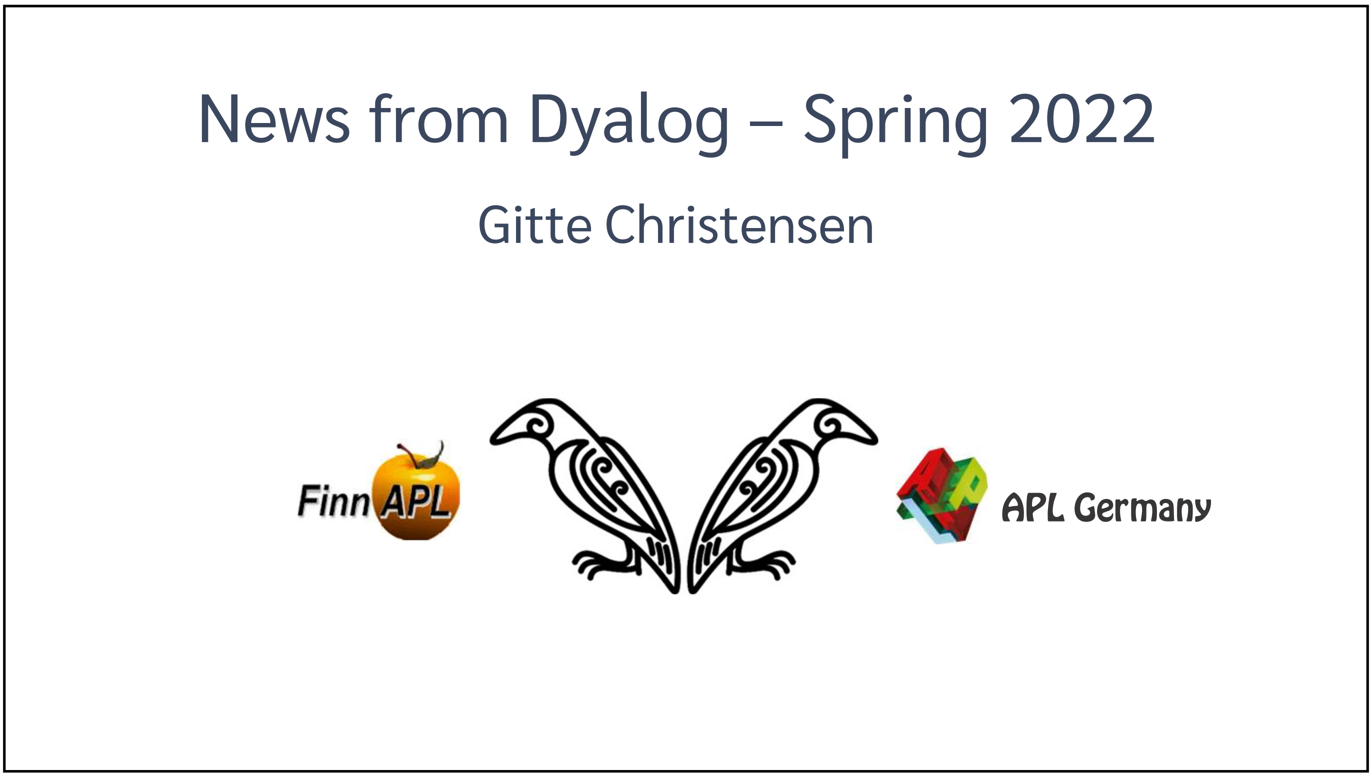
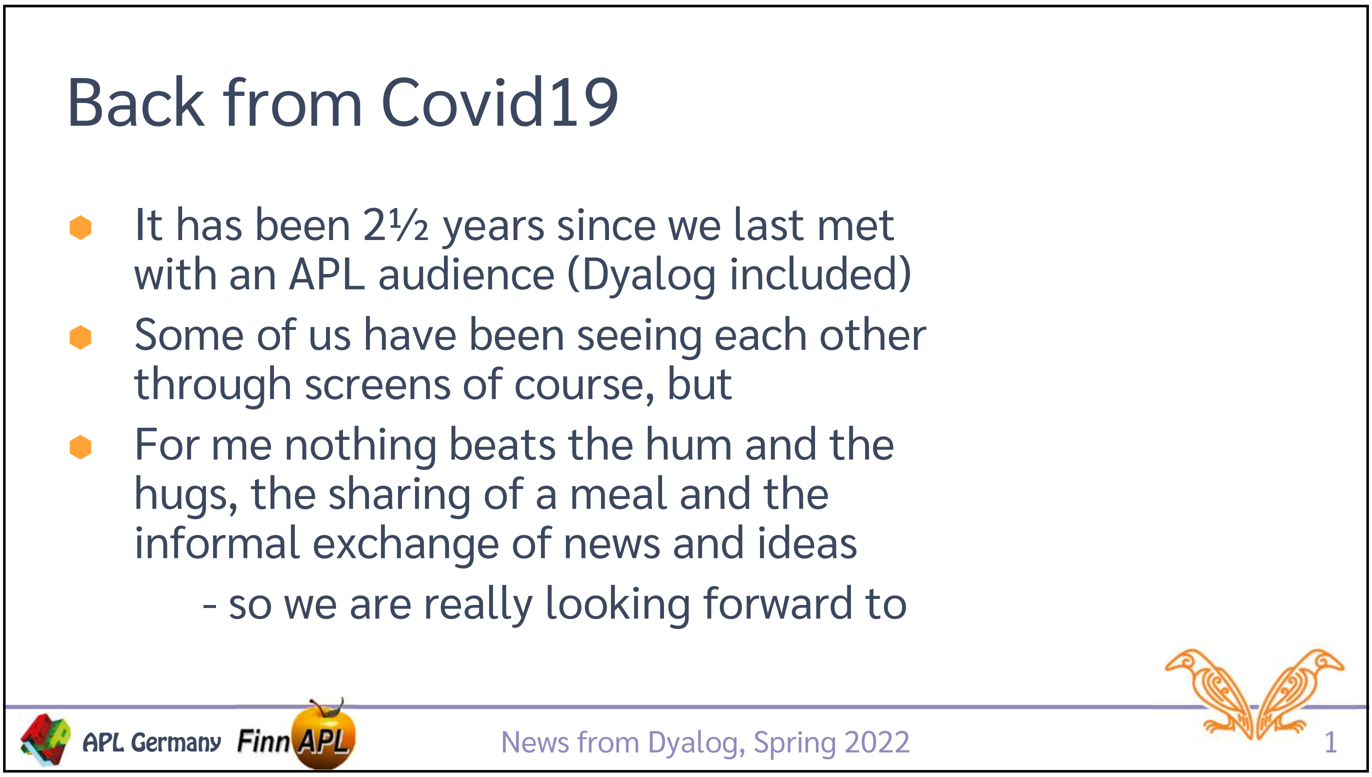


News from Dyalog – Spring 2022

Gitte Christensen

FinnAPL
APL Germany

0



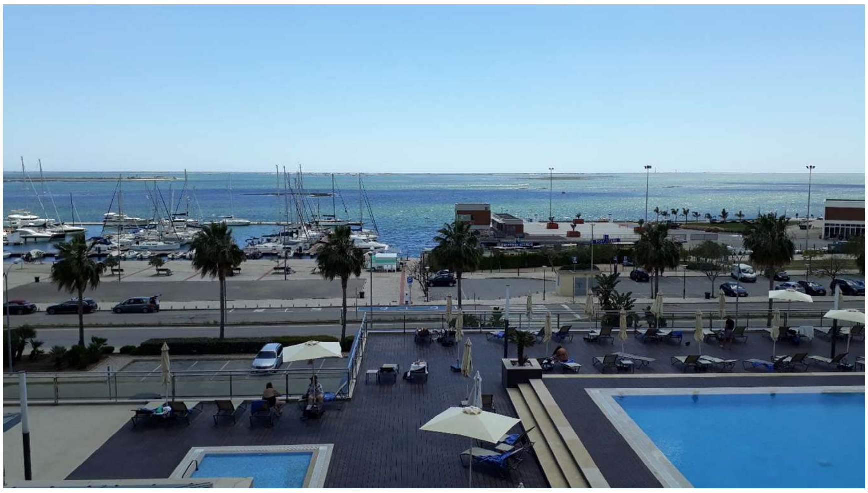
Back from Covid19

- It has been 2½ years since we last met with an APL audience (Dyalog included)
- Some of us have been seeing each other through screens of course, but
- For me nothing beats the hum and the hugs, the sharing of a meal and the informal exchange of news and ideas

