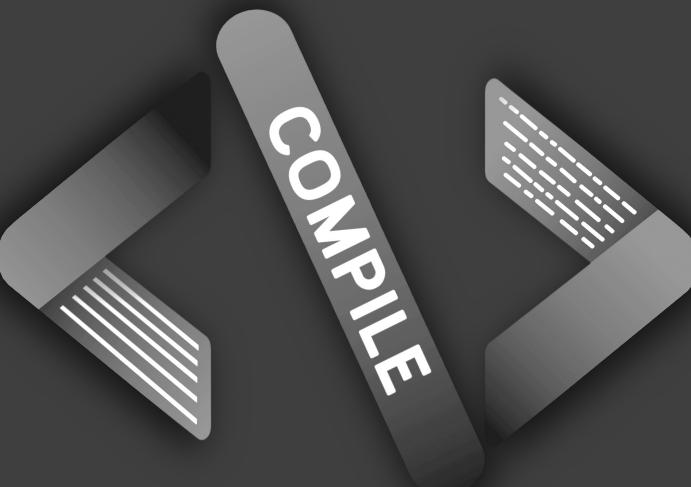


# LESSONS LEARNT FROM TEACHING PYTHON

Hanna Schmück  
& COMPILE





# WHO AM I AND WHY AM I HERE?

- I do academic and linguistic stuff
  - Associate Lecturer & PhD Student at Lancaster University
  - Researching Large-Scale Linguistic Networks
  - Worked on the British National Corpus, syntactic and pragmatic coding projects
- I code
  - 5-ish Years Experience Programming in Python, Some Experience Programming in R, JavaScript
- I teach coding
  - Coordinating the Lancaster branch of COMPILE, teaching *A Practical Introduction to Python Programming* (online, free)

# MORE IMPORTANTLY - WHO ARE WE?

COMPILE collaborates mainly with universities / student societies

The aim is to help social scientists\*, and all sorts of interdisciplinary complete beginners make a start at learning to code



@CompileLearning

# THE TEAM



SAMUEL HOLLANDS



HANNA SCHMÜCK



THEO TOLLET



MAGGIE MI



JAKE SWINNEY



MICHAEL STOREY



IF YOU ARE CODING:  
HOW LONG HAVE YOU  
BEEN CODING FOR?  
ARE YOU SELF-TAUGHT?

