What’s this APL BUG?

APL and its organizations
for SVFIG
24 October 2026

KENNETH E. IVERSON

A Programming Language

JOHN WILEY AND SONS, INC. NEW YORK • LONDON • SYDNEY

Copyright © 1962 by John Wiley & Sons, Inc.

All Rights Reserved
This book or any part thereof
must not be reproduced in any form
without the written permission of the publisher.

FOURTH PRINTING, MAY, 1967

Library of Congress Catalog Card Number: 62-15180
Printed in the United States of America

More generally, if $\odot_1$ and $\odot_2$ are any two operators (whose domains include the relevant operands), then the *generalized matrix product* $X \overset{\odot_1}{\underset{\odot_2}{}} Y$ is defined as follows:

$$(X \overset{\odot_1}{\underset{\odot_2}{}} Y)_j^i = \odot_1/(X^i \odot {}_2Y_j), \quad \begin{cases} i = 1, 2, \ldots, \mu(X) \\ j = 1, 2, \ldots, \nu(Y) \end{cases}.$$

For example, if

$$A = \begin{pmatrix} 1 & 3 & 2 & 0 \\ 2 & 1 & 0 & 1 \\ 4 & 0 & 0 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 4 & 1 \\ 0 & 3 \\ 0 & 2 \\ 2 & 0 \end{pmatrix}$$

then $A \overset{+}{\times} B = \begin{pmatrix} 4 & 14 \\ 10 & 5 \\ 20 & 4 \end{pmatrix}$, $A \overset{\wedge}{=} B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$,

$$A \overset{\vee}{\neq} B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad \text{and} \quad (A \neq 0) \overset{+}{/} B = \begin{pmatrix} 4 & 6 \\ 6 & 4 \\ 6 & 1 \end{pmatrix}.$$

APL type ball - from Google

Picture from

Rex Swain



APL Keyboard



APL\360 arrives

)LOAD 1 CLEANSPACE

SAVED 1966-11-27 15.53.59 (GMT-7)

In APL

```
      A←1 2 4, 3 1 0, 2 0 0,[1.5]0 1 2
      A
1 3 2 0
2 1 0 1
4 0 0 2
      B←4 0 0 2 ,[1.5] 1 3 2 0
      B
4 1
0 3
0 2
2 0
      ⍝ the usual inner product
      A +.× B
4 14
10  5
20  4
      ⍝ What row equals which column?
      A ^.= B
0 1
0 0
1 0
```

In APL

```
    A←1 2 4, 3 1 0, 2 0 0,[1.5]0 1 2
    A
1 3 2 0
2 1 0 1
4 0 0 2
    B←4 0 0 2 ,[1.5] 1 3 2 0
    B
4 1
0 3
0 2
2 0
```

```
    ⍝ the usual inner product
    A +.× B
4  14
10  5
20  4
    ⍝ What row equals which column?
    A ^.= B
0 1
0 0
1 0
```

Year	Location	Title	Date
1969	Binghamton, New York, USA	The Conference on APL	
1972		Fourth International Conference on APL	
1974		Sixth International Conference on APL	
1975		Seventh International Conference on APL	
1976		Eighth International Conference on APL	
1979		APL '79	June
1981		APL '81	September
1983	Heidelberg, Germany	APL '83	September

1983		APL '83	March 1983
1984		APL '84	June 1984
1985		APL and the future	
1986	UK(?)		
1987	Dallas, TX, USA	APL in Transition	May 10-14, 1987
1988	Sydney, Australia	APL '88	February 1-5, 1988
1989	New York City, NY, USA	APL as a Tool of Thought	August 7-10, 1989
1990	Copenhagen, Denmark	APL 90: For the Future	August 13-17, 1990

1991	Palo Alto, CA, USA	APL '91	August 4-8, 1991
1992	St. Petersburg, Russia	APL '92	July 6-10, 1992
1993	Toronto, Ontario, Canada	APL '93	August 15-19, 1993
1994	Antwerp, Belgium	The Language and Its Applications	September 11-15, 1994
1995	San Antonio, Texas, USA	Applied Programming Languages	June 4-8, 1995
1996	Lancaster, UK	Designing the Future	July 29 - August 1, 1996
1997	Toronto, Ontario, Canada	Share knowledge / share success	August 17-20, 1997
1998	Rome, Italy	APL98 Conference on Array Processing Languages	August 27-31, 1998
1999	Scranton, Pennsylvania, USA	All Array Programming Languages	August 10-14, 1999

2000	Berlin, Germany	Think Arrays in a Great City	July 24-27, 2000
2001	New Haven, Connecticut, USA	An Arrays Odyssey	June 25-28, 2001
2002	Madrid, Spain	Array Processing Languages: Lore, Problems, and Applications	July 22-25, 2002
2003	San Diego, California, USA	Stretching the Mind	June 11-14, 2003
2004	Montreal, Canada		
2007	Montreal, Canada	Arrays and Objects	October 21-23, 2007
2010	Berlin, Germany	Array Processing Languages | Learn Parallel Applications	September 13th-16th, 2010
Year	Location	Title	Date

SIGPLAN Chapter on Array-Programming Languages

The array-programming languages chapter operates as a sub-group of ACM's SIGPLAN: the Special Interest Group on Programming LANguages.

What's Happening

Top Choices

Run J in a browser:
Juno on Github

Array languages on social media:
APL/J/K group on Reddit

J language tips on Twitter: #JLangTips

APL Language Bar - (Unicode rendering of) APL characters selectable from a page when you cannot type them!

