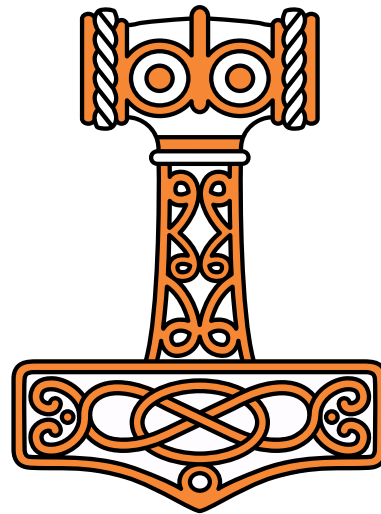


DYALOG

Olhão 2022

Growing APLers

Rich Park



dyalog.tv

APL Seeds '23

arraycast.com

YouTube:

Dyalog Ltd.

Dyalog Usermeeting

Adam's APL

RikedyP

Jeremy Howard

code_report

apl.news

apl.wiki

apl.chat

APL Farm on Discord

[/r/apljk](https://r/apljk)

stackoverflow.com

forums.dyalog.com

tutorial.dyalog.com

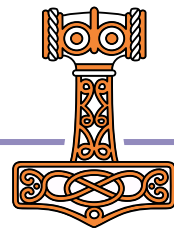
mastering.dyalog.com

course.dyalog.com

xpqz.github.io/learnapl

mooc.fi

dyalog.com/getting-started.htm



Getting Started

Getting started with any new programming language can seem like a daunting task, and the Dyalog application development platform ships with enough features that you might appreciate some guidance to help you get started. The resources on this page are free of charge and aimed at APL novices.



APL Seeds: Events aimed at those who are just starting their APL journey.
APL Seeds '23 will be held in spring 2023.

Download materials/watch recordings from: [APL Seeds '21](#) [APL Seeds '22](#)



Community

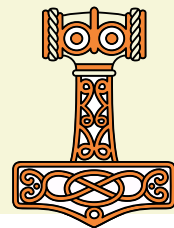
APL has a thriving and enthusiastic community of users who are very happy to answer questions:

- Chat in [the APL Orchard](#), a very active chat room
- Ask a question on [Stack Overflow](#) or the [r/aplj](#) subreddit
- Post in [the Dyalog Forums](#)
- Dyalog social media: [Twitter](#), [Facebook](#), [LinkedIn](#)

Basics

Resources to help you take your first APL steps:

- [Tips](#) is a page of "useful to know" suggestions from previous beginners.
- [Videos: Dyalog Basics](#) is a library of introductory videos.
- [Mastering Dyalog APL](#) by Bernard Legrand is a complete guide to the use of Dyalog, beginning with a thorough introduction to the APL programming language and progressing to worked examples. The book is available for purchase through [Amazon](#); a [free PDF download](#) and an [online revision](#) (currently under development) are also available.
- [TryAPL](#) offers an interactive environment that allows users to play with simple APL expressions. Its [Learn tab](#) includes tutorials in which various scenarios are explored.
- [APL Wiki](#) includes [simple examples](#) of APL in action (as well as some [more advanced ones](#)).
- [APL Cultivation](#) is a series of chat lessons that were run through the [APL Orchard](#) chat room.
- [APL Course](#) is a self-study introduction to Dyalog with exercises.
- [APL Tutor](#) is an online system that takes a complete novice through the terminology, conventions and functionality of APL (not specific to Dyalog's dialect) – it looks a little dated but is a useful introduction.





Getting Data and Code into the Workspace feat.]Get – Dyalog Webinar with Rich Park

Getting Data and Code into the Workspace feat.]Get t

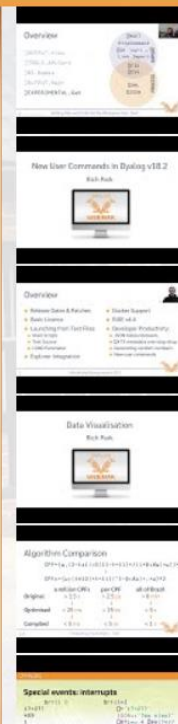
Rich Park

Watch Later

Share



MORE VIDEOS



Getting Data and Code
into the Workspace feat.
]Get

Richard Park

New User Commands in
Dyalog v18.2

Richard Park

Introducing Dyalog
version 18.2

Morten Kromberg

Data Visualisation

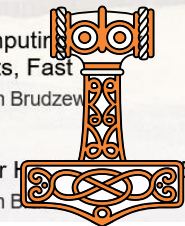
Richard Park

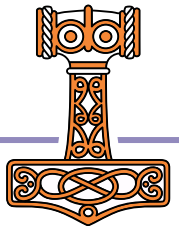
Computing
Digits, Fast

Adam Brudzewski

Error Handling

Adam Brudzewski







↳ > + / ÷ ≠



RikedyP
226 subscribers

SUBSCRIBE

HOME

VIDEOS

PLAYLISTS

CHANNELS

ABOUT



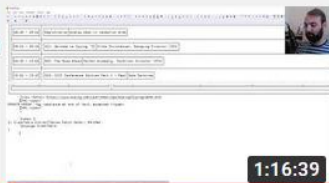
Uploads ▾

≡ SORT BY



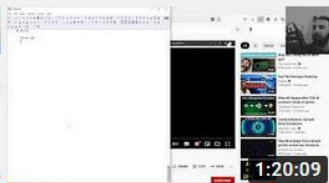
Live APL Programming:
Reading tabular data

64 views • Streamed 7 days ago



Live APL Programming:
Reading Table Data

91 views • Streamed 2 weeks ago



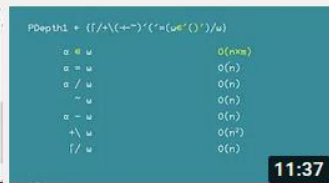
Live APL Programming:
Artificial Life

65 views • Streamed 2 weeks ago



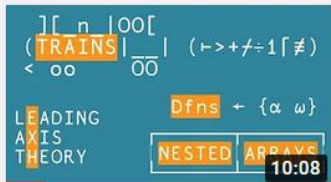
4 more APL solutions in 24
minutes!