  - so we are really looking forward to

1

Dyalog ’22, Olhão



At 5 Star Hotel

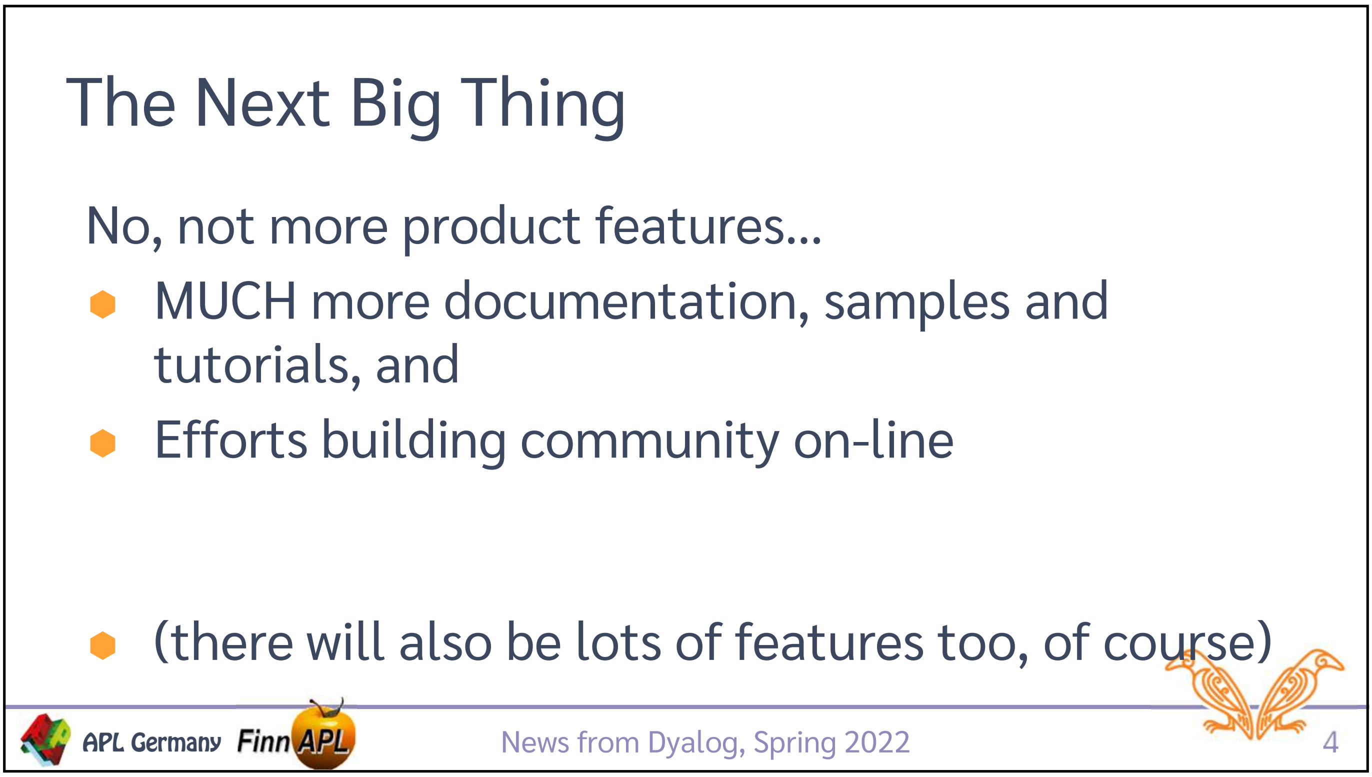
Real Marina

On the South Coast of Portugal

Community

- The APL community is important for members
  - To exchange ideas
  - Inspire each other
  - Help out with algorithms
  - Help with new technological inventions
  - Make friends

  - and most importantly
  - Receive Newcomers

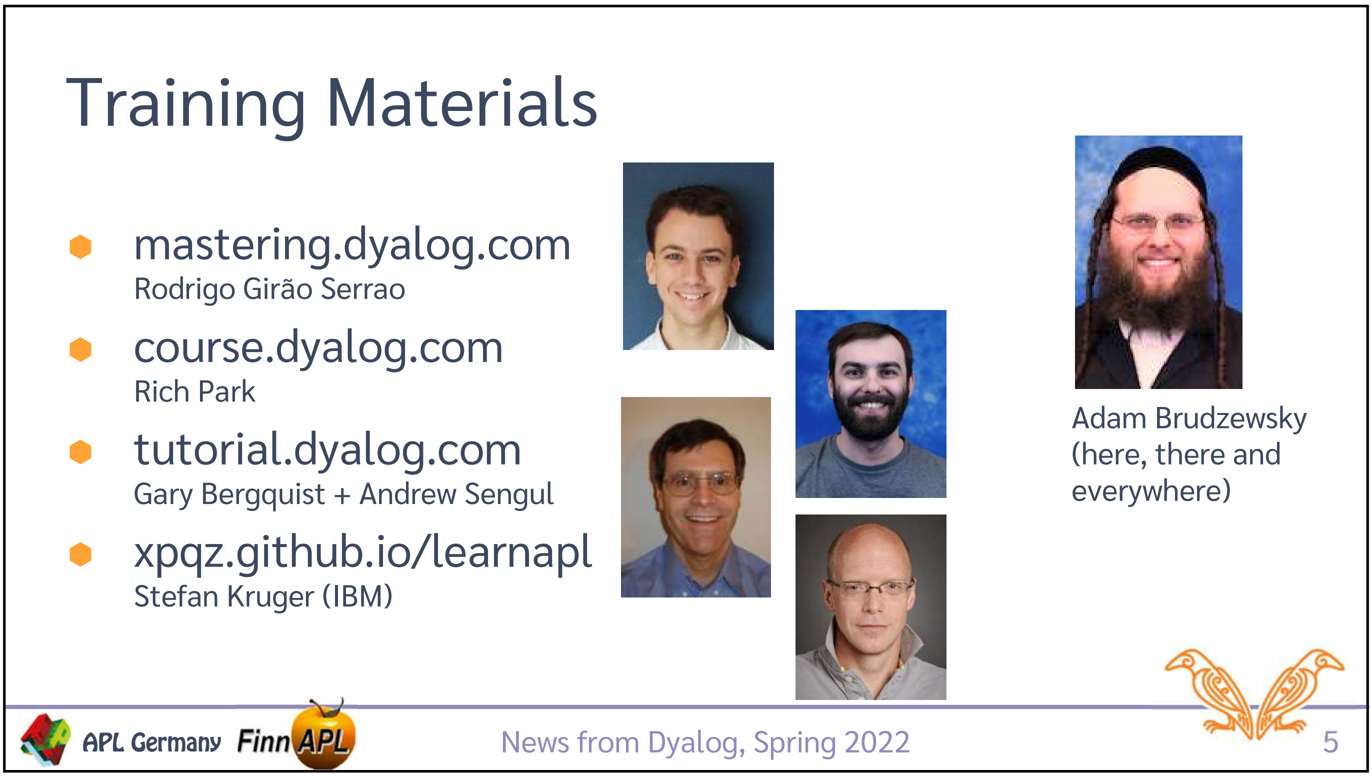


The Next Big Thing

No, not more product features…

- MUCH more documentation, samples and tutorials, and
- Efforts building community on-line
- (there will also be lots of features too, of course)

