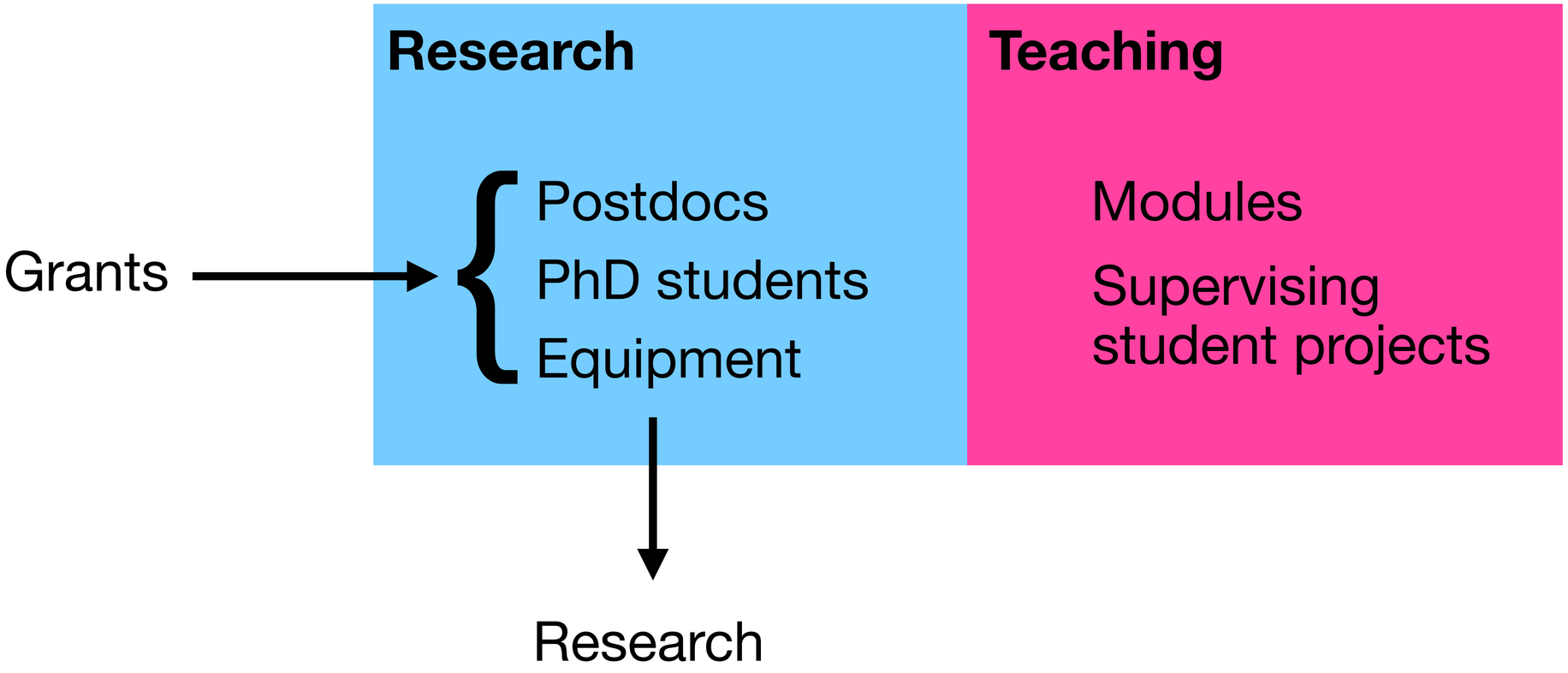
Learning how to think mathematically and computationally

Algorithmic approaches to mathematics
2025-2026

Dhruva V. Raman

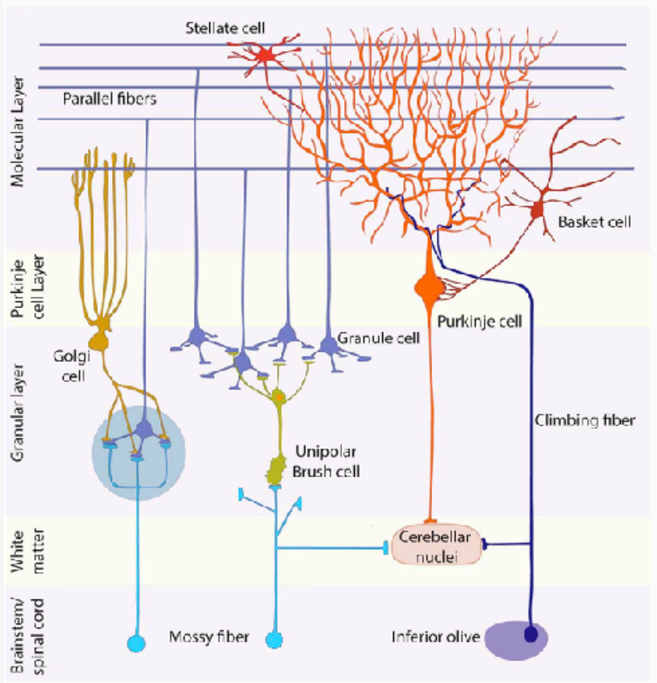
Who are we? A lecturer's life



Dhruva / ~~Sir~~ / ~~Professor~~

(Don’t worry about correct pronunciation)

Design principles in biological systems



What about this design makes this system good for function?

Can we better understand function from design?

Dhruva

XXX raises your risk of cancer by 20%!!!

How can we constructively break the conclusions of statistical models

“Lies, damned lies, and statistics”

Who are you?

Diverse academic backgrounds.
That’s ok!

Making a spectrum of friends will help you academically and personally

- Introduce yourself to neighbours each lecture
- Don’t sit in the same places each week!

Today

Why do this course?

How to do this course?

What should I know
outside this course?

Learning goal: how to digest maths

Learning goal: how to digest and use maths

21st century life

How to learn/communicate/use/think with mathematical concepts

✓ *By learning maths!*

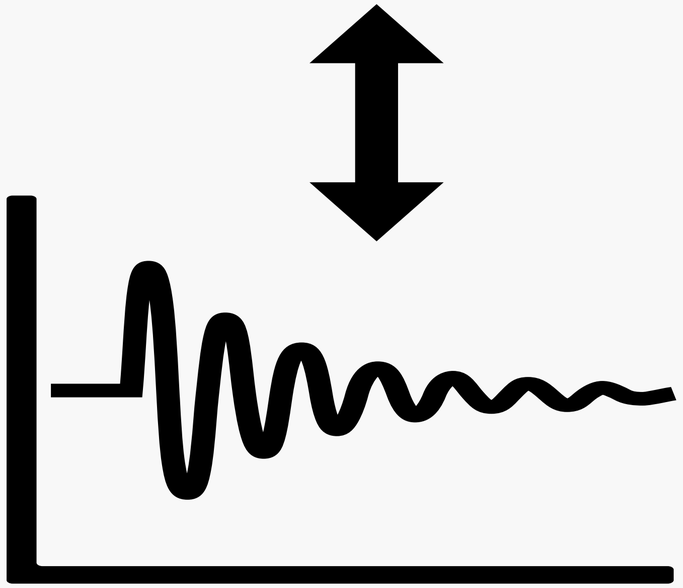
Learn maths all of it!

✗

Requirement 1: Language

Definition of a continuous function

$$\forall \epsilon \in \mathbb{R}^+ : \quad \exists \delta > 0 \text{ s.t.}$$
$$\|x - y\|_2 < \delta \Rightarrow \|f(x) - f(y)\|_2 < \epsilon$$



Mathematics

Reading/writing using LaTeX notation

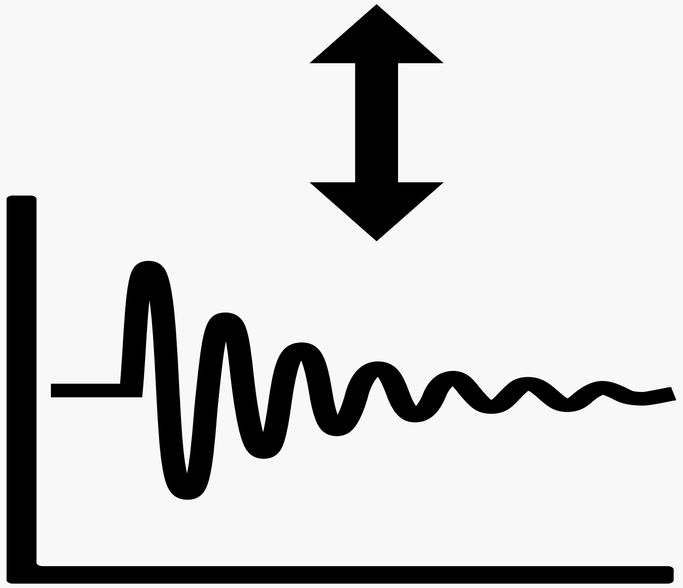
Appreciating/using good grammar

https://kapeli.com/cheat_sheets/LaTeX_Math_Symbols.docset/Contents/Resources/Documents/index

Requirement 1: Language

Definition of a continuous function

$$\forall \epsilon \in \mathbb{R}^+ : \quad \exists \delta > 0 \text{ s.t.}$$
$$\|x - y\|_2 < \delta \Rightarrow \|f(x) - f(y)\|_2 < \epsilon$$



