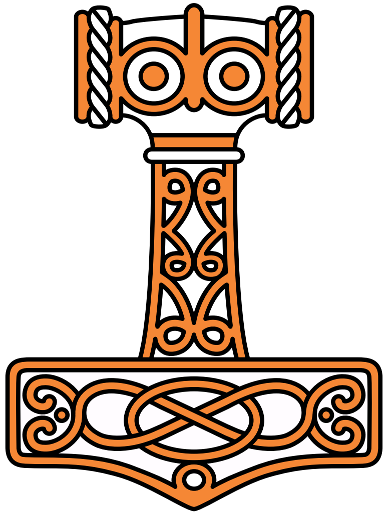
DYALOG

Olhão 2022

The Road Ahead

Morten Kromberg



Ten Lane Highway

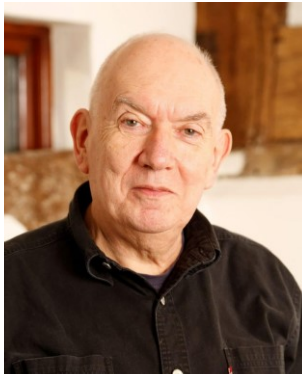
1. Building the Team
2. Training & Evangelism
3. Consulting
4. Source in Text Files
5. Service Orientation
6. Cross-Platform UI
7. [Microsoft].NET
8. New Target Platforms
9. Compiling APL
10. APL Language

Ten Lane Highway

1. Building the Team
2. Training & Evangelism
3. Consulting
4. Source in Text Files
5. Service Orientation
6. Cross-Platform UI
7. [Microsoft].NET
8. New Target Platforms
9. Compiling APL
10. APL Language

1. [Re]Building the Team

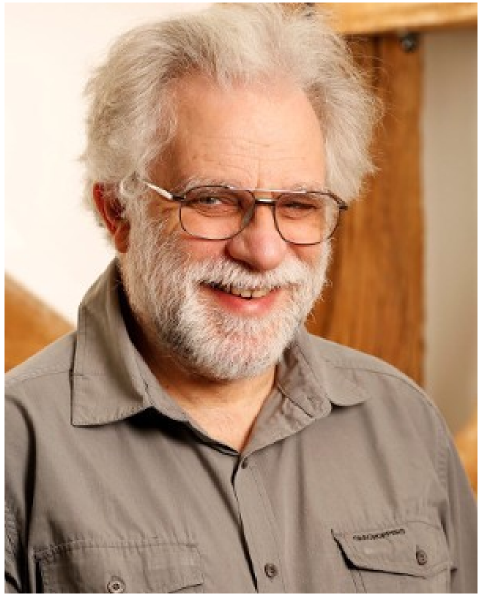
- We stand upon the shoulders of Giants


John Scholes (1948-2019)

Roger Hui (1953-2021)

Still Going Strong...

- With John Scholes, Geoff Streeter wrote Dyalog APL v1.0 in 1981-1983
- Geoff is in good health
  - Now working 3 days per week
  - And still volunteering at night...
- However, Geoff has announced that he intends to retire in April'23
- We hope to welcome him back for a retrospective talk at Dyalog'23

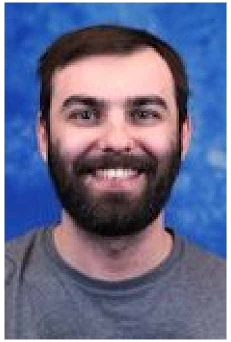


Still Going Strong...

- With John Scholes, Geoff Streeter wrote Dyalog APL v1.0 in 1981-1983
- Geoff is in good health
  - Now working 3 days per week
  - And still volunteering at night...
- However, Geoff has announced that he intends to retire in April'23
- We hope to welcome him back for a retrospective talk at Dyalog'23

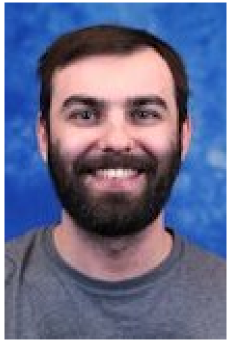
The Queen's Award for Voluntary Service
• THE MBE FOR VOLUNTEER GROUPS •

Dyalog … The Next Generation



🡐 February, May, July & September 2022

Dyalog ... The Next Generation



← Worked with "Plan 9"

Software Security Process

BSIMM

Building Security in Maturity Model

- Compare own routines to industry practices
- Implement and continuously review practices that reduce security risk
  - Dyalog's processes will treat potential **computational errors** as threats on par with **classical security threats**
- Hope to publish an Audit Report in ~~2022~~2023

1. Building the Team

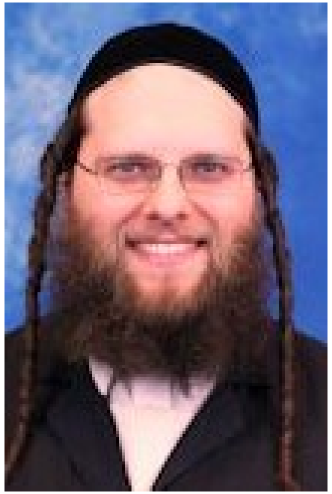
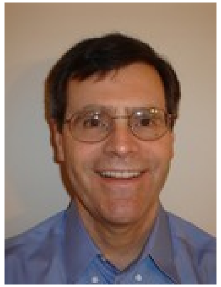
Talks by recent recruits…

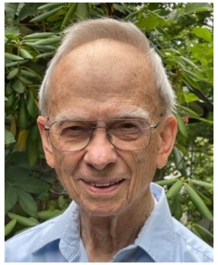
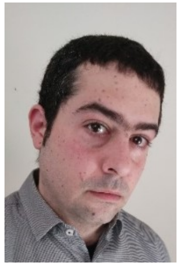
Monday 16:15 Plan 9 from Outer Space (Peter Mikkelsen)

