



**DATA
CULTURE
SOCIETY**

CDCS.ED.AC.UK

SCOTTISH
GRADUATE
SCHOOL FOR
ARTS &
HUMANITIES
Sgoil Ceumachaidh na h-Alba airson
Ealaín agus Daonlachdan

2014-2024

Mr. MacLean, and the Tutor in his family (the Rev. Mr. Ferguson),
having ac-

to point out those marks
by which I was to find my way over the hills to

Sobermorie. We shook hands heartily & parted.

But, I had not proceeded one third of the track, rough
as it was, till I observed the night-clouds gathering on the east,

so that the day would be dark before noon. - I was alone; and in
the sad plight of a traveller benighted. - After a hard
waggoner under the shelter of a rock.

However, with a speech from the light of a candle and a
few the passengers of yesterdays party, I made a
longer stay in the marsh. Yet, even the

water gradually finding land we now left - all

to goope my way the best manner I could.

When I had passed other two mountain streams, I thought I
heard the distant murmur of waves - and, it was no auricular

sensation - it was the roar of a waterfall coursing through

the Lourne of Mull. Hearing this, I naturally concluded

that it was no great distance to the sea. And, now

having met the female that I had seen, - she directed

me, distinctly; and, following her, I soon found myself

at the door of the Inn, into which I entered. I enjoyed that

repose, so sweet & refreshing, after my worn traveller.

Adieu! Day 1, - impossible to say more.

Well, I will say more, -

Adieu! Day 1, - impossible to say more.

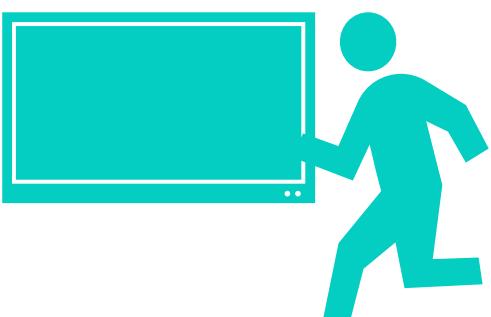
Well, I will say more, -

Adieu! Day 1, - impossible to say more.

DAY 1



ARRANGEMENTS FOR THE WEEK



Time	Monday 10th	Tuesday 11th	Wednesday 12th	Thursday 13th	Friday 14th
09:30 - 09:40	Welcome			Setting Up	
09:40 - 10:40	Introduction to Noteable and Python	Seminar 1	Seminar 2	Interlacing Loops, Functions and Lists	Seminar 3
10:40 - 11:00			Coffee Break		
11:00 - 12:30	Introduction to Noteable and Python	Functions	Collections	Data Importing and Handling	Data Visualisation Basics
12:30 - 13:30			Lunch		
13:30 - 15:00	Conditions and Logic	Functions	List Comprehensions	Data Summaries and Overviews	Data Visualisation in Practice
15:00 - 15:30			Coffee Break		
15:30 - 17:00	Keynote	Dates and Times	Loops	Data Cleaning	Next Steps
Evening	Reception	Ceilidh Club	No Evening Event	Pub Quiz	Garden Drinks