Mathematics **+ programming**

Reading/writing using LaTeX notation

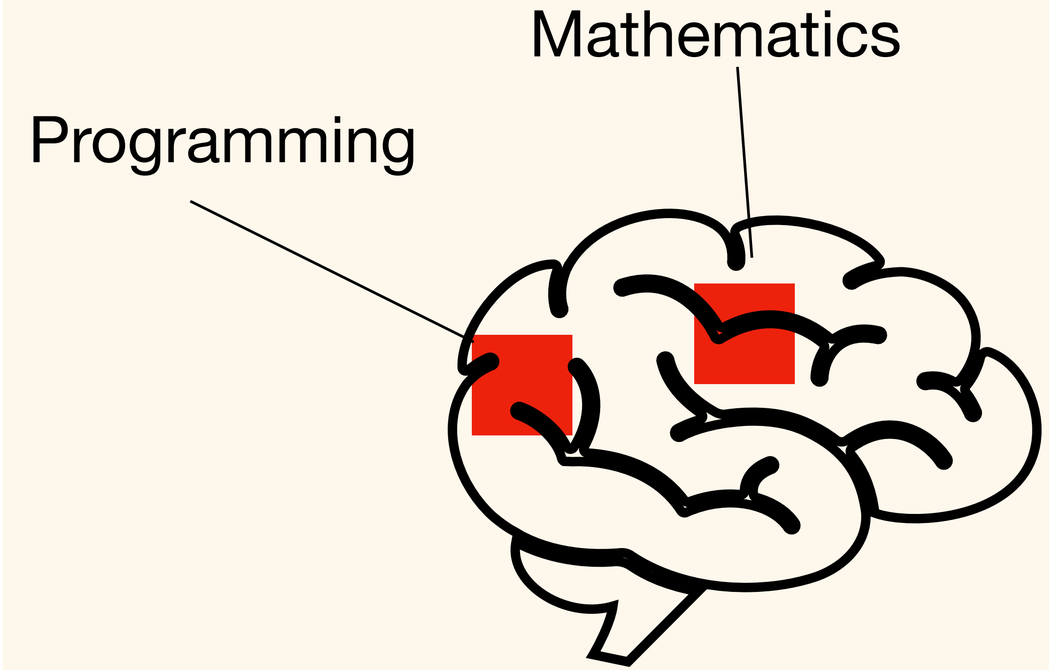
Appreciating/using good grammar

https://kapeli.com/cheat_sheets/LaTeX_Math_Symbols.docset/Contents/Resources/Documents/index

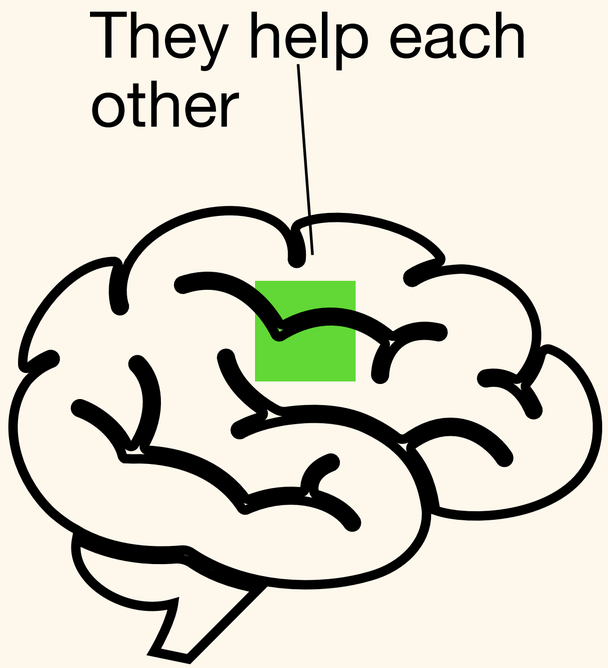
Requirement 2:

Unifying mathematical and computational thinking

Bad



Good



Requirement 3: unlearning

The confidence to experiment, play, distrust and ask questions!

(Bad) school -level understanding

Memorise formulae and recipes

Purpose is to pass exam

Show off knowledge when communicating

Requirement 3: unlearning

The confidence to experiment, play, distrust and ask questions!

(Bad) school -level understanding

Memorise formulae and recipes

Purpose is to pass exam

Show off knowledge when communicating

Understanding enough to use

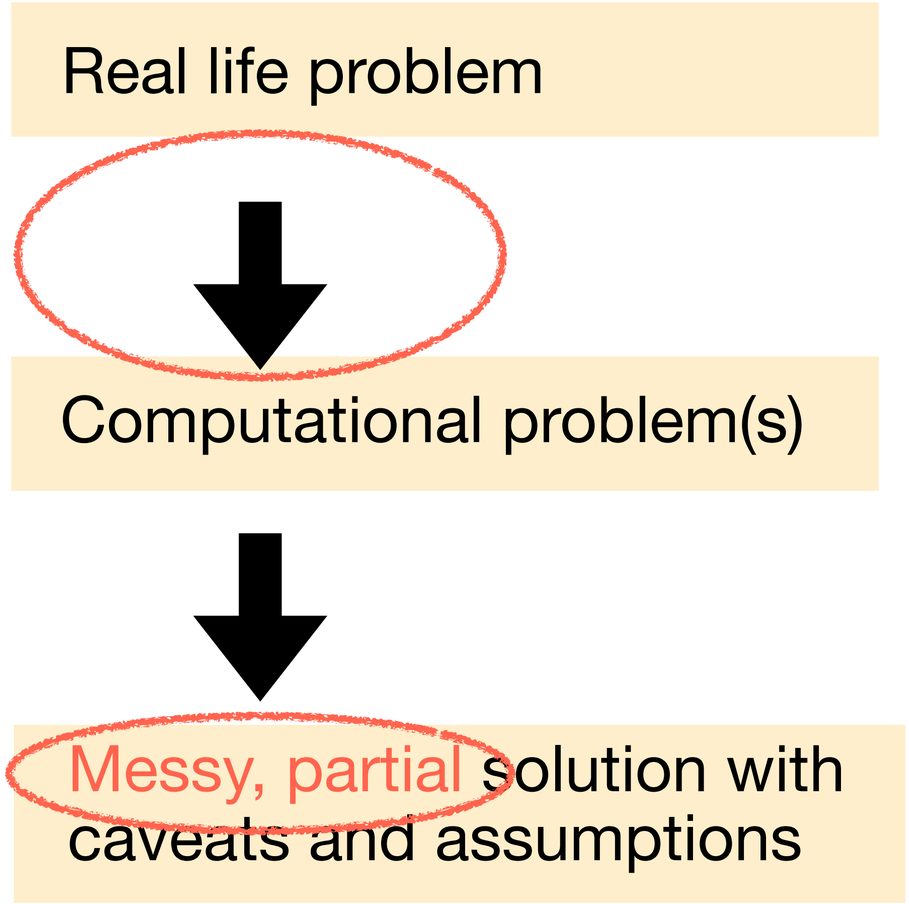
When is it wrong?
What's the point?

Can I use it for something else?
Can I break it?

Can I explain it simply?

Requirement 4:

Maths as a pragmatic tool for complex systems



Requirement 5

Surveying the landscape

I don’t know what things
I don’t know

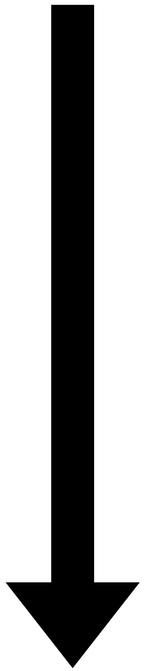
Mathematical tools to solve my problem?

Which tools?

Course structure

Fast survey of main useful topics in applied mathematics/engineering

Giving you the knowledge to teach yourself more advanced topics if/ when you need to



Scientific computing

Linear Algebra

Probability

Dynamical systems

Optimisation theory

Worried?

Look at the prerequisites on canvas

Basic, necessary school-level maths:
https://canvas.sussex.ac.uk/enroll/KYHLP9

Today

Why do this course?

How to do this course?

What should I know
outside this course?

This is not a spectator sport!!!!

Maths is not a spectator sport!!!