4

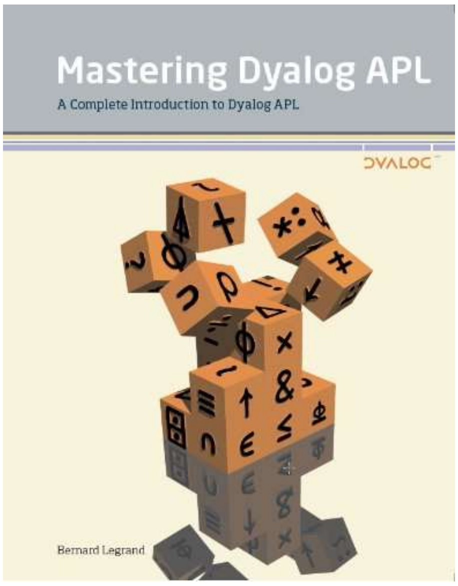


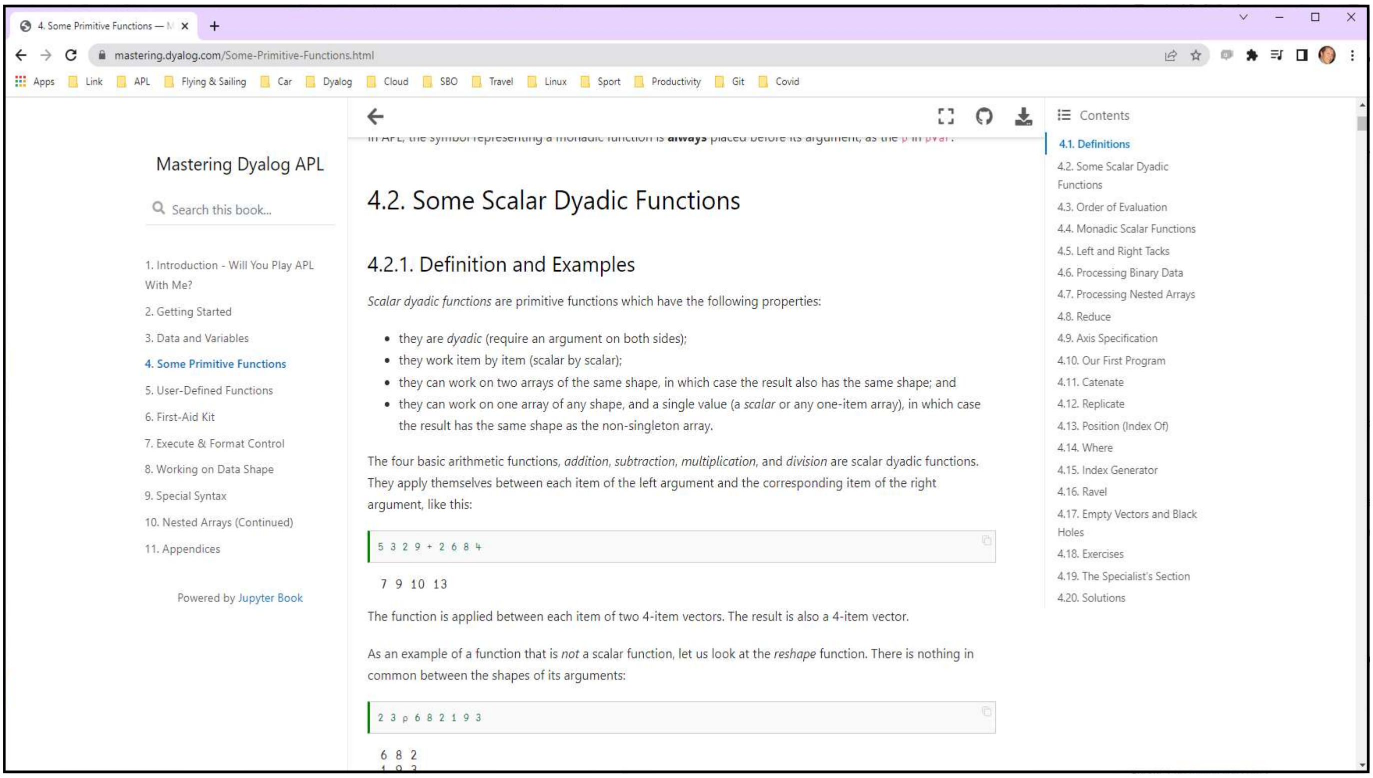
Training Materials

- 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)

Adam Brudzewsky
(here, there and
everywhere)

5




4. Some Primitive Functions — M
mastering.dyalog.com/Some-Primitive-Functions.html
Apps
Link
APL
Flying & Sailing
Car
Dyalog
Cloud
SBO
Travel
Linux
Sport
Productivity
Git
Covid

Mastering Dyalog APL

Search this book...

Powered by Jupyter Book

In APL, the symbol representing a monadic function is **always** placed before its argument, as the ρ in ρvar.

4.2. Some Scalar Dyadic Functions

4.2.1. Definition and Examples

Scalar dyadic functions are primitive functions which have the following properties:

- they are *dyadic* (require an argument on both sides);
- they work item by item (scalar by scalar);
- they can work on two arrays of the same shape, in which case the result also has the same shape; and
- they can work on one array of any shape, and a single value (a *scalar* or any one-item array), in which case the result has the same shape as the non-singleton array.

The four basic arithmetic functions, *addition*, *subtraction*, *multiplication*, and *division* are scalar dyadic functions. They apply themselves between each item of the left argument and the corresponding item of the right argument, like this:

```
5 3 2 9 + 2 6 8 4
```

```
 7 9 10 13
```

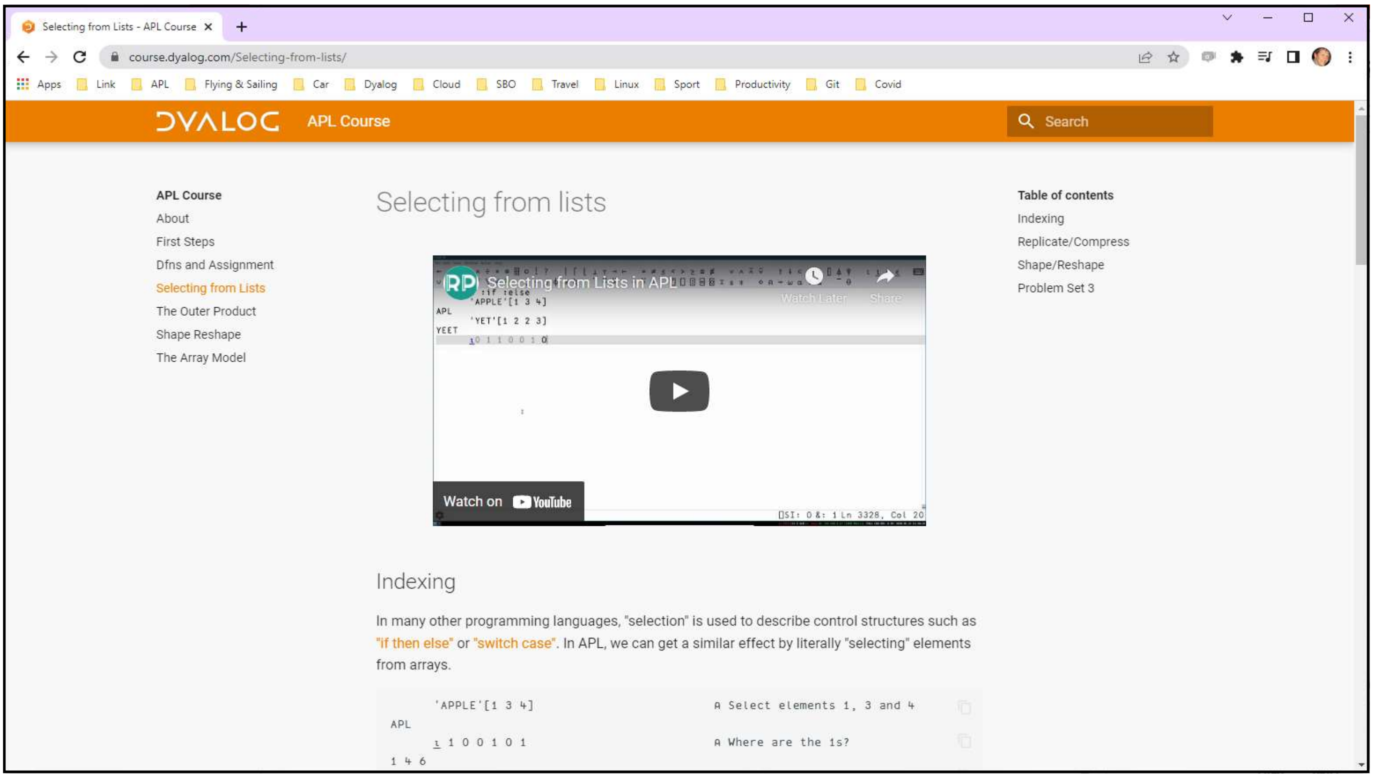
The function is applied between each item of two 4-item vectors. The result is also a 4-item vector.