THE TEAM

INSTRUCTORS

Chris Oldnall



Martin Disley

Aislinn Keogh



Xan Cochran

Pawel Orzechowski



Jess Teed

THE TEAM

SPEAKERS

Ozan Evkaya



University Teacher In Statistics
School of Mathematics
University of Edinburgh

Tuesday

Hannah Claus



Research Assistant
The Ada Lovelace Institute

Wednesday

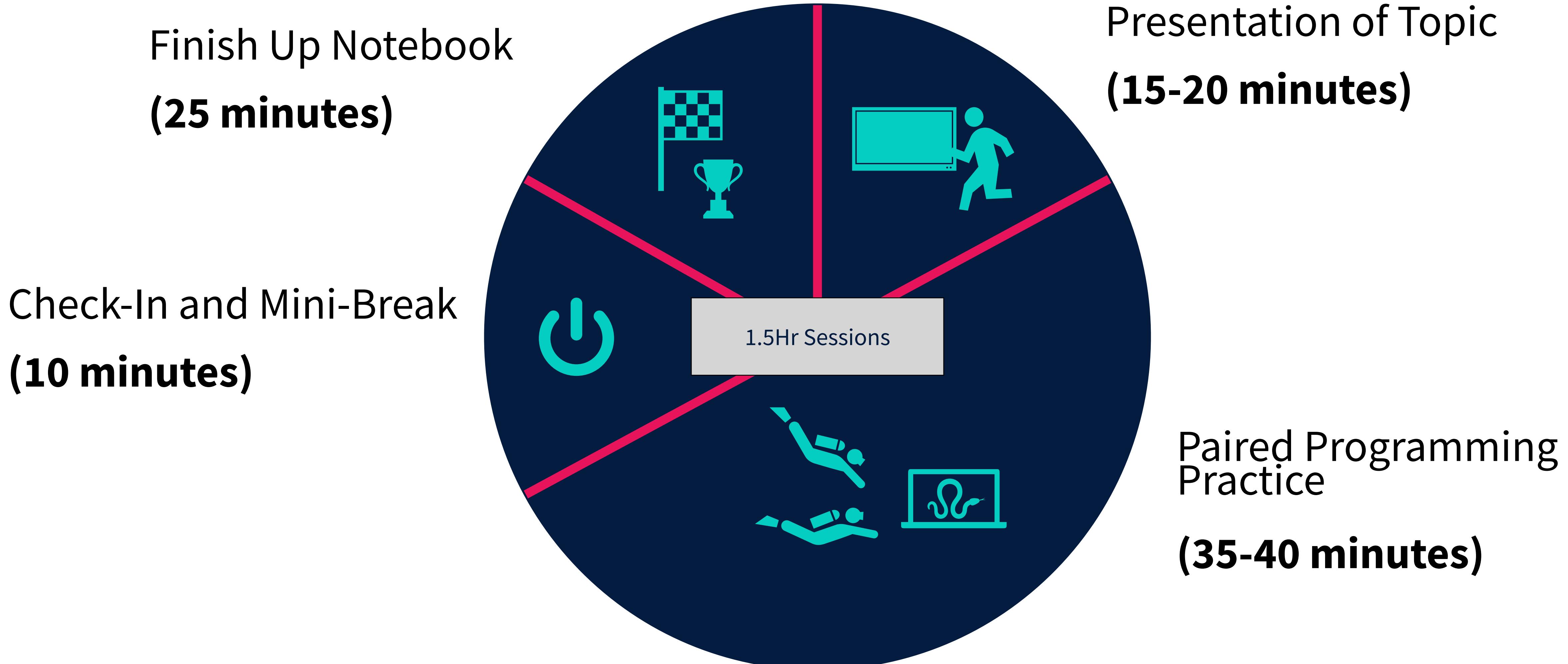
Pawel Orzechowski



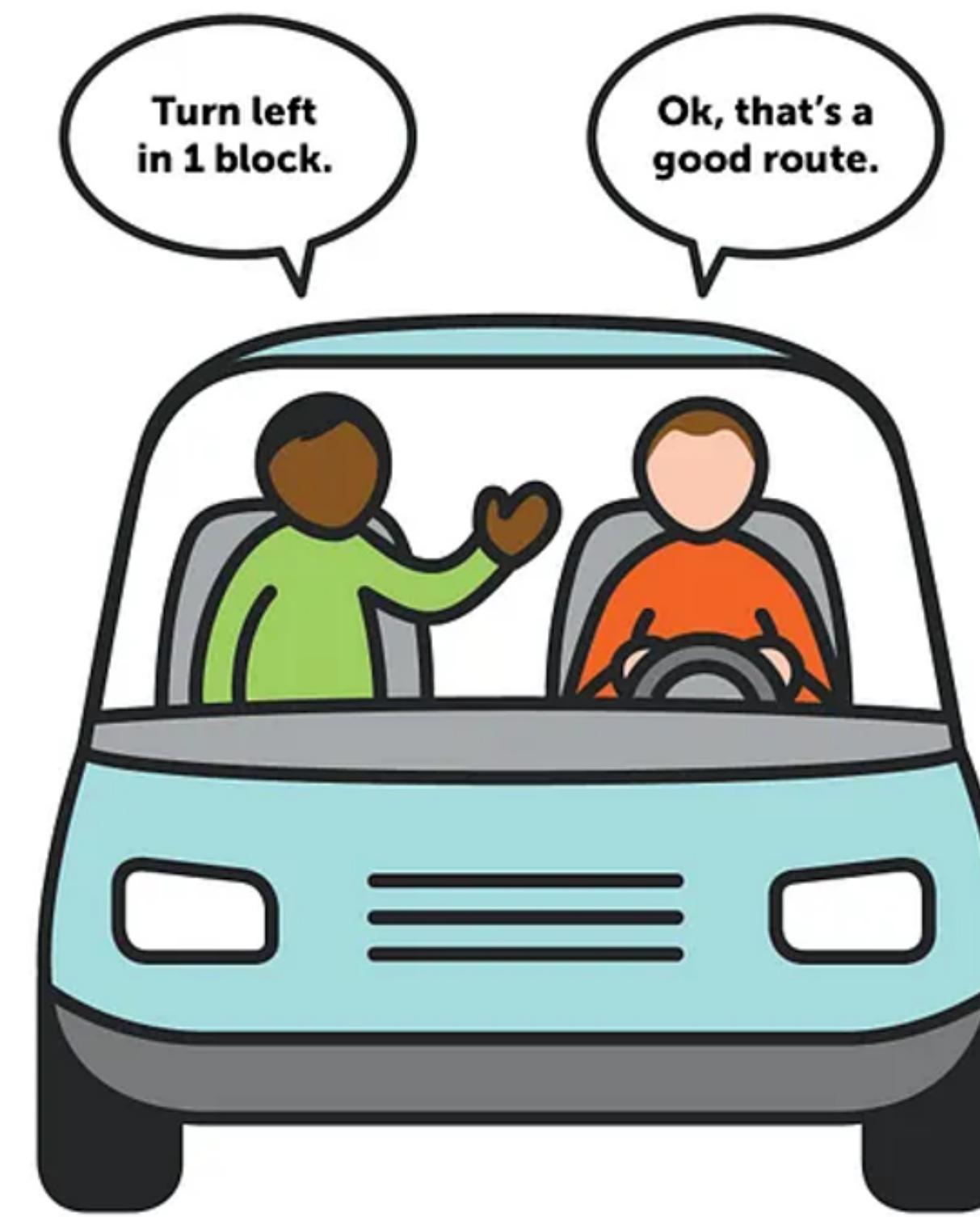
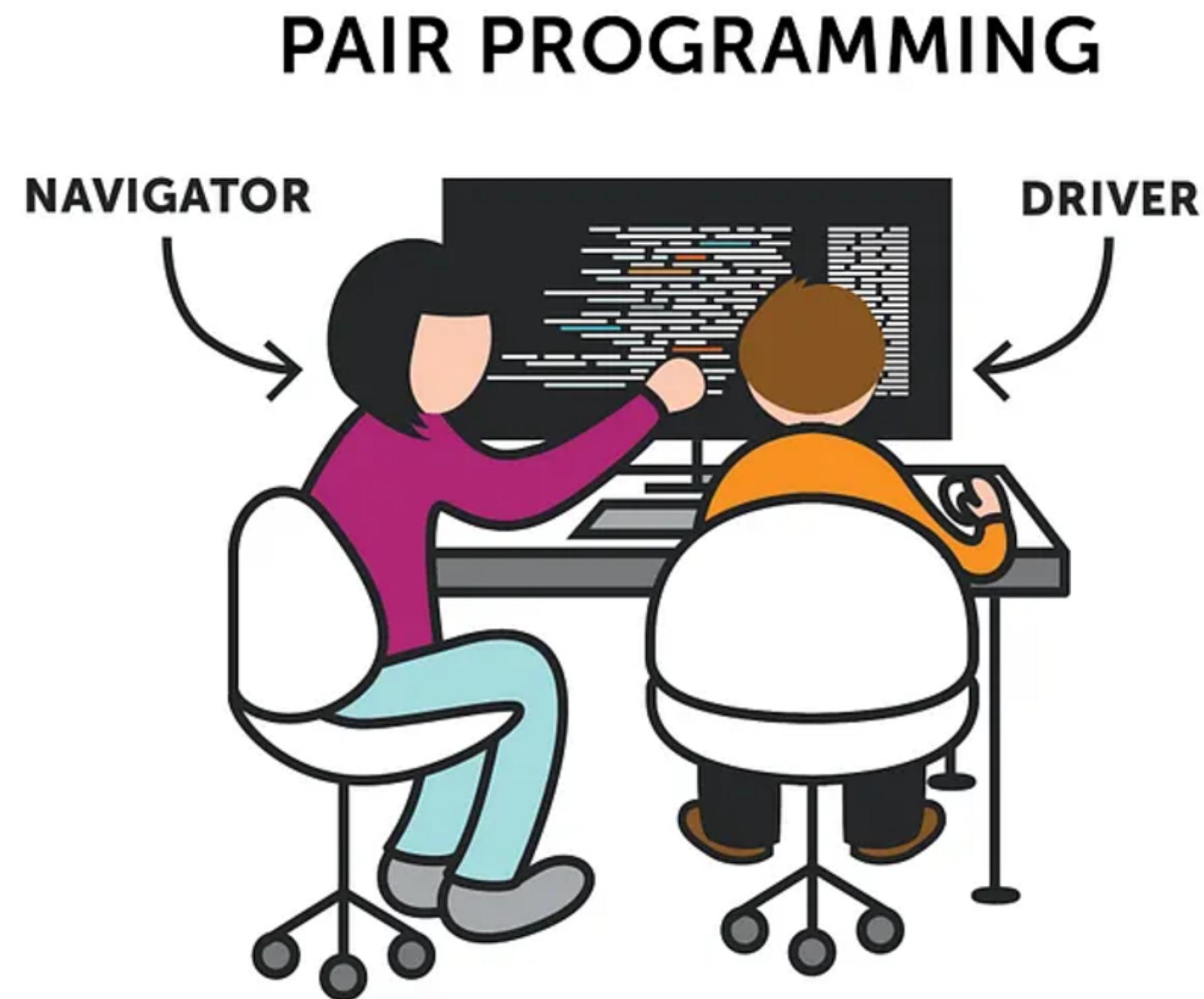
Lecturer in Programming for
Health and Social Care
The Usher Institute