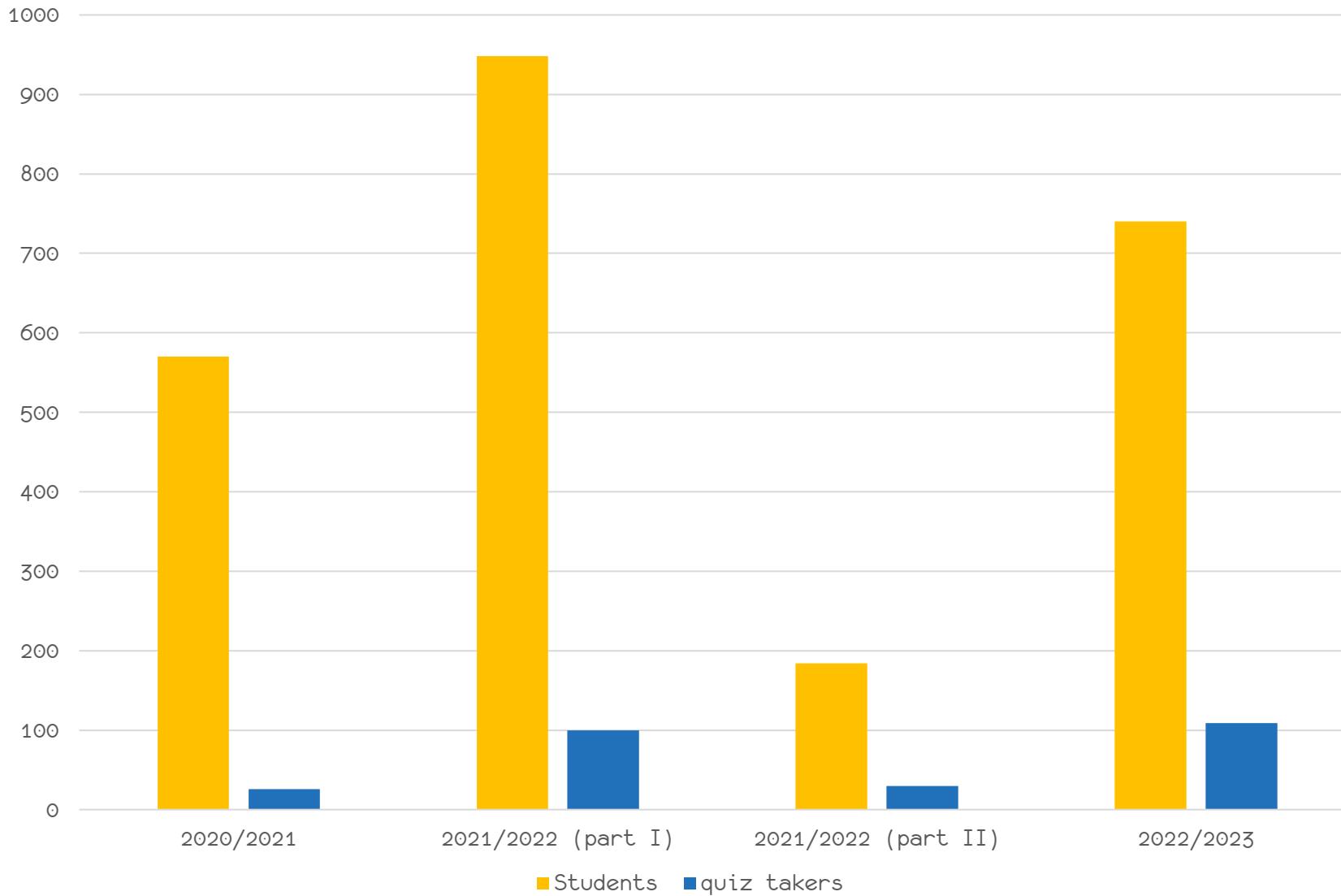


# WHAT DO WE DO?

We focus on:

- Absolute beginners
- People with no tech background
- Learning by doing
- Computational thinking

## PARTICIPANT NUMBERS



# CONTENT

I		II	
Variables and Operations I	W1	Jupyter	W12
Variables and Operations II	W2	Workshop	W13
Lists, Dictionaries, Sets, and Loops I	W3	NLP I	W14
Lists, Dictionaries, Sets, and Loops II	W4	Workshop	W15
Reading and Writing to Files	W5	Reading Week	W16
Invited talk	W6	NLP II	W17
Functions	W7	Workshop	W18
Errors and Exceptions	W8	Pandas	W19
Classes	W9	Workshop	W20
Importing External Libraries	W10	Matplotlib	W21

# SO WHAT ARE WE DOING?

The image displays three screenshots from a digital learning environment:

- Code Editor:** Shows a Python code editor with a script named "fibonacci.py" containing code related to calculating Fibonacci numbers.
- Certificate with Project:** A certificate titled "Certificate with Project" for a "Hangman" project. It includes a "File" icon and a "Project" icon. The text describes the task of creating two files: "hangman.py" and "hangmanClass.py". It specifies that "hangmanClass.py" should be imported into "hangman.py" and that the "startGame()" method in "hangman.py" should trigger the game to run until a win or loss is determined. It also mentions that during each round, the current hangman diagram, number of guesses remaining (out of 6), the current string (e.g., "\_PP\_e"), and the letters already guessed (e.g., "[p, e, z]") should be printed.
- Weekly Challenge:** A challenge titled "Weekly Challenge" with a "Fibonacci" section. It encourages users to take advantage of Teams to share solutions and discuss difficulties. It states that the Fibonacci sequence starts with [0, 1] and defines it as the sum of the previous two numbers. It asks users to calculate the thousandth value of the sequence.

A screenshot of a Python IDE showing code in session3.py. The code includes variables like five, notdoneyet, and thisisamess. Below the code editor is a terminal window displaying a calculator application. The terminal output shows a welcome message and a menu for selecting operators: Multiplication, Division, Addition, and Subtraction. The interface includes standard IDE tools like Run, Structure, and Version Control.

**Question 6**  
Not yet  
answered  
Marked out of  
1.00  
Flag  
question  
Edit  
question

What is the purpose of the re library?  
  
e.g. import re

- a. reactive programming
- b. regressions
- c. regular expressions
- d. recall functions

# Certificate with Project

Hangman - [https://en.wikipedia.org/wiki/Hangman\\_\(game\)](https://en.wikipedia.org/wiki/Hangman_(game))

Create two files `hangman.py` and `hangmanClass.py`. Build a class called `hangman` in `hangmanClass.py` that we can import into `hangman.py` to run our game. Running `startGame()` in `hangman.py` should trigger the game to run until I have either won or lost.

Each round of the game you should print out the current diagram of the hangman, the number of guesses remaining (out of 6), the current string e.g. ("\_pp\_e") and the letters the user has already guessed e.g. ["p", "e", "z"].



A presentation slide with a green header bar. On the right side, there is a video feed of a person. The slide contains two main sections: "Weekly Challenge" and "Fibonacci".