As an example of a function that is *not* a scalar function, let us look at the *reshape* function. There is nothing in common between the shapes of its arguments:

```
2 3 ρ 6 8 2 1 9 3
```

```
 6 8 2
 1 9 3
```

Contents



Selecting from lists

Indexing

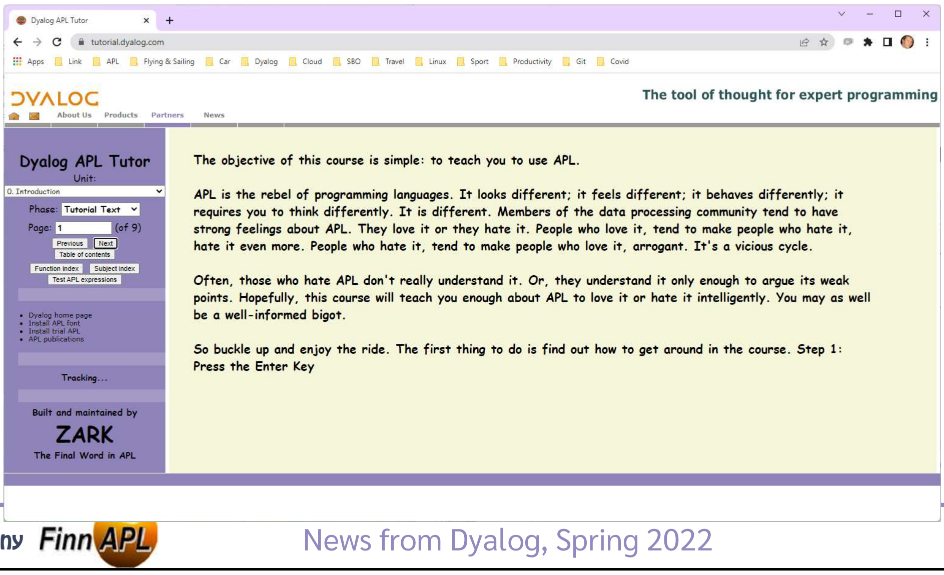
In many other programming languages, "selection" is used to describe control structures such as "if then else" or "switch case". In APL, we can get a similar effect by literally "selecting" elements from arrays.

```
      'APPLE'[1 3 4]                  ⍝ Select elements 1, 3 and 4
APL
      ⍸ 1 0 0 1 0 1                   ⍝ Where are the 1s?
1 4 6
```

8

tutorial.dyalog.com

The Zark (Gary Bergquist) Tutorial



Dyalog APL Tutor

Unit: 0. Introduction

Phase: Tutorial Text

Page: 1 (of 9)

The objective of this course is simple: to teach you to use APL.

APL is the rebel of programming languages. It looks different; it feels different; it behaves differently; it requires you to think differently. It is different. Members of the data processing community tend to have strong feelings about APL. They love it or they hate it. People who love it, tend to make people who hate it, hate it even more. People who hate it, tend to make people who love it, arrogant. It's a vicious cycle.

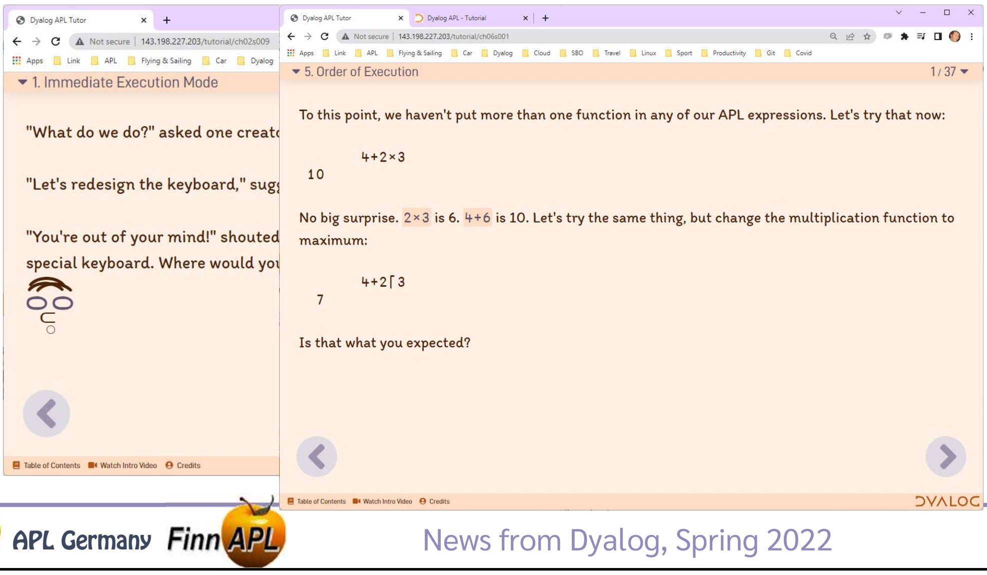
Often, those who hate APL don't really understand it. Or, they understand it only enough to argue its weak points. Hopefully, this course will teach you enough about APL to love it or hate it intelligently. You may as well be a well-informed bigot.

So buckle up and enjoy the ride. The first thing to do is find out how to get around in the course. Step 1: Press the Enter Key

9

tutorial.dyalog.com

The Zark (Gary Bergquist) Tutorial


Dyalog APL Tutor
Not secure | 143.198.227.203/tutorial/ch02s009
Apps Link APL Flying & Sailing Car Dyalog
Dyalog APL Tutor
Dyalog APL - Tutorial
Not secure | 143.198.227.203/tutorial/ch06s001
Apps Link APL Flying & Sailing Car Dyalog Cloud SBO Travel Linux Sport Productivity Git Covid
1/37
Table of Contents Watch Intro Video Credits
Table of Contents Watch Intro Video Credits
DYALOG

1. Immediate Execution Mode

"What do we do?" asked one creat

"Let's redesign the keyboard," sugg

"You're out of your mind!" shouted
special keyboard. Where would you

5. Order of Execution

To this point, we haven't put more than one function in any of our APL expressions. Let's try that now:

```
      4+2×3
10
```

No big surprise. 2×3 is 6. 4+6 is 10. Let's try the same thing, but change the multiplication function to maximum:

```
      4+2⌈3
7
```

Is that what you expected?