679 views • 1 year ago



Overview: Computational
complexity in APL

2.1K views • 1 year ago



Four Features of Modern APL

3.7K views • 1 year ago



PWC: Hash-counting String in
APL



PWC: Rare Numbers in APL

40 views • 1 year ago



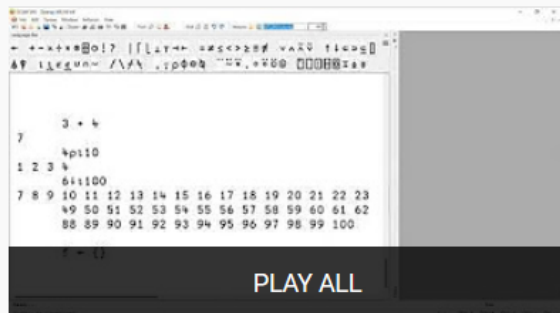
Loopless Snail Sort

72 views • 1 year ago



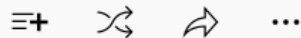
Perl Weekly Challenge i
(and BQN)





Programming Environment Basics

5 videos • 155 views • Last updated on 16 May 2022



DyalogLtd

SUBSCRIBED



1



Useful (Keyboard) Shortcuts in the IDE for Dyalog APL

DyalogLtd

2



Using the Language Bar in the Dyalog APL IDE

DyalogLtd

3



Overview of the Dyalog IDE for Microsoft Windows

DyalogLtd

4



Overview of RIDE – Remote IDE for Dyalog APL

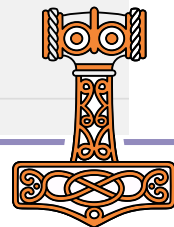
DyalogLtd

5



Editing and Debugging Dyalog APL

DyalogLtd



Format 123

How are they defined/identified?

TRADOP	DOP
Calling syntax in leader:	Presence of <code>as</code> or <code>as in body</code>
<code>f result(Name)</code> <code>case-f s</code>	<code>Name()</code> <code>as s</code>
monadic operator deriving a monadic function	

How to define operators (traditional and dops)

8:10



0:00 / 44:53 • Intr...

User-Defined Operators

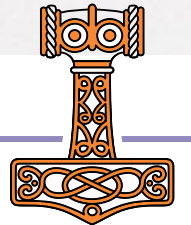


Adám Brudzewsky

APL operators (higher-order functions) offer powerful ways to apply functions to data. While Dyalog provides plenty of powerful operators, you can define your own to accomplish additional tasks. In this webinar, we show the basic syntax of user-defined operators and present a few utilities, compositions, and conceptual tools.

00:00 Introduction

01:42 What is an operator? Operator and derived function valence



Some balanced binary trees in APL

2022-10-03, *Raghu Ranganathan*

EXAMPLE

Futhark on The Array Cast

2022-10-03, *Troels Henriksen*

COMMENTARY

Threading the HTMLRenderer

2022-09-22, *Paul Mansour*

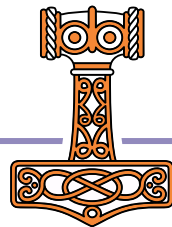
EXAMPLE

A Document Object Model in APL

2022-09-21, *Paul Mansour*

EXAMPLE ; DESIGN

apl.news





Adám's APL

150 subscribers

SUBSCRIBED



HOME

VIDEOS

PLAYLISTS

CHANNELS

ABOUT



Uploads

≡ SORT BY



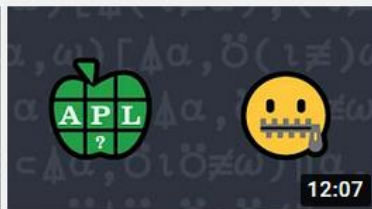
Shorter Ones to the Front
(APL Quest 2016-6)

43 views • 1 day ago



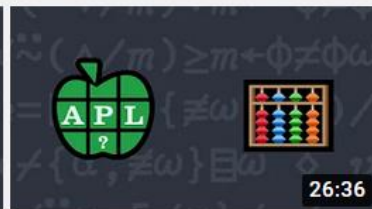
You're Unique, Just Like
Everyone Else (APL Quest...

85 views • 4 days ago



Just Meshing Around (APL
Quest 2016-4)

81 views • 10 days ago



Statistics - Mode (APL Quest
2016-3)

57 views • 10 days ago



Statistics - Median (APL
Quest 2016-2)

102 views • 3 weeks ago



Statistics - Mean (APL Quest
2016-1)

104 views • 1 month ago



Blaise'ing a Trail (APL Quest
2015-10)

95 views • 1 month ago



I'd Like Mine Scrambled
Please (APL Quest 2015-9)

74 views • 1 month ago



Hey! This is My Space (APL
Quest 2015-6) — EXTENDED

47 views • 1 month ago



Unlucky 13 (APL Quest
2015-8)

72 views • 1 month ago

The APL Orchard

general

1.2k sta

The APL Orchard



<https://apl.chat> — Learn, teach, ask, code, golf, & discuss usage.
See https://apl.wiki/APL_Orchard for access and info,
<https://aplcart.info> for simple how-to questions. See also

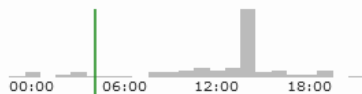


https://apl.wiki/Chat_rooms_and_forums [edit](#)