Attend all lectures

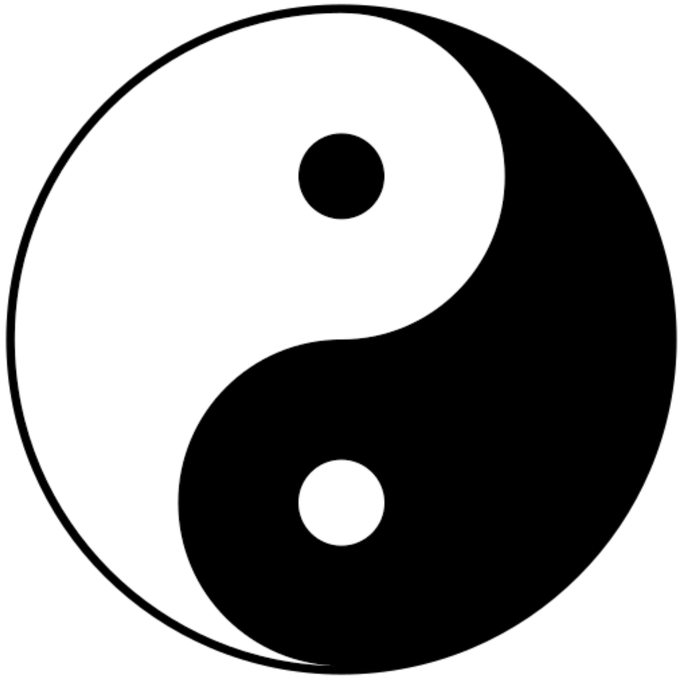
Read and understand all worksheets

Revise all concepts

} FAIL THE EXAM

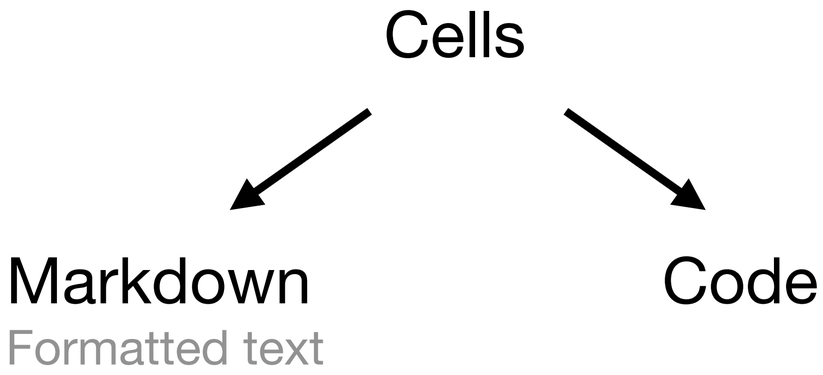
Course structure

Notebooks
that you fill in

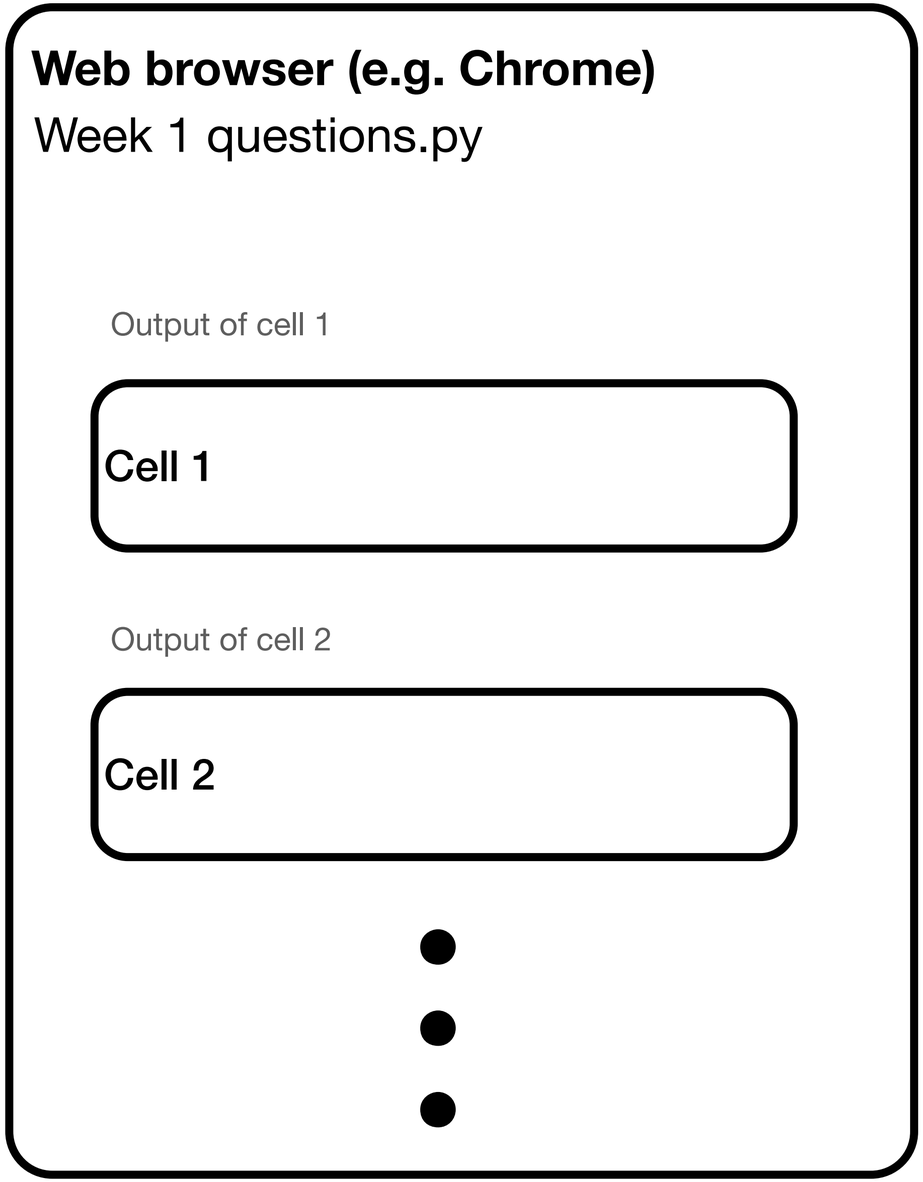


(Lectures)