Many of Ken Iverson's publications and others on J and APL

Three Principles of Coding Clarity

Upcoming APL, J, & other Functional Programming Events

APLBUG meeting - Monday, July 13th, 2026, 10:00 PDT
The Northern California APL ACM Chapter (or APL Bay Area Users' Group or APL BUG) will have its next round-table discussion at 10 a.m. Pacific Daylight Time (5 p.m. UTC) on Monday the 13th of July 2026. Full information may be found here.

NYCJUG - Tuesday, July 14th, 2026, 18:30 EDT
The New York City J Users Group will hold its usual monthly meeting on Tuesday, July 14th at 6:30 pm EDT. Please take a look at our Meetup site for further information. To see notes on this and other past meetings, take a look at our meeting notes .

BAA - Thursdays, 2026, 16:00 BST (UTC+1)
The British APL Association hosts an online open session - look here for more information.

ArrayCast - the array languages podcast
This hour long podcast has many episodes, including interviews with numerous notable people in the array-language community.

The APL Show
Inspired by Ken Iverson's paper "Notation as Tool of Thought", this podcast explores various aspects of notation in the context of the APL programming

News

KDB-X Free Download Available
KX has announced "The world's fastest analytics for time-series and vector data" with their new release of KDB-X. A full version of this powerful tool is available for personal and commercial use. Visit their site for more information

J version 9.7 Released
The latest release of J, version 9.7, has several new language features and performance improvements; it has been available from November 2025. See J9.7 Release Notes. J is available for the following platforms: Windows, macOS, Linux, Android, Raspberry Pi, the web and in the cloud.

Sessions for Dyalog '26 posted
The text of presentations from the Dyalog 2026 conference held April 27th in New York are here.

Array Cast
This series of pod-casts discusses array programming languages, usually with a different guest from the array-languages community. The hour-long episodes are listed here.

APL2000 announces APL64.
The production release of APL64 is was made available on

Chapter Name, Location, Year Founded

- 6 Dec 1982 - First meeting of the APL Bay Area Users' Group (APL BUG)
  10 Jan 1983 - First issue of the newsletter.
- 1988 - Affiliated with the ACM SIGAPL as the Northern California SIGAPL.
- 1989 - Joint meeting with SF Bay Area ACM
  Distinguished lecturer Donald McIntyre.
- 1991 - APL91, the SIGAPL annual conference, was at Stanford.
- 2020 - ACM administrative shift from chapter of SIGAPL to a
  "regular professional chapter".

Contact Information, Social Media Handles, and Chapter Leaders

Chair:	Chuck Kennedy	charles-kennedy@comcast.net
Vice Chair:	Charles Schulz	caschulz@earthlink.net
Sec./Treas	Curtis A. Jones	jonesca@ieee.org

Not an officer, but contributor to planning: Paul Jackson plj541@gmail.com

Meeting notices are posted at http://www.sigapl.org/

Past Events

- 12 Sep. 2022 - Round-Table Discussion
- 14 Nov. 2022 - Round-Table Discussion (Namespaces)
- 9 Jan. 2023 - Round-Table Discussion
- 10 Apr. 2023 - Round-Table Discussion (... APL or J in browser…)
- 19 Jun. 2023 - Data I/O and Officer Election (Richard Smith of Dyalog)

Future Goals

Continue providing means for APL and J users to support each other.

Inform people of news in APL and J world.

Webinar Schedule 2026

Posted by John Jacob | Jun 26, 2025 | Events | 31

Each Webinar lasts around an hour and is mostly an open session inviting discussions on APL related topics. The floor is always open to anyone who would like to share something with the group. On occasion the floor may be given over to individual who has said they would like to present a particular item.

The webinars are recorded and any interesting highlights may be published subsequently on YouTube.

Calendar

Dates for webinars are published in a Google Calendar

GET INVOLVED WITH APL

The British APL Association will continue it's series of bi-weekly webinars via Zoom indefinitely. The published **Schedule** has details for up-coming events.

BAA London live meetings will restart when UK lock down measures permit and run along side the webinars.

JOIN OUR MAILING LIST

By subscribing to the British APL Association you will be among the first to find out

Ex-Fauci adviser pleads | (54,469 unread) - curtis | calendar.yahoo.com | APLBUG4SVFIG - Goog | Webinar Schedule 202
britishaplassociation.org/webinar-schedule-2026/
All Bookmarks

May 28	16:00-17:00 BST, 15:00-16:00 UTC	Open Session
Jun 11	16:00-17:00 BST, 15:00-16:00 UTC	Possible AGM
Jun 25	16:00-17:00 BST, 15:00-16:00 UTC	Possible AGM
Jul 9	16:00-17:00 BST, 15:00-16:00 UTC	Open Session
Jul 23	16:00-17:00 BST, 15:00-16:00 UTC	Open Session
Aug 6	16:00-17:00 BST, 15:00-16:00 UTC	Open Session
Aug 20	16:00-17:00 BST, 15:00-16:00 UTC	Open Session
Sep 3	16:00-17:00 BST, 15:00-16:00 UTC	Open Session
Sep 17	16:00-17:00 BST, 15:00-16:00 UTC	Open session
Oct 1	16:00-17:00 BST, 15:00-16:00 UTC	Open Session
Oct 15	16:00-17:00 BST, 15:00-16:00 UTC	Open Session
Oct 29	16:00-17:00 BST, 15:00-16:00 UTC	Open Session
Nov 12	16:00-17:00 GMT, 16:00-17:00 UTC	Open Session
Nov 26	16:00-17:00 GMT, 16:00-17:00 UTC	Open session
Dec 10	16:00-17:00 GMT, 16:00-17:00 UTC	Open Session

