare la documentazione partite da qui:

<https://jsoftware.com/NuVoc>

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Primo ordine

Una *parte del discorso* è una delle seguenti:

- ▶ Sequenza numerica separata da spazi bianchi.
- ▶ Stringa di caratteri circondata da ' '
- ▶ Simbolo o identificatore seguito da *inflessioni* (. o :).

Sostantivi

0 'a' a (numero, carattere, variabile)

0 1 2 'hello' a. a: (vari array)

Verbi

+ - * % (plus, minus, times, divide)

i. # |: (integers, tally, transpose)

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Commento

NB . ignora fino alla fine della riga

Copula

a =: 10 assegnamento a variabile globale

a =. 20 assegnamento a variabile locale

Esempi

i verbi possono essere applicati in modo prefisso

```
      - 2      NB. `-'` è un verbo  
_2      NB. `_'` è parte di un numero
```

o infisso

```
      3 - 1  
2
```

tutte le applicazioni di verbo hanno la stessa precedenza e associano a destra

```
      % 5 - 2 * 4  
_0.333333  
      % (5 - (2 * 4))  
_0.333333
```

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Primo ordine

i verbi accettano array come argomenti

```
1 + 3 4 5
4 5 6
10 20 30 + 4 5 6
14 25 36
```

e generano array come risultati

```
a =. i. 12
a
0 1 2 3 4 5 6 7 8 9 10 11
5 6 7 { a
5 6 7
```

in questo senso in J tutto è un array.

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Modello degli array

Modello degli array

Un array è una collezione omogenea di valori descritta dalle seguenti proprietà

a	Tipo datatype a	Forma \$ a	Rango # \$ a	Atomi , a
1.0	floating	' '	0	1.0
1 2 3 4	integer	4	1	1 2 3 4
1.5 0.5 0.5 1.5	boxed	' '	0	1.5 0.5 0.5 1.5
abcd efgh ijkl mnop qrst vwxx	literal	2 3 4	3	abcdefghijkl...

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Rango di verbi

Verbi scalari (rango 0)

+ * ^
- % ^.
<. >. |
< = >

tutti i verbi aritmetici sono scalari: agiscono su atomi dei loro argomenti mappando implicitamente sulla struttura

Verbi strutturali (rango >0)

{. |. |:
/: \: {
, # \$

I verbi di rango maggiore interpretano i loro argomenti come collezioni di valori del rango specificato.

[Storia](#)

[Prime impressioni](#)

[One-liners e Code Golf](#)

[Linguaggio J](#)

[Primo ordine](#)

[Modello degli array](#)

[Secondo ordine](#)

[Ordine 2.5](#)

[Definire verbi](#)

[Strutture di controllo](#)

[Leggere un programma point-free](#)

[Game of Life again](#)

[Glossario](#)

Regole per l'applicazione

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

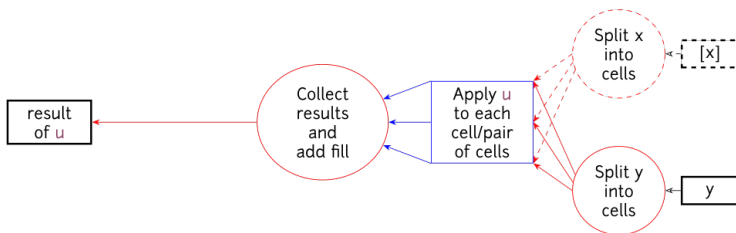
Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario



- ▶ `x` è opzionale
- ▶ `x` e `y` sono separati in celle se eccedono il rango di `u`
- ▶ `u` è applicato tra coppie di celle corrispondenti

Regole per l'applicazione

- ▶ `%.` ha rango 2
 - ▶ `x` ha rango 3: è visto
come array di rango 1 di
array di rango 2
 - ▶ `%.` applicato a ciascun
elemento
 - ▶ i risultati sono
collezionati in un array
di rango 3.
- | | | |
|--|-----------------|--------------------------------|
| | | <code>x</code> |
| | <code>1</code> | <code>0</code> |
| | <code>0</code> | <code>1</code> |
| | <code>0</code> | <code>_1</code> |
| | <code>1</code> | <code>0</code> |
| | | <code>%.</code> <code>x</code> |
| | <code>1</code> | <code>0</code> |
| | <code>0</code> | <code>1</code> |
| | <code>0</code> | <code>1</code> |
| | <code>_1</code> | <code>0</code> |

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Regole per l'applicazione

- ▶ `+` ha rango sinistro 0, rango destro 0.
- ▶ `x` ha rango 2: è visto come array di rango 2 di array di rango 0.
- ▶ `y` ha rango 1: è visto come array di rango 1 di array di rango 0.
- ▶ `3 2` è compatibile con `3` (`3` è prefisso di `3 2`).
- ▶ ogni elemento di `y` corrisponde a una riga di `x`

```
      x
0 1
2 3
4 5

      y
10 20 30
$ x
3 2
$ y
3

      x + y
10 11
22 23
34 35
```