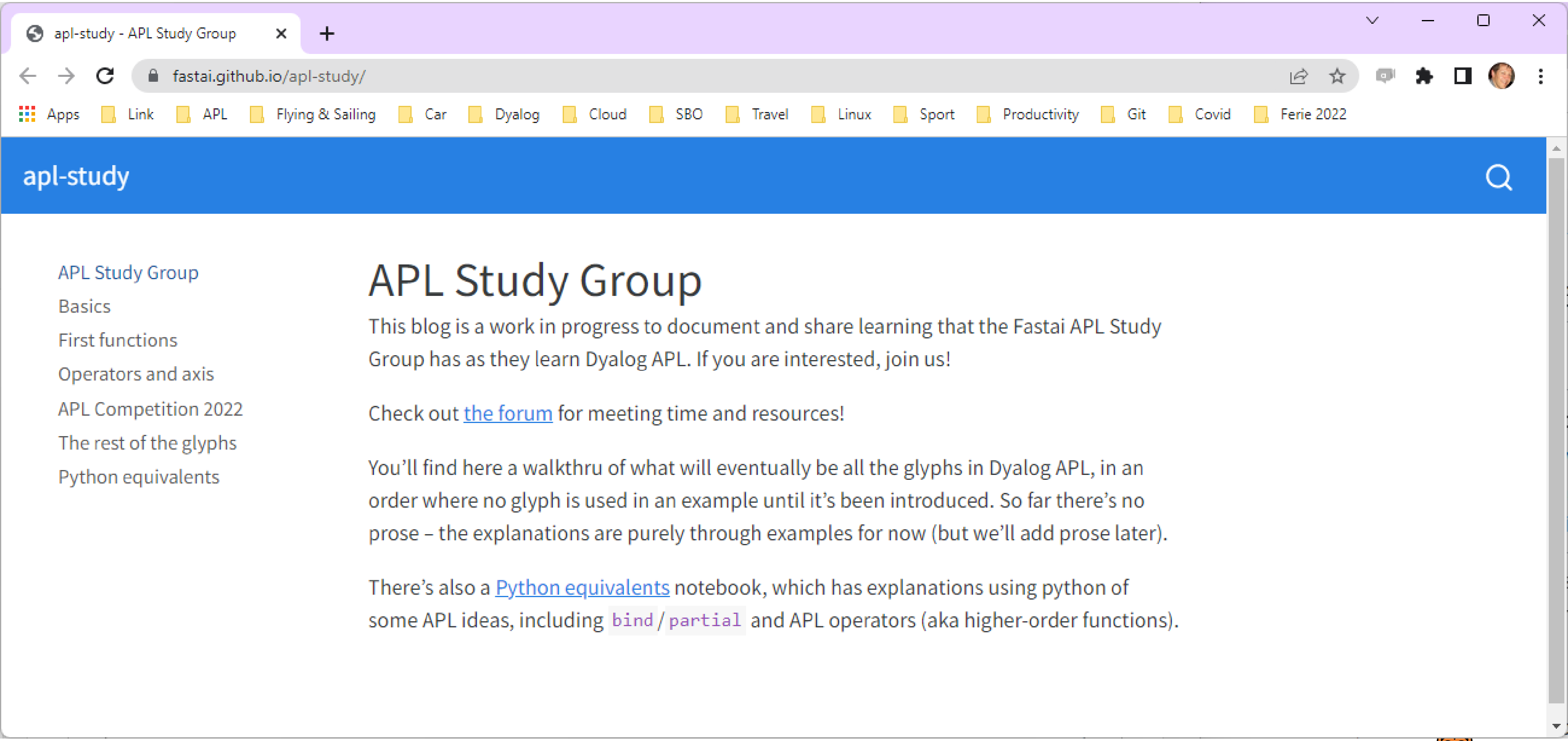
Tuesday 15:00 Performance Improvements in Set Operations (Karta Kooner)

2. Training & Evangelism

- mastering.dyalog.com
  Rodrigo Girão Serrao
- course.dyalog.com
  Rich Park
- tutorial.dyalog.com
  Gary Bergquist + Andrew Sengul
- xpqz.github.io/learnapl
  Stefan Kruger (IBM)


apl-study - APL Study Group

fastai.github.io/apl-study/

apl-study

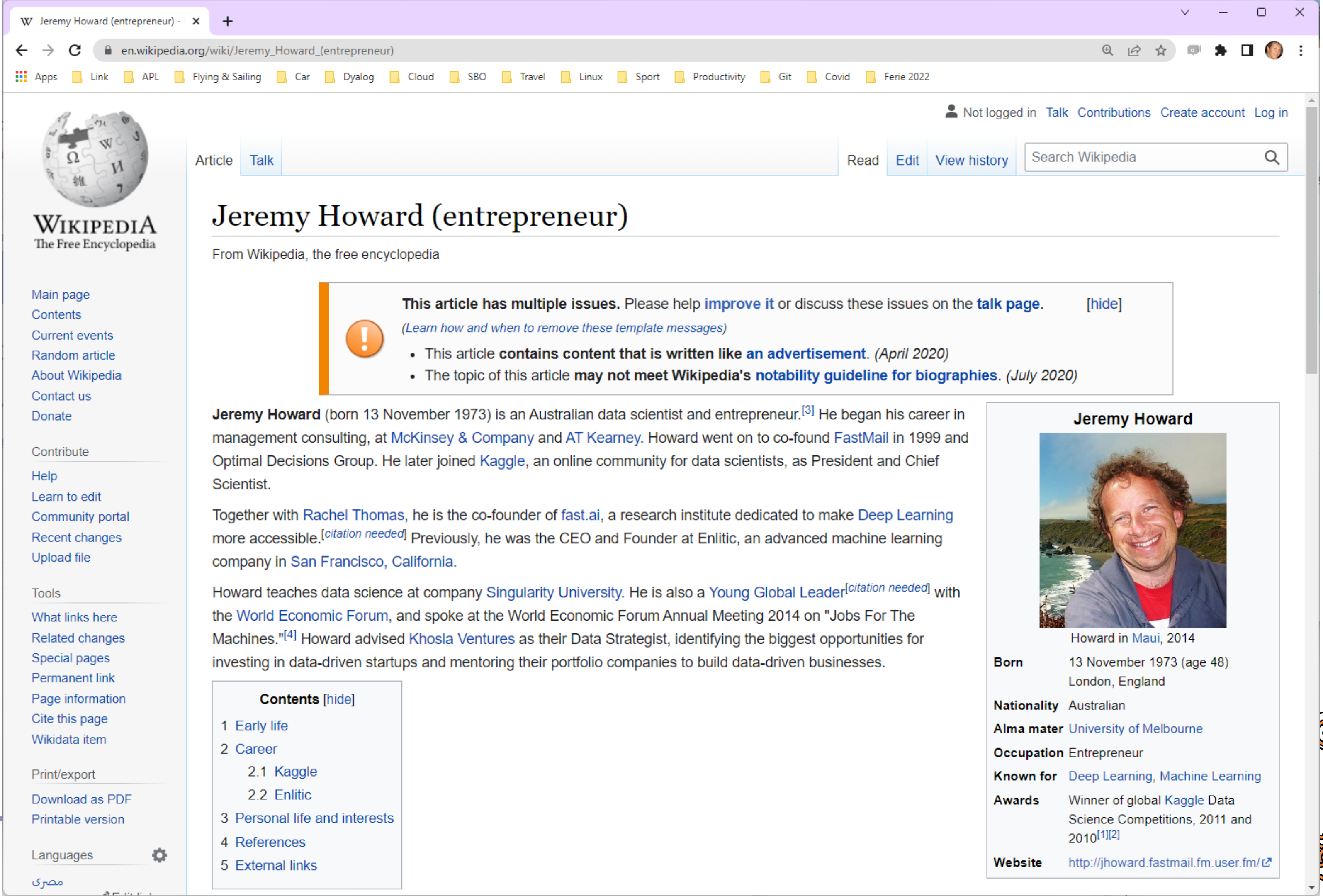
APL Study Group

This blog is a work in progress to document and share learning that the Fastai APL Study Group has as they learn Dyalog APL. If you are interested, join us!

Check out the forum for meeting time and resources!

You'll find here a walkthru of what will eventually be all the glyphs in Dyalog APL, in an order where no glyph is used in an example until it's been introduced. So far there's no prose – the explanations are purely through examples for now (but we'll add prose later).

There's also a Python equivalents notebook, which has explanations using python of some APL ideas, including `bind`/`partial` and APL operators (aka higher-order functions).



Jeremy Howard (entrepreneur) -

en.wikipedia.org/wiki/Jeremy_Howard_(entrepreneur)

Apps · Link · APL · Flying & Sailing · Car · Dyalog · Cloud · SBO · Travel · Linux · Sport · Productivity · Git · Covid · Ferie 2022

Not logged in Talk Contributions Create account Log in

Article Talk · Read Edit View history · Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF
Printable version

Languages
مصرى

Jeremy Howard (entrepreneur)

From Wikipedia, the free encyclopedia

This article has multiple issues. Please help improve it or discuss these issues on the talk page. [hide]

(Learn how and when to remove these template messages)

- This article **contains content that is written like an advertisement**. *(April 2020)*
- The topic of this article **may not meet Wikipedia's notability guideline for biographies**. *(July 2020)*

Jeremy Howard (born 13 November 1973) is an Australian data scientist and entrepreneur.[3] He began his career in management consulting, at McKinsey & Company and AT Kearney. Howard went on to co-found FastMail in 1999 and Optimal Decisions Group. He later joined Kaggle, an online community for data scientists, as President and Chief Scientist.