**Weekly Challenge**  
You are strongly encouraged to take advantage of Teams to share solutions and discuss difficulties you are having!

**Fibonacci**  
The fibonacci sequence is denoted as starting with [0,1..] after this every following number should be the sum of the two previous numbers. Calculate fibonacci to the thousandth value.

# VARIABLES AND OPERATIONS

```
# Testing a Print Function - Introduce Strings
print("Hello World")

# Writing a comment
print("Hello World")      # Prints "Hello World"

# Printing some Arithmetic
print("Addition", 2 + 2)
print("Subtraction", 2 - 1)
print("Division", 2 / 4)
print("Multiplication", 2 * 4)
print("Exponent (To the power of)", 2 ** 2)
print("Modulus (remainders)", 13 % 5)

# Float vs Int
print("Integer", 2*2)    # Whole number (without decimal places, even .0!)
print("Float", 2/4)       # Decimal (Even .0!)

# Storing my Data
my_number = 5
print("Variables", my_number)
print("Comparatives", 5 == 5)  # Remember 1x "=" is variable assignment, 2x "=" is comparative
```



# LISTS, DICTIONARIES, SETS, AND LOOPS

```
# List Indexing

myList = ["Apple", "Pear", "Lime", "Banana"]

myPear = myList[1] # How to get an item by index value, remember Python (like most programming Languages) starts counting from 0 not 1!

mySelection = myList[1:3] # Selecting a range of objects from a Python List

myLast = myList[-1] # Get the Last item in a List regardless of list size

myLength = len(myList) # Gets the Length of myList as an integer

myRange = range(myLength) # Creates an iterable object that contains every countable whole number between 0 and myLength excluding the last value; range(3) == 0, 1, 2

# Dictionaries
# Dictionaries are also used to store multiple items in one variable, but they have a special shape. Dictionaries contain unique keys
# which are assigned values (these don't have to be unique). Their key characteristics are:
# that they contain only unique keys, are changeable, and are ordered (since Python 3.7).

myShopping = {"Apples": 5, "Grapes": 10, "Lemons": 5}
applePrice = myShopping["Apples"]
```





WHICH CONCEPTS DO YOU  
WISH YOU HAD LEARNT  
ABOUT SOONER?

FOR ME, IT'S CLASSES, REGEX FLAVOURS, AND DECORATORS



# LESSONS LEARNT

- There is a lot more demand than we anticipated
- Learners value the social component, 'even' for coding
- Asynchronous options are key
- A cluttered IDE is a killer

```
# Below is an import we have seen before, recall that we can use the "as" keyword to give it an alias
import numpy as np

# if I run this in an environment where I don't have numpy installed it will throw an error
# why is this?

# Python comes with certain builtin packages that we can import from the get go
# e.g. I can write the following without any errors on an out-of-the-box python install:
import json
import pickle

# however we mentioned that the power of python lies in the ability for anyone and everyone
# so for external libraries -- ones not included with python -- we need some way of accessing them
# In your terminal: pip install numpy
# or ..... pip3 install numpy

# now if we try import numpy again it should be a success!
import numpy as np

# and if we try create a matrix of zeros as Sam did, everything works great:
print(np.zeros((5,5)))

# so what did we do?
# pip is Python's built-in package manager
# And a package manager is simply an efficient way to interface with a huge collection of packages
# as we just did with numpy!

#~~~ SLIGHT DISCLAIMER ~~~#
# Pip has no filtering measures in place, which means anyone can build any package and make it available
# We do need to bear in mind that externally sourced code can be malicious
# You shouldn't face any issues with popular libraries
# but inspecting source code is a good habit to get into even just as a learning exercise
#~~~~~#"

# In practice, being able to import packages is very very powerful.

# PROBLEM STATEMENT: Given a sentence and a list of other sentences, return the sentence with the most similar meaning
# Time for some NLP magic.
# In Natural Language Processing there are often many operations we want to apply to texts
# so that they can be used/analysed down stream, such as Stopword Removal and Tokenization.
# Googling "stop words python" we find a solution to this using the NLTK library

# pip install nltk

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

# GET INVOLVED!

Are there any skills you wish more people in your field had?

Is there a challenge you are often faced with, and you have found a good solution to overcome it?

Get in touch with

Sam Hollands [shollands1@sheffield.ac.uk](mailto:shollands1@sheffield.ac.uk)

or yours truly [h.schmueck@lancaster.ac.uk](mailto:h.schmueck@lancaster.ac.uk)

You could teach a specialist session (45-minutes) and help people get even more excited about coding.

# THANK YOU

Hanna Schmück

[hannaschmueck.github.io](https://hannaschmueck.github.io)

[@fediscience.org](mailto:@hanna_schmueck)