Currently.com - AT&T Ya | (54,491 unread) - curtis_ | calendar.yahoo.com | J Dynamic Functional Pr

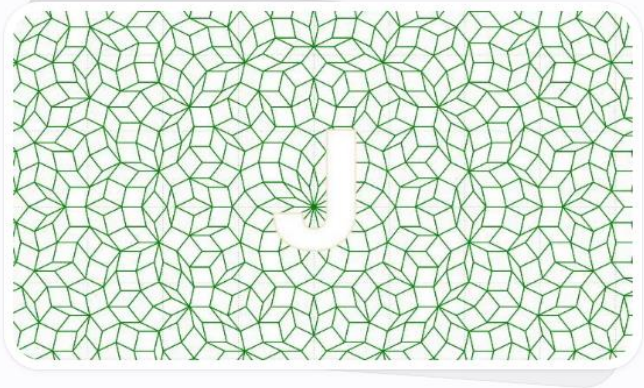
meetup.com/j-dynamic-functional-programming/

All Bookmarks

meetup

Search events... Mountain View, CA

English Log in Sign up



J Dynamic Functional Programming

4.8 • 126 ratings

New York, NY, USA

335 members

Share

About Events Members Photos Discussions

Join this group

About us

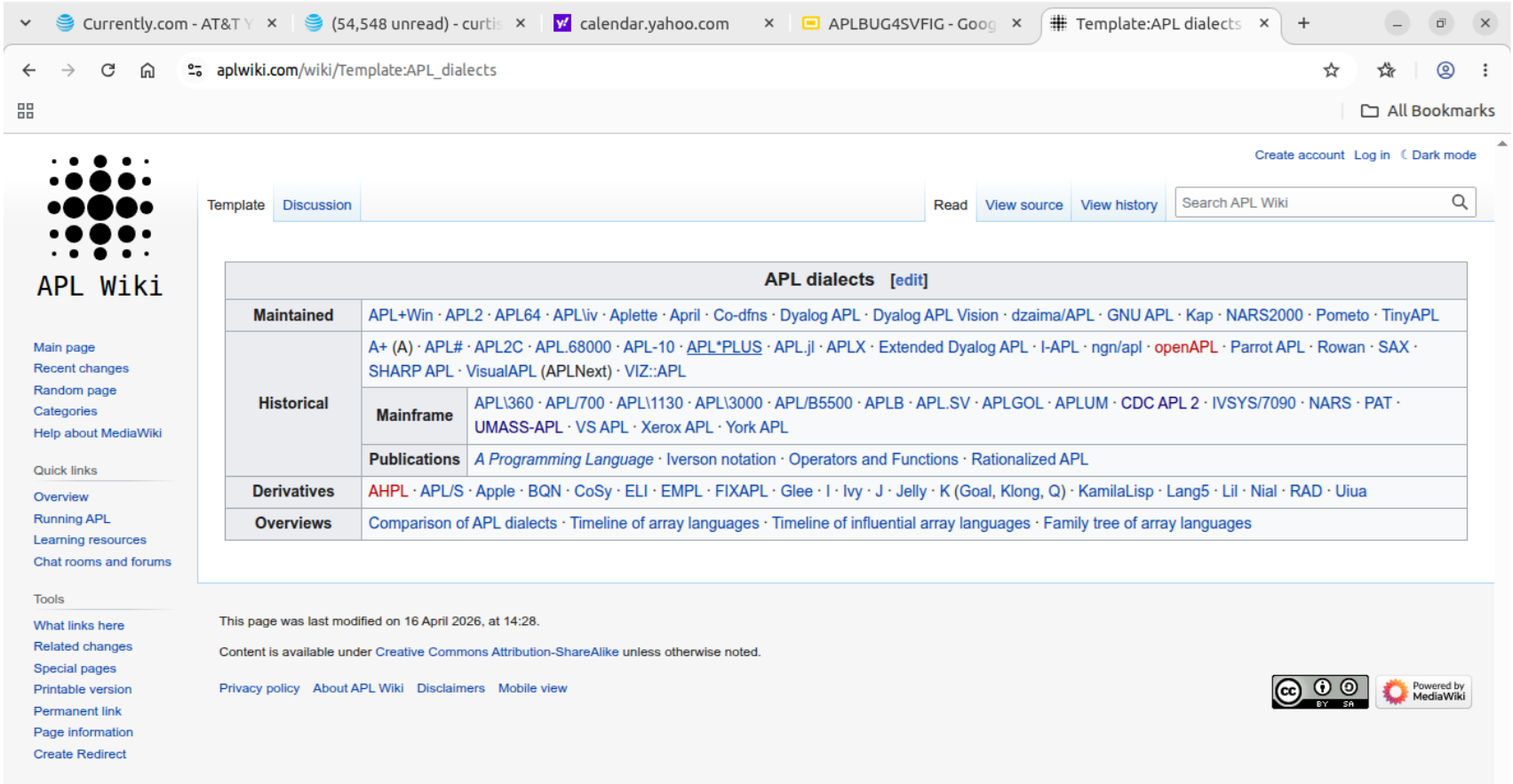
The J programming language is unlike any other because it is a computational notation. That is, it has the form of a mathematical notation while incorporating