[apl](#) [array-manipulation](#) [j](#) [tips](#)

first message 2017-01-24

last message 5 hours 35 minutes ago



0

today

10

yesterday

18

per day



111

this week

154

last week

167

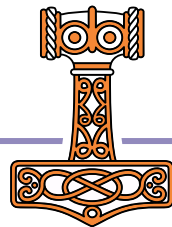
per week

[join this room](#)[view transcript](#)[search for messages containing](#)

406

225017

apl.chat



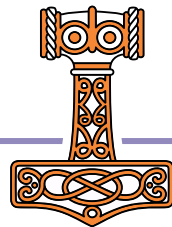
The APL Farm on



Discord

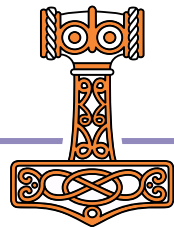
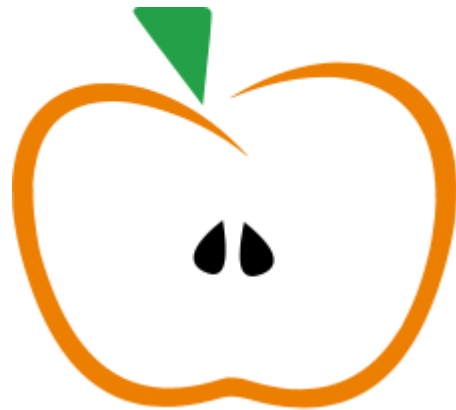
discord.gg/cYbMMw5D

reddit.com/r/apljk



APL Seeds '23

- Spring 2023
- New and prospective APL users
- What is your APL story? We'd love to hear
- aplseeds@dyalog.com



FUNCTIONAL CONF 2022

Asia's Premiere Functional Programming Conference




Beginner Talk

Why APL is a language
worth knowing



Rodrigo Girão Serrão

Consultant
Dyalog

 @mathspblog



0:00 / 48:25

• Chapters >

functionalconf.com 24-26 March - Online



Why APL is a language worth knowing by Rodrigo Girão Serrão #FnConf 2022

726 views • 31 Mar 2022



30



DISLIKE



SHARE



CLIP



SAVE



ConfEngine

12.1K subscribers

SUBSCRIBE



APL New York Meetup (In Person)



Hosted By
Conor H. and 2 others



WED, SEP 7 · 11:00 PM BST
APL New York Meetup (In Person)



Programming Languages Virtual
Meetup

Public group



Wednesday, September 7, 2022 at
6:00 PM to Wednesday, September 7,
2022 at 8:00 PM EDT

[Add to calendar](#)



WeWork Office Space & Coworking
750 Lexington Ave · New York, NY

How to find us
750 Lexington Ave, 12th Floor.
Someone will be waiting to let you in at
the 12th floor.

Share



Past event





A Look at Array Languages


APL vs BQN vs J

<https://github.com/codereport/Content>

<https://github.com/codereport/array-language-comparisons/>

Conor Hoekstra

 code_report

 codereport

0:37 / 42:43

APL vs BQN vs J vs Q vs NumPy vs Julia vs R

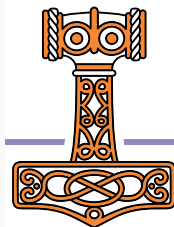
4,388 views • Premiered on 17 Sept 2022

 198  DISLIKE  SHARE  CLIP  SAVE ...



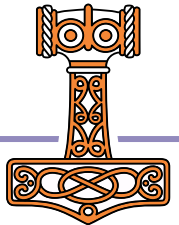
code_report
41.5K subscribers

SUBSCRIBED



FinnAPL spring meeting

APL Germany spring meeting





Thread



Jeremy Howard
@jeremyphoward



Change of plan. We switched to using [@dyalogapl](#) APL for learning sequences and series in our math lesson, and the kid's like it *so* much better than Python list comprehensions!



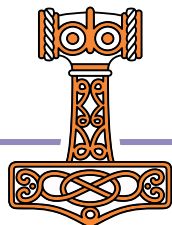
CLEAR WS - Dyalog APL/W-64

File Edit View Window Session Log Action

Language Bar



```
⍳10
1 2 3 4 5 6 7 8 9 10
2×⍳10
2 4 6 8 10 12 14 16 18 20
+/2×⍳10
110
```





[fast.ai APL Study Group](#)

fast.ai APL study session 1

5,907 views • 5 Jul 2022

145 DISLIKE SHARE CLIP SAVE ...



Jeremy Howard
66.4K subscribers

All are welcome to join our APL study group! Details here: <https://forums.fast.ai/t/apl-array-pr...>
Discuss this session here: <https://forums.fast.ai/t/apl-study-se...>

SUBSCRIBE

fast.ai APL Study Group

Jeremy Howard · 1/17



- fast.ai APL study session 12**
Jeremy Howard
- Fast.ai APL study session 13**
Jeremy Howard
- Fast.ai APL study session 14**
Jeremy Howard
- Fast.ai APL study session 15**
Jeremy Howard
- Fast.ai APL study session 16**
Jeremy Howard
- Fast.ai APL study session 17**
Jeremy Howard

All

Deep learning

Computer programming



Algorithms as a Tool of Thought // Conor Hoekstra // APL Seed...

Dyalog Usermeeting
12K views • 1 year ago




This Spiderman Stunt took 156 Takes. Can we Beat Them?

Corridor Crew
520K views • 2 days ago

All are welcome to join our APL study group! Details here: <https://forums.fast.ai/t/apl-array-pr...>
Discuss this session here: <https://forums.fast.ai/t/apl-study-se...>

SHOW MORE

16 Comments  SORT BY



Add a comment...