A notebook is a personal ‘code’ diary



www.markdownguide.org/basic-syntax/



Parallel notebooks in Python and Julia

Python

Less mathematical background

Best for quickly learning the essential mathematical concepts

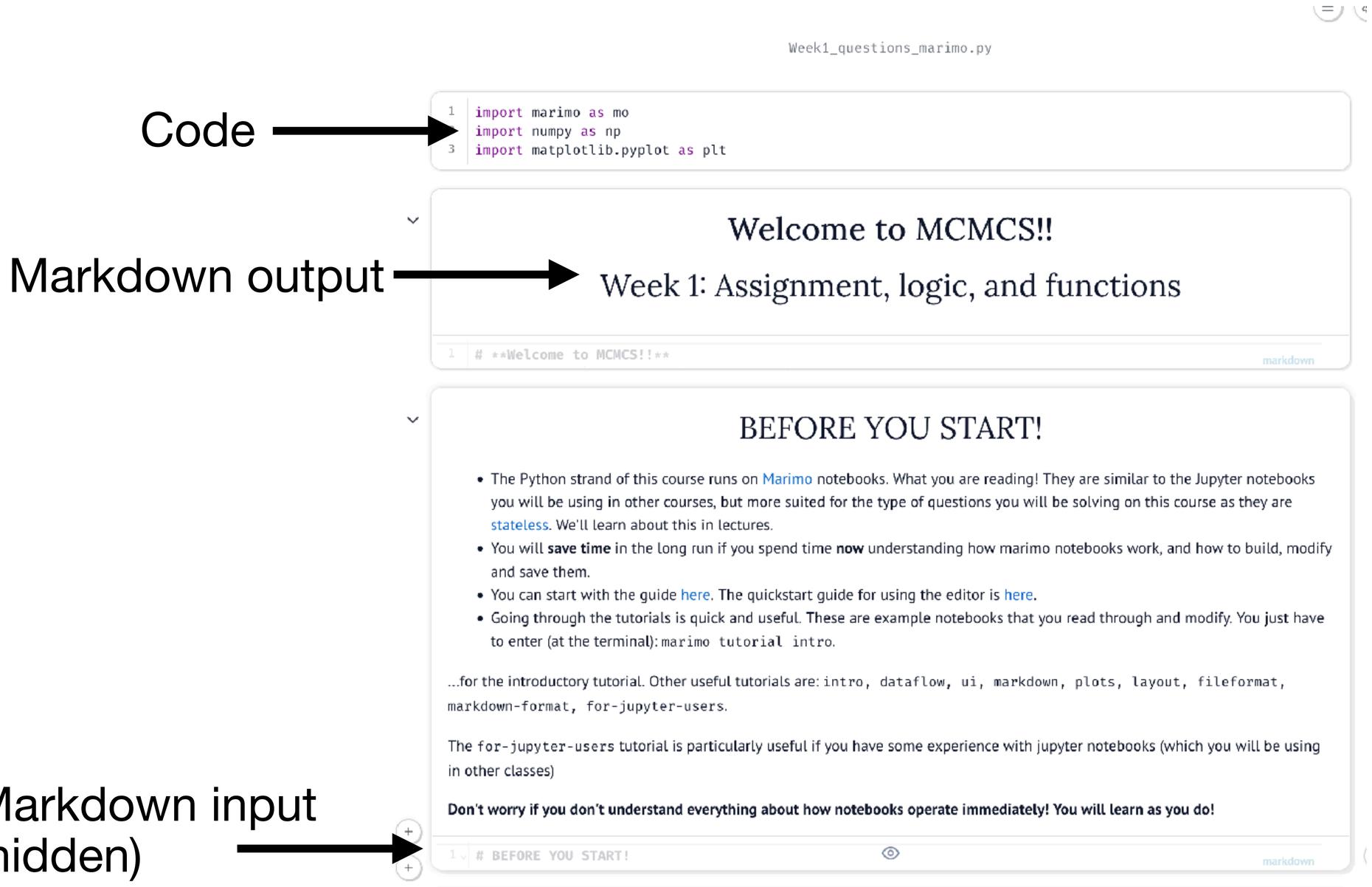
Julia

Mathematical background

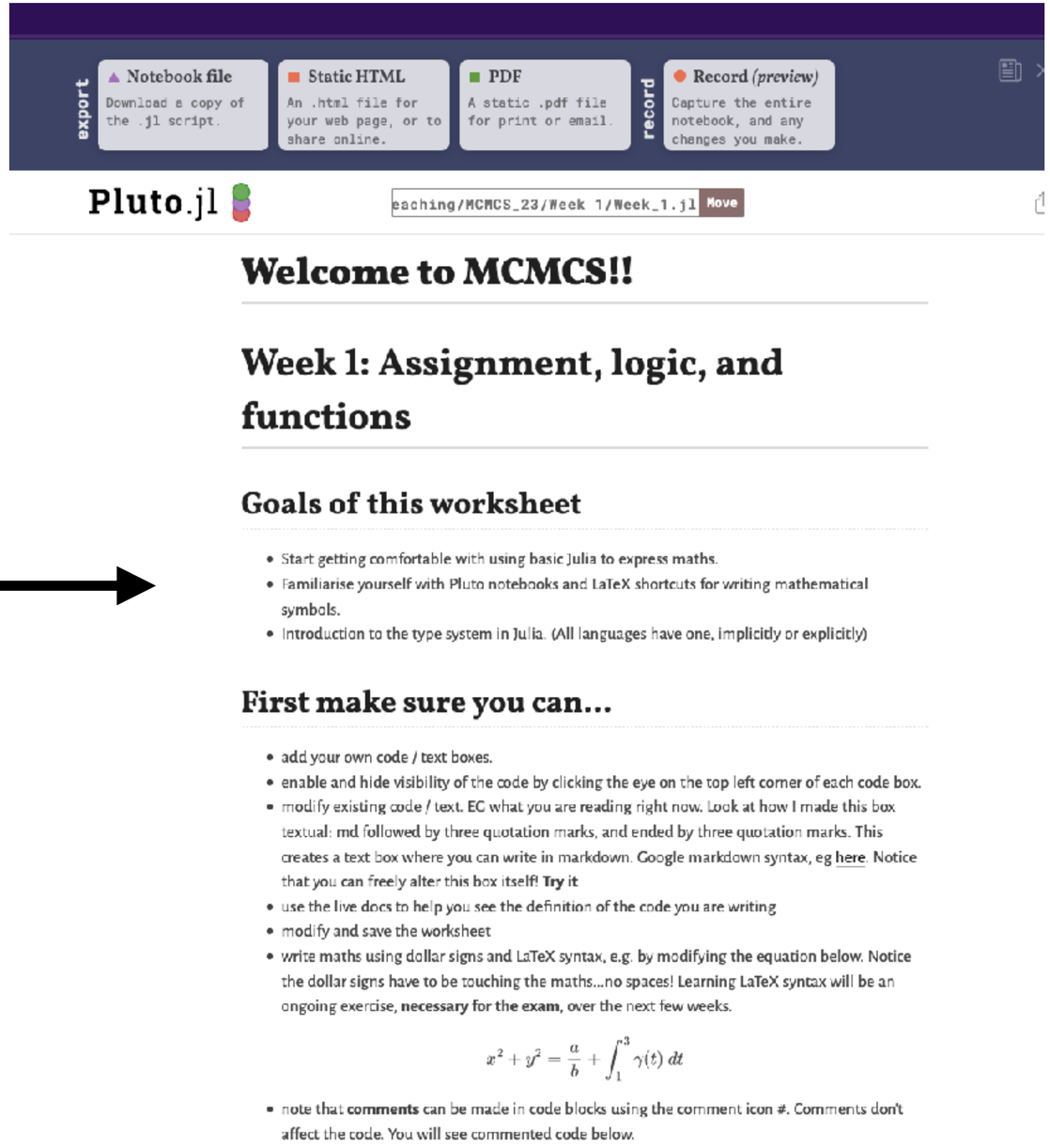
Best for improving programming and problem-solving

Similar to Python at this level

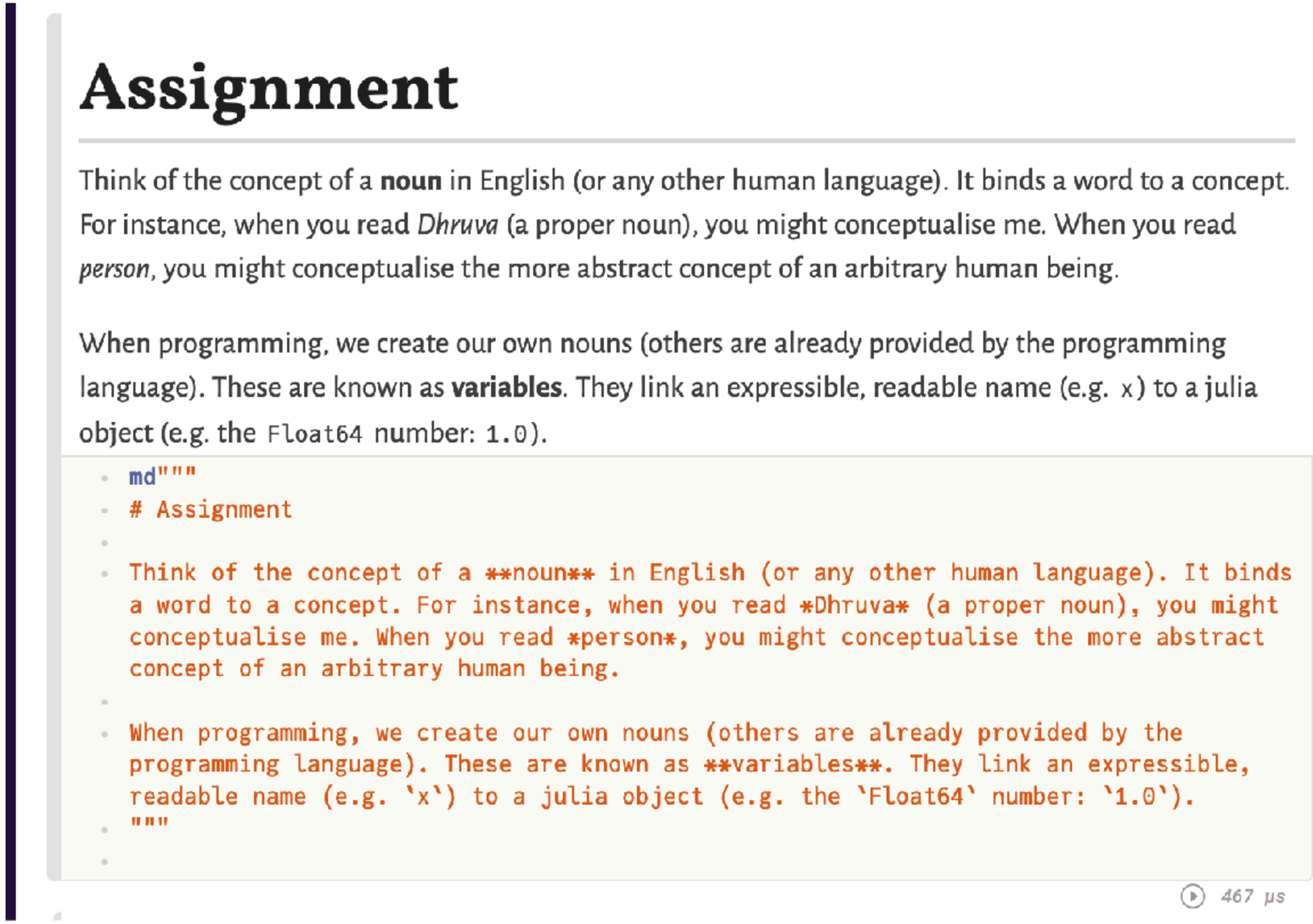
Bilingual is better


Week1_questions_marimo.py
Code
1 import marimo as mo
2 import numpy as np
3 import matplotlib.pyplot as plt
Markdown output
Welcome to MCMCS!!
Week 1: Assignment, logic, and functions
1 # **Welcome to MCMCS!!**
markdown
BEFORE YOU START!
• The Python strand of this course runs on Marimo notebooks. What you are reading! They are similar to the Jupyter notebooks you will be using in other courses, but more suited for the type of questions you will be solving on this course as they are stateless. We'll learn about this in lectures.
• You will save time in the long run if you spend time now understanding how marimo notebooks work, and how to build, modify and save them.
• You can start with the guide here. The quickstart guide for using the editor is here.
• Going through the tutorials is quick and useful. These are example notebooks that you read through and modify. You just have to enter (at the terminal): marimo tutorial intro.
...for the introductory tutorial. Other useful tutorials are: intro, dataflow, ui, markdown, plots, layout, fileformat, markdown-format, for-jupyter-users.
The for-jupyter-users tutorial is particularly useful if you have some experience with jupyter notebooks (which you will be using in other classes)
Don't worry if you don't understand everything about how notebooks operate immediately! You will learn as you do!
Markdown input (hidden)
1 # BEFORE YOU START!
markdown