Friday

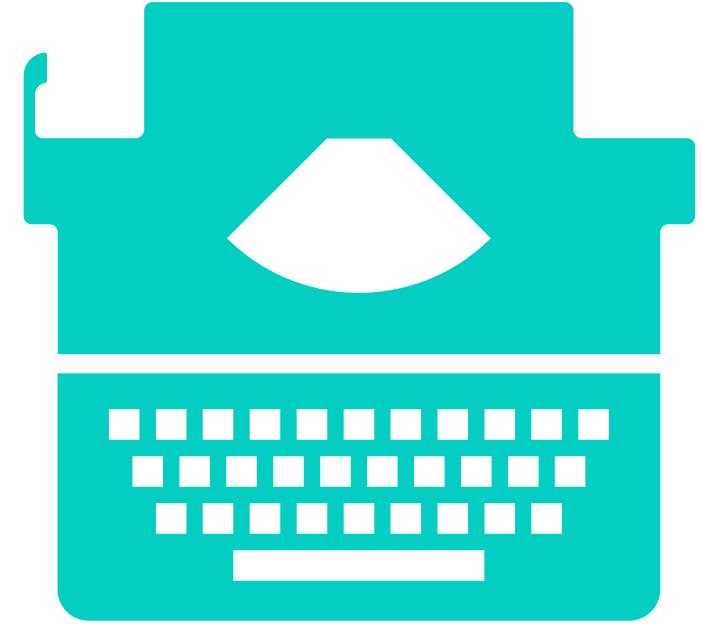
MOST SESSIONS THROUGH THE WEEK



PAIRED PROGRAMMING



OTHER THINGS YOU WILL SEE THROUGH THE WEEK



DEMONSTRATIONS

Sometimes you might see the typewriter symbol. This means we are going to demonstrate something in Python/Noteable.

Bear with us if it takes a moment to switch windows.

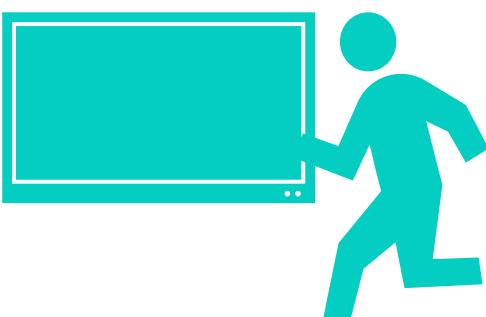
```
variable_name = sensible  
print(variable_name)  
  
“sensible”
```

CODE CHUNK TEXT

In the slides we may see text which is ‘pink’ in colour and a different font. This is to indicate it is a chunk of text, written in Python. The colour/font don’t matter just noticing it is code is important!

INTRODUCTION TO NOTEABLE AND PYTHON

PART 1



Programming = Telling the computer what to do

You need to speak a language the computer will understand,

e.g. **Python**



HOW DO WE ‘SPEAK’ PYTHON?

In order to ‘speak’ Python we will need two things...

1. The syntax (equivalent to the grammar of a language),