Adam's APL 3 months ago (edited)

17:45 — Turn off the docked windows by default through saving your session via the menu: Session > Save

15:55 — Remove the word "Language Bar": Right-click on the Language Bar > Hide Caption

32:50 — Turn boxing on permanently by saving your session, as described above

40:20 — Real backticks with the in-browser language bar: Backtick,Space



 9  REPLY



shunya shanti 3 months ago

Thank you Jeremy for your work(democratising education)

I just came across fastchan and i thank u agn for it. Made the environment setup and dependency installation much less nightmarish

 2  REPLY



Ken Chang 3 months ago

OMG, APL! I got exposed to it back in the early 70's along with BASIC. I enjoyed APL 

  REPLY



aeamaea 3 months ago

Apparently Ken Iverson used the "Iverson Notation" (later to become APL) to create the formal spec for IBM's 360 mainframe CPU.


 2  REPLY



Joe Kanaan 3 months ago

Hi Jeremy! When will the part 2 of the tutorial be released?



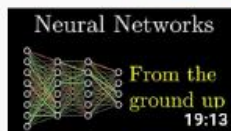
Corridor Crew  520K views • 2 days ago
New




You're Unique, Just Like Everyone Else (APL Quest...
Adám's APL
88 views • 5 days ago
New



I Cycled 750km Across Japan in a Week | Ft. @CDawgVA
Abroad in Japan  677K views • 2 days ago
New



Neural Networks
3BLUE1BROWN SERIES S3 E1
But what is a neural network? | Chapter 1, Deep learning
3Blue1Brown  12M views • 5 years ago



Mix - Jeremy Howard
More from this channel for you



fast.ai APL study session 2
Jeremy Howard
1.4K views • 3 months ago



Book Review - Deep Learning with fastai Cookbook
DigitalSreeni
2.5K views • 11 months ago



Lesson 1 - Deep Learning for Coders (2020)
Jeremy Howard
315K views • 2 years ago

[Register](#)[Mission](#)[FAQ](#)[Team](#)[Contact](#)

Paradigm Conference



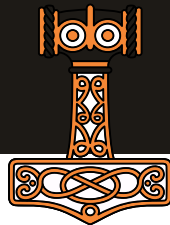
Conference has concluded! Check back in 2023!

Apply for the 2023 team here!

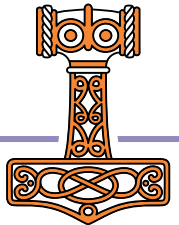
esolangconf.com

Our Mission

The high school computer science curriculum is entrenched in old, boring programming languages. Students have no idea of the diversity of programming paradigms out there: from functional, to array-based, to knowledge-based programming. Most have never even heard of pattern matching, macros, or higher-order functions! We want to change this.



Teaching APL



[About](#)[First Steps](#)[Dfns and Assignment](#)[Selecting from Lists](#)[The Outer Product](#)[Shape Reshape](#)[The Array Model](#)[Search, Sort and Select](#)[Loops and Recursion](#)[Extra Assignment](#)[Workspace Basics](#)[Namespaces and Other Objects](#)[Getting Help](#)[User-defined Functions](#)[Operators](#)[Quad Names](#)[Error Handling and Debugging](#)[Data IO](#)[Code IO](#)[External Interfaces](#)[Historical Quirks](#)[Interpreter Internals](#)[Further Reading](#)

APL Course

A self-study introduction to Dyalog APL with exercises.

This course consists of a series of short instructional notes interleaved with problem sets. The first few sessions are accompanied by video tutorials.

Status

The course is undergoing changes in structure and content. We do not currently recommend linking to specific pages or sections at the current time, as those links are likely to break in future updates.

Getting Started

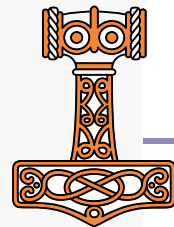
If you're just starting out, you can use the [TryAPL](#) online interpreter. Put TryAPL on one half of your screen, and [view the course](#) on the other half.

If you need help typing APL glyphs (e.g. $\times\neq$) then see [the APL Wiki](#).

Audience

This course assumes high-school / secondary level mathematics knowledge, and some familiarity with basic programming terminology (e.g. *function*, *variable*, *recursion*). It is not currently intended as a general introduction to programming, but more of a fast-track to getting

Table of contents

[Status](#)[Getting Started](#)[Audience](#)[Feedback](#)

About

First Steps

Dfns and Assignment

Selecting from Lists

The Outer Product

Shape Reshape

The Array Model

Search, Sort and Select

APL Course

A self-study introduction to Dyalog APL with exercises.

This course consists of a series of short instructional notes interleaved with problem sets. The first few sessions are accompanied by video tutorials.

Table of contents


Status

Getting Started

Audience

Feedback



@Adám If you just want some basics, check out <https://course.dyalog.com/> which isn't ready yet, but already tackles the basics. 



Jadefalke



Today at 9:15 AM

The first page was rather easy, managed to solve everything

User-defined Functions

Operators

Quad Names

Error Handling and Debugging

Data IO

Code IO

External Interfaces

Historical Quirks

Interpreter Internals

Further Reading

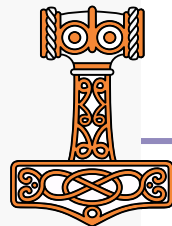
Getting Started

If you're just starting out, you can use the [TryAPL](#) online interpreter. Put TryAPL on one half of your screen, and [view the course](#) on the other half.

If you need help typing APL glyphs (e.g. $\times\neq$) then see [the APL Wiki](#).

Audience

This course assumes high-school / secondary level mathematics knowledge, and some familiarity with basic programming terminology (e.g. *function*, *variable*, *recursion*). It is not currently intended as a general introduction to programming, but more of a fast-track to getting



x (+/ xö-1)ö1 2 +-y

Learning APL

Search this book...

Introduction

It's arrays all the way down

Indexing

Glyphiary

Direct functions and operators

Iteration

The Key operator: ⌘

The At operator: ⌠

The Rank/Atop operator: ⌞

The Stencil operator: ⌘

The Over operator: ⌵

Dyadic transpose: ⌵⌵

Encode decode: ⌵⌵

Products

Trainspotting

Finding things

Partitions

Error handling

The APL Way

Namespaces ⌵⌵



Introduction

A language that doesn't affect the way you think about programming is not worth knowing.