Markdown output →

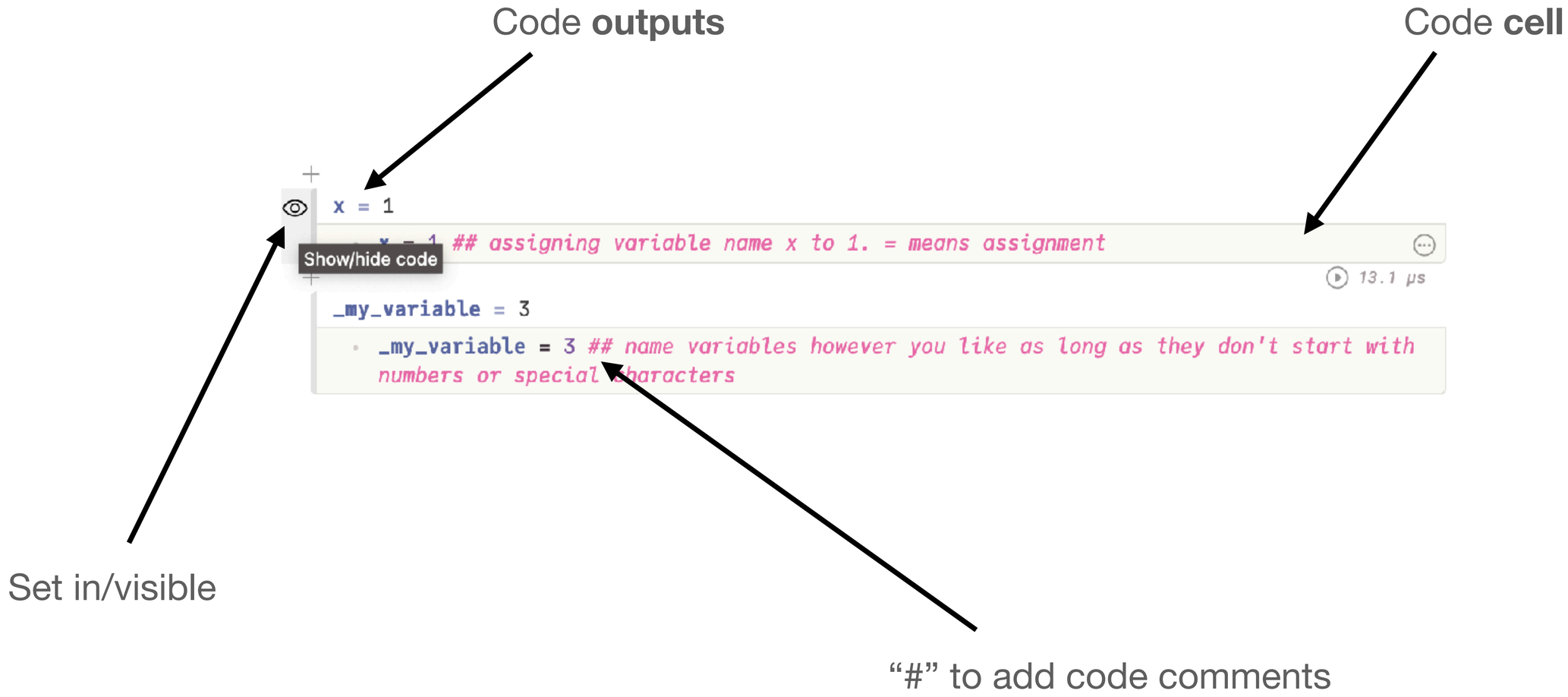


Markdown cells (Julia/Python)



www.markdownguide.org/basic-syntax/

Code blocks


Code **outputs**
Code **cell**
x = 1
x = 1 ## assigning variable name x to 1. = means assignment
Show/hide code
13.1 µs
_my_variable = 3
_my_variable = 3 ## name variables however you like as long as they don't start with numbers or special characters
Set in/visible
"#" to add code comments

Annotate and save your notebooks!

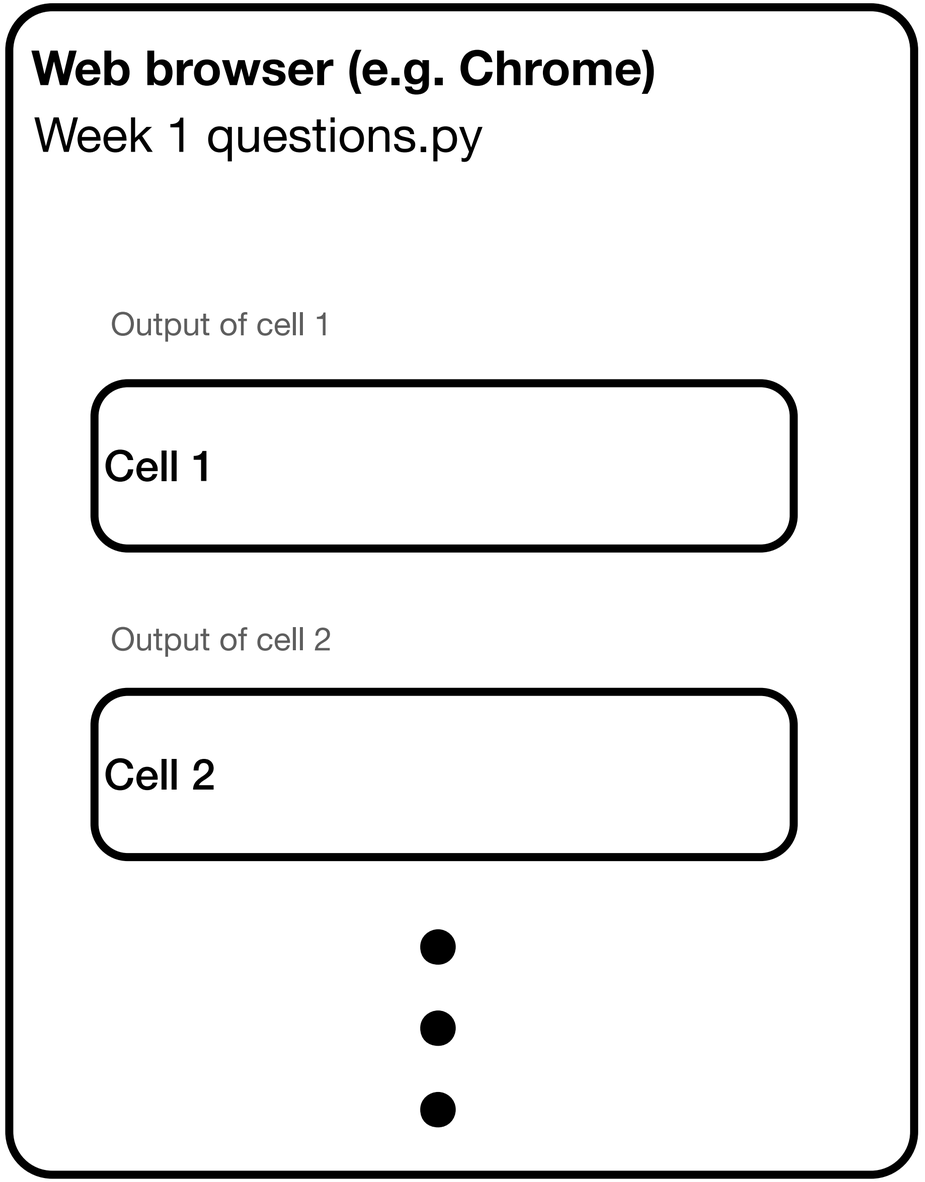
How?

Inspect/modify the cells I have written!

Why?

Answer provided questions

Make your own course notes



Why not Jupyter notebooks?

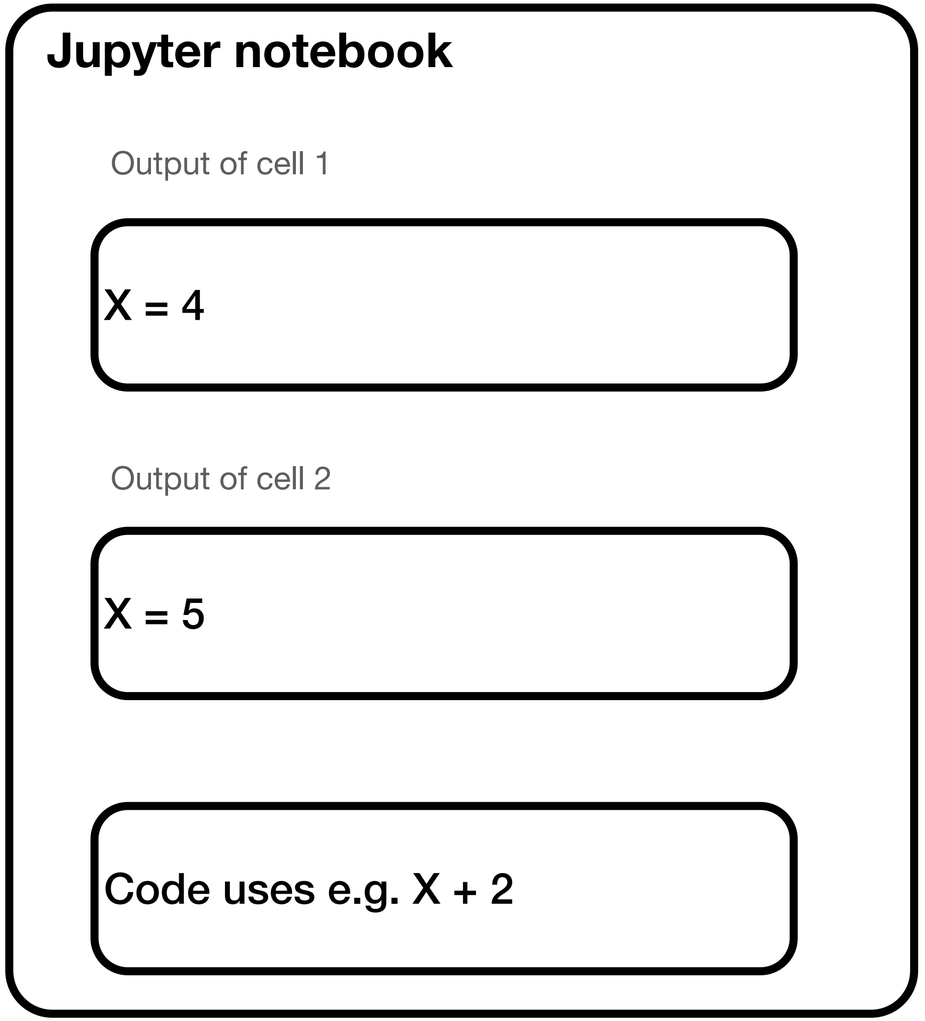
More elegant?

