Learn APL

DYALOG

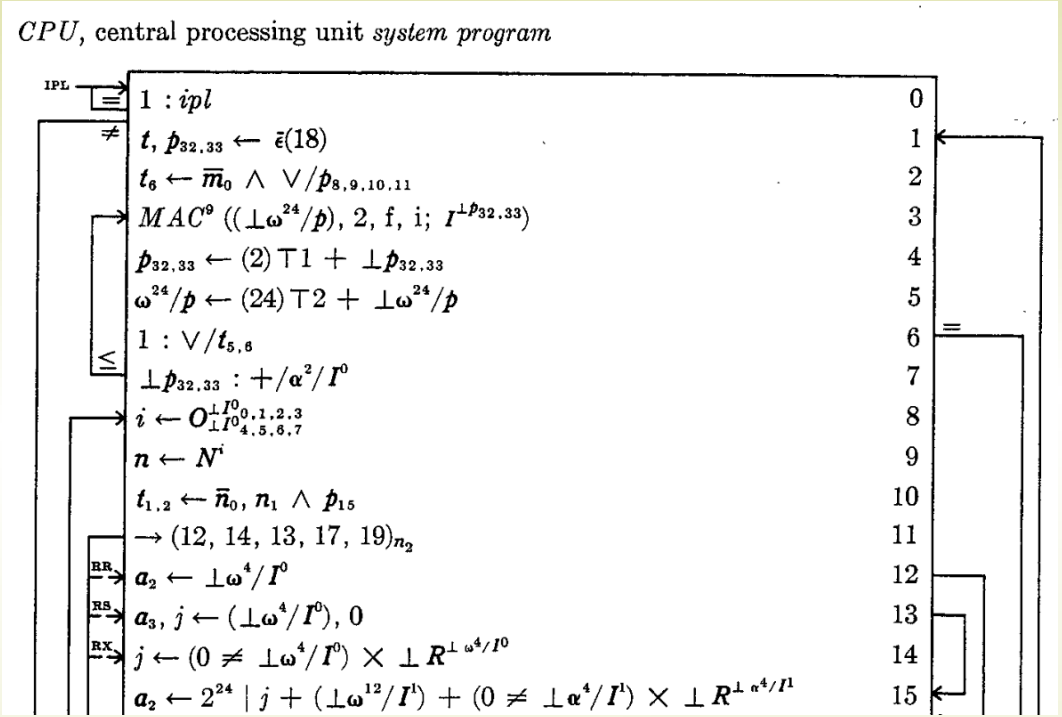
APL Seeds 2021

Please indicate your current APL knowledge:

Year	Event
1962	A Programming Language describes Ken Iverson's notation
1966	APL\360 completed and used within IBM
1968	APL\1130 released as an IBM Type-III Library[W]
1970	APL*PLUS is offered by IPSA and STSC as part of a time-sharing service
1971	"A Generalization of APL", Jim Brown's Ph.D. thesis, describes a nested APL which later becomes the basis of APL2
1973	APL.SV introduces shared variables
1976	VS APL is IBM's first APL to interact with the host system
1976 (approximate)	SHARP APL splits from APL*PLUS
1981	NARS is the first nested APL
1981	SHARP APL adds boxes to the flat array model
1983	Rationalized APL, a paper on Iverson's latest thinking on APL, includes the Rank operator added to SHARP APL later that year
1983	Dyalog APL, based on NARS and the to-be-released APL2, is released
1984	APL2 is IBM's take on nested array programming
1985	A is Arthur Whitney's APL, the first to be based on leading axis theory
1987	A Dictionary of APL describes a precursor to J
1988	A+ adds a GUI[W] to A and replaces it
1990	J is Ken Iverson's new array language, with ASCII[W] characters and based on leading axis theory
1992	K ("K0"), Arthur Whitney's list-based language, first becomes available
1997	Dyalog APL adds dfns
2006	NARS2000, a successor to NARS, is released
2009	VisualAPL is the first .NET-based APL, but soon stops development
2010	APL#, a .NET-based language, is released by Dyalog Ltd., to be abandoned in 2012
2013	GNU APL is the first fully scriptable[W] APL