—Alan Perlis

Who is this for?

I wrote this to be the book I would have wanted to read when I started to learn APL. An introduction to APL for an experienced practitioner from a different programming language or two. We all learn in different ways, and I prefer the fundamental concepts laid bare first, and then learn by example.

I came to APL after discovering a file of [solutions](#) to the [Advent of Code](#) 2015 challenge in [K](#), an APL derivative. That's around 100 lines of actual code, and whilst I didn't understand any of it, I kept looking at it, trying to figure out which of the 50 problems (well, 49) this was a solution to. Each of my Python solutions typically ran to 50-100 lines+ for the bulk of the problems.

Turns out it was the whole lot. That blew my mind.

What is APL?

APL is an [array language](#), and one of the oldest programming languages still in use today, next to [FORTRAN](#), [Lisp](#) and [COBOL](#). APL uses its own curious-looking symbols, like $\uparrow\phi\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow$, rather than reserved words written out in English like most other languages, like [C](#) or [Python](#). As a language, APL sits at a very high level of abstraction, making it well suited to ultra-concise formulations of algorithms.



Contents

Who is this for?

What is APL?

Why should I learn APL?

...but it's unreadable!

Don't I need a lot of mathematics?

A note on our APL subset

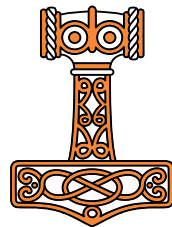
Is terser better?

Other resources

Ok, I'm convinced, how do I get started?

Our first tentative steps

Valence



1. Introduction - Will You Play APL With Me?
2. Getting Started
3. Data and Variables
4. Some Primitive Functions
5. User-Defined Functions
6. First-Aid Kit
7. Execute & Format Control
8. Working on Data Shape
9. Special Syntax
10. Nested Arrays (Continued)
11. Operators
12. Appendices



Mastering Dyalog APL

The “[Mastering Dyalog APL](#)” book is the *de facto* standard for people who are looking to learn Dyalog APL from a book. In today’s world technology changes rapidly, so a printed book about a programming language is at risk of becoming outdated.

! Attention

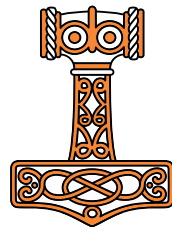
This online version is a **work in progress**. It is missing chapters and sections and the content here may undergo heavy revision.

Feel free to [open a new issue](#) on GitHub if you would like to give feedback or to suggest corrections to any eventual mistakes/typos you encounter. Alternatively, you can send an email to mdapl@dyalog.com.

The first edition of “Mastering Dyalog APL” is an excellent resource but is becoming more and more outdated as the years go by and Dyalog APL evolves. For that matter, an updated and more modern version of the book is being created out of Jupyter Notebooks (available in [this GitHub repository](#)) to provide for a more interactive learning experience for those who like to read and experiment. A [static online version](#) also exists, and a printed version will be made available for those of you who prefer to hold a paper book in their hands.

The first edition dates back to November of 2009 and was written by Bernard Legrand, with most grateful acknowledgements to the contributors:

- Kim S. Andreasen
- Daniel Baronet



Dyalog APL Tutor

Unit:

1. Immediate Execution Mode

Phase: **Tutorial Text**

Page: **9** (of 23)

[Previous](#)
[Next](#)

[Table of contents](#)

[Function index](#)
[Subject index](#)

[Test APL expressions](#)

- Dyalog home page
- Install APL font
- Install trial APL
- APL publications

Tracking...

Built and maintained by

ZARK

The Final Word in APL

"What do we do?" asked one creator.

"Let's redesign the keyboard," suggested a second.

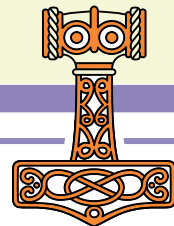
"You're out of your mind!" shouted a third. "No one will use APL if you need a special keyboard. Where would you get it?"

"Relax," said the second. "All we have to do is make some minor modifications to a regular keyboard. First, we can replace the keytops with new ones that have our symbols on them. Or, in a pinch, we can provide little stickers people can stick to the top side or front side of the keys.

"Second, we can rip out whatever mechanism displays the normal symbols, and replace it by a mechanism that displays our symbols."



