

1. Title: **Introduction to Julia**
2. Overview

What is Julia

3. Goal: SciComp & Prototyping a) **high performance** b) **high-level** c) **Often C/C++**
4. Combine both: Julia a) **3 main features** b) **history**
5. Resources

Installation & REPL

6. Installation
7. REPL
Have a first, empty terminal ready to start julia in
a) **name & examples (copy a few)** b) **shortcuts**
8. Modes a) **? (illustrate with sqrt)** b) **| c) ;**

Main features

9. General Philosophy & format a) **Philosophy** b) **Format**
10. Prequel: Start a notebook
Prepare that second terminal and the browser with Pluto open upfront
a) **add** b) **using** c) **run** d) **or both together** e) **we continue**
11. Control Flow I: for & while a) **for 1** b) **for 2** c) **for 3**
d) **compr** e) **while**
12. Control Flow II: Conditionals a) **if** b) **lazy** c) **inline**
13. Defining functions a) **naming** b) **doc** c) **type** d) **return**
e) **shorter**

14. More on functions I: args&kwargs a) **pos** b) **kwargs** c) **pass on**
15. More on functions II: Broadcast & Mutation a) **first class & anonymous** b) **broadcast** c) **multiple** d) **modify** e) **naming**
16. Data structures a) **abstract** b) **naming** c) **struct** d) **is immutable** e) **but fields** f) **efficient (2 artificial breaks!)** g) **mutable** h) **can change** i) **less**
17. Parametric types a) **motivation** b) **II** c) **III** d) **def/clumsy** e) **nicer** f) **concrete vector** g) **own constructor**
18. Multiple Dispatch a) **best fitting** b) **Def** c) **we get** d) **most fitting** e) **multiple** f) **avoid** g) **resolve**
19. Operators are functions a) **Def** b) **then** c) **also parametric**
20. Functors a) **poly struct** b) **functor** c) **we get**
21. Python a) **end** b) **indent** c) **1** d) **string** e) **loops fast** f) **abstract arrays** g) **range** h) **imag** i) **matmul** j) **dispatch**
22. R a) **single quotes** b) **vectors** c) **ops on diff lengths** d) **assignment** e) **anon fct** f) **mat mul** g) **pass by ref** h) **abstract range** i) **no vectorize code** j) **indexing**
23. Matlab a) **arrays in []** b) **copy arrays** c) **modify** d) **1-dim** e) **int/float** f) **use broadcast** g) **no auto broadcast**

Packages

24. Namespaces & Modules a) **namespace** b) **use others** c) **export** d) **others with prefix** e) **! name clash** f) **default packages**
25. Installing & using Packages a) **Registry & Pkg** b) **default** c) **package mode again** d) **add** e) **installs** f) **resolves versions** g) **status** h) **update** i) **using**
26. Package environments a) **env defined** b) **default** c) **ac-**

tivate d) current folder, easy! e) reproducible f) project
g) manifest h) reproducible(!)

Pluto Notebooks

27. Pluto.jl a) intro b) cells & execute c) hide code d) live docs e) status f) start
28. Differences to Jupyter a) Script b) includable c) output d) git e) versions intern f) own env on start g) exact versions h) reproducible i) persistent state j) dependencies k) update all dependents l) global state m) no need to rememebr right order
29. Live Demo (create something with Makie and sliders?)
30. Further topics

workshop!