Year	Event
1962	A Programming Language describes Ken Iverson's notation
1966	APL\360 completed and used within IBM
1968	APL\1130 released as an IBM Type-III Library[W]
1970	APL*PLUS is offered by IPSA and STSC as part of a time-sharing service
1971	"A Generalization of APL", Jim Brown's Ph.D. thesis, describes a nested APL which
1973	APL.SV introduces shared variables
1976	VS APL is IBM's first APL to interact with the host system
1976 (approximate)	SHARP APL splits from APL*PLUS
1981	NARS is the first nested APL

CPU, central processing unit *system program*



A notation made executable

$$u \overset{+}{\underset{\times}{}} v \quad \longleftrightarrow \quad u+.\times v$$

Beautiful Squiggles

Unicode

U+2356
U+2357
U+2358
U+2359
U+235A
U+235B
U+235C
U+235D
U+235E
U+235F
U+2360
U+2361
U+2362
U+2363
U+2364
U+2365
U+2366
U+2367
U+2368
U+2369
U+236A
U+236B
U+236C
U+236D
U+236E
U+236F
U+2370
U+2371
U+2372
U+2373
U+2374
U+2375

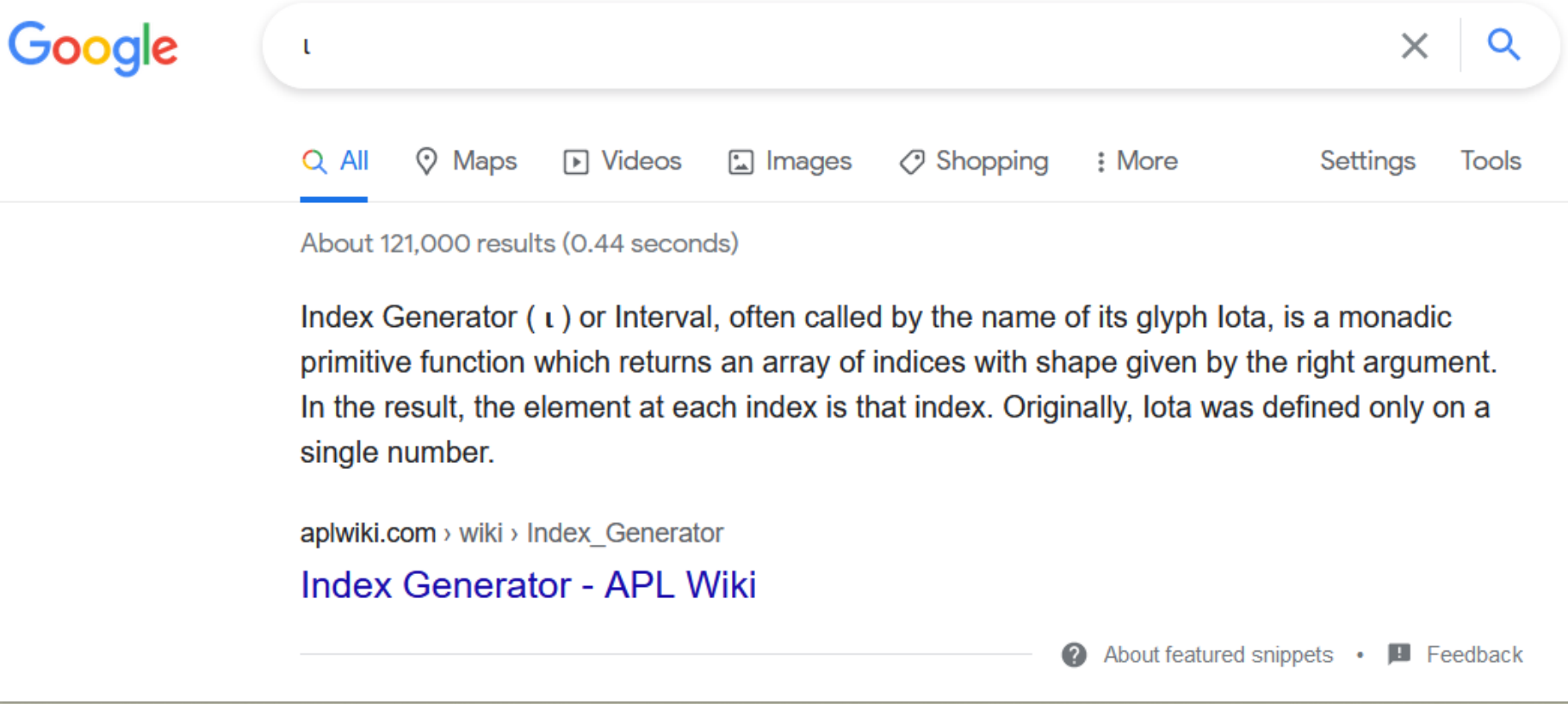
Dialect	Documentation	Download	Try It Online	Interactive
APLX[1]	APLX Archive	APLX Archive		
Dyalog APL	Centre	website	#apl-dyalog	TryAPL · Gitpod · Jupyter Notebook: Binder, Gitpod
dzaima/APL	docs	GitHub	#apl-dzaima	Android app
GNU APL	info manual	gnu.org		try-GNU-APL
NARS2000	wiki	website		
ngn/apl[2]		GitHub	#apl-ngn	Scripted · Session · REPL