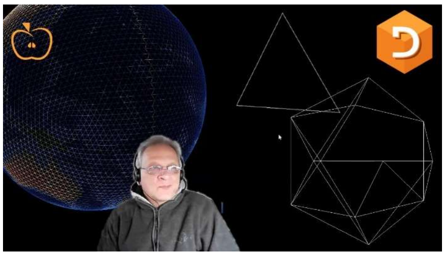
Facelift by
Andrew Sengul

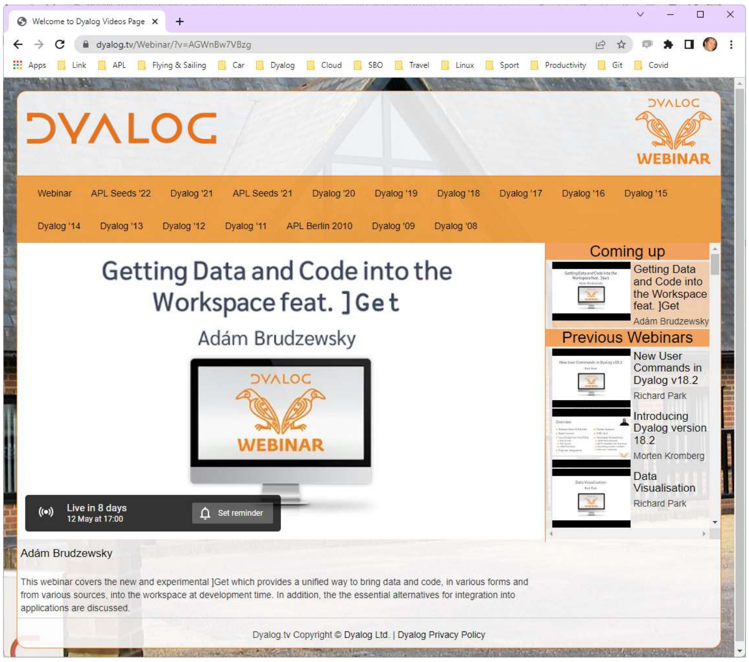
Will be released
soon

Documentation and Training Materials

- Reworking Mastering Dyalog APL
- New and improved TryAPL
- New and improved APL Tutor
- Maintain high frequency of webinars
  - And webcasts

Dyalog.tv/APLSeeds21





Getting Data and Code into the Workspace feat.]Get

Adám Brudzewsky

Coming up

Getting Data and Code into the Workspace feat.]Get
Adám Brudzewsky

Previous Webinars

New User Commands in Dyalog v18.2
Richard Park

Introducing Dyalog version 18.2
Morten Kromberg

Data Visualisation
Richard Park

Adám Brudzewsky

This webinar covers the new and experimental]Get which provides a unified way to bring data and code, in various forms and from various sources, into the workspace at development time. In addition, the the essential alternatives for integration into applications are discussed.

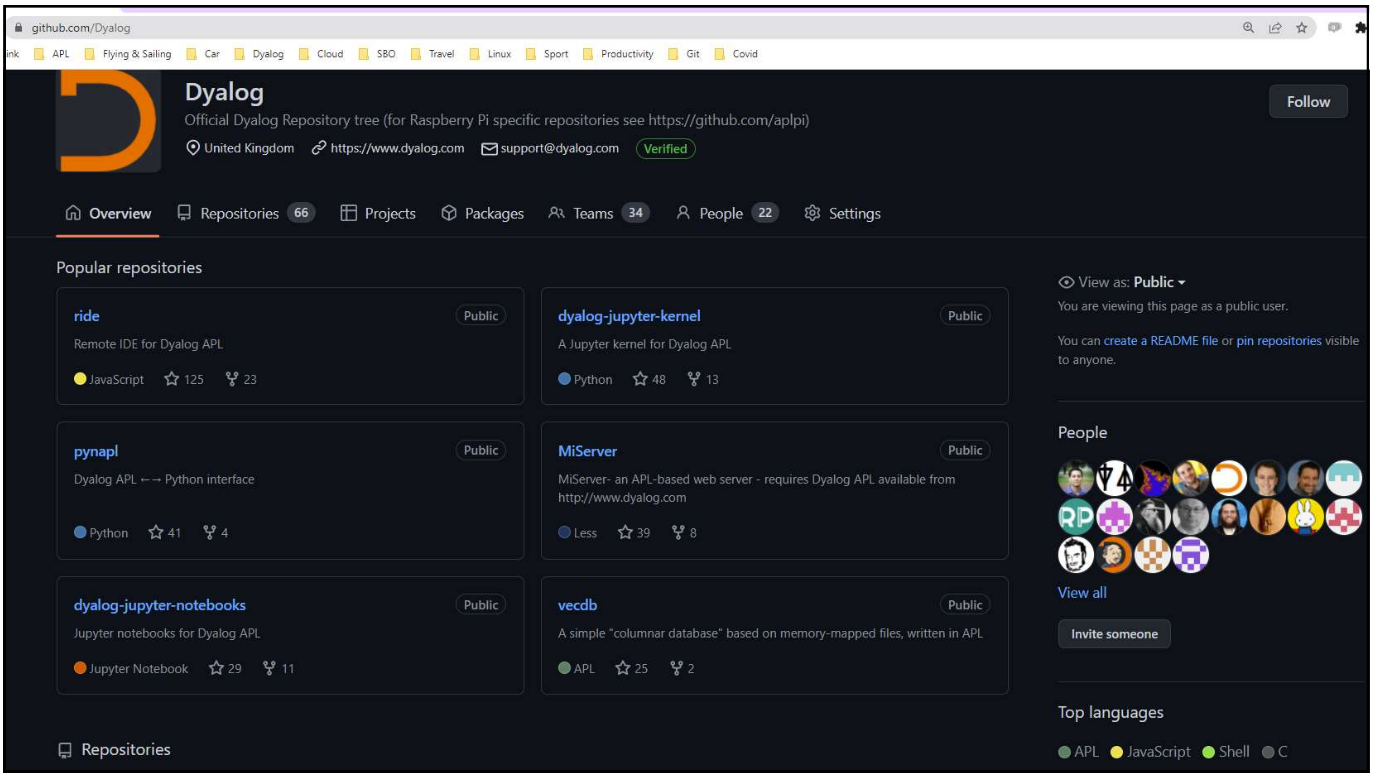
Dyalog.tv Copyright © Dyalog Ltd. | Dyalog Privacy Policy

Documentation and Training Materials

- Open Source Project Templates
  - Running APL applications as services / daemons under Windows and Linux
  - Deploying and operating secure and load-balanced solutions in Dyalog APL
  - Continuous integration, generating containers from APL code in GitHub, deploying to the cloud

APL Germany FinnAPL News from Dyalog, Spring 2022 14

14



Dyalog

Official Dyalog Repository tree (for Raspberry Pi specific repositories see https://github.com/aplpi)

United Kingdom https://www.dyalog.com support@dyalog.com Verified

Popular repositories

Repository	Visibility	Description	Language	Stars	Forks
ride	Public	Remote IDE for Dyalog APL	JavaScript	125	23
dyalog-jupyter-kernel	Public	A Jupyter kernel for Dyalog APL	Python	48	13
pynapl	Public	Dyalog APL ←→ Python interface	Python	41	4
MiServer	Public	MiServer- an APL-based web server - requires Dyalog APL available from http://www.dyalog.com	Less	39	8
dyalog-jupyter-notebooks	Public	Jupyter notebooks for Dyalog APL	Jupyter Notebook	29	11
vecdb	Public	A simple "columnar database" based on memory-mapped files, written in APL	APL	25	2

Repositories

View as: Public

You are viewing this page as a public user.