Creator 2



APL TUTOR

0. Introduction

1. Immediate Execution Mode

2. Variables

3. Scalars and Vectors

4. Number Manipulation Functions

5. Order of Execution

6. Monadic Number Manipulation Functions

7. Vector Generating Functions

8. Matrices

9. Character Data

10. Shape and Rank

13. Selection

14. Reductions

15. Scans

16. Searching

17. Programming

18. User-Defined Functions

19. Workspace Management

20. Branching and Looping

21. Control Structures

22. Debugging

23. Writing Interactive Functions

26. Files

27. Shuffling Data

28. Inner and Outer Products

29. Advanced Numerical Functions

30. Workspace Storage

31. Full Screen Input

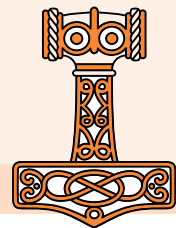
32. Readability

33. Exception Handling

34. Efficiency

35. Boolean Techniques

36. File Design



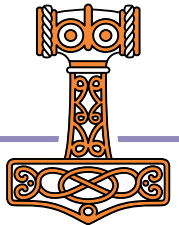
tutorial.dyalog.com

mastering.dyalog.com

course.dyalog.com

xpqz.github.io/learnapl

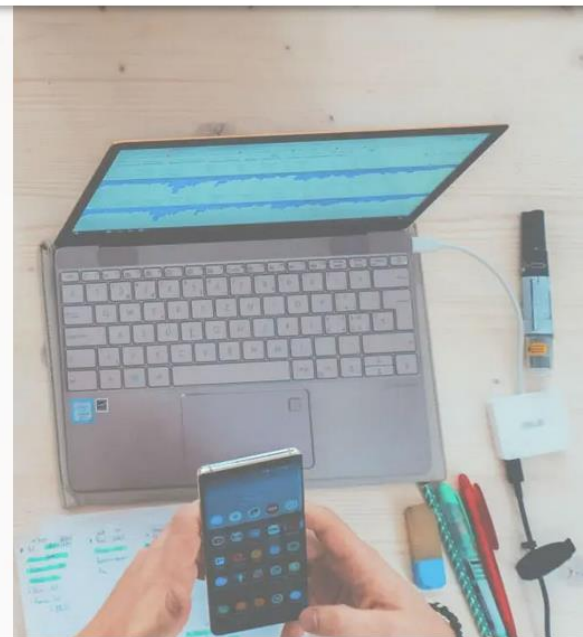
aplclass.com/book



High-quality, open and free courses for everyone!

The courses are hosted by University of Helsinki MOOC Center. No prior knowledge is required — beginners can start to learn programming basics from the Programming with Java course, or start to get familiar with artificial intelligence from the course Elements of AI.

Our courses



Help the victims of the war in Ukraine.

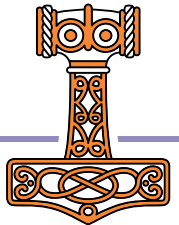
Support students and researchers affected by the invasion of Ukraine

[Donate to the Finnish Red Cross Disaster Relief Fund](#)  [University of Helsinki Ukraine appeal](#) 

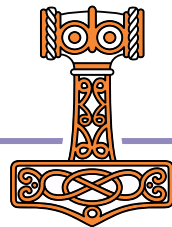


"Libraries are super useful but I am fascinated that APL has sort of them come this far without having a huge package repository which everyone uses I think it's really interesting that if when solving something in APL the first thing you do is not go search on the internet for some ready-made solution but actually looking at the problem and saying can I do this myself and usually you can"

- Alve Björk - Young APLers' Panel, Dyalog '19



Get ans fastly



Home

PUBLIC

Questions

Tags

Users

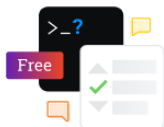
Companies

COLLECTIVES

Explore Collectives

TEAMS

Stack Overflow for Teams – Start collaborating and sharing organizational knowledge.



Create a free Team

Why Teams?

Questions tagged [apl]

Ask Question

APL (named after the book A Programming Language) is an interactive array-oriented language. It is based on a mathematical notation developed by Kenneth E. Iverson. Do not use this tag for Alexa Presentation Language; use [alexa-presentation-language] instead.

Watch tag

Ignore tag

[Learn more...](#) [Improve tag info](#) [Top users](#) [Synonyms](#)

305 questions

Newest

Active

Bountied

Unanswered

More

Filter

3 votes

2 answers

57 views

Read column of CSV file as array

I am trying to read a column from a CSV file into an array. So far I have successfully read the file with the code below. load 'csv' data =: readcsv '/Users/max/Desktop/prices.csv' Typing data in ...

csv j apl

M4X_397 asked Sep 28 at 14:42

6 votes

2 answers

122 views

What is the "j" expression for the "maximum consecutive ones" problem?

In the paper "Combinatory Logic and Combinators in Array Languages" they give a solution in APL: $vec \leftarrow 1101110001$ split (partition) on zeroes $\vec{0}vec$

j apl

madeofmistake 508 asked Sep 13 at 0:00

1 vote

1 answer

22 views

How to change Dyalog's UCMD source folders on Linux

On Windows the answer is Options>Configure>"User commands"

configuration apl dyalog

awagga 136 asked Sep 6 at 11:36

0 votes

Specifying Non-Absolute Paths to Libraries in GNU APL

The Overflow Blog

- How to earn a million reputation on Stack Overflow: be of service to others
- The right way to job hop (Ep. 495)

Featured on Meta

- Bookmarks have evolved into Saves
- Inbox improvements: marking notifications as read/unread, and a filtered...
- Reviewer overboard! Or a request to improve the onboarding guidance for new...
- Collectives Update: Recognized Members, Articles, and GitHub
- Should we burninate the [script] tag?

Hot Meta Posts

- Is someone botting API key removal suggested edits?

Custom Filters

Create a custom filter

Watched Tags

[[Moderator Control Panel](#)]

Dyalog on a Chromebook

POSTREPLY ↩

Search this topic...

Search

13 posts • [Page 1 of 2](#) • [1](#) [2](#)

DYALOG ON A CHROMEBOOK

* [EDIT](#)     [QUOTE](#)

 by [paulmansour](#) on Fri Dec 10, 2021 10:34 pm

Does Dyalog run on Chromebook? I assume these things are Linux of some sort. If so, is it simple to install? I have zero experience with Chromebooks, Linux, Ride et.

paulmansour

Posts: 398

Joined: Fri Oct 03, 2008 4:14 pm

One Click Ban: [[One Click Ban](#)]



RE: DYALOG ON A CHROMEBOOK

* [EDIT](#)     [QUOTE](#)

 by [paulmansour](#) on Mon Dec 20, 2021 3:25 pm

Isn't this a simple yes or no question? Or is it a really stupid question?

paulmansour

Posts: 398


Joined: Fri Oct 03, 2008 4:14 pm

One Click Ban: [[One Click Ban](#)]

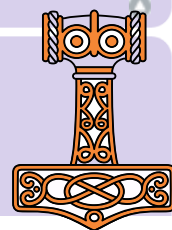


RE: DYALOG ON A CHROMEBOOK

* [EDIT](#)     [QUOTE](#)

 by [Morten|Dyalog](#) on Tue Dec 21, 2021 8:14 am

Hi Paul, sorry for the slow response. I believe the simple answer is "Yes", at least the TTY version starts and runs in the "Linux App". Geoff did the testing a while ago and I wanted to ask him about whether he got RIDE

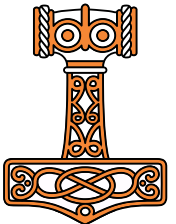


About 0 results (0.21 seconds)

Your search - **site:forums.dyalog.com chromebook** - did not match any documents.