Together with Rachel Thomas, he is the co-founder of fast.ai, a research institute dedicated to make Deep Learning more accessible.[*citation needed*] Previously, he was the CEO and Founder at Enlitic, an advanced machine learning company in San Francisco, California.

Howard teaches data science at company Singularity University. He is also a Young Global Leader[*citation needed*] with the World Economic Forum, and spoke at the World Economic Forum Annual Meeting 2014 on "Jobs For The Machines."[4] Howard advised Khosla Ventures as their Data Strategist, identifying the biggest opportunities for investing in data-driven startups and mentoring their portfolio companies to build data-driven businesses.

Contents [hide]

Jeremy Howard

Howard in Maui, 2014

Born	13 November 1973 (age 48) London, England
Nationality	Australian
Alma mater	University of Melbourne
Occupation	Entrepreneur
Known for	Deep Learning, Machine Learning
Awards	Winner of global Kaggle Data Science Competitions, 2011 and 2010[1][2]
Website	http://jhoward.fastmail.fm.user.fm/

Training & Evangelism

- We trained a group of 24 developers in India last year (via Zoom)
- The materials prepared for this exercise are available free of charge at course.dyalog.com
  - This will be the case for all training materials that we are creating
- We are also considering running on-line courses – contact us if you need training!

Basic Licence

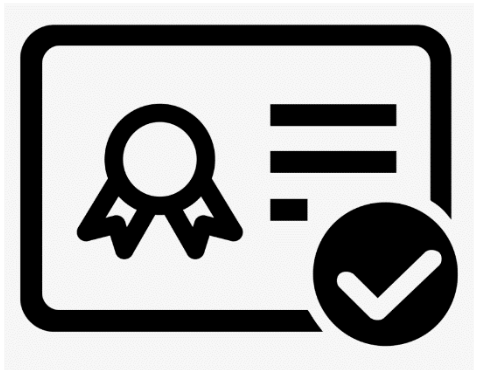
- Replaces *non-commercial licence*
- Allows distribution of Dyalog along with your work, under the default royalty licence
    - Fee is 2% of gross APL-based revenue
    - No fees due if revenue < GBP 5,000 in a calendar year
    - Multiple alternative commercial licence schemes are available

- For GBP 150 per year, you can subscribe to the Dyalog Support Service (DSS)

Basic Licence

Perhaps the most important "feature" of v18.2 – intended for

- non-commercial use
- education
- personal projects & experiments
- sharing your experience
- proof of concepts / trials
- participating in programming competitions for cash prizes
- fun



Keyboarding on all platforms

Issues:

- Dyalog IME does not work with Windows Universal Windows Platform applications
- APL keyboards do not work in RIDE (backtick still works) under Wayland (Linux)
- New users report that "ctrl" is problematic as the APL key

Immediate Solutions:

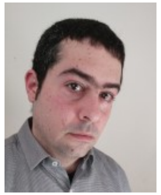
- Keyboards for Windows which use different "APL" keys (Alt, AltGr, etc)
- Backtick-style keyboards for all platforms

Longer Term:

- A new IME which offers a similar experience across supported platforms and works in and out of the IDEs (this will take a bit longer)

2. Training & Evangelism

Thursday 11:15 Dyalog and Academia
(Jesús Galàn López and Gitte Christensen)

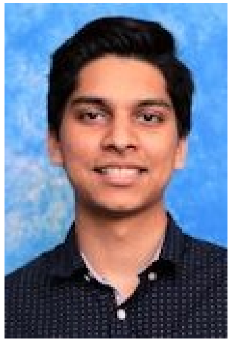


Thursday 11:45 What – Another APL Book?
(Ray Polivka)

Thursday 12:05 Growing APLers
(Rich Park)

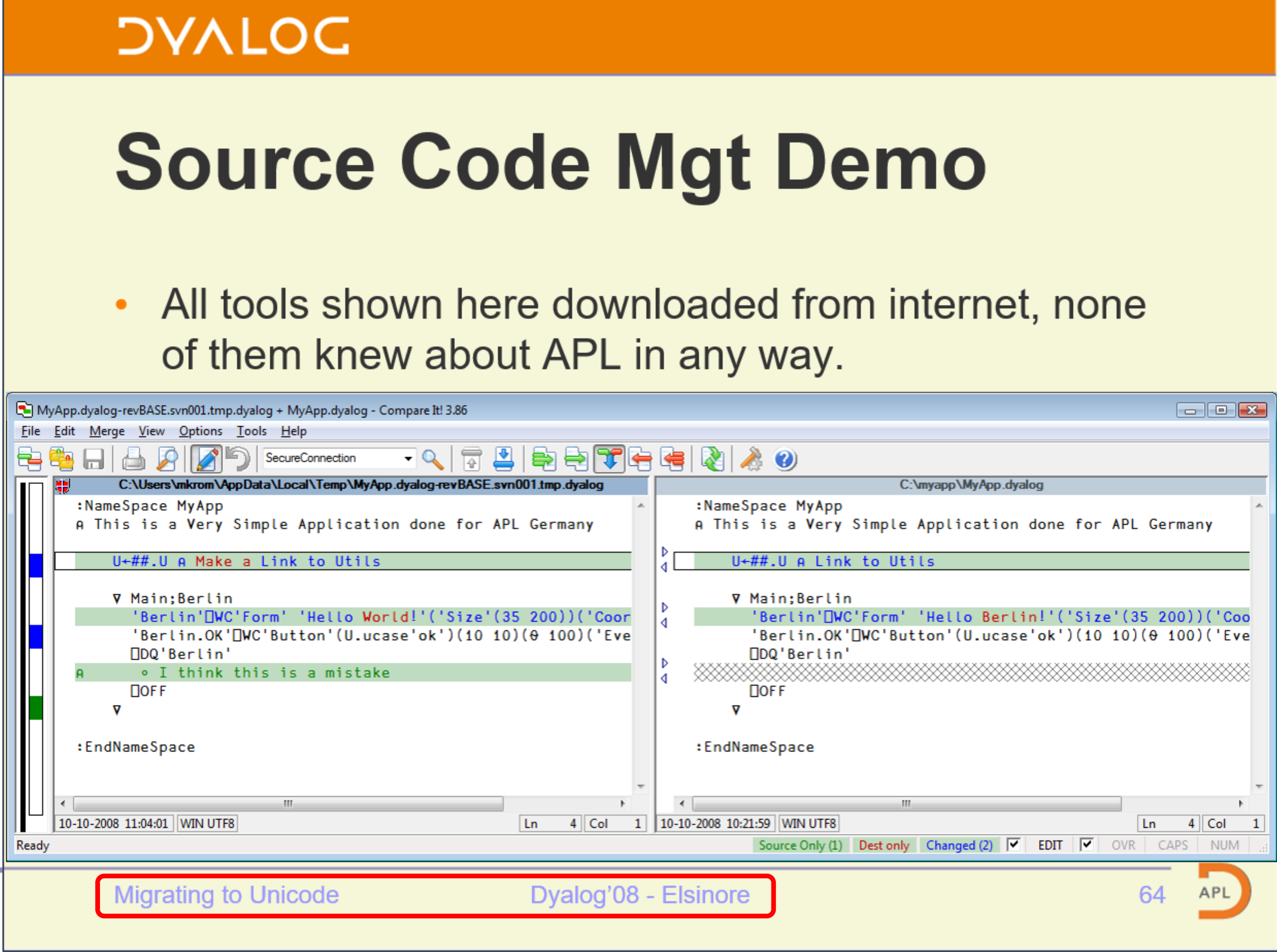
3. Consulting

- We started building a consulting group in the USA in 2019
  - Paused due to Covid and other factors
- We expect to resume recruiting APL consultants in the USA next year
- Get in touch
  - ... if you need consulting (also outside the USA)
  - ... know someone interested in an APL career



4. Source

- Here he goes again



Source Code Mgt Demo

- All tools shown here downloaded from internet, none of them knew about APL in any way.