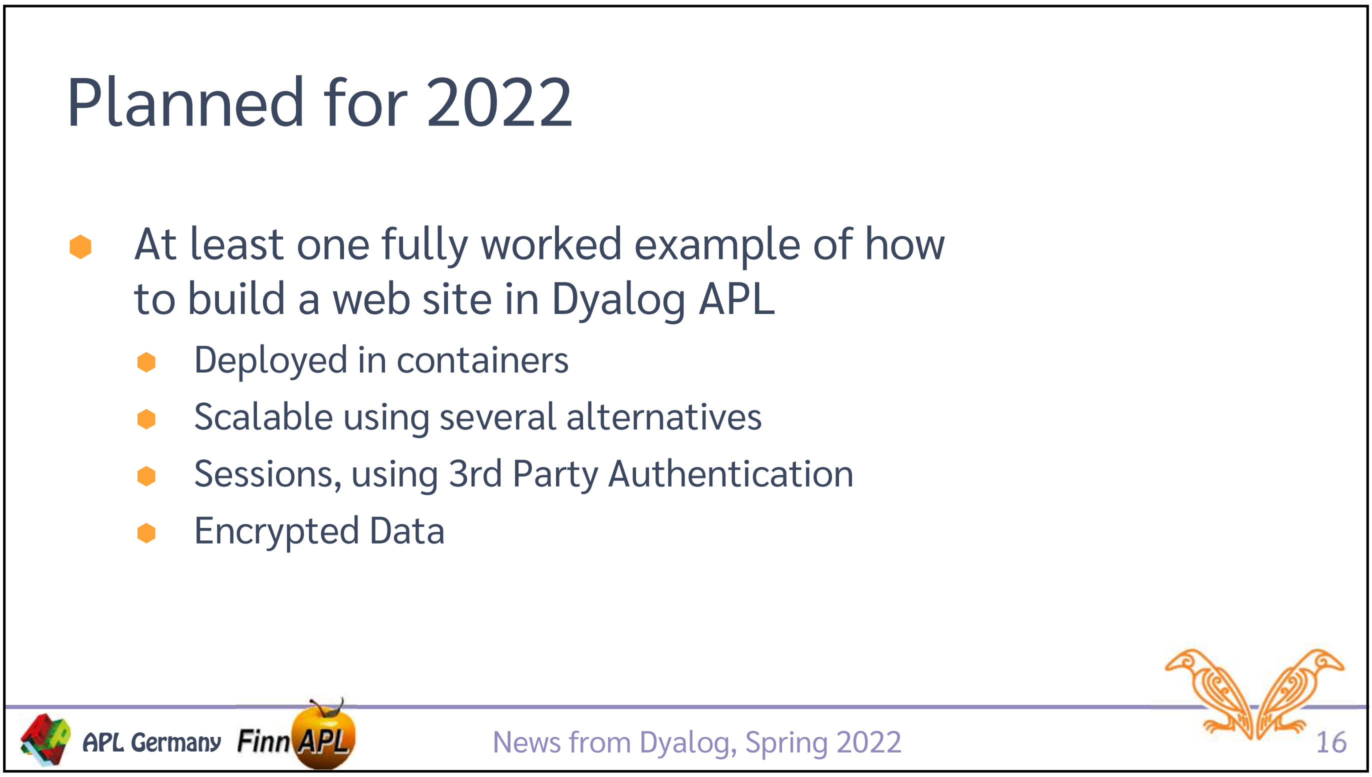
You can create a README file or pin repositories visible to anyone.

People

Top languages

● APL ● JavaScript ● Shell ● C

15

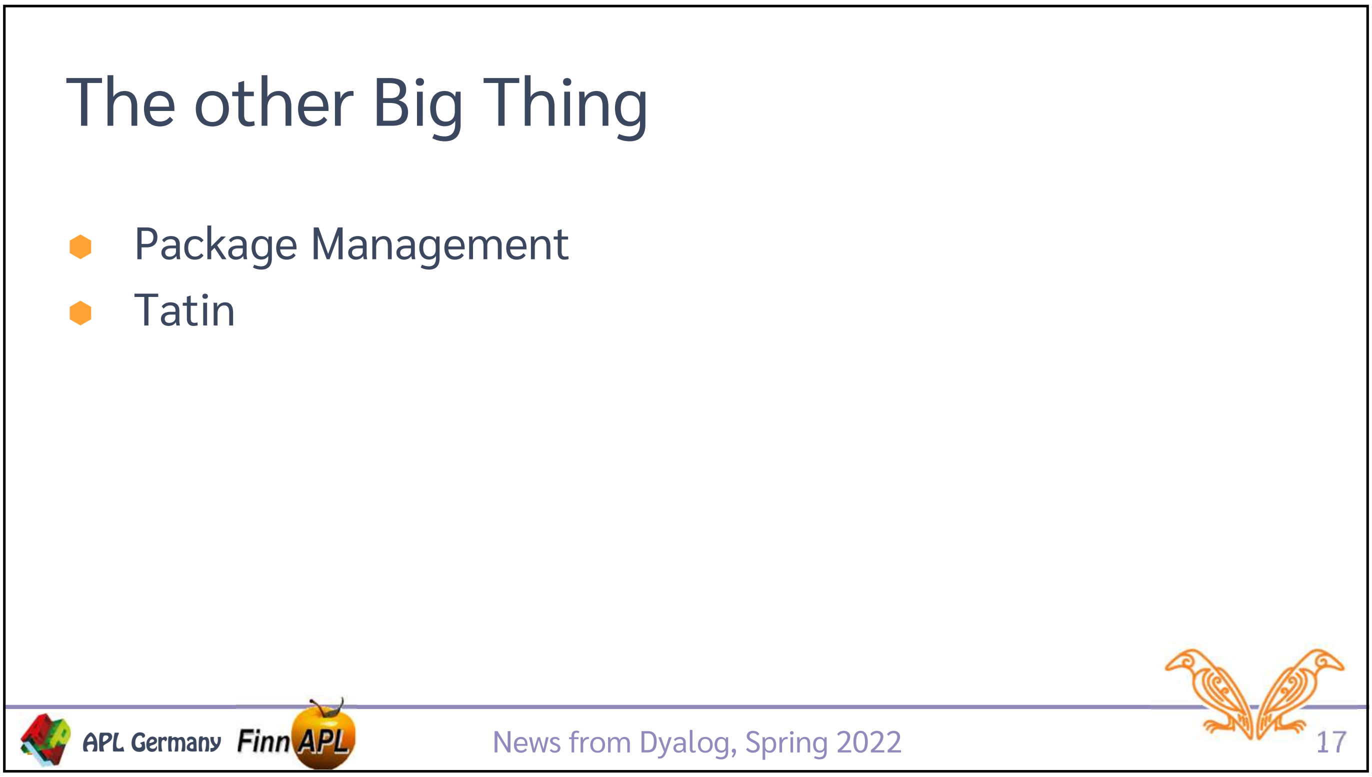


Planned for 2022

- At least one fully worked example of how to build a web site in Dyalog APL
  - Deployed in containers
  - Scalable using several alternatives
  - Sessions, using 3rd Party Authentication
  - Encrypted Data