2. an interpreter (an interface to take in our Python syntax).



SYNTAX

Count from 1 to 10.

```
for number in range(1,11):  
    print(number)
```



Time	Monday 10th	Tuesday 11th	Wednesday 12th	Thursday 13th	Friday 14th
09:30 - 09:40	Welcome			Setting Up	
09:40 - 10:40	Introduction to Noteable and Python	Seminar 1	Seminar 2	Interlacing Loops, Functions and Lists	Seminar 3
10:40 - 11:00			Coffee Break		
11:00 - 12:30	Introduction to Noteable and Python	Functions	Collections	Data Importing and Handling	Data Visualisation Basics
12:30 - 13:30			Lunch		
13:30 - 15:00	Conditions and Logic	Functions	List Comprehensions	Data Summaries and Overviews	Data Visualisation in Practice
15:00 - 15:30			Coffee Break		
15:30 - 17:00	Keynote	Dates and Times	Loops	Data Cleaning	Next Steps
Evening	Reception	Ceilidh Club	No Evening Event	Pub Quiz	Garden Drinks

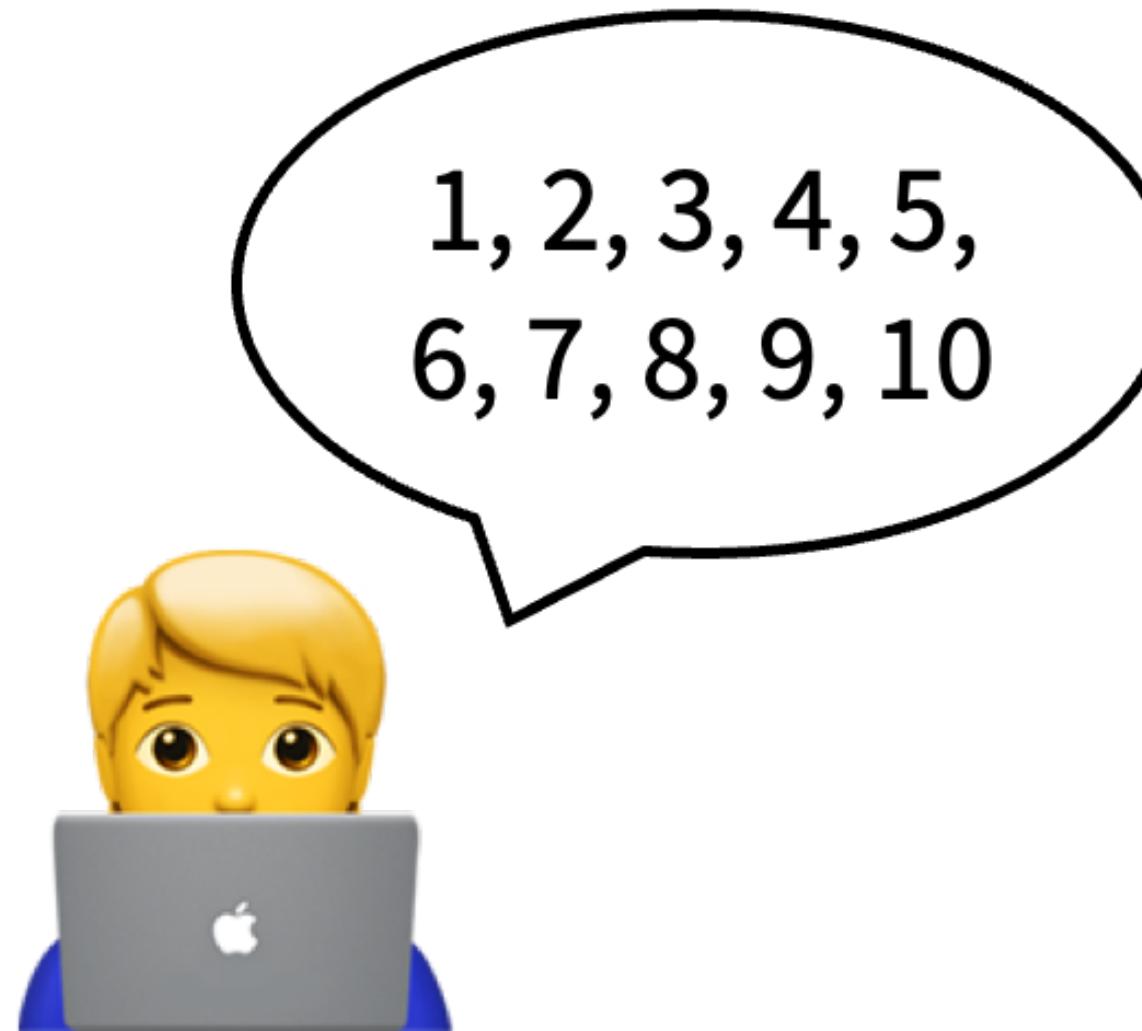
We'll spend a lot of the week learning the **syntax** of Python.

