Programming 101

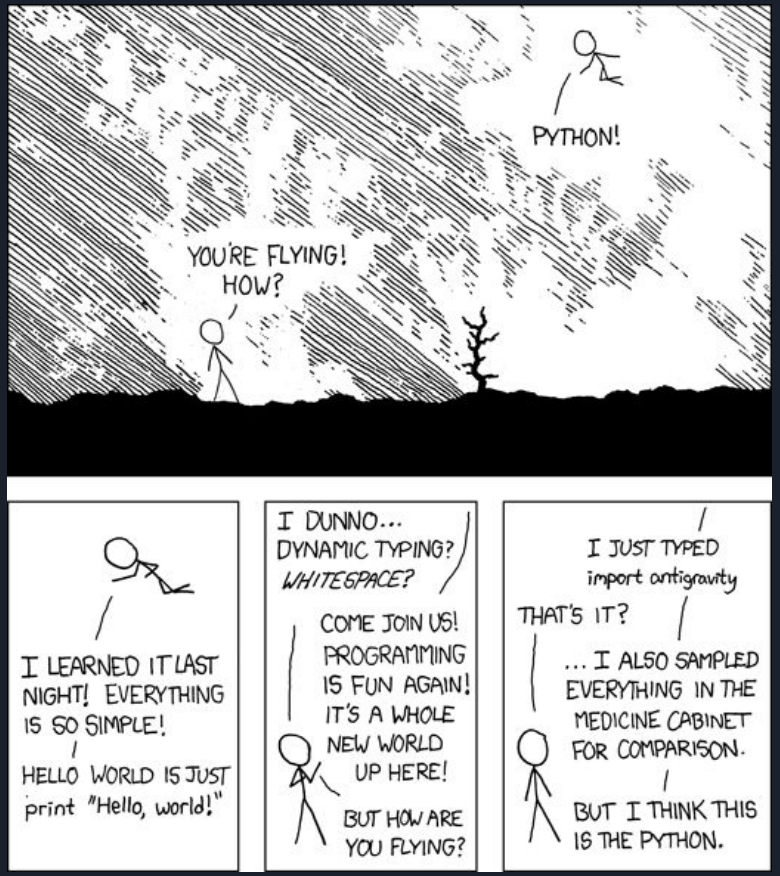
Patrick Faley, 9/15/2021

What We'll Be Covering

- Installing Python and pip
- Choosing a text editor
- Python Basics
  - Simple data types
  - Basic operations
  - Variables
  - Control Flow (if/elif/else, for, while)
- Challenge Problems!

What is Python?

- Interpreted, dynamic language
- Tons of potential applications
  - Web backend
  - Machine learning
  - Scientific computing
  - Embedded systems
- Great first language!



Installing Python

- Option 1: Anaconda - https://www.anaconda.com/products/individual-d
  - Works on Windows, MacOS, and Linux
  - Click the checkbox for “add to path” when prompted
- Option 2: for MacOS
  - Open terminal
  - “curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py”
  - “python get-pip.py”
  - “pip install ipython”
- Option 3: for Linux
  - Use your distribution’s package manager
    - apt for Ubuntu/Debian, etc.
    - yum for RHEL/CentOS
    - pacman for Arch/Manjaro, etc.
  - Install ipython (“pip install ipython”)

Choosing a Text Editor

- Visual Studio Code: https://code.visualstudio.com/Download
  - Great GUI-based text editor
- Terminal Based: useful for programming on a Raspberry Pi
  - nano - user friendly
  - vim - very powerful, steep learning curve
  - Emacs - highly customizable, also has a steep learning curve
  - Talk to me if you want help learning any of these
- IDEs: PyCharm, Eclipse, Visual Studio, etc.
  - Lots of great features, but can be intimidating to start
- Your favorite editor (if you already have one)

Getting Started with Python - Terminal

- Open your terminal
  - Powershell on Windows
  - Terminal on MacOS
  - Whatever terminal you use on Linux
  - Ctrl + “`” to open from VSCode
- Terminal: you exist at a directory
- Useful Commands:
  - “cd <directory>” - go to the specified directory
  - “ls” - show the files in the current directory
  - “pwd” - show the current path
  - “mkdir <directory>” - make a new directory

Python REPL

- Open your terminal
  - Powershell on Windows
  - Terminal on MacOS
  - Whatever terminal you use on Linux
- Type “ipython”
- REPL: Read, Evaluate, Print Loop
- Can type commands and get results back

Data Types

- Boolean: True or False
- String: collection of characters surrounded by quotes
- Integer: numbers without decimals
- Floats: numbers with decimals

Numeric Operators

Operation	Syntax	Example
Addition	x + y	3 + 4 == 7
Multiplication	x * y	3 * 4 == 12
Subtraction	x - y	3 - 4 == -1
Division	x / y	3 / 4 == 0.75
Exponent	x ** y	3 ** 4 == 81
Remainder	x % y	3 % 4 == 0
Integer Division	x // y	3 // 4 == 0

Comparison Operators

Operation	Syntax	Example
Equality	x == y	3 == 3 (True)
Inequality	x != y	3 != 4 (True)
Greater Than/Less Than	x > y, x < y	3 > 4 (False), 3 < 4 (True)
Greater/Less Than or Equal To	x >= y, x <= y	3 >= 2 (True), 3 >= 3 (True)

Boolean Operators

A	B	A and B	A or B	not A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Variables

- Kind of like variables in math
- Any variable can take any value
- Use identically to the values they have
- Note: typing print(<some value>) will print that value to the screen

```
1    x = 3
2    y = 4
3    print(x + y)    # equal to 7
```

Conditionals

- Want to run different code depending on some condition

```
5  if x < y:
6      print("x is less than y")
7  elif x > y:
8      print("x is greater than y")
9  else:
10     print("x is equal to y")
```

While Loops

- Test a condition, run some code, then repeat.

```
12    x = 0
13    while x < 5:
14        print(x)
15        x = x + 1
```

Writing a Script

- Using your favorite text editor, make a file called “hello.py”
- Save the file somewhere you can access it
- Go to that file in your terminal
  - Type “cd <directory name>”
- Start ipython there
- Type “run hello.py” to run the program

Challenges

1. Write a program to solve a quadratic equation.
2. Write a program to find the sum of the first “n” numbers, where “n” is a variable you can set.
3. (Advanced) Write a guessing game. You should let the user type a guess between 1 and 100, and then tell the user if they are too high, too low, or correct.
    a. For this one, you’ll need the “input” function to get user input, and the “random” library to generate random numbers. Talk to me for help with this one.