Other full-featured implementations are only available for a fee.

Dialect	Documentation	Website
APL2[3][4]	Literature	IBM Marketplace
APL2000		website

Learning Resources

APL Wiki


Google
⍳
All
Maps
Videos
Images
Shopping
More
Settings
Tools
About 121,000 results (0.44 seconds)
Index Generator (⍳) or Interval, often called by the name of its glyph Iota, is a monadic primitive function which returns an array of indices with shape given by the right argument. In the result, the element at each index is that index. Originally, Iota was defined only on a single number.
aplwiki.com › wiki › Index_Generator
Index Generator - APL Wiki
About featured snippets
Feedback

Year	Location	Title	Date
1969	S.U.N.Y Binghamton, New York, USA	The Conference on APL "The March On Armonk"	July 11–12, 1969
1970	Goddard Space Flight Center, Greenbelt, MD, USA	APL II	June 19–20, 1970
1971	Berkeley, CA, USA	APL III	April 20–21, 1971
1972	Atlanta, Georgia	APL IV	June 15–16, 1972
1973	Toronto, Ontario	APL V	May 15–18, 1973
1974	Anaheim, CA, USA	APL 6	May 14–17, 1974
1975	Pisa, Italy	APL75	June 11–13, 1975
1976	Ottawa, Canada	APL76	September 22–24, 1976
1978	Foothills College, Los Altos Hills, CA, USA	"APL78" (unofficial)	March 29, 1978
1979	Rochester, NY, USA	APL '79	May 30–June 1, 1979
1980	Leeuwenhorst, Noordwijk, Holland	APL '80	June 24–26, 1980
1981	San Francisco, CA, USA	APL '81	October 21–23, 1981
1982	Heidelberg, Germany	APL '82	September 1982
1983	Washington, D.C	APL '83	April 10–13, 1983
1984	Helsinki, Finland	APL '84	June 11–15, 1984
1985	Seattle, WA, USA	APL and the future	May 12–16, 1985
1986	Manchester, UK	APL in Action	July 7–11, 1986

```
     ∇ Z←P PANDRED V
[1]    Z←(V≤P)/P ◇ Z←(Z/1⌽Z)∧P/V
     ∇

     ∇ Z←P PORRED V
[1]    Z←(V∨P)/P ◇ Z←(Z/1⌽Z)≤P/V
     ∇

     ∇ Z←P PEQRED V
[1]    Z←'=' N⍀(1⌽P)/=⍀V
     ∇

     ∇ Z←P PNERED V
[1]    Z←'≠' N⍀(1⌽P)/≠⍀V