All the sessions dedicated to this are in shaded to the left, on the timetable for the week.



THE INTERPRETER

Count from 1 to 10.



```
for number in range(1,11):  
    print(number)
```

```
for number in range(1,11):  
    print(number)
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```



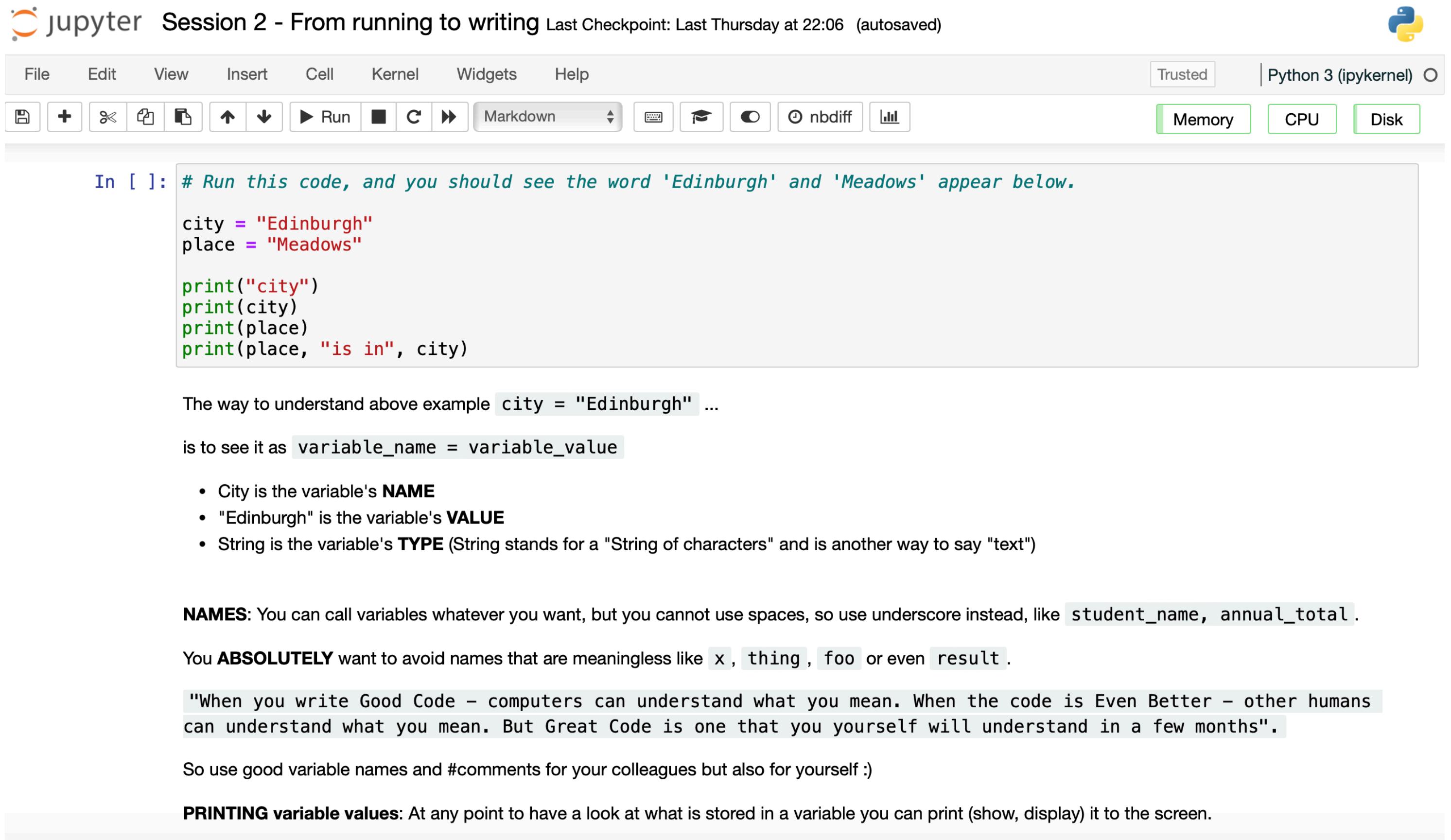
JUPYTER NOTEBOOKS

Our choice of interpreter throughout the week will be Jupyter Notebooks. These have many benefits:

1. Easy to use and share with others,
2. can be run on the Noteable service,
3. very hard to break anything (properly).



JUPYTER NOTEBOOKS



jupyter Session 2 - From running to writing Last Checkpoint: Last Thursday at 22:06 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) ○

Memory CPU Disk

In []: # Run this code, and you should see the word 'Edinburgh' and 'Meadows' appear below.