MyApp.dyalog-revBASE.svn001.tmp.dyalog + MyApp.dyalog - Compare It! 3.86

C:\Users\mkrom\AppData\Local\Temp\MyApp.dyalog-revBASE.svn001.tmp.dyalog

```
:NameSpace MyApp
⍝ This is a Very Simple Application done for APL Germany

    U←##.U ⍝ Make a Link to Utils

    ∇ Main;Berlin
      'Berlin'⎕WC'Form' 'Hello World!'('Size'(35 200))('Coor
      'Berlin.OK'⎕WC'Button'(U.ucase'ok')(10 10)(⍬ 100)('Eve
      ⎕DQ'Berlin'
⍝      ∘ I think this is a mistake
      ⎕OFF
    ∇

:EndNameSpace
```

10-10-2008 11:04:01 WIN UTF8 Ln 4 Col 1

C:\myapp\MyApp.dyalog

```
:NameSpace MyApp
⍝ This is a Very Simple Application done for APL Germany

    U←##.U ⍝ Link to Utils

    ∇ Main;Berlin
      'Berlin'⎕WC'Form' 'Hello Berlin!'('Size'(35 200))('Coo
      'Berlin.OK'⎕WC'Button'(U.ucase'ok')(10 10)(⍬ 100)('Eve
      ⎕DQ'Berlin'

      ⎕OFF
    ∇

:EndNameSpace
```

10-10-2008 10:21:59 WIN UTF8 Ln 4 Col 1

Ready

File Edit Merge View Options Tools Help
SecureConnection
Source Only (1) Dest only Changed (2) EDIT OVR CAPS NUM

Why is Text Source IMPORTANT?

MuleSoft

eclipse

SUBVERSION

mercurial

Jenkins

Visual Studio Code

GitHub

git

circleci

Emacs

Vim

4. Source in Text Files

Done:

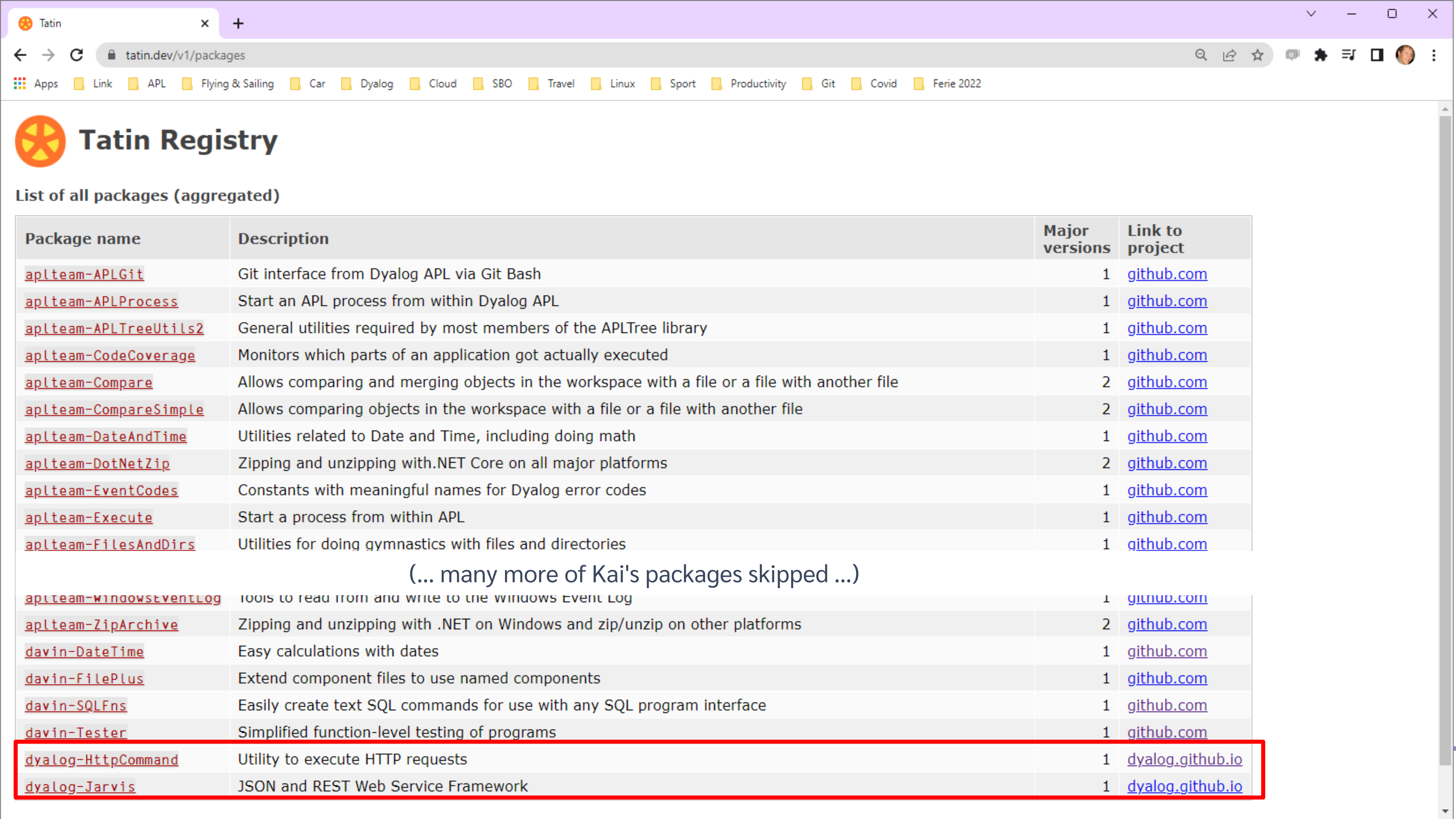
- Link 3.0 included with v18.2
    - Compatible with v18.0
    - Will replace SALT
- Launch APL from text source
    - No workspace required
    - Right-click on a function or namespace source file in Windows Explorer and run it
    - Also supported in containers
- HashBang/Shebang scripting

Project Managers:

- Acre
- Dado
- Cider

Package Manager:

- Tatin



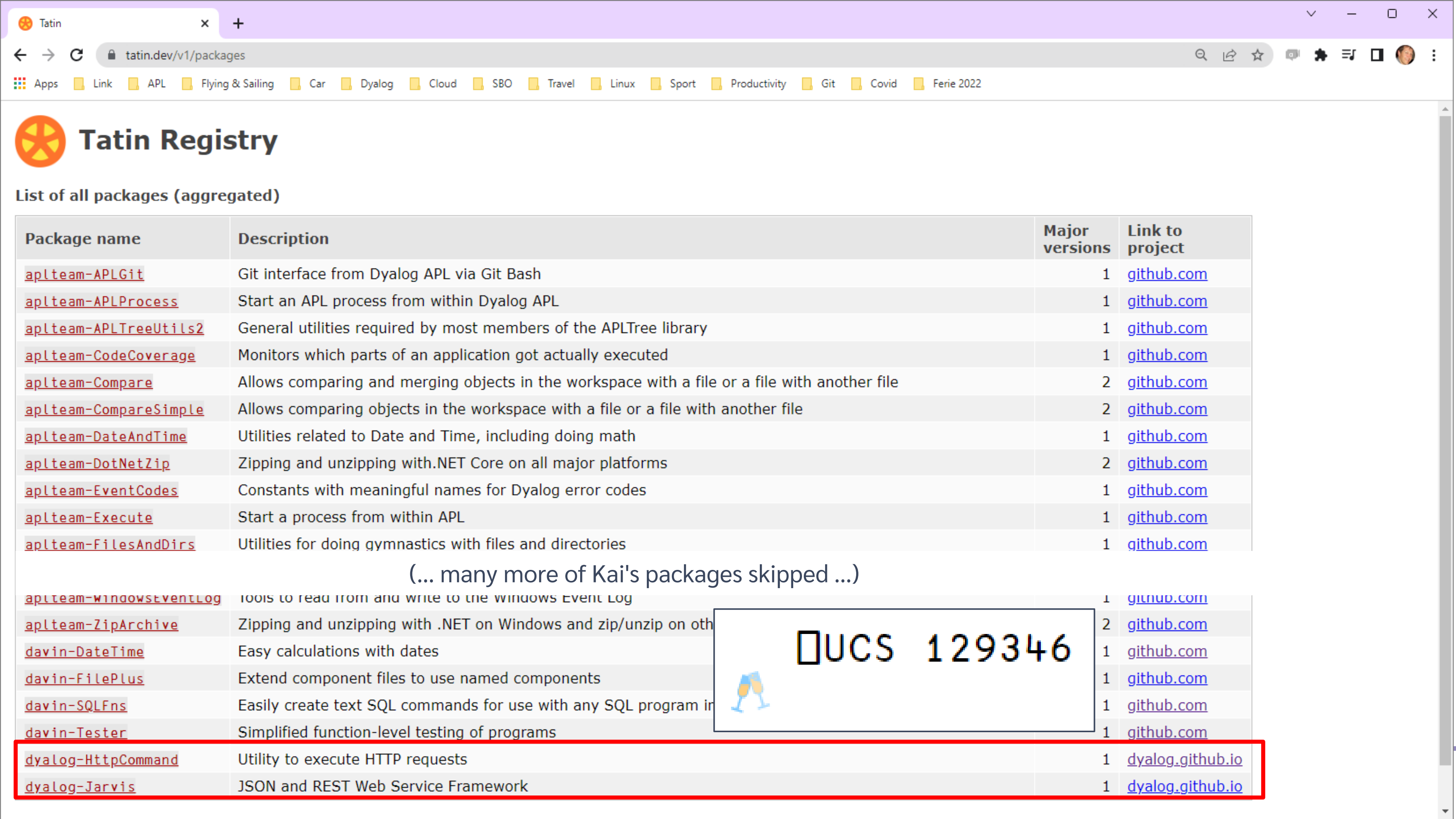
Tatin Registry

List of all packages (aggregated)

Package name	Description	Major versions	Link to project
aplteam-APLGit	Git interface from Dyalog APL via Git Bash	1	github.com
aplteam-APLProcess	Start an APL process from within Dyalog APL	1	github.com
aplteam-APLTreeUtils2	General utilities required by most members of the APLTree library	1	github.com
aplteam-CodeCoverage	Monitors which parts of an application got actually executed	1	github.com
aplteam-Compare	Allows comparing and merging objects in the workspace with a file or a file with another file	2	github.com
aplteam-CompareSimple	Allows comparing objects in the workspace with a file or a file with another file	2	github.com
aplteam-DateAndTime	Utilities related to Date and Time, including doing math	1	github.com
aplteam-DotNetZip	Zipping and unzipping with.NET Core on all major platforms	2	github.com
aplteam-EventCodes	Constants with meaningful names for Dyalog error codes	1	github.com
aplteam-Execute	Start a process from within APL	1	github.com
aplteam-FilesAndDirs	Utilities for doing gymnastics with files and directories	1	github.com

(... many more of Kai's packages skipped ...)

Package name	Description	Major versions	Link to project
aplteam-WindowsEventLog	Tools to read from and write to the Windows Event Log	1	github.com
aplteam-ZipArchive	Zipping and unzipping with .NET on Windows and zip/unzip on other platforms	2	github.com
davin-DateTime	Easy calculations with dates	1	github.com
davin-FilePlus	Extend component files to use named components	1	github.com
davin-SQLFns	Easily create text SQL commands for use with any SQL program interface	1	github.com
davin-Tester	Simplified function-level testing of programs	1	github.com
dyalog-HttpCommand	Utility to execute HTTP requests	1	dyalog.github.io
dyalog-Jarvis	JSON and REST Web Service Framework	1	dyalog.github.io



Tatin Registry

List of all packages (aggregated)

Package name	Description	Major versions	Link to project
aplteam-APLGit	Git interface from Dyalog APL via Git Bash	1	github.com
aplteam-APLProcess	Start an APL process from within Dyalog APL	1	github.com
aplteam-APLTreeUtils2	General utilities required by most members of the APLTree library	1	github.com
aplteam-CodeCoverage	Monitors which parts of an application got actually executed	1	github.com
aplteam-Compare	Allows comparing and merging objects in the workspace with a file or a file with another file	2	github.com
aplteam-CompareSimple	Allows comparing objects in the workspace with a file or a file with another file	2	github.com
aplteam-DateAndTime	Utilities related to Date and Time, including doing math	1	github.com
aplteam-DotNetZip	Zipping and unzipping with.NET Core on all major platforms	2	github.com
aplteam-EventCodes	Constants with meaningful names for Dyalog error codes	1	github.com
aplteam-Execute	Start a process from within APL	1	github.com
aplteam-FilesAndDirs	Utilities for doing gymnastics with files and directories	1	github.com

(... many more of Kai's packages skipped ...)

aplteam-WindowsEventLog	Tools to read from and write to the Windows Event Log	1	github.com
aplteam-ZipArchive	Zipping and unzipping with .NET on Windows and zip/unzip on oth	2	github.com
davin-DateTime	Easy calculations with dates	1	github.com
davin-FilePlus	Extend component files to use named components	1	github.com
davin-SQLFns	Easily create text SQL commands for use with any SQL program ir	1	github.com
davin-Tester	Simplified function-level testing of programs	1	github.com
dyalog-HttpCommand	Utility to execute HTTP requests	1	dyalog.github.io
dyalog-Jarvis	JSON and REST Web Service Framework	1	dyalog.github.io

```
⎕UCS 129346
🥂
```

4. Source in Text Files

Still to do:

- Publish more [Dyalog] packages on the Tatin server
- Cider Project Manager
- Array Notation

Current Use:

- Major customers have moved to text source
- New users tend to start with text source
- All new Dyalog tools are [open] text source on GitHub
  - Taking advantage of Continuous Integration for automated testing

Literal Array Notation

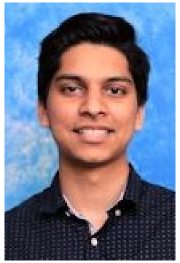
- Constants can form part of the "source" of an application
  - Enumerations
  - [Translated] strings
  - Conversion tables
- A notation for constants is an important piece of the "text source puzzle"
  - (In addition to being generally useful in code)

```
File Errors.apla
```

```
[ 2 'SYNTAX'
  3 'INDEX'
  4 'RANK'
  5 'LENGTH'
  6 'VALUE' ]
```

4. Source in Text Files

Sunday 09:30 SA3: Link, Text-Based Source,
and Source Code Management
(Morten Kromberg and Josh David)



Monday 14:15 The P words… Project and Packages
(Morten Kromberg)

(also used in many other workshops and presentations)

5. Service Orientation

It must be easy to run APL as a service
and call it from other environments.