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Leggere un programma con soli verbi

```
nums =. 29 41 68 83 79 46 60 19 73 19  
target =. 128
```

```
compl =. target - nums  
hasCompl =. (#nums) > nums i. compl  
result =. hasCompl # nums ,. compl
```

Legenda

y Tally: conta gli elementi di y
x - y Minus
x # y Copy: elementi in y in corrispondenza di 1 in x
x i. y Index-of: indice in x di elementi di y
x ,. y Stitch: affianca x e y come colonne

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Prestazioni di un programma con soli verbi

La complessità di quasi tutti i verbi primitivi è funzione della dimensione degli argomenti: annotiamo la dimensione dei valori intermedi (con $n = |\text{nums}|$)

$$\begin{array}{c} \overbrace{\text{target}}^1 - \overbrace{\text{nums}}^n \\[1em] \overbrace{(\# \text{nums})}^1 > \overbrace{\underbrace{\text{nums}}_n \text{ i. } \underbrace{\text{compl}}_n}^n \end{array}$$

complessità totale: $T = O(n \log n)$, $S = O(n)$

Sapendo che

$$T(x \text{ i. } y) = O((|x| + |y|) \log |x|)$$

$$S(x \text{ i. } y) = O(|x| + |y|)$$

$$TS(x \text{ v } y) = O(|x| + |y|) \text{ se } v \in \{- \# , . >\}$$

$$TS(\# x) = O(1)$$

[Storia](#)
[Prime impressioni](#)
[One-liners e Code Golf](#)
[Linguaggio J](#)
[Primo ordine](#)
[Modello degli array](#)
[Secondo ordine](#)
[Ordine 2.5](#)
[Definire verbi](#)
[Strutture di controllo](#)
[Leggere un programma
point-free](#)
[Game of Life again](#)
[Glossario](#)

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Secondo ordine

Avverbi

/ \ ~ (insert, prefix, reflex)

se v è un verbo o sostantivo e a è un avverbio,
 $v\ a$ è un verbo.

Congiunzioni

@ & & . " (atop, compose, dual, rank)

se u e v sono verbi o sostantivi e c è una congiunzione,
 $u\ c\ v$ è un verbo.

Secondo ordine

l'applicazione di avverbio o congiunzione ha precedenza maggiore dell'applicazione di verbo

10 +/\ 1 2 3 4 NB. 1 + 2 + 3 + 4

25 3 +&* : 4 NB. (* : 3) + * : 4

tutte le applicazioni di avverbi e congiunzioni hanno la stessa precedenza e associano a sinistra

1 3 6 10 +/\ 1 2 3 4

1 3 6 10 (+/)\ 1 2 3 4

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

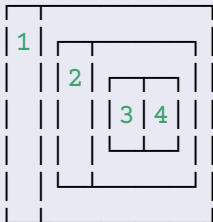
Glossario

Insert e Prefix

u/ Insert

applica u tra celle
dell'argomento

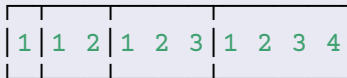
,&</ 1 2 3 4



u\ Prefix

applica u a ogni prefisso non
vuoto

<\ 1 2 3 4



	+	-	*	<.	+%
/	sommatoria	sommatoria alternata	produttoria	minimo	frazione continua
/\	sommatoria cumulativa	sommatoria alternata cum.	produttoria cumulativa	minimo cumula- tivo	convergenti frazione continua

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

La congiunzione Rango

`u"m` è un verbo analogo a `u`, ma avente rango dettato da `m`.

```
x =. i. 3 2
x
0 1
2 3
4 5
x + 10 20
|length error, executing dyad +
|shapes 3 2 and 2 do not conform
|  x      +10 20
  x +"(1) 10 20
10 21
12 23
14 25
```

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Ordine 2.5

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Because explicit definition [...] was relatively complex, it was not implemented until some months had elapsed.

Therefore, for some months all verbs were tacit.

– Roger Hui

Programmazione point-free

[...] to write $f + g$ as in calculus, one writes $f + g$ in J.