Force you to code in a different style

Better for interactive graphs and data analysis

Fewer bugs

Why not Jupyter notebooks?



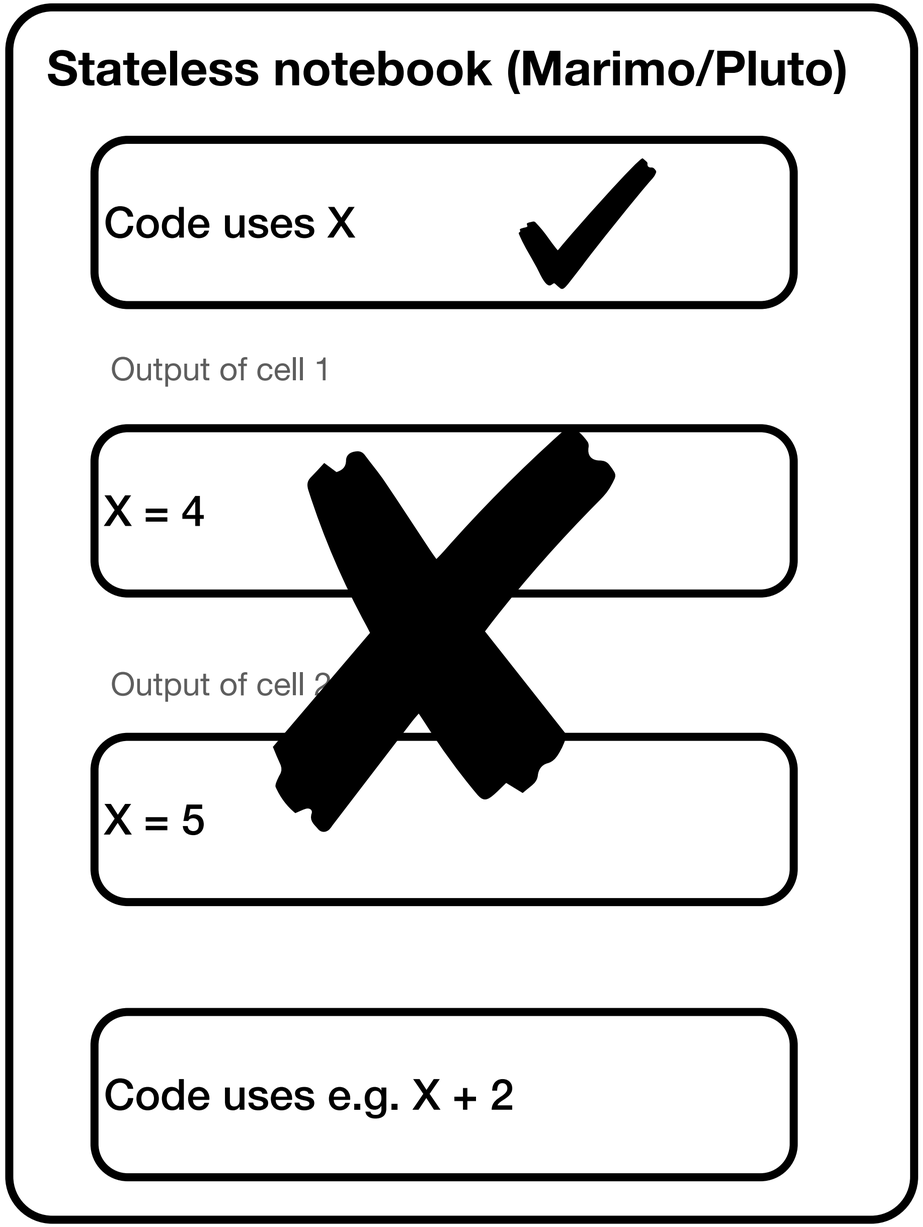
Why not Jupyter notebooks?

Stateless notebook:

No top-to-bottom order

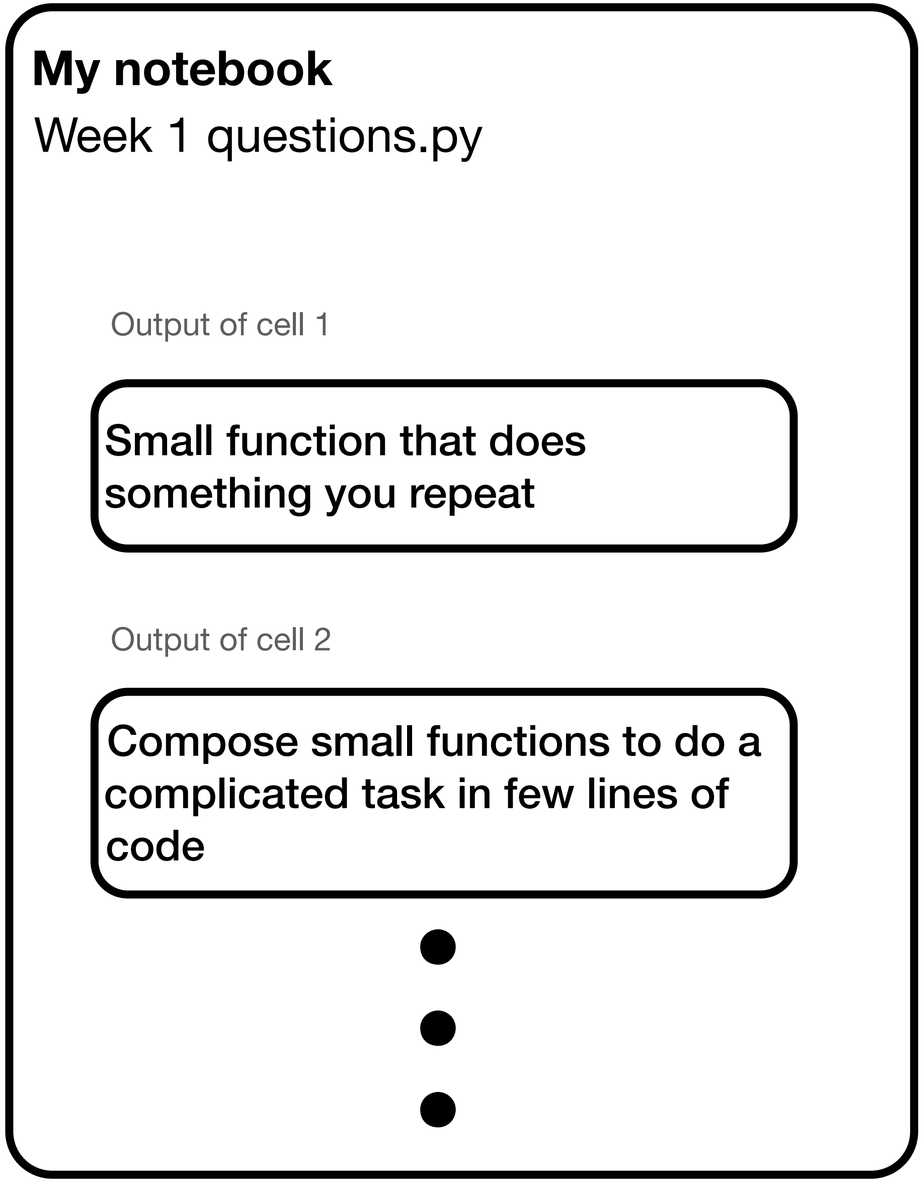
No ambiguity about variable values

Dependent cells refresh



Why not Jupyter notebooks?