Running APL as a Service

Imagine you have two pieces of business logic written in APL:

```
      sum←+/
      reverse←⌽
```

If you start Jarvis with a reference to the namespace containing the functions, Jarvis makes them available as a "Web Service":

```
      Server←Jarvis.Run 8080 #
```

Six different examples of calling "sum":

JavaScript

```
var xhr = new XMLHttpRequest();
xhr.open("POST", http://localhost:8080/sum);
xhr.setRequestHeader("content-type", "application/json");
xhr.send("[1,2,3,4]");
xhr.response;
```

PowerShell

```
$url = http://localhost:8080/sum
$hdrs = @{'content-type' = 'application/json'}
$body = '[1,3,5,7,9,11]'
Invoke-WebRequest -Method Post -URI $url -Headers $hdrs -Body $body
```

Python

```
url = 'http://localhost:8080/sum'
hdrs = {"content-type":"application/json"}
array = [2,4,6,8]
resp = requests.post(url, data=json.dumps(array), headers=hdrs)
print(resp.json())
```

curl

```
curl -d "[1,2,3,4,5]" -H "content-type:application/json" http://localhost:8080/sum
```

APL

```
HttpCommand.GetJSON 'post' 'localhost:8080/sum' (⍳5)
```

5. Service Orientation

It must be easy to run APL as a service and call it from other environments.

Jarvis Web Service Framework

- Replaces "JSONServer"
- Supports REST and "plain" HTTP/JSON services
- Widely available

  https://github.com/dyalog/jarvis
  https://hub.docker.com/r/dyalog/jarvis
  https://tatin.dev

Related Improvements

- Version 18.2 runs headless comfortably
  - Easier to use in containers
- RIDE 4.4 supports debugging threaded code
- ⎕JSON support for converting matrices to JSON tables

5. Service Orientation

- Continue work to make platforms more similar
  - Develop under Windows/macOS, deploy under Linux
- .NET Bridge provides cross-platform libraries & frameworks
- Unify configuration across platforms
  - All settings configurable via text files
  - Remove need for the Windows Registry
    - (Except perhaps to configure Windows IDE)

5. Service Orientation

- Materials developed for Dyalog'22 workshops will be extended over the next weeks and months
- We will publish a fully worked example of how to build a web service in Dyalog APL
  - Deployed in containers to the cloud
  - Scalable using several alternative mechanisms
  - User Sessions using 3rd Party Authentication
  - Encrypted Data
- More webinars, webcasts & samples to come

5. Service Orientation

Provide interface to monitor the state of APL processes

- CPU consumption
- Memory usage, Compaction counts, etc
- Are any threads suspended?
-)SI and Error information
- Available via API and / or protocols like SNMP

First version planned for v19.0

5. Service Orientation

Sunday 09:30 SA2: Building Web Services with Jarvis
(Brian Becker)

Sunday 14:00 SP2: Deploying Services
(Brian Becker & Morten Kromberg)

Tuesday 09:30 Automatic Application Builds with AWS
(Norbert Jurkiewicz)

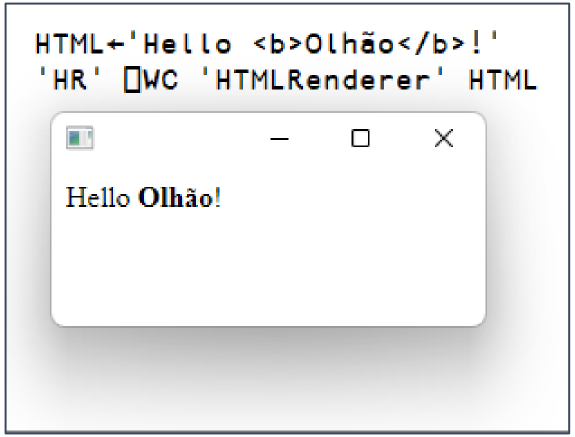
Wednesday 09:00 Simplifying Secure, Scalable Web Services
(Brian Becker)

(and used in many other workshops and presentations)

6. Cross Platform UI

When APL Services behind other GUI won't do...

- Adoption of the HTMLRenderer is growing as the delivery mechanism for new UI
  - Appears in four [user] presentations this week
- No clear choice of tool to generate HTML/JS
  - DUI/MiServer still has a small user base
  - Users are experimenting with writing own tools
  - ... and integrating HTML/JS generated by 3rd party tools or developers



```
HTML←'Hello <b>Olhão</b>!'
'HR' ⎕WC 'HTMLRenderer' HTML
```

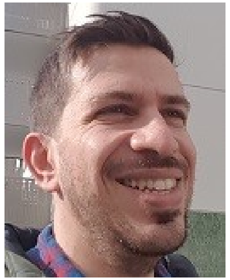
Hello **Olhão**!

HTMLRenderer improvements

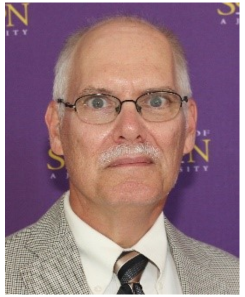
- Most important: Find a way to easily upgrade the Chromium Embedded Framework
  - In the medium term, turn the HTMLRenderer into an Open Source project to allow community participation
- Enhancement in v19.0
  - Support Multiple windows that take turns being modal

6. Cross Platform UI

Monday 11:40 Lift-Off from APL2 Mainframe to Dyalog in the Cloud
(Gilgamesh Athoraya – Tiamatica AB)



Monday 14:45 A Modern APL Workbench
(Kimmo Linna - Finnair)



Wednesday 09:30 TAMPA – Taming Mathematical Programming in APL
(Stephen Mansour – Misericordia University)

Wednesday 10:00 Integrating HTMLRenderer Into Existing Applications
(Norbert Jurkiewicz – The Carlisle Group)

7. [Microsoft].NET

As .NET celebrates 20 years of existence, Microsoft is pushing everyone to move from proprietary Microsoft.Net Framework to the new open source, cross-platform .NET.

Name	Platforms	Version Numbers
Microsoft.NET Framework	`Windows`	1 2 4
.NET (previously ".NET Core")	`Windows Linux macOS`	3 5 6 7

Dyalog v18.0 added a bridge to .NET 3, to complement the 20 year old bridge to the .NET framework.

.NET Bridge

- Add support for .NET 5, 6 & 7
- Export APL code as .NET assemblies
  - v18 .NET bridge only allows USING .NET classes
- Generate APL-based applications under
  - Linux: Amd/Intel x64 and Pi/AWS on Arm64
  - macOS (x64 and M1/M2)
  - Windows (x64 – maybe Arm64 later)
- Work on support for Async features

.NET 6 is the current Long Term Support version of .NET

7. [Microsoft].NET

In a nutshell, the specification is that the new bridge it will work exactly the same way as the old one.

But there will be some enhancements.