     ∇

     ∇ Z←P PLTRED V
[1]    Z←(P≥V=1⌽P)/P
[2]    Z←(Z/1⌽Z)∧P/V=1⌽P
     ∇

     ∇ Z←P PMAXRED V
[1]    Z←V[⍒⍒V)[P/⍋(+⍀P)[⍒V]]]
     ∇

     ∇ Z←P PMINRED V
[1]    Z←V[⍒⍋V)[P/⍋(+⍀P)[⍋V]]]
     ∇

     ∇ Z←P PGRADEUP V
[1]    Z←⎕IO+(⍋V)[⍋(+⍀P)[⍋V]]-⌈⍀P×⍳⍴P
     ∇

     ∇ Z←P PPLRED V
[1]    Z←'-' N⍀(1⌽P)/+⍀V
     ∇

     ∇ Z←P PPLREDB V
[1]    Z←((V∨P)/P),1
[2]    Z←(1+'-' N⍀ Z/⍳⍴Z)-~P/V
     ∇
```

```
     ∇ Z←P PANDSCAN V
[1]    Z←~≠⍀(V≤P)⍀'≠' N⍀~(V≤P)/V
     ∇

     ∇ Z←P PORSCAN V
[1]    Z←≠⍀(V∨P)⍀'≠' N⍀(V∨P)/V
     ∇

     ∇ Z←P PEQSCAN V
[1]    Z←=⍀V≠P⍀'≠' N⍀~P/=⍀¯1↓1,V
     ∇

     ∇ Z←P PNESCAN V
[1]    Z←≠⍀V≠P⍀'≠' N⍀ P/≠⍀¯1↓0,V
     ∇

     ∇ Z←P PLTSCAN V
[1]    Z←(V∧P)∨(V∨P)⍀'>' N⍀(V∨P)/V
     ∇

     ∇ Z←P PMAXSCAN V
[1]    Z←⍋(⍋V)[⍋(+⍀P)[⍋V]]
[2]    Z←V[Z⍳⌈⍀Z]
     ∇

     ∇ Z←P PMINSCAN V
[1]    Z←⍋(⍒V)[⍋(+⍀P)[⍒V]]
[2]    Z←V[Z⍳⌈⍀Z]
     ∇

     ∇ Z←P PGRADEDOWN V
[1]    Z←⎕IO+(⍒V)[⍋(+⍀P)[⍒V]]-⌈⍀P×⍳⍴P
     ∇

     ∇ Z←P PPLSCAN V
[1]    Z←+⍀V-P⍀'-' N⍀ P/+⍀¯1↓0,V
     ∇

     ∇ Z←P PREVERSE V
[1]    Z←V[⊖⍒+⍀P]
     ∇
```

Learn

dyalog.com/getting-started.htm

apl.wiki/learning_resources

mastering.dyalog.com

tryapl.org

dyalog.tv

Ask questions

stackoverflow.com/questions/tagged/apl

forums.dyalog.com

The APL Orchard (apl.chat)

The APL Farm (Discord)

APL Online

BritishAPLAssociation.org

Dyalog webinar (dyalog.tv)

apl.wiki/community

Why should I start learning today?

Annual APL Problem Solving Competition

Open now

Compete for $$$

Previous winners learned APL *while* participating

Referral awards

dyalogaplcompetition.com

Today's Programme

Welcome to APL Seeds '21

Gitte Christensen and Richard Park

Will You Play APL With Me?

Rodrigo Girão Serrão

How an APL Prototype Helped Designing a Service

Martin Janiczek

Algorithms as a Tool of Thought

Conor Hoekstra

Simulation and Gaming with APL

Tomas Gustafsson