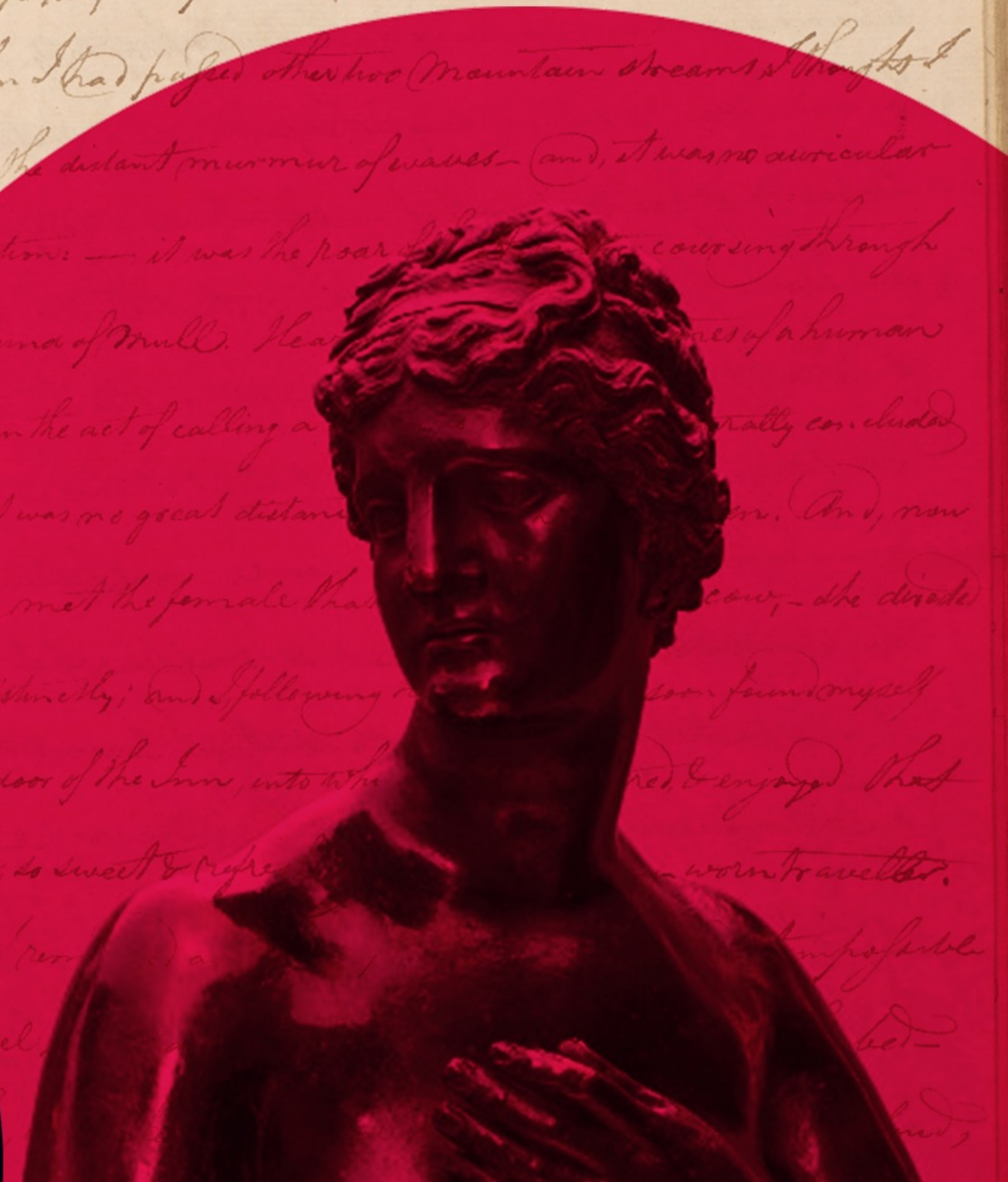
DATA CULTURE SOCIETY
CDCS.ED.AC.UK

SCOTTISH GRADUATE SCHOOL FOR ARTS & HUMANITIES
Sgoil Ceumnachaidh na h-Alba airson Ealain agus Daonnachdan
2012-2024

DAY 2



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.