8. New target Platforms

- 64-bit ARM
  - This low power RISC processor is gaining traction
  - We expect to support v19.0 on ARM64 (specifically M1 & M2 Macs)

- Web Assembly (WASM)
  - Co-dfns will target WASM as an execution platform (no release date)
  - We are likely to look at whether a cut-down interpreter engine could run in the browser (no timeframe)

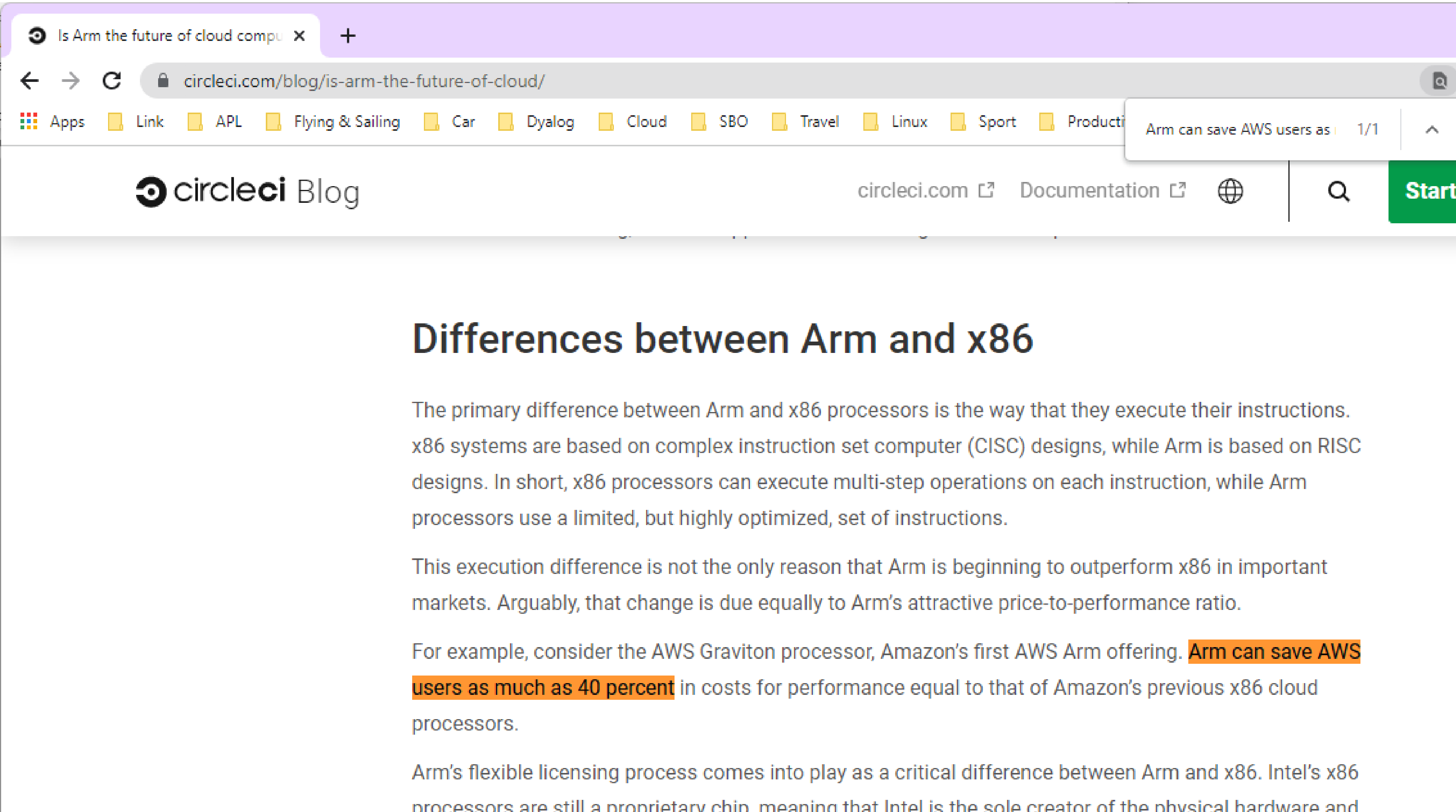
Arm64

64-bit ARM chips are appearing in places that Dyalog should support:

- M1 & M2 Macs
- Raspberry Pi – 64 Bit
- Amazon Web Services "Graviton"

C6g

Best price performance for compute-intensive workloads



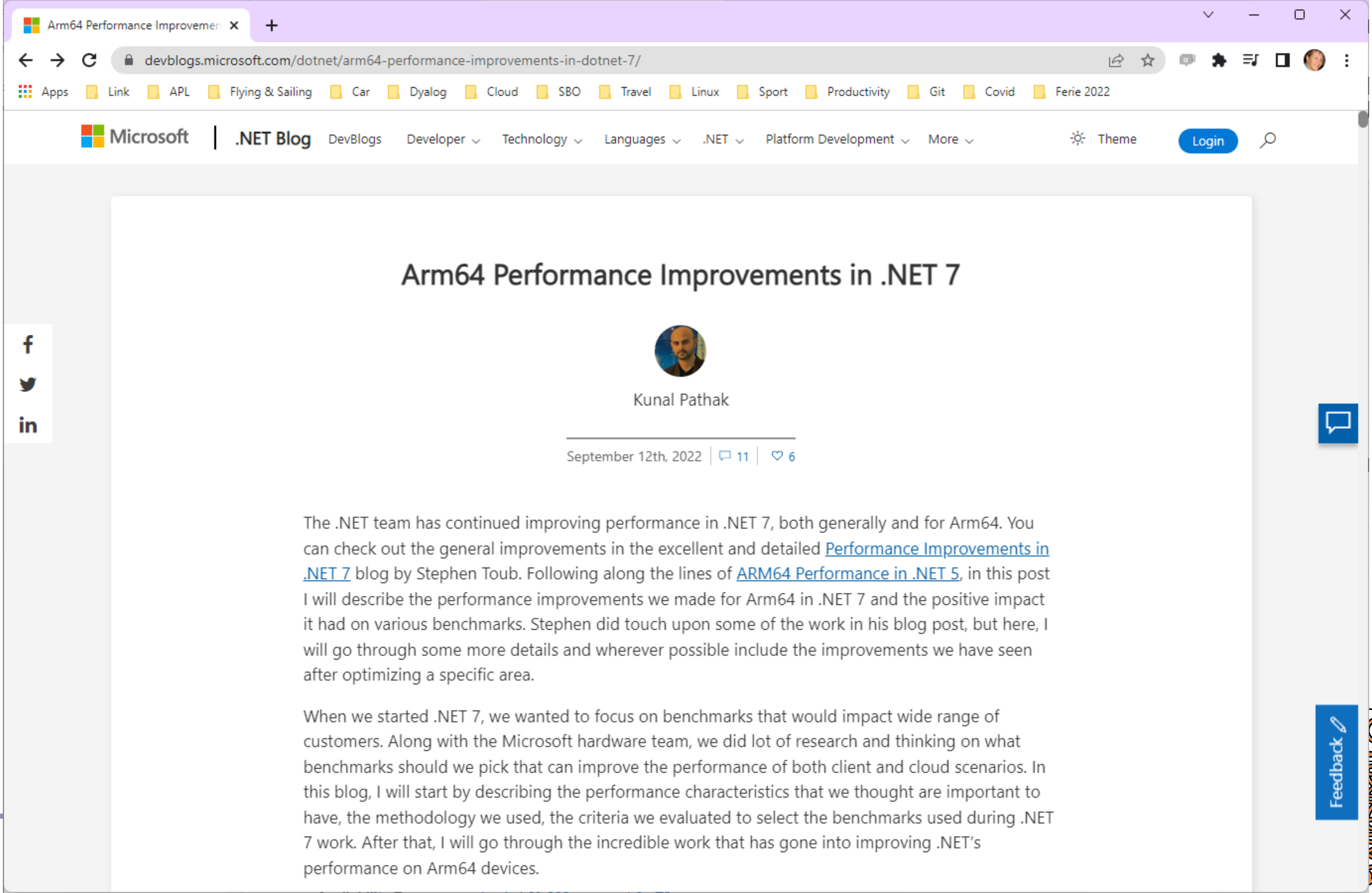
Differences between Arm and x86

The primary difference between Arm and x86 processors is the way that they execute their instructions. x86 systems are based on complex instruction set computer (CISC) designs, while Arm is based on RISC designs. In short, x86 processors can execute multi-step operations on each instruction, while Arm processors use a limited, but highly optimized, set of instructions.