```
city = "Edinburgh"
place = "Meadows"

print("city")
print(city)
print(place)
print(place, "is in", city)
```

The way to understand above example `city = "Edinburgh"` ...
is to see it as `variable_name = variable_value`

- City is the variable's **NAME**
- "Edinburgh" is the variable's **VALUE**
- String is the variable's **TYPE** (String stands for a "String of characters" and is another way to say "text")

NAMES: You can call variables whatever you want, but you cannot use spaces, so use underscore instead, like `student_name, annual_total`.
You **ABSOLUTELY** want to avoid names that are meaningless like `x, thing, foo` or even `result`.
"When you write Good Code – computers can understand what you mean. When the code is Even Better – other humans can understand what you mean. But Great Code is one that you yourself will understand in a few months".
So use good variable names and #comments for your colleagues but also for yourself :)

PRINTING variable values: At any point to have a look at what is stored in a variable you can print (show, display) it to the screen.

}

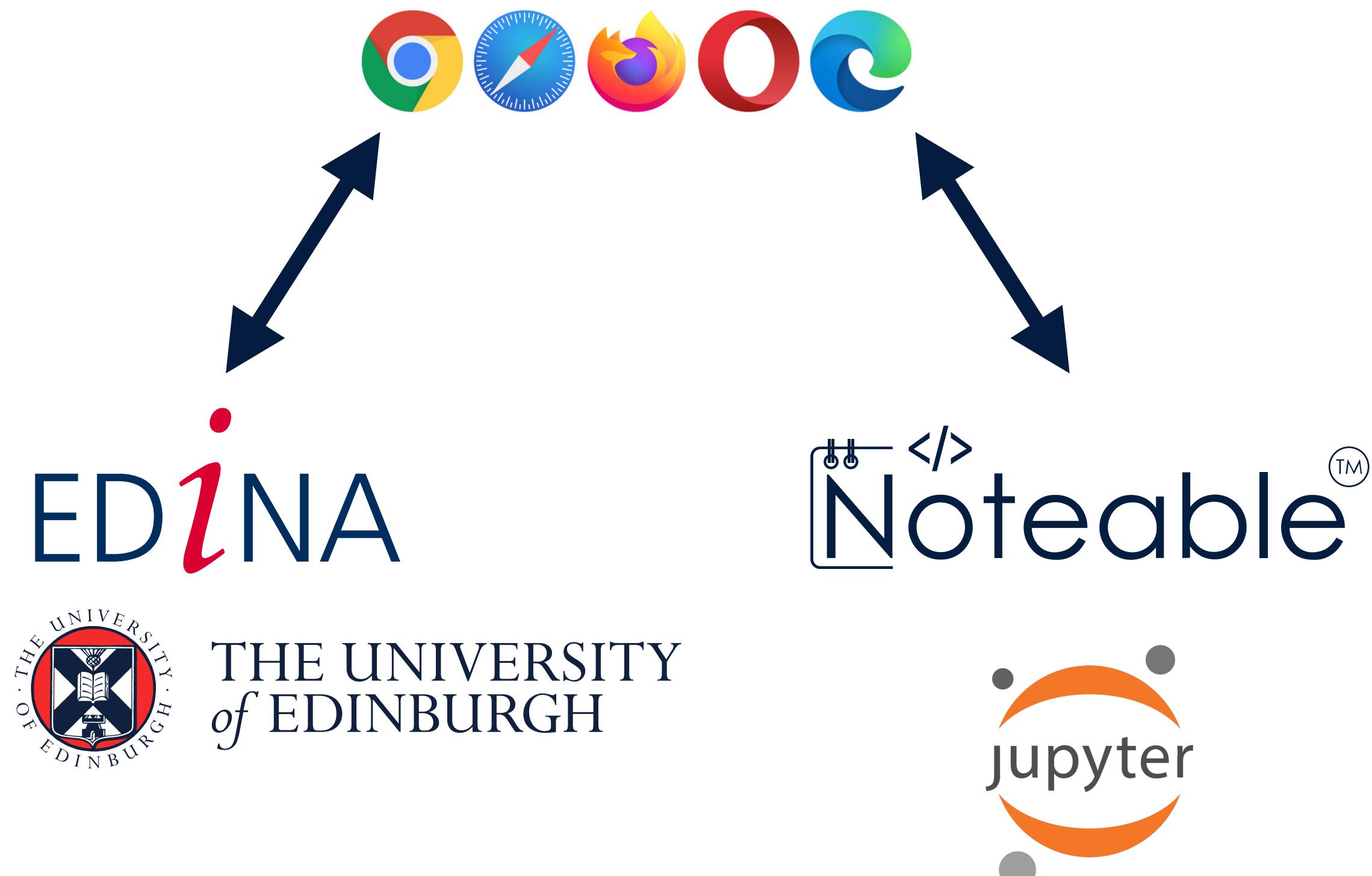
Code (Python)

}

Text (Markdown)



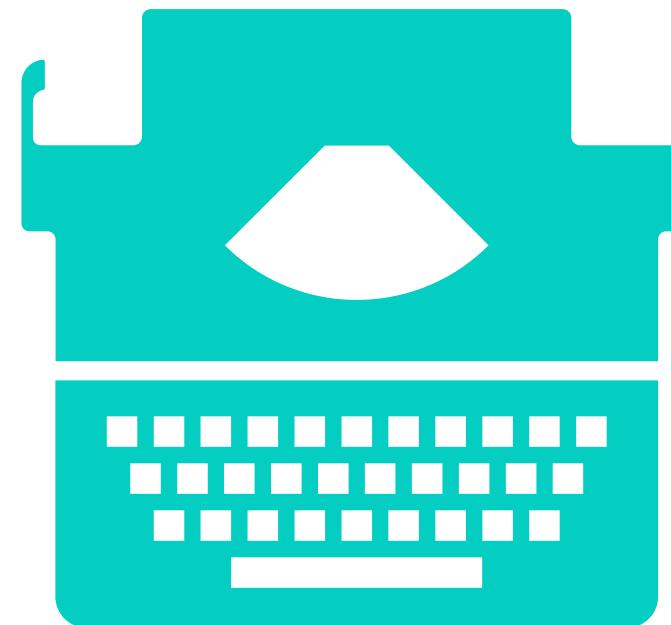
HOW DOES NOTEABLE COME INTO ALL OF THIS?



Noteable is a university of Edinburgh service which provides online web-based access to a Jupyter Notebook ‘server’. As it is not local this makes it great and easy for research and teaching!



LETS LEARN HOW TO ACCESS NOTEABLE AND JUPYTER NOTEBOOKS



Start Noteable

1. Open the following link in a new tab: <https://noteable.edina.ac.uk/login>.
2. Login with your EASE credentials (either your Edinburgh university login, or those you were provided with).
3. Under 'Standard Notebook (Python 3)' click 'Start'

Download the files to Noteable.

1. From the Noteable home page, click on the '+GitRepo' button at the top right of the screen.
2. In the 'Git Repository URL' field copy the link to this GitHub repository, <https://github.com/DCS-training/summerschool2024-stream1>. Ignore all other fields.
3. Once filled in, click the 'clone' button. After a few moments, you will then see a new folder appear with the files.



LETS GET PROGRAMMING

Session 1: Getting to know
Jupyter Notebooks





COFFEE BREAK

**WE ARE GOING TO RESTART AT
11:00**

INTRODUCTION TO NOTEABLE AND PYTHON

PART 2



TOP TO BOTTOM

Code is written in lines. Each line does something, and they are executed from top to bottom.