SEMINAR 1

Ozan Evkaya

University Teacher In Statistics
School of Mathematics
University of Edinburgh

THE UNIVERSITY of EDINBURGH
Centre for Data, Culture & Society

DATA CULTURE SOCIETY

COFFEE BREAK

WE ARE GOING TO RESTART AT 11:00

FUNCTIONS

PART 1

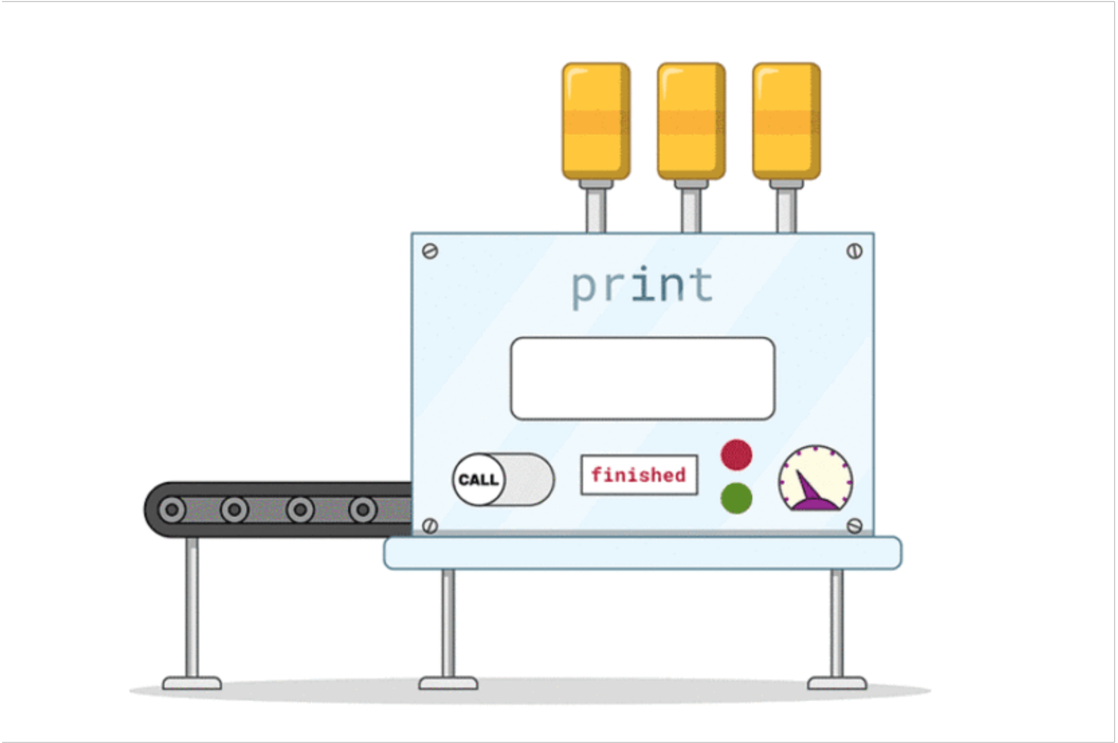
QUESTIONS TO ASK…

1.What is a function?

2. Why do we use functions?

3. How do I make a function in Python?

WHAT IS A FUNCTION?



WHAT IS A FUNCTION?

A way to generalise a process that will need to be done over and over again.

WHY DO WE USE FUNCTIONS?

- Reduce lines of code,
- Enhance computing performance,
- Make life easier!

WHY DO WE USE FUNCTIONS?

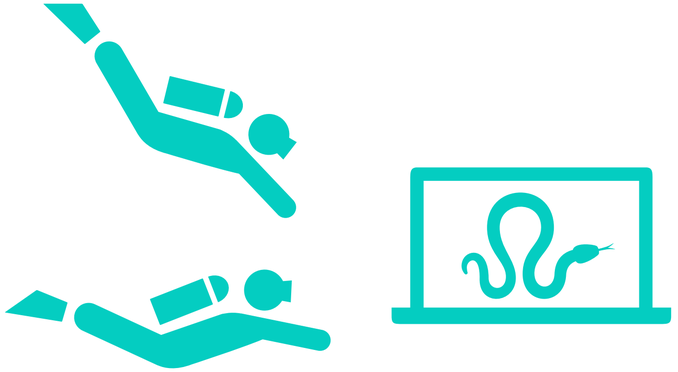
WHAT IS THE RECIPE FOR A FUNCTION IN PYTHON?