Organizers

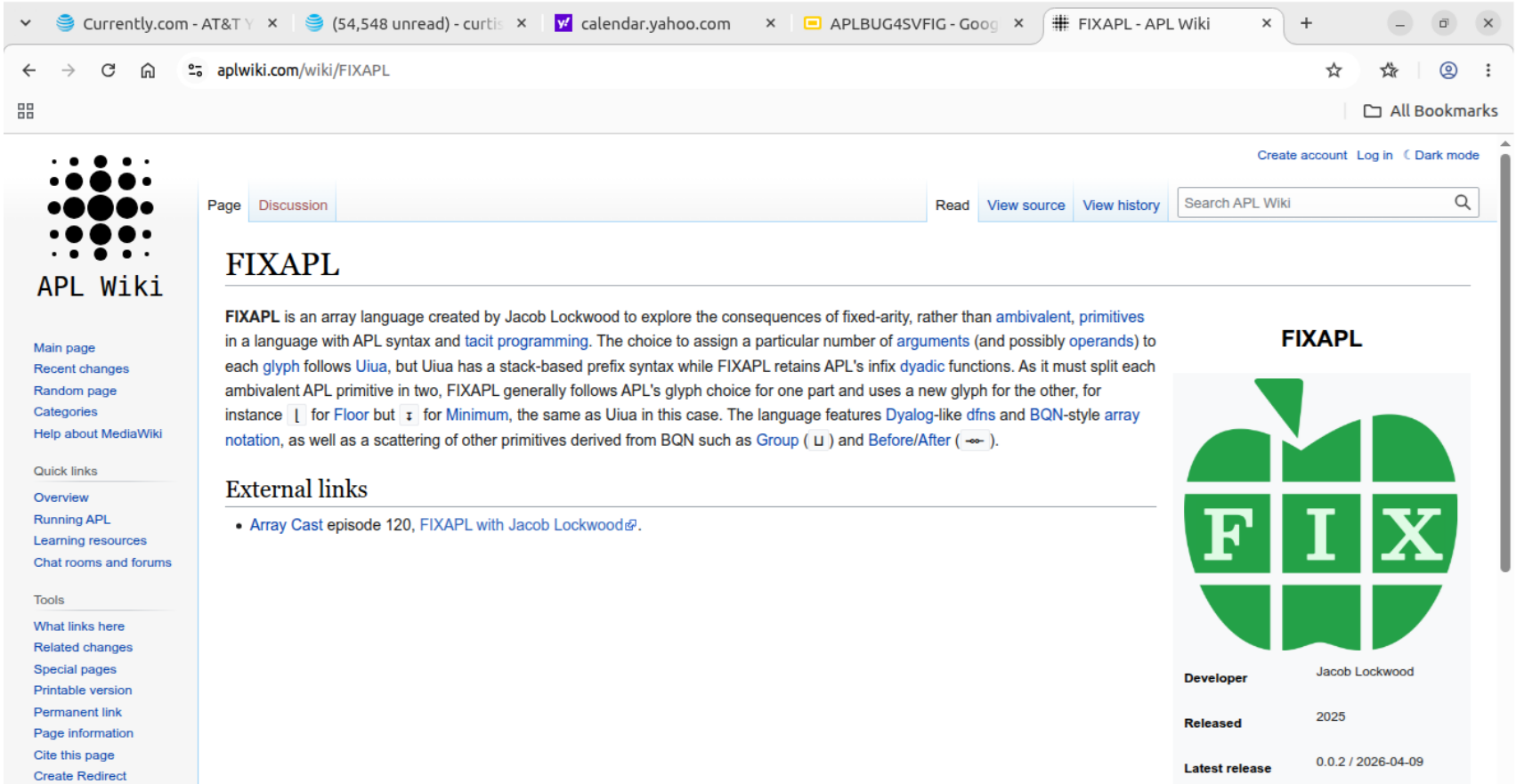
Devon McCormick

Message

APL Dialects



A dialect: FIXAPL



Currently.com - AT&T Y | (54,548 unread) - curtis | calendar.yahoo.com | APLBUG4SVFIG - Goog | AP8L

lexaloffle.com/bbs/?pid=177016

All Bookmarks

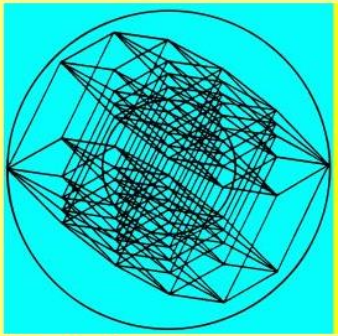
PICO-8

PICO-8 FAQ Resources Schools Submit Forum Carts Log In

AP8L

RubenVerg • 2025-11-06 02:35* • BBS > PICO-8 > Work in Progress





CoSy

Bob Armstrong Vita

Coherent Systems 1980 | CoSy.com since 19960526

#Blog #BobA PlanetaryPhysics

email to be added to CoSy NewsLetter list

αlphas . Simplicity . #CoSyAnyDay . NL202503 - polish . Tekydome . Daily Blog

Learn about *CoSy*: Saturday Morning Zoom (complementary to SV-FIG FORTH Sat)

FIRST OF ALL

>>> DON'T FORGET <<<

CoSy **Your TimeStamped Diary**

spanning openly from the Math to the ` Chip

One language (timeless as Math) for the Everyday Business of Life

A TRUE LANGUAGE

A NOTEBOOK in language you can make your own , word by word

(rather than writing ` *programs*)

Simplicity : The Rules

Currently.com | (54,493 unre | calendar.yah | GNU APL | APLBUG4SVF | New Tab | ngn apl - Goo

gnu.org/software/apl/

All Bookmarks

GNU APL



Rho, rho, rho of X
Always equals 1
Rho is dimension, rho rho rank.
APL is fun!