Suggestions:

- Make sure that all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.





site:forums.dyalog.com chromebook



[All](#)

[Shopping](#)

[Images](#)

[News](#)

[Videos](#)

[More](#)

[Tools](#)

About 3 results (0.29 seconds)

<https://forums.dyalog.com> › [viewtopic](#)

Dyalog Forums • View topic - Dyalog on a Chromebook

Some **Chromebooks** run low-end Intel processors, some **Chromebooks** run Qualcomm processors, some **Chromebooks** run realtek processors.

You've visited this page 2 times. Last visit: 9/9/22

<https://forums.dyalog.com> › [viewforum](#)

Dyalog Forums • View forum - Platforms and Interfaces

Forum: Topics: Posts: Last post. Windows: GUI, COM/OLE/ActiveX. Using (or providing) components based on the "Win32" framework: 89 Topics: 408 Posts: Last ...

You've visited this page 3 times. Last visit: 8/30/22

<https://forums.dyalog.com> › [viewforum](#)

Dyalog Forums • View forum - UNIX and Linux

Dyalog on a **Chromebook** · 1, 2by paulmansour on Fri Dec 10, 2021 10:34 pm: 12 Replies: 4381 Views: Last post by RichardP|Dyalog View the latest post

Portugal

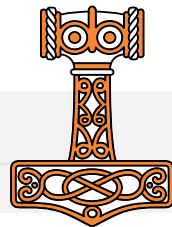
● Olhão - Based on your past activity - [Update location](#)

[Help](#)

[Send feedback](#)

[Privacy](#)

[Terms](#)



[illegible]

time	11	22	33	12	13	23	iter	esh
0.00000E+000	0.00000E+000	0.00000E+000	0.00000E+000	0.00000E+000	0.00000E+000	0.00000E+000	0	0
0.20000E-002	-0.15347E+000	-0.14970E+000	0.30317E+000	0.20687E-002	-0.65533E-002	-0.55885E-002	2	0
0.40000E-002	-0.30695E+000	-0.29939E+000	0.60634E+000	0.41375E-002	-0.13107E-001	-0.11177E-001	2	0
0.60000E-002	-0.46042E+000	-0.44909E+000	0.90951E+000	0.62062E-002	-0.19660E-001	-0.16766E-001	2	0
0.80000E-002	-0.61386E+000	-0.59875E+000	0.12126E+001	0.82699E-002	-0.26227E-001	-0.22350E-001	7	12
0.10000E-001	-0.76489E+000	-0.74601E+000	0.15109E+001	0.10293E-001	-0.32746E-001	-0.27841E-001	17	21
0.12000E-001	-0.87989E+000	-0.85736E+000	0.17373E+001	0.11782E-001	-0.38893E-001	-0.31963E-001	24	25
0.14000E-001	-0.92867E+000	-0.90298E+000	0.18316E+001	0.12292E-001	-0.43981E-001	-0.33563E-001	19	19
0.16000E-001	-0.94871E+000	-0.92026E+000	0.18690E+001	0.12384E-001	-0.48157E-001	-0.34048E-001	14	14
0.18000E-001	-0.96191E+000	-0.93095E+000	0.18929E+001	0.12361E-001	-0.51748E-001	-0.34257E-001	12	12
0.20000E-001	-0.97347E+000	-0.94018E+000	0.19136E+001	0.12329E-001	-0.54893E-001	-0.34375E-001	9	12
0.22000E-001	-0.98450E+000	-0.94904E+000	0.19335E+001	0.12289E-001	-0.57678E-001	-0.34464E-001	9	12
0.24000E-001	-0.99509E+000	-0.95771E+000	0.19528E+001	0.12215E-001	-0.60141E-001	-0.34555E-001	9	9
0.26000E-001	-0.10055E+001	-0.96630E+000	0.19718E+001	0.12133E-001	-0.62344E-001	-0.34639E-001	9	9
0.28000E-001	-0.10156E+001	-0.97478E+000	0.19904E+001	0.12044E-001	-0.64323E-001	-0.34718E-001	9	9
0.30000E-001	-0.10256E+001	-0.98316E+000	0.20087E+001	0.11947E-001	-0.66111E-001	-0.34795E-001	9	9
0.32000E-001	-0.10353E+001	-0.99143E+000	0.20268E+001	0.11842E-001	-0.67733E-001	-0.34870E-001	9	9
0.34000E-001	-0.10449E+001	-0.99957E+000	0.20445E+001	0.11729E-001	-0.69213E-001	-0.34946E-001	9	9
0.36000E-001	-0.10543E+001	-0.10076E+001	0.20619E+001	0.11609E-001	-0.70567E-001	-0.35023E-001	9	9
0.38000E-001	-0.10634E+001	-0.10155E+001	0.20789E+001	0.11474E-001	-0.71804E-001	-0.35107E-001	9	9
0.40000E-001	-0.10724E+001	-0.10233E+001	0.20957E+001	0.11334E-001	-0.72949E-001	-0.35194E-001	9	9
0.42000E-001	-0.10812E+001	-0.10309E+001	0.21122E+001	0.11187E-001	-0.74010E-001	-0.35285E-001	9	9
0.44000E-001	-0.10899E+001	-0.10385E+001	0.21284E+001	0.11033E-001	-0.74999E-001	-0.35379E-001	9	9
0.46000E-001	-0.10984E+001	-0.10459E+001	0.21444E+001	0.10873E-001	-0.75923E-001	-0.35478E-001	9	9
0.48000E-001	-0.11068E+001	-0.10533E+001	0.21601E+001	0.10706E-001	-0.76790E-001	-0.35581E-001	9	9
0.50000E-001	-0.11151E+001	-0.10605E+001	0.21755E+001	0.10533E-001	-0.77605E-001	-0.35689E-001	9	9
0.52000E-001	-0.11232E+001	-0.10675E+001	0.21907E+001	0.10352E-001	-0.78373E-001	-0.35802E-001	9	9
0.54000E-001	-0.11311E+001	-0.10745E+001	0.22056E+001	0.10165E-001	-0.79099E-001	-0.35921E-001	9	9
0.56000E-001	-0.11389E+001	-0.10814E+001	0.22203E+001	0.99713E-002	-0.79786E-001	-0.36046E-001	9	9
0.58000E-001	-0.11466E+001	-0.10881E+001	0.22347E+001	0.97708E-002	-0.80439E-001	-0.36177E-001	9	9
0.60000E-001	-0.11542E+001	-0.10947E+001	0.22489E+001	0.95636E-002	-0.81059E-001	-0.36315E-001	9	9
0.62000E-001	-0.11616E+001	-0.11013E+001	0.22629E+001	0.93498E-002	-0.81649E-001	-0.36461E-001	9	9
0.64000E-001	-0.11689E+001	-0.11077E+001	0.22766E+001	0.91294E-002	-0.82211E-001	-0.36613E-001	9	9