APL Germany FinnAPL News from Dyalog, Spring 2022 16

16



The other Big Thing

- Package Management
- Tatin

APL Germany FinnAPL News from Dyalog, Spring 2022 17

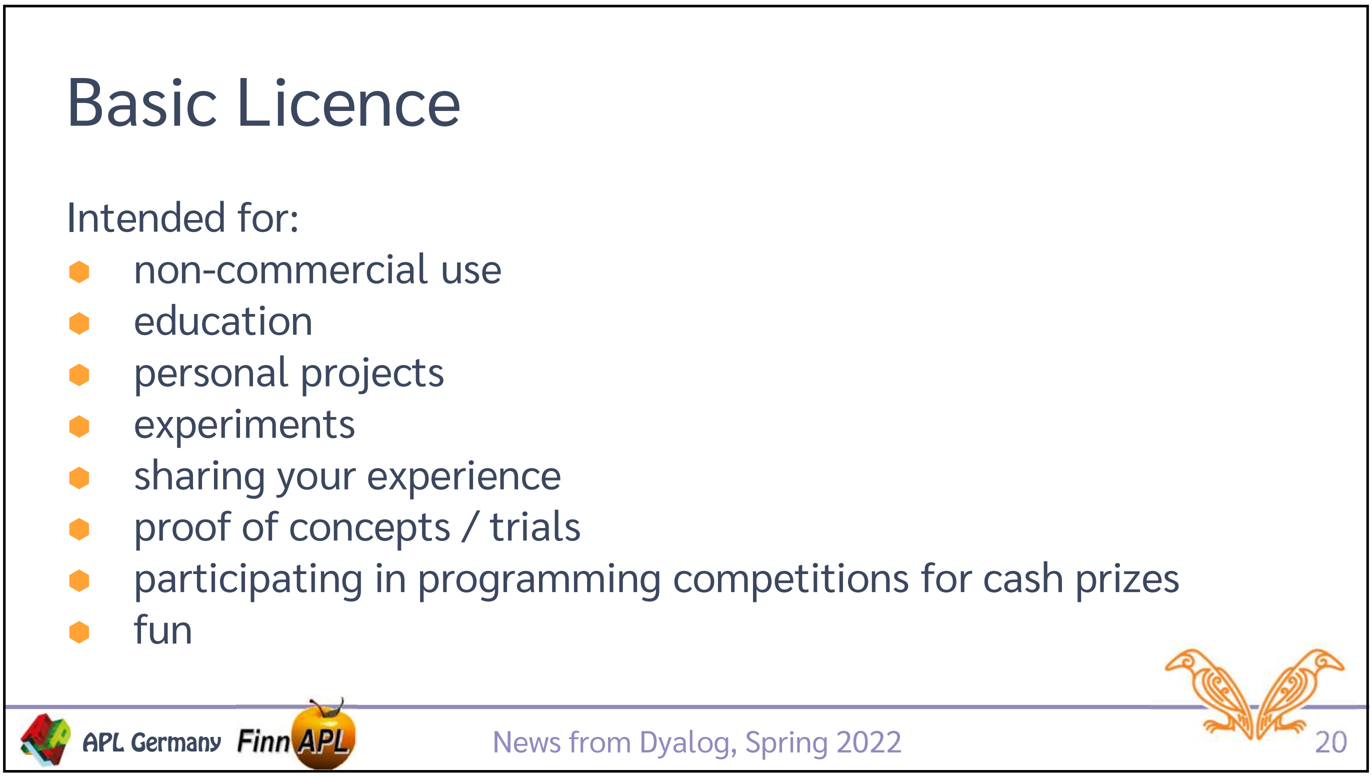
17

The APL Competition

- https://www.dyalog.com/student-competition.htm

New Basic License

- From Dyalog v18.2, the Non-commercial licence is replaced by a Basic licence
- A Basic Licence is a **free licence** that allows APL users to have a copy of the latest Dyalog technology **for personal or non-commercial use** and experimentation.
- Allows **distribution of Dyalog along with your work** under the terms of the Royalty-Based Run-Time Licence, which will apply as the default run-time licence.
  - Fee is 2% of gross APL-based revenue
  - No fee if revenue < GBP 5,000 in a calendar year
  - Multiple alternative commercial license schemes are available
- For GBP 150 per year, you can subscribe to the Dyalog Support Service (DSS)

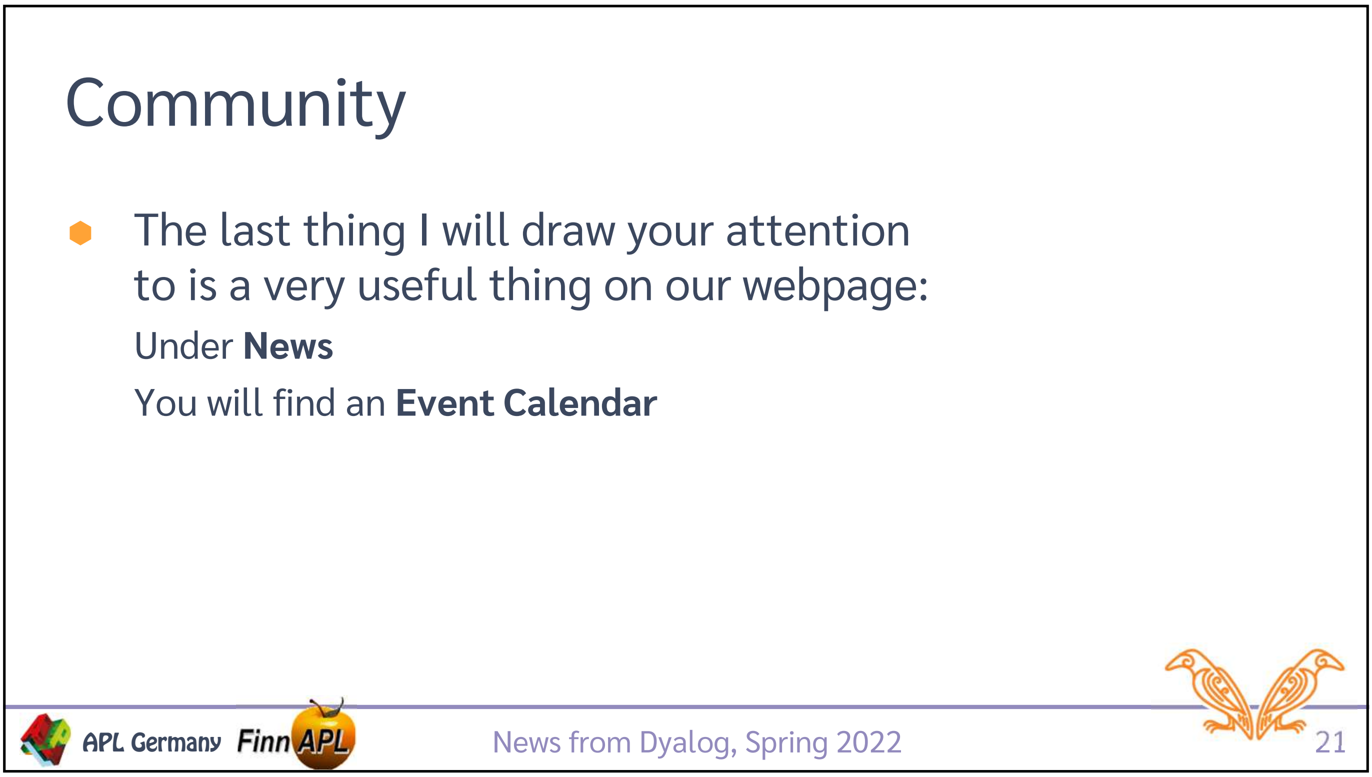


Basic Licence

Intended for:

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

20



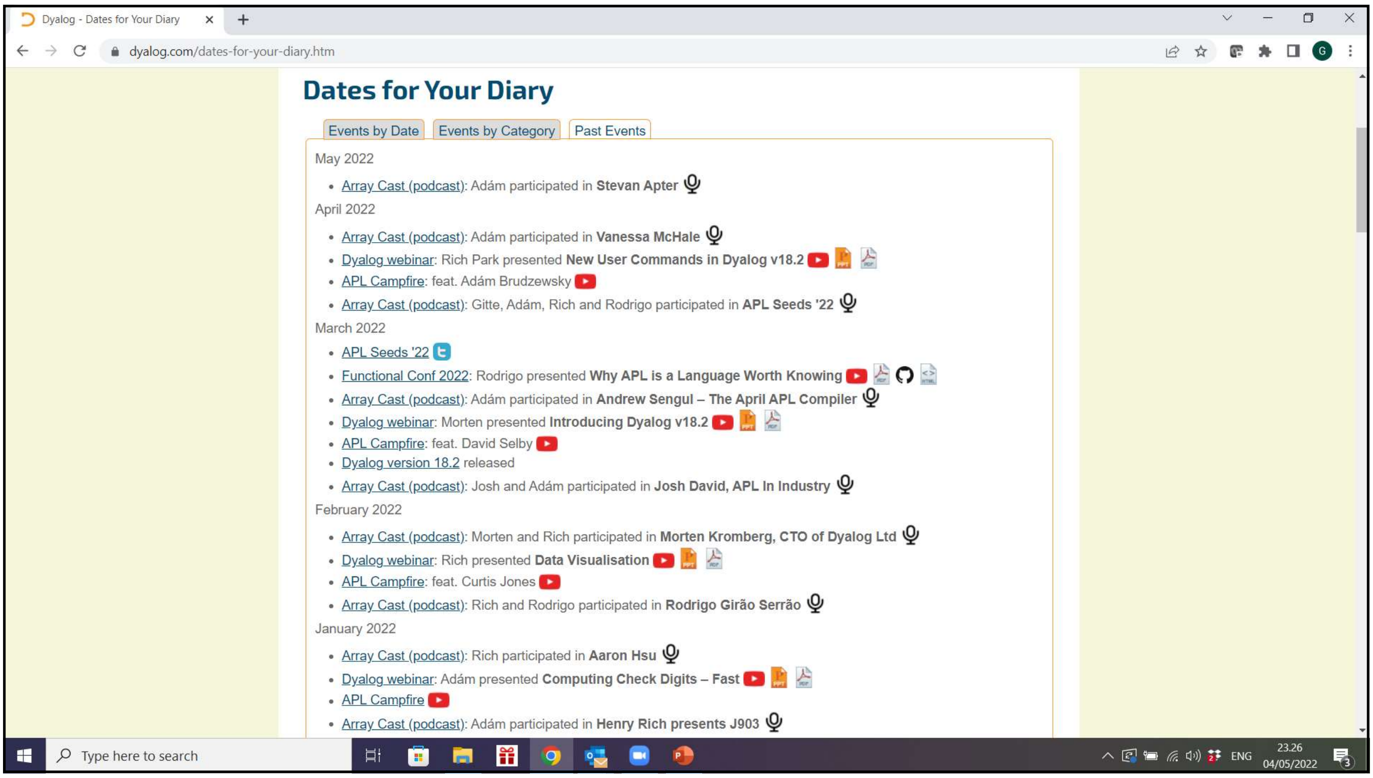
Community

- The last thing I will draw your attention to is a very useful thing on our webpage:

  Under **News**

  You will find an **Event Calendar**

21


Dyalog - Dates for Your Diary
dyalog.com/dates-for-your-diary.htm
Type here to search
ENG
23:26
04/05/2022

Dates for Your Diary

Events by Date | Events by Category | Past Events

May 2022

- Array Cast (podcast): Adám participated in **Stevan Apter**

April 2022

- Array Cast (podcast): Adám participated in **Vanessa McHale**
- Dyalog webinar: Rich Park presented **New User Commands in Dyalog v18.2**
- APL Campfire: feat. Adám Brudzewsky
- Array Cast (podcast): Gitte, Adám, Rich and Rodrigo participated in **APL Seeds '22**

March 2022

- APL Seeds '22
- Functional Conf 2022: Rodrigo presented **Why APL is a Language Worth Knowing**
- Array Cast (podcast): Adám participated in **Andrew Sengul – The April APL Compiler**
- Dyalog webinar: Morten presented **Introducing Dyalog v18.2**
- APL Campfire: feat. David Selby
- Dyalog version 18.2 released
- Array Cast (podcast): Josh and Adám participated in **Josh David, APL In Industry**

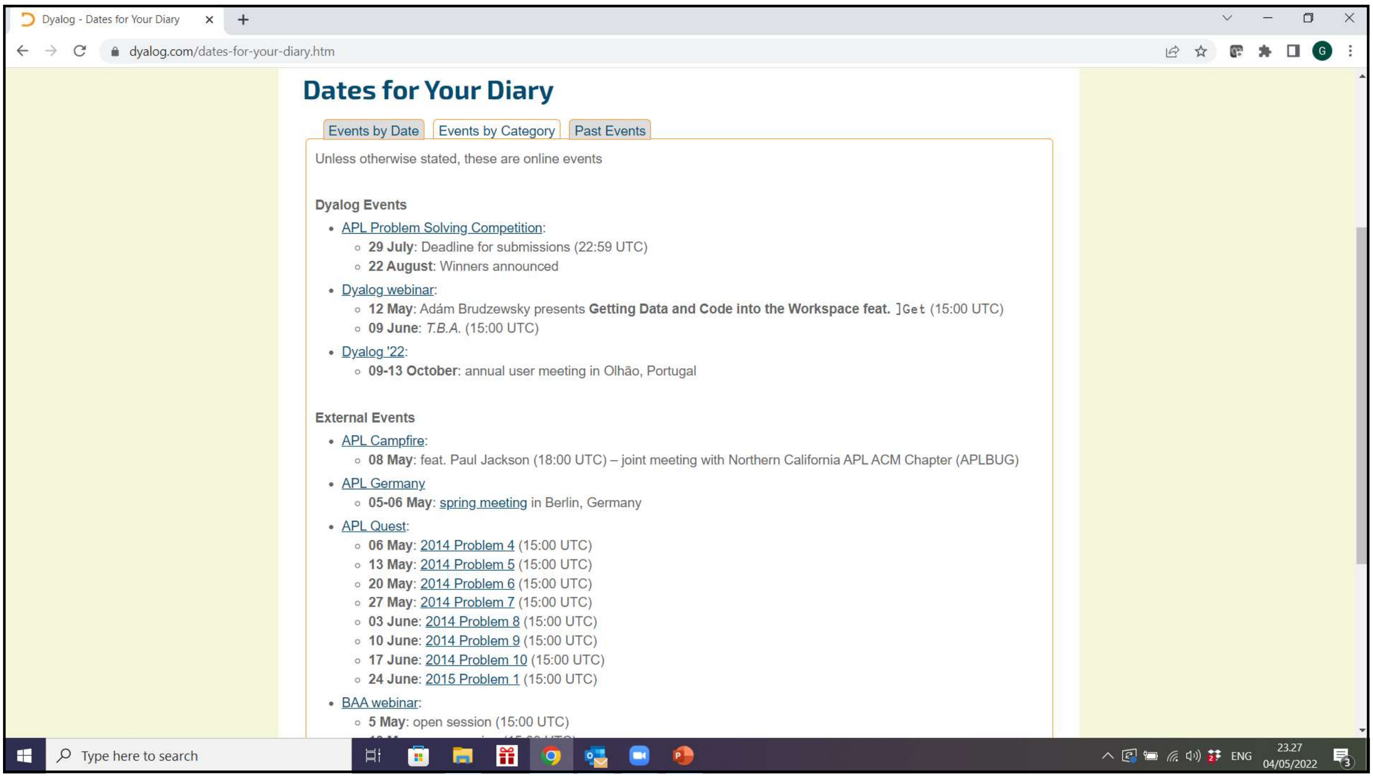
February 2022

- Array Cast (podcast): Morten and Rich participated in **Morten Kromberg, CTO of Dyalog Ltd**
- Dyalog webinar: Rich presented **Data Visualisation**
- APL Campfire: feat. Curtis Jones
- Array Cast (podcast): Rich and Rodrigo participated in **Rodrigo Girão Serrão**

January 2022

- Array Cast (podcast): Rich participated in **Aaron Hsu**
- Dyalog webinar: Adám presented **Computing Check Digits – Fast**
- APL Campfire
- Array Cast (podcast): Adám participated in **Henry Rich presents J903**

22


Dyalog - Dates for Your Diary
dyalog.com/dates-for-your-diary.htm
Type here to search
ENG
23:27
04/05/2022

Dates for Your Diary

Events by Date | Events by Category | Past Events

Unless otherwise stated, these are online events

Dyalog Events

- APL Problem Solving Competition:
  - **29 July**: Deadline for submissions (22:59 UTC)
  - **22 August**: Winners announced
- Dyalog webinar:
  - **12 May**: Adám Brudzewsky presents **Getting Data and Code into the Workspace feat. `]Get`** (15:00 UTC)
  - **09 June**: *T.B.A.* (15:00 UTC)
- Dyalog '22:
  - **09-13 October**: annual user meeting in Olhão, Portugal

External Events

- APL Campfire:
  - **08 May**: feat. Paul Jackson (18:00 UTC) – joint meeting with Northern California APL ACM Chapter (APLBUG)
- APL Germany
  - **05-06 May**: spring meeting in Berlin, Germany
- APL Quest:
  - **06 May**: 2014 Problem 4 (15:00 UTC)
  - **13 May**: 2014 Problem 5 (15:00 UTC)
  - **20 May**: 2014 Problem 6 (15:00 UTC)
  - **27 May**: 2014 Problem 7 (15:00 UTC)
  - **03 June**: 2014 Problem 8 (15:00 UTC)
  - **10 June**: 2014 Problem 9 (15:00 UTC)
  - **17 June**: 2014 Problem 10 (15:00 UTC)
  - **24 June**: 2015 Problem 1 (15:00 UTC)
- BAA webinar:
  - **5 May**: open session (15:00 UTC)

23

Dyalog ’22, Olhão

Dyalog ’22, Olhão

Hope to

See you all there!!

Questions?

26