– Roger Hui

Hooks

se f e g sono verbi, $f \ g$ è un verbo e

$(f \ g) \ y \leftrightarrow y \ f \ (g \ y)$

$x \ (f \ g) \ y \leftrightarrow x \ f \ (g \ y)$

Trains

se f e g sono verbi, $[: f \ g]$ è un verbo e

$([: f \ g) \ y \leftrightarrow f \ (g \ y)$

$x \ ([: f \ g) \ y \leftrightarrow f \ (x \ g \ y)$

Forks

se f , g e h sono verbi, $f \ g \ h$ è un verbo e

$(f \ g \ h) \ y \leftrightarrow (f \ y) \ g \ (h \ y)$

$x \ (f \ g \ h) \ y \leftrightarrow (x \ f \ y) \ g \ (x \ h \ y)$

Programmazione point-free

APL e
programmazione
ad array

```
a =. 7 8 9 10
```

```
+/ a
```

34

```
# a
```

4

```
NB. f =. +/
```

```
NB. g =. %
```

```
NB. h =. #
```

```
NB. (f y) g (h y)
```

```
(+/ a) % (# a)
```

8.5

```
(+/ % #) a
```

8.5

```
mean =. +/ % #
```

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Definire verbi

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Definire verbi

arcaico

```
mean =. 3 : 0  
    (+/ y) % (# y)  
)
```

vecchio

```
mean =. monad define  
    (+/ y) % (# y)  
)
```

diretto

```
mean =. {{ (+/ y) % (# y) }}
```

point-free

```
mean =. +/ % #
```

[Storia](#)

[Prime impressioni](#)

[One-liners e Code Golf](#)

[Linguaggio J](#)

[Primo ordine](#)

[Modello degli array](#)

[Secondo ordine](#)

[Ordine 2.5](#)

[Definire verbi](#)

[Strutture di controllo](#)

[Leggere un programma
point-free](#)

[Game of Life again](#)

[Glossario](#)

Strutture di controllo

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Strutture di controllo

In altri linguaggi ricevono molta attenzione. In J è possibile scrivere programmi che non usino alcuna struttura di controllo.

for./while.

```
for. 10 do.
```

```
    NB. ...
```

```
end.
```

```
for_i. 1 2 3 do.
```

```
    NB. i readonly
```

```
end.
```

```
while. cond do.
```

```
    NB. ...
```

```
end.
```

if./else.

```
if. cond1 do.
```

```
    NB. ...
```

```
elseif. cond2 do.
```

```
    NB. ...
```

```
end.
```

altre...

```
select. case.
```

```
label_x. goto_x.
```

```
try. catch. return.
```

```
break. continue.
```


Sostituire strutture di controllo

for.

- ▶ ripetere n volte ($f \wedge n$)
- ▶ iterare strutture ($f \ " \ n$)
- ▶ accumulare valori ($f/$)

while.

- ▶ ripetere fino a convergenza ($f \wedge _$)

if.

- ▶ eseguire condizionalmente ($f \wedge \text{cond}$)

select.

- ▶ dispatch dinamico ($f \backslash g \ @. \text{discr}$)

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

**Leggere un programma
point-free**

Game of Life again

Glossario

Leggere un programma point-free

Leggere un programma point-free

```
text =. fread 'input.txt'
(#~ CR&= *: 1 |. LF&=) text
```

suddivisione even-odd

```
(Reflex) #~ ┌ (Hook)
             │ * : ───┐ (Fork)
             └──┬──┐
CR&= (Bond)    ┌──┤ |. ───┐ (Fork)
                │ 1    LF&= (Bond)
```

Legenda

$x \ f \sim \ y$ Reflex: $y \ f \ x$

$x \&f \ y$ Bond: $x \ f \ y$

$x \# \ y$ Copy

$x \ *: \ y$ Nand

$x \ |. \ y$ Rotate: rotate y to the left by x positions

[Storia](#)

[Prime impressioni](#)

One-liners e Code Golf

[Linguaggio J](#)

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

[Glossario](#)

Aiuto dall'interprete

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

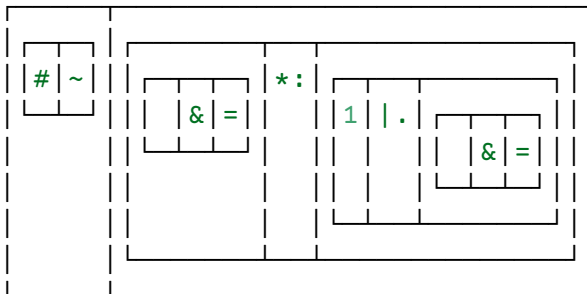
Strutture di controllo

**Leggere un programma
point-free**

Game of Life again

Glossario

```
win2unix =. #~ CR&= *: 1 |. LF&=
5!:2 < 'win2unix'
```



Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Game of Life again

Game of Life again

Una cella è viva al turno $n + 1$ se al turno n :

- ▶ ha 3 vicine vive, oppure
- ▶ è viva con 4 vicine vive

(una cella conta come vicina di se stessa)

```
input =. _6 _6 {. 3 3 $ 1 1 1 1 0 0 0 1 0
input
```

```
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 1 1 1
0 0 0 1 0 0
0 0 0 0 1 0
```