Tell me about: something to achieve/improve/optimize



X,Y,Z:any M,N:num I,J:int A,B:Bool C,D:char f,g,h:fn ax:axis s:scal v:vec m:mat

aplcart.info



\emptyset



$\vdash Y$



X DOP Y \vdash Z



X \vdash Y



X \vdash Y



$\neg Y$



X \neg Y



X \neg Y



+Y



+N



M+N



-N



M-N



\times N



Empty Numeric Vector



Same (I-combinator): Y



Separate right operand of dyadic operator (DOP) from its right argument (same as (X DOP Y)Z)



Right (K-combinator): Y



Church Boolean false (X if false, else Y)



Same (I-combinator): Y



Left (KI-combinator): X



Church Boolean true (X if true, else Y)



Conjugate ('Identity' if Y not complex)



Mirror complex N across x-axis



Adding N to M



Negate: 0-N



Subtracting N from M



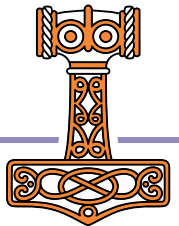
Direction ('Signum' if N is real)



```

table ← ⚡NGET'/t/proc.out'1
d←↓⊞↑{⚡{^/α:ω ⚡ ⚡SIGNAL 7}/⚡VFI'0'@('⚡ '+'')'-'@('⚡ '-'')ω}''1↓table
⚡CSV⚡'Separator' ' '⚡('^\s+' '\s+'⚡R' ' '⚡table)'S' 4

```



Tutorials

Tools and Interface Catalogue

Application Development and Deployment

External Language Interfaces and Standard Libraries

File handling and data processing

Binary files or arbitrary file types

Text Files

Tabular data

CSV

JSON

XML

OR/OS

MAP

JSON

Component files

Downloading data from the internet

SQL Interface

Features

Getting Started

Web communications

CSV

The Comma Separator Values system function `CSV` can read tabular data from `.csv` files as APL matrices, Comma separated values are a very common and convenient. While we encourage you to [read the documentation](#) for a full description, here is an overview of features of `CSV`:

- Read data from and write data to files directly

```
data ← CSV '/path/to/file.csv'
```

- Separate the header (first row) from the rest of the data

```
(data header) ← CSV '/path/to/file.csv' 0 0 1
```

- Treat specific columns of input as numeric or text, depending on the options provided.
The `4` here indicates to convert numeric values if possible, else keep the value as text.

```
numeric_if_possible ← CSV '/path/to/file.csv' 0 4
```

- Use a separator other than commas, using the "Separator" variant option, for example using tabs (`UCS 9`) for Tab Separated Values (.tsv).

```
tsv ← CSV 'Separator' (UCS 9) '-'/path/to/file.csv'
```

- Read data chunks at a time so as to not fill the workspace, using the "Records" variant option.



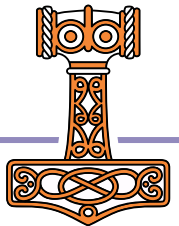
Example




```

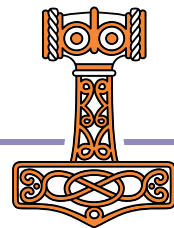
table ← ⎵NGET'/t/proc.out'1
d←↓⊞↑{⊃{^/α:ω ⋄ ⎵SIGNAL 7}/⎵VFI'0'@('° '+' )'-'@('° '-' )ω}¨1↓table
⎵CSV⎵'Separator' ' '⌊('^\s+' '\s+'⎵R' ' '⌊table)'S' 4
ReadTable '/t/proc.out'

```



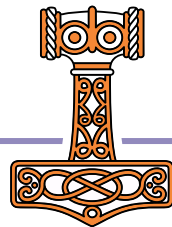
```
'LoadTEXT'∘CY'loaddata'
∘←3↑t←LoadTEXT '\tmp\proc.out'
```

time	11	22	33	12	
0.00000E+000	0.00000E+000	0.00000E+000	0.00000E+000	0.00000E+000	0
0.20000E-002	-0.15347E+000	-0.14970E+000	0.30317E+000	0.20687E-002	-0



What else do we need?

- ✧ Practical APL
- ✧ Show the rest of the world that we can do
- ✧ Tools for leveraging APL in the context of modern software



APL as a Tool of Thought