Richard M. Stallman, 1969

GNU APL is a free interpreter for the programming language APL.

The APL interpreter is an (almost) complete implementation of ***ISO standard 13751*** aka. ***Programming Language APL, Extended.***

The APL interpreter has implemented:

- nested arrays and related functions
- complex numbers, and
- a shared variable interface

In addition, **GNU APL** can be scripted. For example, this GNU APL home page was produced by a CGI script written in APL (see **APL demo**).

GNU APL was written and is being maintained by Jürgen Sauermann.

Downloading and Installing GNU APL

GNU APL should be available on every GNU mirror (in directory **apl**) and at ftp://ftp.gnu.org.

Paul Jackson’s NGN APL in a browser

From https:org:

Run APL in a browser and use APL in web-page development

Paul L. Jackson's browser-based APL can be installed locally and used to design, display, and test web pages from the APL development environment.//sigapl.

https://plj541.github.io/APL.js/

NARS2000: An Experimental APL Interpreter

On this page:

- Introduction
- Language Features
- Datatypes
- Miscellaneous Features
- Platforms
- Documentation
- What It Isn't
- Get The Code
- If You Want To Help
- Discussion Forum
- Fonts
- Changes
- Author

Welcome to NARS2000

Introduction

The first NARS (Nested Arrays Research System) was designed and implemented in the early 1980s as a testbed for new ideas in APL, principally with nested arrays. A quarter of a century later, it's time to try again.

The goal of this effort, as with the last one, is to foster new ideas about the language and its implementation.

Language ideas include new functions, operators, and datatypes. To aid in getting your ideas into an implementation, there will be a mechanism for telling NARS that a user-defined APL function represents a new primitive function, operator, or datatype. In order to experiment with your ideas for new extensions, all you need to know is how to write APL programs.

Implementation ideas include algorithms to enhance code space (smaller executable), time space (faster execution times), or expressibility (clearer and more understandable code). A goal of the existing implementation is to be open and easy for others to change so that new ideas may be inserted smoothly.

Moreover, this project is made available as Free Software so, among other things, any developer may pick it up and add new features, and any end user may use it free of charge.