Force you to code in a different style



Writing LaTeX in markdown

This is a nonsensical mathematical expression

$$\int_0^T \frac{a(t)}{b(t)} \mathrm{d} t \in \mathbb{R}$$

- []e

```
md"""

##### This is a nonsensical mathematical expression

$$\int_{0}^T \frac{a(t)}{b(t)} \mathrm{d} t \in \mathbb{R}$$


"""
```

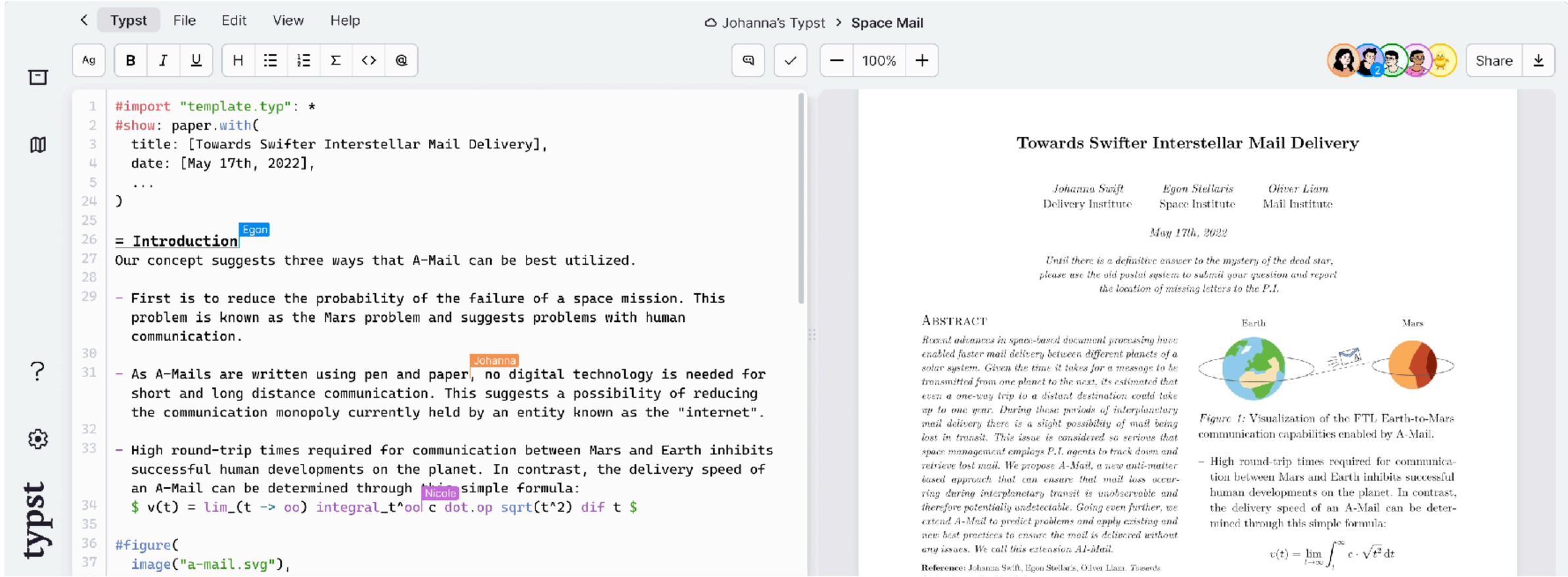
322 μs

https://kapeli.com/cheat_sheets/LaTeX_Math_Symbols.docset/Contents/Resources/Documents/index

https://www.overleaf.com/

(https://typst.app/)

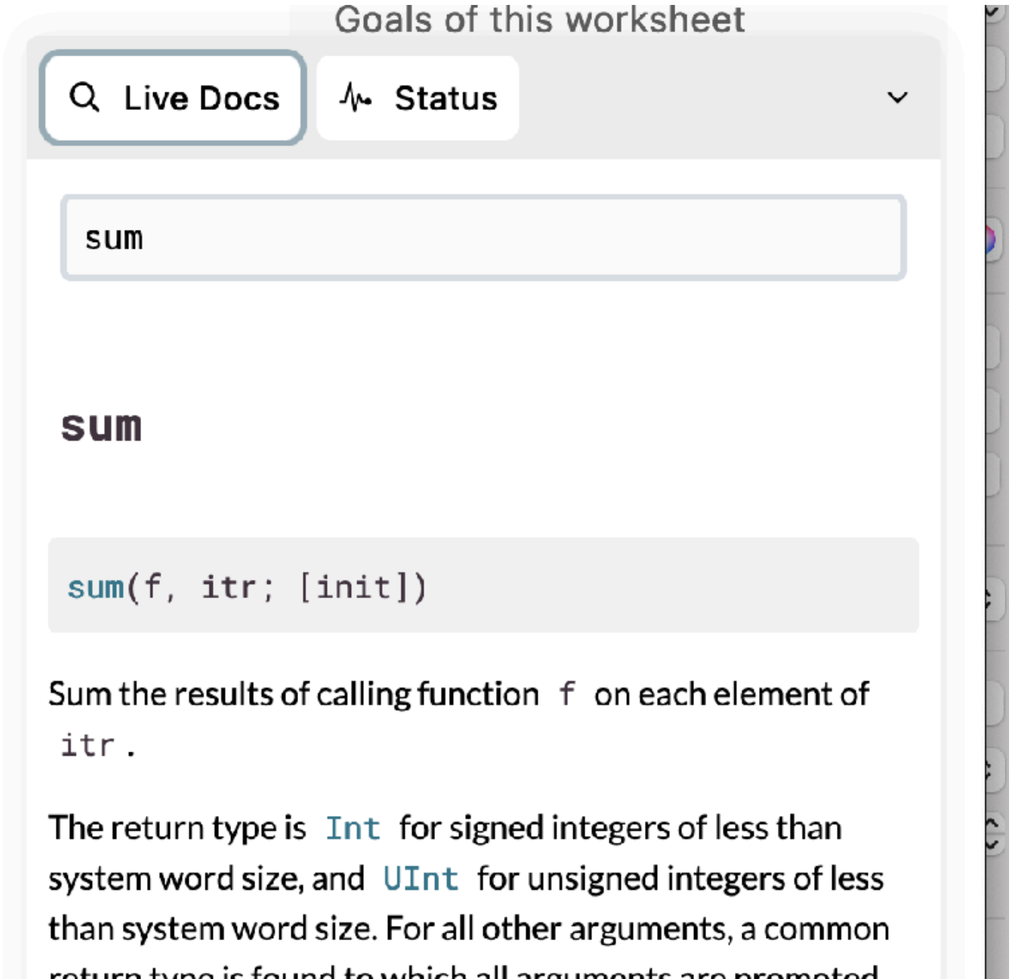
….or download locally



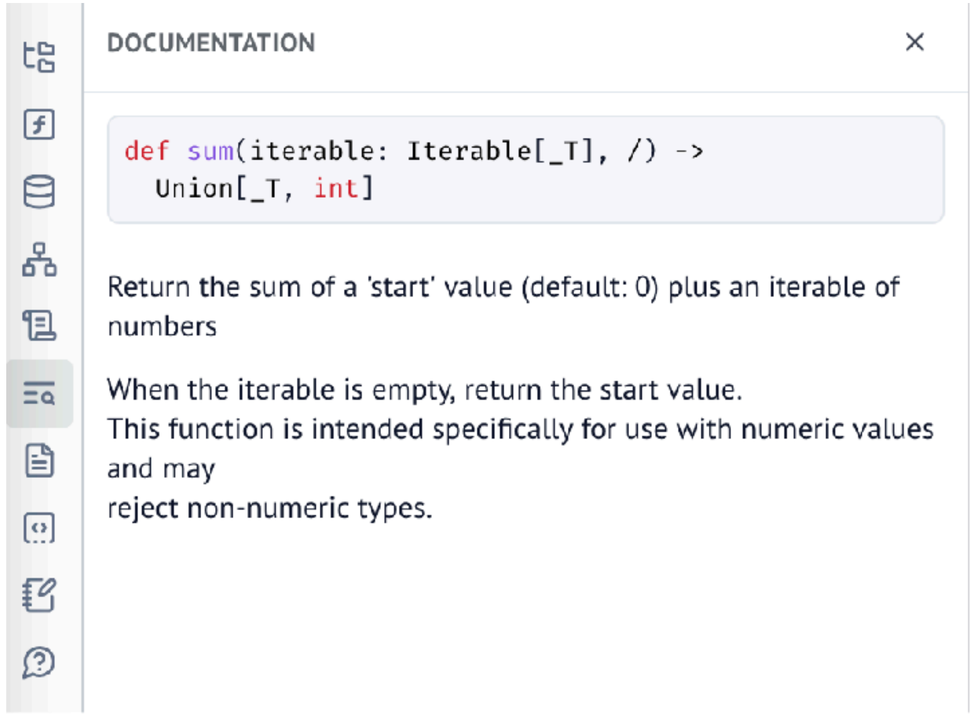
Use the live docs

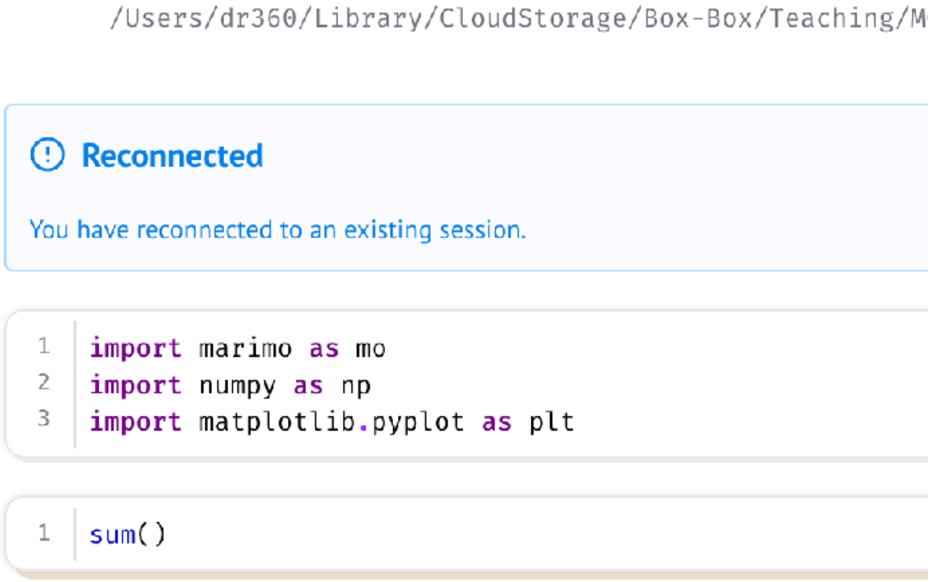
```
sum()
```

67.0 µs



Use the live docs





https://youtube.com/watch?v=Hgd2F2QNfEE

Repeat before each lab

This is my notebook. There are many like it,
but this one is mine

My notebook is my best friend. It is my life.
I must master it as I must master my life.