This execution difference is not the only reason that Arm is beginning to outperform x86 in important markets. Arguably, that change is due equally to Arm's attractive price-to-performance ratio.

For example, consider the AWS Graviton processor, Amazon's first AWS Arm offering. Arm can save AWS users as much as 40 percent in costs for performance equal to that of Amazon's previous x86 cloud processors.

Arm's flexible licensing process comes into play as a critical difference between Arm and x86. Intel's x86 processors are still a proprietary chip, meaning that Intel is the sole creator of the physical hardware and


Arm64 Performance Improvemen
devblogs.microsoft.com/dotnet/arm64-performance-improvements-in-dotnet-7/
Apps Link APL Flying & Sailing Car Dyalog Cloud SBO Travel Linux Sport Productivity Git Covid Ferie 2022
Microsoft | .NET Blog DevBlogs Developer Technology Languages .NET Platform Development More Theme Login
Feedback

Arm64 Performance Improvements in .NET 7

Kunal Pathak

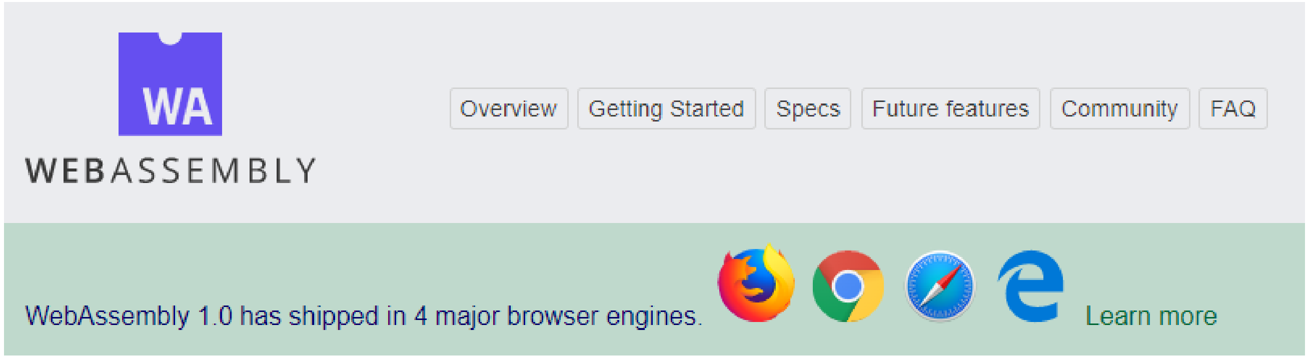
September 12th, 2022 11 6

The .NET team has continued improving performance in .NET 7, both generally and for Arm64. You can check out the general improvements in the excellent and detailed Performance Improvements in .NET 7 blog by Stephen Toub. Following along the lines of ARM64 Performance in .NET 5, in this post I will describe the performance improvements we made for Arm64 in .NET 7 and the positive impact it had on various benchmarks. Stephen did touch upon some of the work in his blog post, but here, I will go through some more details and wherever possible include the improvements we have seen after optimizing a specific area.

When we started .NET 7, we wanted to focus on benchmarks that would impact wide range of customers. Along with the Microsoft hardware team, we did lot of research and thinking on what benchmarks should we pick that can improve the performance of both client and cloud scenarios. In this blog, I will start by describing the performance characteristics that we thought are important to have, the methodology we used, the criteria we evaluated to select the benchmarks used during .NET 7 work. After that, I will go through the incredible work that has gone into improving .NET's performance on Arm64 devices.

Web Assembly (WASM)

APL running in the browser…



WebAssembly 1.0 has shipped in 4 major browser engines.

WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

8. New Target Platforms

Tuesday 16:45 Report on Co-dfns (Aaron Hsu)

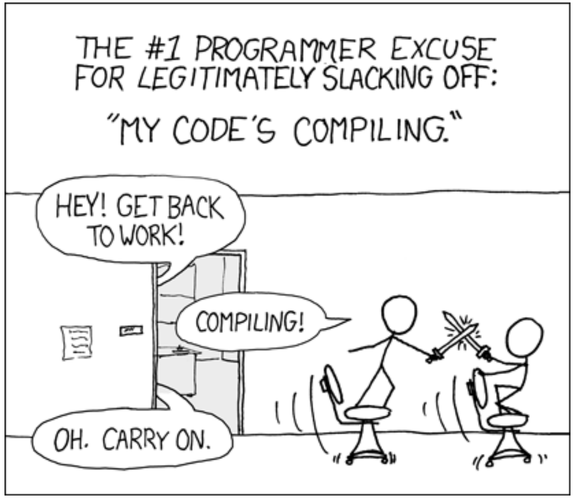
And talk to Ron about the ARM64 Port

9. Compiling APL

Work on Co-dfns continues. 2023 targets:

- Almost complete language support, including Control Structures & TradFns
- Characters, Mixed Arrays, Complex Numbers
- New backend targets: WASM/Javascript, Scheme/Lisp, Java/C#, Python
- Tracing and debugging

Emphasis is as much on making APL accessible for new applications in new environments, as on compiling existing applications



Source: xkcd.com

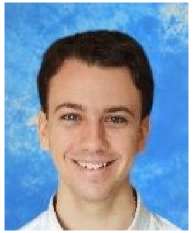
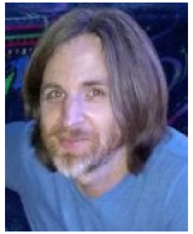
9. Compiling APL

Tuesday 11:30 Implementing the Convolutional Neural Network U-net in APL
(Rodrigo Girão Serrão)

Tuesday 12:00 APL on the Side
(Justin Dowdy – Semantic Arts)

Tuesday 14:00 Scheduling Array Operations
(Juuso Haavisto – University of Oxford)

Tuesday 16:45 Report on Co-dfns
(Aaron Hsu)

10. APL Language

- Literal Array Notation
- Multiple Numeric Towers, so we have a unified model which supports
  - 64-bit integers
  - Rational numbers
- Carefully considering which primitives are most important to add next. Not in a hurry.
  - Depth, Behind, Select, Under/Dual, etc…

Primitive Candidates

Select	(X⊇Y)
Depth	(f⍥k)
Behind	(f⍛h)
Under	(f⍢g)
Obverse	(f⍫g)

10. APL Language

Thursday 10:00 Filling the Core Language Gaps (Adám Brudzewsky)

JD might present some ideas today and tomorrow...

... but I'm not allowed to say

Ten Lane Highway

1. Building the Team
2. Training & Evangelism
3. Consulting
4. Source in Text Files
5. Service Orientation
6. Cross-Platform UI
7. [Microsoft].NET
8. New Target Platforms
9. Compiling APL
10. APL Language

DYALOG

Olhão 2022

The Road Ahead

Morten Kromberg