1. 'def'
2. Name
3. What goes into it (arguments)
4. What it does (the steps)
5. What it gives back (return value)

```python
def bake_a_cake(cake_type, cake_size, cake_flavor, cake_filling, cake_frosting):
  """This function bakes a cake of the specified type, size, flavor, filling, and frosting.

  Args:
    cake_type: The type of cake to bake, e.g. "chocolate", "vanilla", "red velvet".
    cake_size: The size of the cake to bake, e.g. "small", "medium", "large".
    cake_flavor: The flavor of the cake to bake, e.g. "chocolate", "vanilla", "strawberry".
    cake_filling: The filling for the cake, e.g. "chocolate ganache", "vanilla buttercream", "strawberry jam".
    cake_frosting: The frosting for the cake, e.g. "chocolate ganache", "vanilla buttercream".

  Returns:
    A cake of the specified type, size, flavor, filling, and frosting.
  """

  print(f"Baking a {cake_type} {cake_size} {cake_flavor} cake...")

  # Prepare the cake batter
  # ...

  # Pour the batter into a cake pan
  # ...

  # Bake the cake
  # ...

  # Let the cake cool
  # ...

  # Fill the cake if specified
  if cake_filling:
    # Fill the cake
    # ...

  # Frost the cake if specified
  if cake_frosting:
    # Frost the cake
    # ...

  print("Cake is ready!")
  return f"{cake_type} {cake_size} {cake_flavor} cake with {cake_filling} and {cake_frosting}"
```

LETS GET PROGRAMMING

Session 4: Write the Recipe



THE UNIVERSITY of EDINBURGH
Centre for Data, Culture & Society

DATA CULTURE SOCIETY

LUNCH BREAK

WE ARE GOING TO RESTART AT 13:30

FUNCTIONS

PART 2

LOCAL AND GLOBAL

Local:
Only those close to him *(within the same function)* who know about what he can do.

Global:
Can be accessed by anyone, anywhere – everyone knows what he can do!

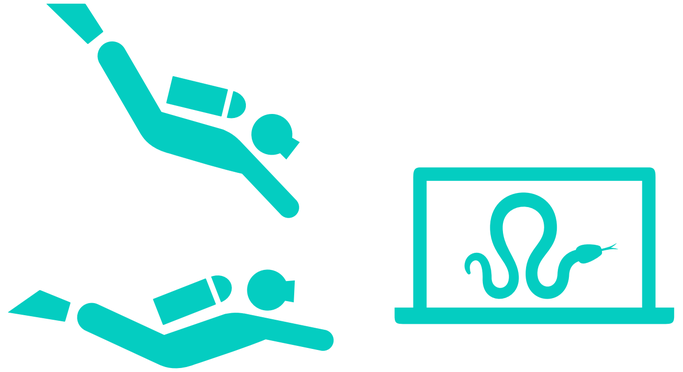
THE SCOPE RULES

Rule 1: Anything inside a function is mysterious to the outside...
You are not able to peek inside of a function elsewhere in code.
Only things returned will become available to the 'global' environment.

Rule 2: Functions can look outside, but shouldn't...
Things can get complicated when a function looks outside.
We tackle this by carefully specifying arguments with relevant names.

LETS GET PROGRAMMING

Session 5: SCOPE



THE UNIVERSITY of EDINBURGH
Centre for Data, Culture & Society

DATA CULTURE SOCIETY

COFFEE BREAK

WE ARE GOING TO RESTART AT 15:30

DATA CULTURE SOCIETY
CDCS.ED.AC.UK

SCOTTISH GRADUATE SCHOOL FOR ARTS & HUMANITIES
Sgoil Ceumnachaidh na h-Alba airson Ealain agus Daonnachdan
2014 - 2024