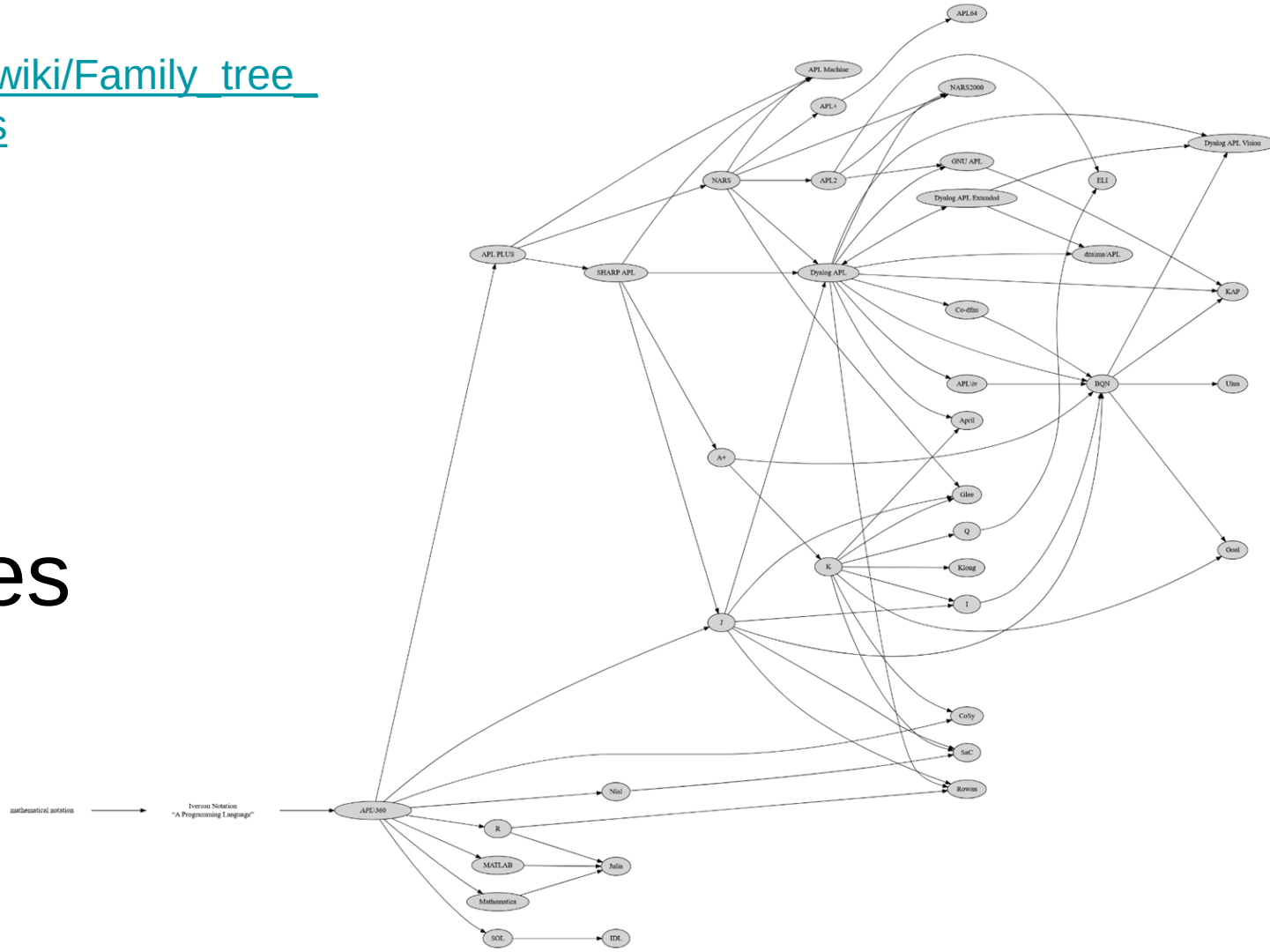
Language Features

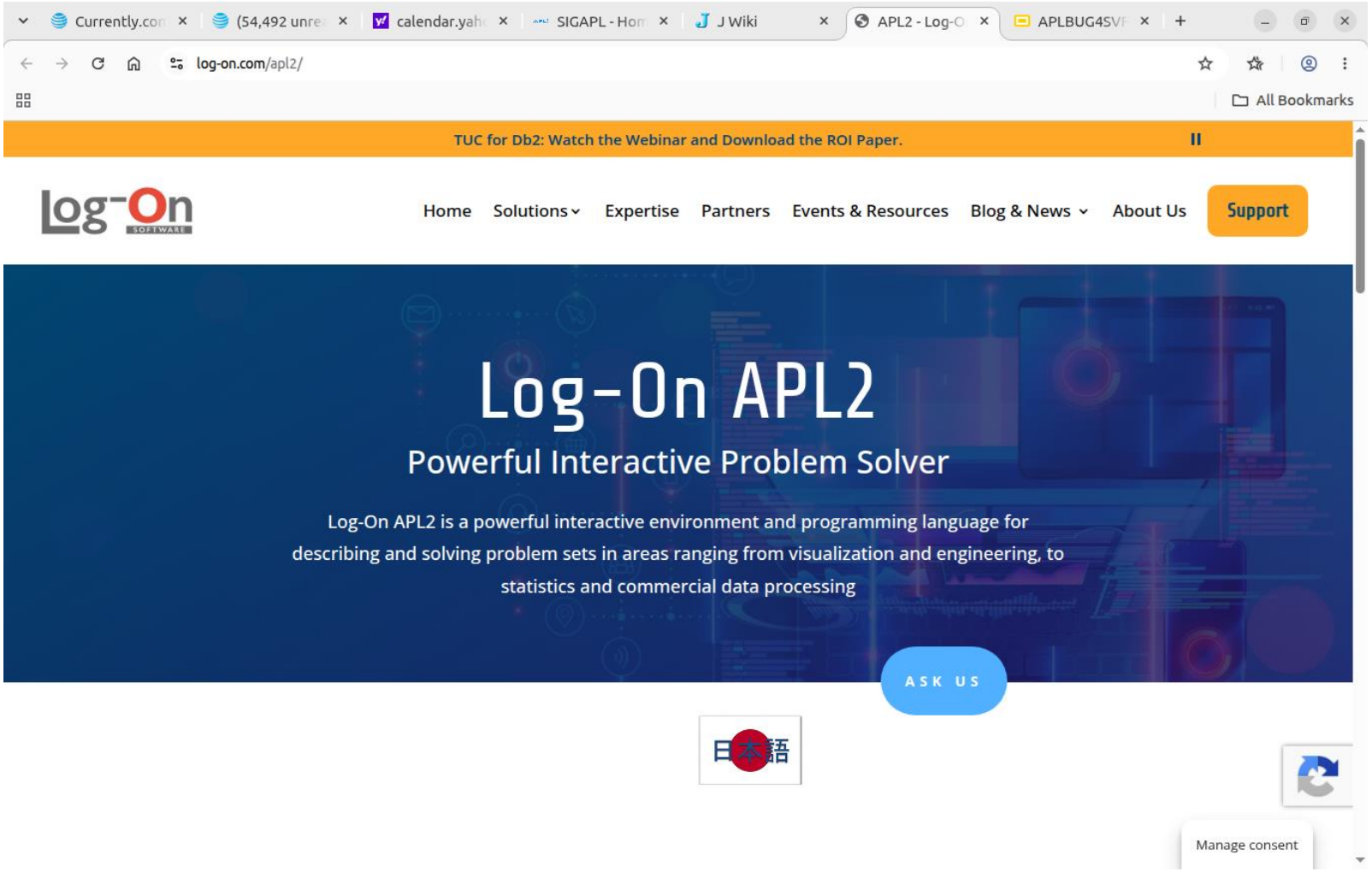
Current Language Features

- ISO/IEC 13751 Extended APL Compliant
- Consistent Extensions to ISO/IEC 13751
  - Strand Assignment (e.g., **(A B)←1 2** is the same as **A←1** and **B←2**)
  - Modify Assignment (e.g., **A+←1** is the same as **A←A+1**)
  - Modify Strand Assignment (e.g., **(A B)+←1** is the same as **A←A+1** and **B←B+1**)
  - Function/operator assignment (e.g., **F←+.×** and **F←¨** and **F←∘**)
  - Sink (monadic **←A** to suppress the display of **A**)
  - Unified index reference, assignment, and modify assignment: **R[L]**, **R[L]←A**, and **R[L]*f*←A** allow both Reach and Scatter indexing — that is,
    - if **L⊃R** is valid, it is equivalent to **⊃R[⊂L]**, and
    - if **L⊃¨⊂R** is valid, it is equivalent to **R[L]**, and