Without me, my notebook is useless.
Without my notebook, I am useless

Course etiquette

Start the notebooks early

Annotate the notebooks massively

Don’t give up!!!

Course etiquette

Use the padlet:

Attend labs
(attendance will be recorded)

Missing the labs?

Send excuses **during missed lab**

(Timed email)

Relaxed, creative, perseverance

Ironclad ego

Spend time on a question without worrying

Ask naive questions

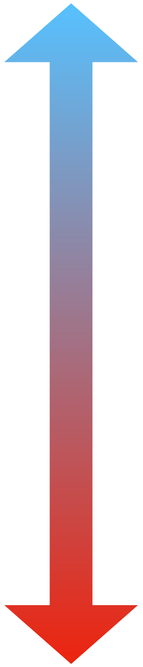
Playfulness

Maths is a game! Take breaks

Don’t get in a hole: change tactics

How to collaborate and use the answers

Using answers when completely stuck



Sharing approaches and perspectives

Using answers when mildly bored

Blindly copying

Doing it wrong?

The only person who gets hurt is you

Asking questions

During/after the lecture

During seminars

Summary

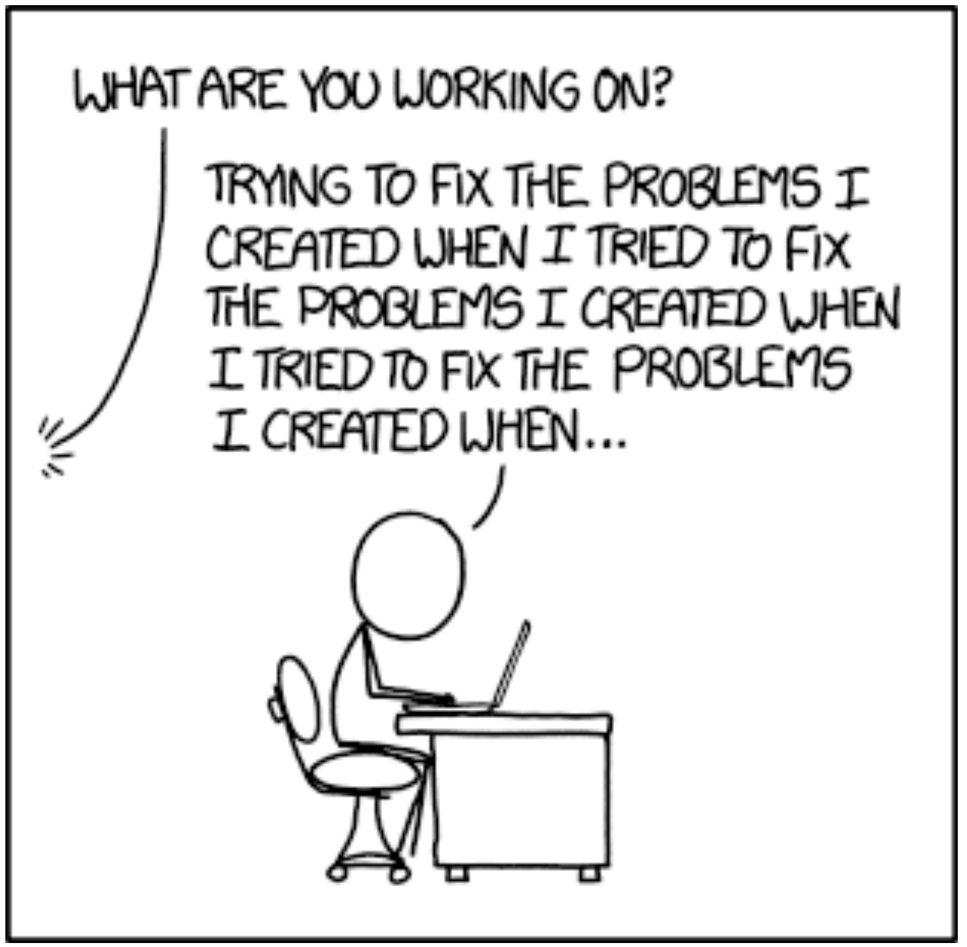
You learn when you're patient
and when you have fun

Do the notebooks
conscientiously

Ask questions and
talk to colleagues

Be proactive!!

The most important part of programming

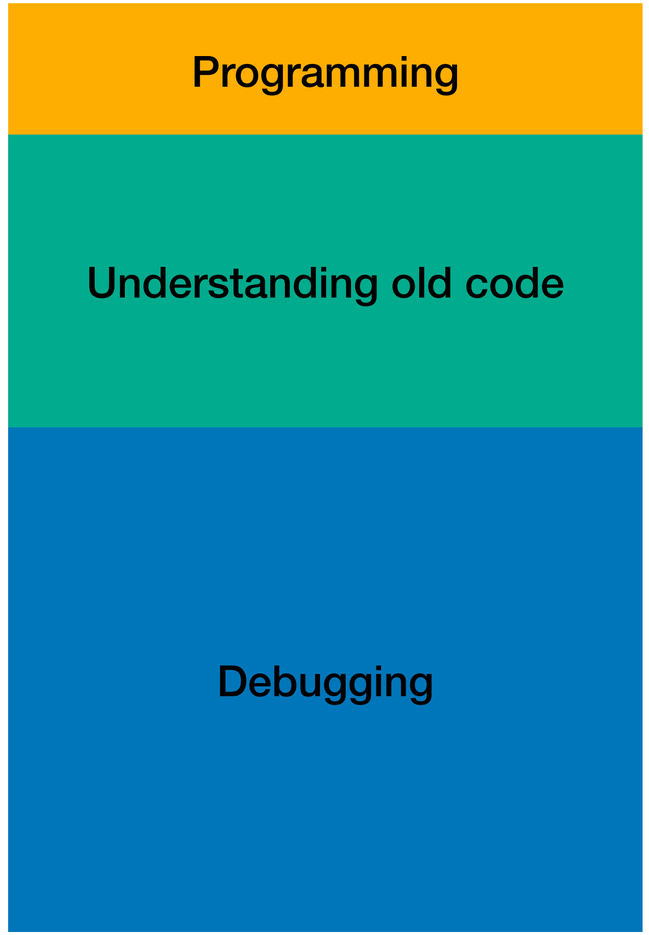


Debugging

Good debugger

>

Good programmer



Comment all code extensively

Future you will be thankful

Debugging

Wikipedia
The Free Encyclopedia

Search Wikipedia | Search

Rubber duck debugging

Article | Talk

From Wikipedia, the free encyclopedia

Contents [hide]

(Top)

The Duck House Brighton
https://www.theduckhousebrighton.online

Home | The Duck House Brighton | Rubber Duck Shop

Welcome to The Duck House! Here you will find over 400 different types of imaginative **rubber ducks**. There's one for everyone! Browse our website or come and ...

All Ducks
There's a Rubber Duck for Everyone! Black Facebook Icon ...

Most Popular
There's a Rubber Duck for Everyone! · DC- Catwoman ...

About Us
At The Duck House you will find over four hundred different ...

Special Occasions
There's a Rubber Duck for Everyone! · Love Eco Rubber ...

More results from theduckhousebrighton.online »

Good programming is a craft

Good programmers are always in high demand

Write code you're proud of

Continuously improve style

Use the padlet to ask questions

https://uofsussex.padlet.org/draman2/questions-board-to0eux5irituizt4

And to make lecture requests!

Make sure you've read the canvas/course website info!

Now…

Go through the Marimo learn notebooks
if you're unfamiliar with Python

https://marimo-team.github.io/learn/

Start familiarising with mathematical
notation on the cheatsheet

Then get working on notebook 1
(Python / Julia)

Today

Why do this course?

How to do this course?

What should I know outside this course?

I’m jealous of my friend

Simple tools that work together.
Advanced furniture-making + DIY

Handiness
Literacy with tools allows for creativity, nonstandard solutions and fixing things

Mastery is enjoyable

Better quality more bespoke than IKEA / builders but takes ages :(

My friend is jealous of me

Simple tools that work together. On a computer

Handiness

Literacy with tools allows for creativity, nonstandard solutions and fixing things

Mastery is enjoyable

Quicker and better than using monolithic tools!

What can I do on my setup?

What

Write/reuse/move/search/
visualise content and data

Where

Websites, slides,
documents, databases…

All my tools work together.

Not working / no functionality?
I can sort it

You should be “handy” with a computer

Slow and iterative process

Worth it for any computer-based job

https://missing.csail.mit.edu/

“The missing semester of your CS education”

Critical skills I won’t teach you

Git

SQL (e.g. in marimo)

Confidence
using the shell

Touch typing

Modal editing
(Most tools allow
vim keybindings)