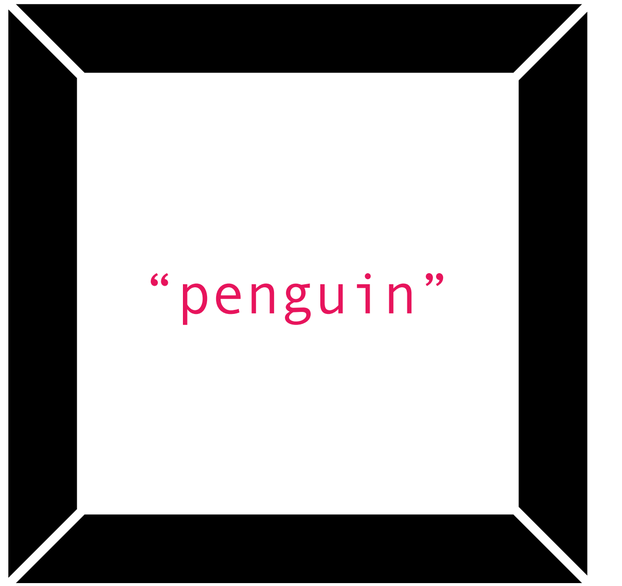
DATES AND TIMES

RECAP: 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*

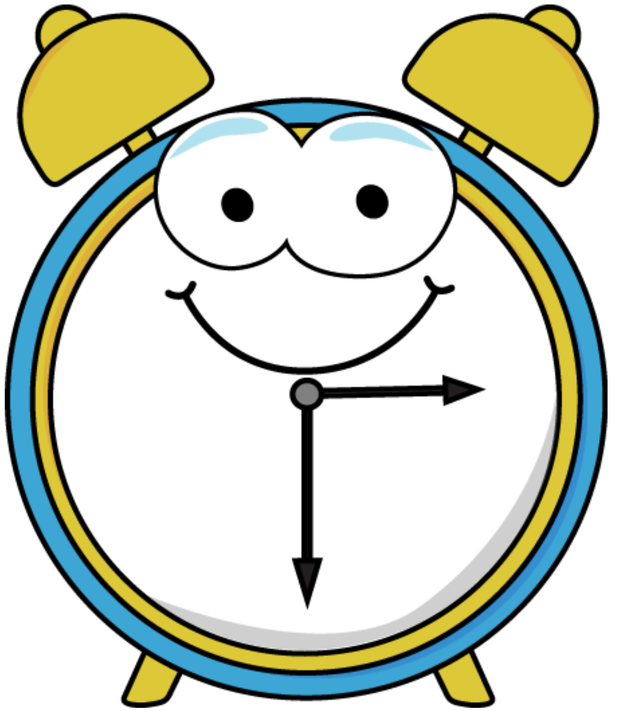
```
my_favourite_animal = “penguin”
```



ANOTHER DATA TYPE

Using the package ‘datetime’ we can introduce a special data type: **datetime**

With this we can manipulate times and dates, which is often useful for work across the arts, humanities and sciences!



WHAT IS A PACKAGE?

We mentioned in the previous slide that we would ‘import’ the ‘datetime’ package.

We’ll talk more about packages later in the week, but for now what you need to thing is a package is a collection of functions all to do with one purpose written and bundled together by an author.

HOW DO I IMPORT A PACKAGE?

```
import datetime as dt
```

- `import` is a special Python command for telling the interpreter to search for a package,
- `datetime` is the name of the package,
- `as` tells the interpreter we are going to ‘rename’ the package when using it,
- `dt` is the ‘nickname’ we will use in our code.

We only have to write (and run) this once per notebook.

WHAT CAN I DO WITH DATETIME?

Firstly I may want to get the current time…

```
dt.datetime.now()
```

- We firstly say we want to look at the package datetime by '`dt`'.
- Then we add '`.datetime`' to get to the functions within it.
- Finally we add '`.now()`' to get the current date-time.

WHAT CAN I DO WITH DATETIME?

We may instead want to create a specific date time object…

```
summer_school_start = dt.datetime(2024, 6, 10, 9, 0, 0)
print("The summer school starts on:", summer_school_start)
```

```
The summer school starts on: 2024-06-10 09:00:00
```

LETS GET PROGRAMMING

Session 6: Date-Time