https://aplwiki.com/wiki/Family_tree_of_array_languages

Family Tree of Array Languages




Currently.com
(54,492 unre
calendar.yah
SIGAPL - Hom
J Wiki
APL2 - Log-O
APLBUG4SVF
log-on.com/apl2/
All Bookmarks
TUC for Db2: Watch the Webinar and Download the ROI Paper.
Log-On SOFTWARE
Home
Solutions
Expertise
Partners
Events & Resources
Blog & News
About Us
Support
Log-On APL2
Powerful Interactive Problem Solver
Log-On APL2 is a powerful interactive environment and programming language for describing and solving problem sets in areas ranging from visualization and engineering, to statistics and commercial data processing
ASK US
日本語
Manage consent

Currently.com - A | (54,492 unread) - | calendar.yahoo.c | SIGAPL - Home | APL2000 » Home | APLBUG4SVFIG -

Not secure apl2000.com

All Bookmarks

- Home
- What is APL?
- Products
- Pricing
- Services
- Training
- Free APL Tutorial
- Software Downloads
- Forum
- APLDN Archive
- Press Releases
- Contact Us

APL2000℠
Rapid Application Development

Welcome to APL2000

Announcing the Release of APL64

It's Here!

APL2000 is excited to announce the latest production release of APL64 2026.1 on Jul 28, 2026. APL2000 is moving APL forward with APL64, our newest product, developed for the 64-bit environment and designed to simplify the transition for APL+Win users. Take advantage of the new features in this next- generation APL development platform to enhance your development experience.

APL2000 Subscriptions include APL64 and APL+Win software.

Click here for more details

There are Many Substantial Advantages to APL64.

Click here for more details

Recent History

Currently.com - A | (54,492 unread) - | calendar.yahoo.c | SIGAPL - Home | Home - Dyalog | APLBUG4SVFIG -

dyalog.com

All Bookmarks

DYALOG

The tool of thought for **software solutions**

Contact Us

Home Business Learning Community Resources News About Us



Products Downloads Support

Featured Videos

Working with Array Notation in Dyalog APL
Write and manipulate literal arrays



Dyalog delivers an APL-based development environment that allows both subject matter experts and IT specialists to efficiently convert ideas into software solutions.

Inline Tracing in Dyalog
Evaluate expressions primitive-by-primitive



Using the Behind Operator in Dyalog APL
Completing the set of function composition operators



DYALOC v20.0
Available Now

APL CHALLENGE
Enter Now!

Dyalog '26 Eastbourne
Monday 12 - Friday 16 October 2026

Currently.com - A | (54,492 unread) - | calendar.yahoo.c | SIGAPL - Home | Download Dyalo | APLBUG4SVFIG -

dyalog.com/download-zone.htm

All Bookmarks

DYALOG The tool of thought for software solutions

Contact Us

Home Business Learning Community Resources News About Us

- Products
- Dyalog Services
- Prices and Licences
- Support (DSS)
- Download Dyalog – Free
- Dyalog '26



Home >> Resources >> Download Dyalog for Free

Download Dyalog for Free

Dyalog is free for non-commercial use but it is not free software. A Basic Licence is a free licence which allows APL users to have a copy of the latest Dyalog technology for personal or non-commercial use. It can be used for experiments and proof of concept until the point in time that it is of value.