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Generare l'intorno (orizzontale)

```
hneighbors =. _1 0 1 (|. "1"0 _) input  
<"2 hneighbors
```

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	1	1	1	0	0	1	1
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0

Generare l'intorno (verticale)

APL e programmazione ad array

Ordine 2.5

Definire verbi

Leggere un programma
point-free

Game of Life again

```
vneighbors =. _1 0 1 (|. "2"0 _) hneighbors
<"2 vneighbors
```

0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0

Contare l'intorno

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

```
neighbors =. +/+/ vneighbors  
neighbors
```

```
0 0 0 1 1 1  
0 0 0 0 0 0  
1 0 1 2 3 2  
1 0 2 3 4 2  
1 0 2 4 5 3  
0 0 1 2 2 1
```

Confronti

APL e
programmazione
ad array

Primo ordine

Modello degli array

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

```
cmp =. 3 4 =/ neighbors
<"2 cmp
```

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Valutare la proposizione

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

disgiunzione (+./) di congiunzioni (*.)

```
output =. +./ (1 ,: input) *. cmp
```

output

```
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 1 0
0 0 0 1 1 0
0 0 0 1 0 1
0 0 0 0 0 0
```

Conclusione

```
input =. _6 _6 {. 3 3 $ 1 1 1 1 0 0 0 1 0
hneighbors =. _1 0 1 (|. "1"0 _) input
vneighbors =. _1 0 1 (|. "2"0 _) hneighbors
neighbors =. +/+ vneighbors
cmp =. 3 4 =/ neighbors
output =. +./ (1 ,: input) *. cmp
```

Programmazione concatenativa

il sostantivo a cui si assegna il risultato della riga sopra è usato a destra della riga sotto.

dato che in J i verbi associano a destra, concatenare le righe elide la variabile.

```
+./ (1 ,: input) *. 3 4 =/ +/+ ...
_1 0 1 (|. "2"0 _) _1 0 1 (|. "1"0 _) input
```

[Storia](#)

[Prime impressioni](#)

[One-liners e Code Golf](#)

[Linguaggio J](#)

[Primo ordine](#)

[Modello degli array](#)

[Secondo ordine](#)

[Ordine 2.5](#)

[Definire verbi](#)

[Strutture di controllo](#)

[Leggere un programma
point-free](#)

[Game of Life again](#)

[Glossario](#)

Glossario

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

Glossario

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

mat

```
0 1 2
3 4 5
6 7 8
```

\$ Shape/Reshape

```
$ 1
```

```
$ 1 2 0
```

```
3
```

```
$ mat
```

```
3 3
```

```
2 2 $ 1 2 0
```

```
1 2
```

```
0 1
```

+ - * % Aritmetica

```
1 2 0 + 10 * mat
```

```
1 11 21
```

```
32 42 52
```

```
60 70 80
```

Tally

```
# 1
```

```
1
```

```
# 1 2 0
```

```
3
```

```
# mat
```

```
3
```

Glossario

APL e
programmazione
ad array

{ From

```
1 2 0 { 'apl'  
pla  
1 2 0 { mat  
3 4 5  
6 7 8  
0 1 2
```

|: Transpose

```
|: mat  
0 3 6  
1 4 7  
2 5 8
```

i. Integers/Indices

```
i. 5  
0 1 2 3 4  
'apl' i. 'plan'  
1 2 0 3
```

< Box

```
< 'apl'
```

```
[apl]
```

> Open

```
> 1 2 3 ; 4 5 6  
1 2 3  
4 5 6
```

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario

@ Atop

```
(i. @ #) 'hello'  
0 1 2 3 4
```

& Bond/Compose

```
(- & 1) 1 2 3  
0 1 2  
3 (+ & *: ) 4  
25
```

&. Dual

```
3 (+ &. *: ) 4  
5  
(*: &. {.) 3 4 5  
9 4 5
```

" Rank

```
(< " 1) mat
```

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

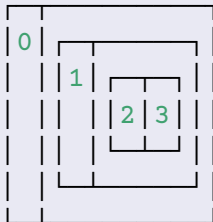
Game of Life again

Glossario

Glossario

/ Insert

(+ /) 4 5 6
15
(, < /) i. 4



\ Prefix

(+ / \) 4 5 6
4 9 15
(< \) i. 4



~ Reflex

(, < ~) 3



1 (, < ~) 2



Storia

Prime impressioni

One-liners e Code Golf

Linguaggio J

Primo ordine

Modello degli array

Secondo ordine

Ordine 2.5

Definire verbi

Strutture di controllo

Leggere un programma
point-free

Game of Life again

Glossario