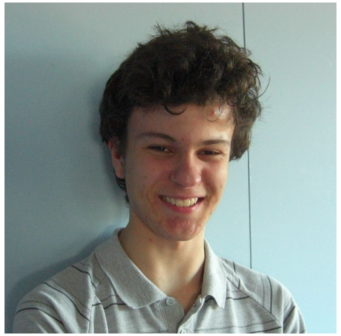
#CodeYork

Session 1

Who? What? When?



Graham Campbell

University of York

gjc510@york.ac.uk

https://github.com/GrahamCampbell

Jack Romo

University of York

jr1161@york.ac.uk

https://github.com/jackromo

Weekly Classes

3:30-4:30PM

Weekly Assignments

Optional, but good preparation

Game Driven Teaching

Battle against your friends

What’s a Python?

- We’ll be using Python 3.5
- Interpreted scripting language
- Simple and readable
- “Holy Grail” of prototyping and teaching
- Used in data science, AI, sciences, web, scripting

https://python.org

Down to busyness

- Quick recap of Python's features
    - Variables, data types, if-elif-else, functions....
- Not much practical work today
- Use these slides as reference
    - Unlike Java or C#, indentation has meaning; be careful when copying into editor.
    - Indent with 4 spaces, or tabs, but whatever you do, keep it the same, or your code may not even run. We will be using 4 spaces throughout our code samples.

Primitive Data Types

- Words and letters are strings (str)
  - ‘egg’ or “spam”
- Numbers with a decimal point are floating point numbers (float)
  - 124.0, 124e2, -0.123
- Numbers with no decimal point are integers (int)
  - 1, 124, 0, -5
- Booleans give us logic
  - True, False
  - Must be capitalized, false is not False!
- Some things are nothing
  - ‘None’ (not ‘null’ like in Java!)

Operators and Conditionals

- Arithmetic is pretty straightforward
  - 1 + 2 - 3 * 4 / 5 ** 6 (try and figure out what this evaluates to!)
- Booleans have algebra too
  - True and False, True or False
  - not True
- Conditionals let us check for truth
  - 1 == 1 (True), 1 > 1 (False)
  - <= means ‘less than or equal to’
- Some operations happen before others (operator precedence)
- If two operators have equal precedence, left goes first
  - Think about what this means for subtraction

Variables

- Variables let you store stuff
  - x = 5, y = “hey”
- No types, Python variables can hold anything
  - Not like Java, where you have to say what type a variable is (eg. ‘int x;’)
- Variables are changeable (mutable)
  - x = x + 1 makes sense! (x becomes 1 greater than before this)
  - x += 1 is shorthand for above, also x /= 1, x -=1, x *=1…
    - Note that x++ is not supported
  - Variables may also change type
    - Think about why this might not actually be such a good idea

I/O

- Can display stuff to the user…
  - print(5) - prints 5 to the screen.
  - Strings are printed without quotes; each type has its own way of being printed, strings themselves being the most trivial example!
- ...and ask for things from the user
  - x = input(“Tell me your name: ”)
    - Prints prompt, then reads anything user types before pressing ‘Enter’
- Read/write files, or indeed any stream (such as from a website or database).
  - open()

Control Structures

- If blocks are the the most common control structure you will encounter.
- The syntax is as follows:

```
if foo == 3:
    print('Variable foo was 3!')
else:
    print('Variable foo was NOT 3!')
```

- The else clause is optional.

While Loops

- Some things need to be done several times
- While loops are if statements on repeat
  - If statement runs code if true, while loop keeps running code while condition is true
  - Useful to avoid retyping repeated code
- Good if you don’t know when your code should stop running
  - Eg. Game engine (can’t predict when user shuts down game!)
  - Infamous ‘while True:’ loop (what does this do?)

Lists

- Lists can hold several ordered items
  - ls = [1, 2, ‘hello’, 3.4] - different types in the list are fine
- Lists can be indexed (get element at location)
  - ls[0] is 1, ls[2] is ‘hello’
  - Lists are 0-indexed; we count from 0, not 1
  - Lists of length n have elements 0 through n-1
- Lists can be ‘spliced’ for a sublist
  - ls[1:3] gives [2, ‘hello’]
  - First index included, second one not
  - ls[0:4] is entire list, ls[0:4:2] gives every 2nd item in list, [start:end:jump]
  - ls[::] is entire list, ls[::-1] is list in reverse! (Why is this?)
- Lists can be sorted in place with list_name.sort()

Other Data Structures

- Dictionaries let you index things with more than just a number
  - One example could be using someone’s Twitter handle as a key, and a list of their tweets as a value.
- Tuples are similar to lists, but are of immutable size
  - (1, 2, ‘red’, True)
  - Use cases could be a pair of coordinates
- Python also supports “set”s for distinct unordered items
  - All the expected set operations such as union and intersection are supported