For a commercial or enterprise licence, please contact sales@dyalog.com

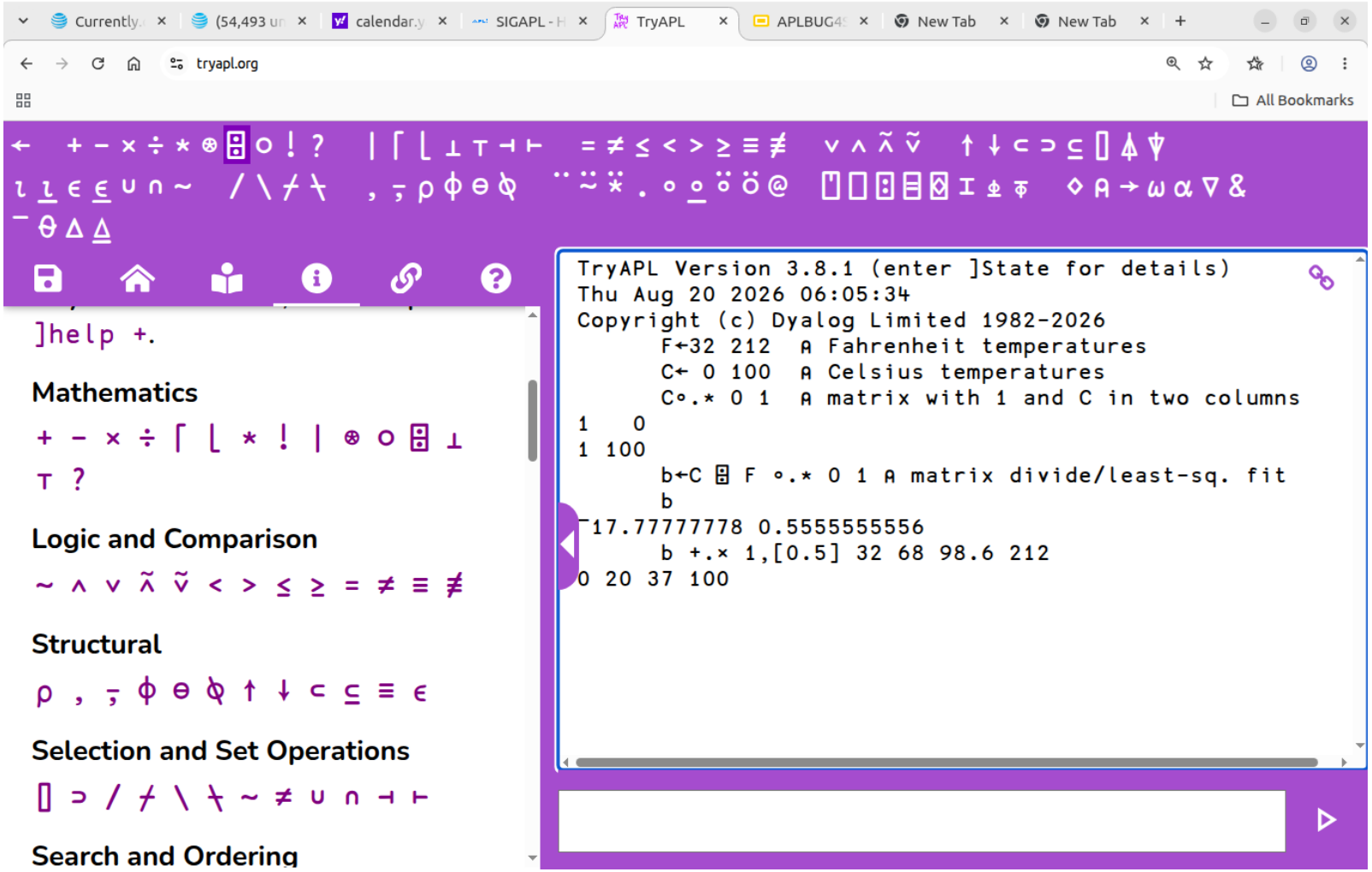
I understand this licence is subject to the conditions described in the licence agreement and in the Terms and Conditions.

- I will not make unauthorised copies of the software.
- I will not give the software to anyone else.
- I will not use the software for commercial purposes.

I accept the conditions ☐

QUICK LINKS

MyDyalog	On-line Help	Dyalog Tools	Try APL on-line	Dyalog User Meetings
Prices and Licences	On-line Manuals	dfns Home Page	Mastering Dyalog APL	Videos
Download Dyalog	Dyalog Forum	Tutorials & Samples	APL Challenge	External Publications


Currently
(54,493 ur
calendar.y
SIGAPL - H
TryAPL
APLBUG4
New Tab
New Tab
tryapl.org
All Bookmarks
← + − × ÷ * ⍟ ⌹ ○ ! ? | ⌈ ⌊ ⊥ ⊤ ⊣ ⊢ = ≠ ≤ < > ≥ ≡ ≢ ∨ ∧ ⍲ ⍱ ↑ ↓ ⊂ ⊃ ⊆ ⍷ ⍋ ⍒
⍳ ⍸ ∈ ⍷ ∪ ∩ ~ / \ ⌿ ⍀ , ⍪ ⍴ ⌽ ⊖ ⍉ ¨ ⍨ ⍣ . ∘ ⍤ ⍥ @ ⍞ ⎕ ⍠ ⌸ ⌺ ⍧ ⍎ ⍕ ◇ ⍝ → ⍵ ⍺ ∇ &
¯ ⍬ ⍙ ⍫
]help +.
Mathematics
+ − × ÷ ⌈ ⌊ * ! | ⌹ ○ ⍟ ⊥
⊤ ?
Logic and Comparison
~ ∧ ∨ ⍲ ⍱ < > ≤ ≥ = ≠ ≡ ≢
Structural
⍴ , ⍪ ⌽ ⊖ ⍉ ↑ ↓ ⊂ ⊆ ≡ ∈
Selection and Set Operations
⍸ ⊃ ⌿ ⍀ \ ⍋ ~ ≠ ∪ ∩ ⊣ ⊢
Search and Ordering
TryAPL Version 3.8.1 (enter]State for details)
Thu Aug 20 2026 06:05:34
Copyright (c) Dyalog Limited 1982-2026
F←32 212 ⍝ Fahrenheit temperatures
C← 0 100 ⍝ Celsius temperatures
C∘.* 0 1 ⍝ matrix with 1 and C in two columns
1 0
1 100
b←C ⌹ F ∘.* 0 1 ⍝ matrix divide/least-sq. fit
b
¯17.77777778 0.5555555556
b +.× 1,[0.5] 32 68 98.6 212
0 20 37 100

Thank you for watching.

Questions?