```
count = 43  
print(count)
```

```
count = 44  
print(count)
```

```
count = 44  
print(counter)
```

```
count = 44  
print(counter)
```

```
count = count + 1  
print(count)
```



TOP TO BOTTOM

Code is written in lines. Each line does something, and they are executed from top to bottom.

```
count = 43  
print(count)
```

43

```
count = 44  
print(count)
```

44

```
count = 44  
print(counter)
```

ERROR

```
count = 44  
print(count)
```

44

```
count = count + 1  
print(count)
```

45



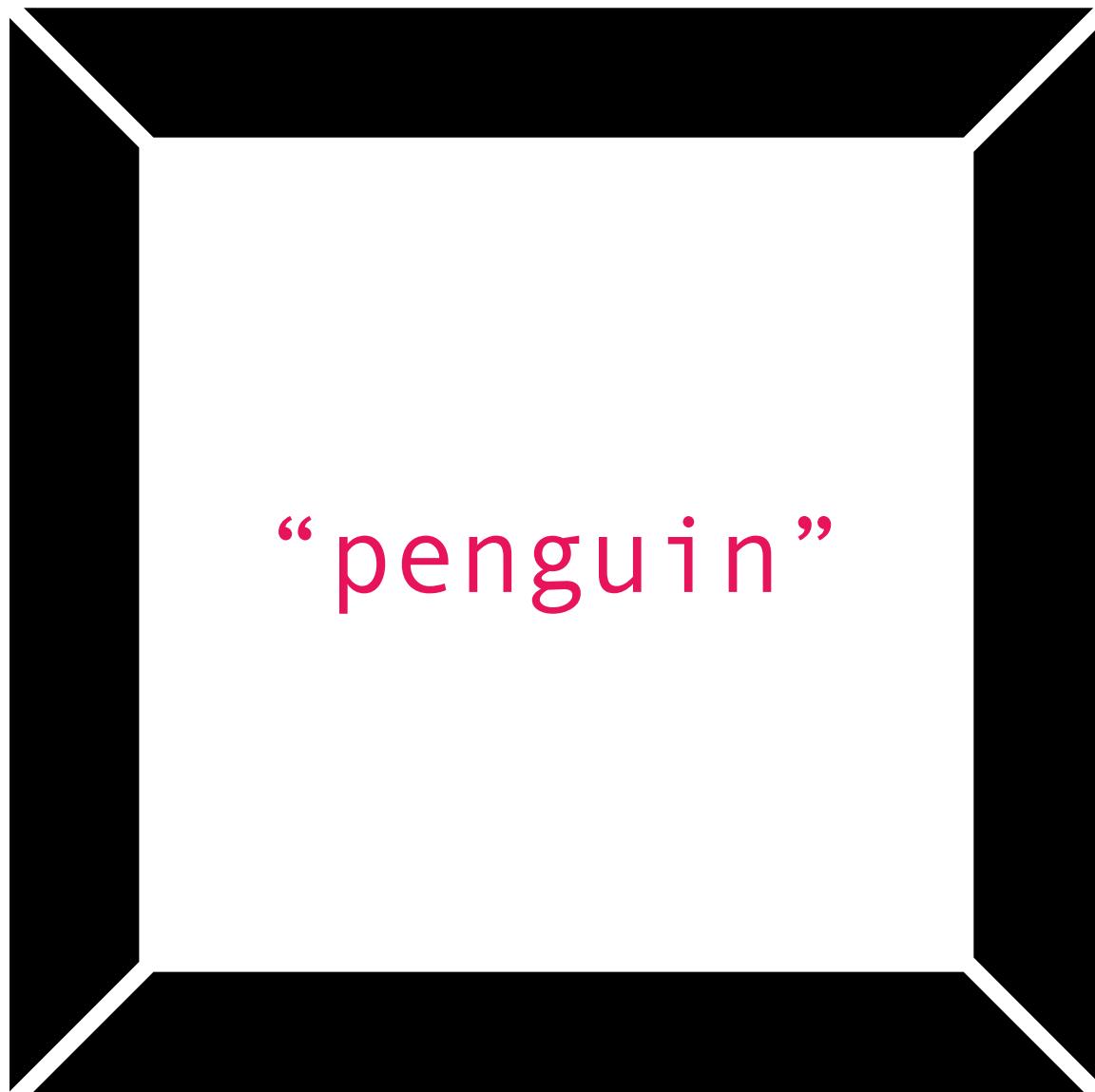
VARIABLES

Variables are places to store values for later.
There are different types of variables:

- **string** for text, e.g. “*penguin*” or “*I like Python*”
- **int** (integer) for whole numbers, e.g. 1, 5, 2014
- **float** for decimal numbers, e.g. 2.25, 6.1246, 16.2
- **bool** (Boolean) for logic values: *True*, *False*

```
my_favourite_animal = "penguin"
```

`my_favourite_animal`



LETS GET PROGRAMMING

Session 2: From Running to Writing





THE UNIVERSITY of EDINBURGH
Centre for Data, Culture & Society



LUNCH BREAK

**WE ARE GOING TO RESTART AT
13:30**

CONDITIONS AND LOGIC



COMPARING VARIABLES

```
2 == 2  
2 == 3
```

```
"penguin" == "penguin"  
"penguin" == "whale"  
"penguin" == "Penguin"
```

```
"2024" == 2024
```



COMPARING VARIABLES

2 == 2
2 == 3

True
False

"penguin" == "penguin"
"penguin" == "whale"
"penguin" == "Penguin"

True
False
False

"2024" == 2024

False



COMPARING VARIABLES

2 < 2

2 < 3

3 > 2

2 <= 2

3 != 2

3 != 3



COMPARING VARIABLES

$2 < 2$ False

$2 < 3$ True

$3 > 2$ True

$2 \leq 2$ True

$3 \neq 2$ True

$3 \neq 3$ False



CONDITIONALS

Controlling what part of your code gets executed based on some conditions can be very useful in capturing real life conditionals.

E.g. “If the traffic light is green, go, otherwise, wait.”

```
traffic_light = "green"

if traffic_light == "green":
    print('go')
else:
    print('wait')
```

go

```
traffic_light = "red"

if traffic_light == "green":
    print('go')
else:
    print('wait')
```

wait



LETS GET PROGRAMMING

Session 3: If (This AND That)





COFFEE BREAK

WE WILL RESTART AT 15:30 FOR THE KEYNOTE TALK BY PROFESSOR MELISSA TERRAS
"HOW DO THE HUMANITIES KEEP UP WITH AI? OPPORTUNITIES AND ISSUES FOR RESEARCH"