This course includes an Introduction to Python from an unconventional approach. Inspired by Al Sweigart’s “Automate the Boring Stuff with Python,” this course will show students just how quickly programming can be useful in just about anyone’s life. The course will span 7 weeks, from week 2 of MT18 onward.

This document will further summarize the material covered in each class.

<https://automatetheboringstuff.com/>

The idea is to make the knowledge from the course applicable as soon as possible.

**Week 1**

On week 1, we should invite people to come an hour early if they would like help setting up Anaconda, Jupyter notebooks (We are not stopping to help people set up) great idea!

To compensate for the extra time some people spend, we’ll try to keep class 1 to an hour and a half.

Concepts

Quarter 1 ---- William

* Programming
  + Why is it a useful skill?
* The Python programming language
  + Why is it useful?
  + How does it compare to other languages?

Quarter 2 -- Hazem

* Using Python as a calculator
  + Order of operations and ()
* Variable Assignment
  + Clear confusion about reassignment
    - Reassignment
  + += (increment)
  + The ‘?’ which returns docstring in jupyter notebook

Quarter 3 -- William

* Data types
  + Ints, floats, strings
  + Convert between them
  + String slicing
  + Methods, dir() function

Quarter 4 -- Hazem

* Built-in functions
* Defining a function
* Stack Overflow
* StackOverflow
* GitHub

Activities

*In all the below activities I think it’s a good idea to explain the structure of the code, so people have a good idea of what’s to come*

* Compare a python script to a C++ script to show just how much more compact Python can be Think this is potentially a waste of time, could instead emphasis that python is very great to start with. It’s also open source!! (Take that, MATLAB!)
* Demonstrate useful scripts in python, some of which (or all?) we will write in the course
  + Send a personalised and randomised email to everyone using smtplib
  + We could for instance have everyone fill out a google form to register for CodeSoc officially and then email everyone a custom “Thank you for submitting” based on their submission.

-- Snack Break --

* Google Forms Quiz
* 1-2 project Euler challenge(s) using the methods introduced and functions defined. Show how fast python can count to 1 million ( or do some other seemingly difficult task ) Through an Project Euler problem

**Week 2**

Concepts:

* Lists, Dictionaries (William)
* Booleans(Hazem)
* If statements
* For loops, while loops. (William)
* Control Flow / Order of Execution
* Importing modules(Hazem)

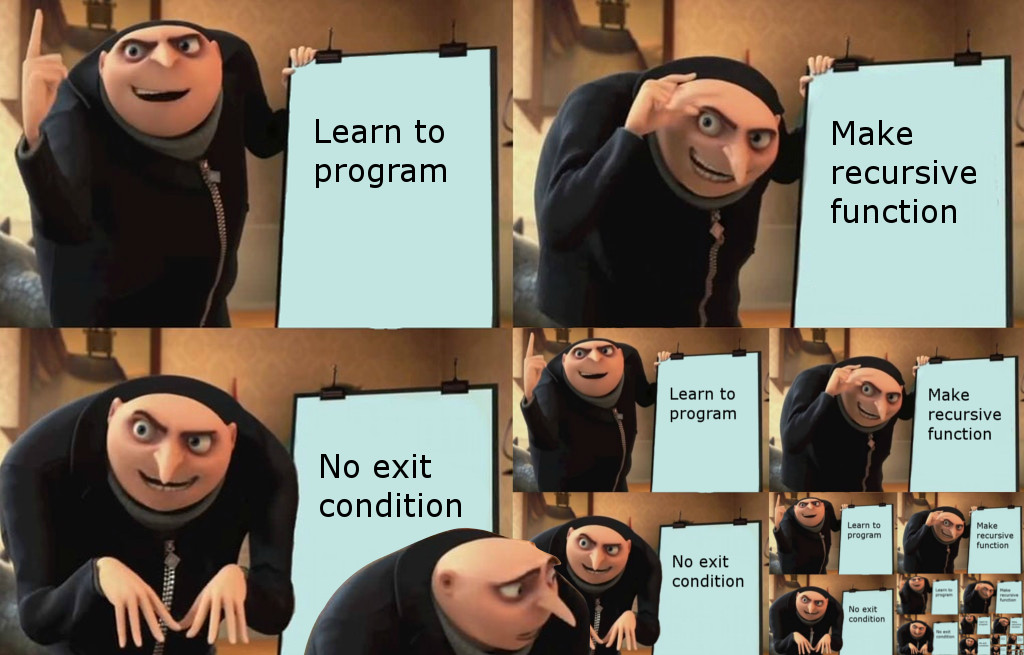
Activities

* Be clear that this is class in particular is kind of very intense, and give homework for it.
  + To make it easier, should we print out notes for everyone to take / Make slides available for everyone during the class?
  + I’m not sure if you’ve heard of this, but there’s a great book called automate the boring stuff with python, i think we should tell people about that. There’s like problems on it too
* Easier than project Euler exercises “Write a program that…” (exercises at the end of ATBSWP) (Also consider exercises from Think Python: Think like a computer scientist

-- Snack Break --

* Google Forms Quiz
* Start out with a Project Euler challenge (leave it up there for the early arrivals too)
* At the end of the class promise that we’ll automate some boring stuff next week!!

**Week 3**



Concepts:

* Recursion
* String formatting ( str.format() str.split() str.join() )
* Mostly hands on work from the week before. Example: https://www.w3resource.com/python-exercises/python-conditional-statements-and-loop-exercises.php
* SMTP, smtplib

Activities

* Write a fib(n) function using recursion
* People make fake Gmail accounts.
* Right as people learn to use SMTP, let’s get everyone to send custom emails to [example1@gmail.com](mailto:example1@gmail.com) and show the inbox on the screen (we want to make sure everyone know it’s working)
* When people send an email to [examplebot@gmail.com](mailto:examplebot@gmail.com) There are two teams. Everyone receives an email with their team number of (1,2) and a string.
* The first n fibonacci numbers of the string are a hint, like a 20-questions-answer, the two teams compete to guess what the thing is.
* I reckon the 20 questions exercise would take no more than 20 minutes. We should use the passwords to play some type of game, maybe we could choose one that’s easier and has fewer points of failure.
* After 10 minutes of the 20 questions game, if the progress is slow, we can show everyone an example function that filters all the fibs from a string.
* Prize for the team that gets the hints!!

-- Snack Break --

* Coding Quiz (Should review lists, dictionaries)
* The str.fomat() method
* Everyone sends a custom email to everyone
* With some minor adjustments, now everyone has a script that sends customized emails using a template and spreadsheet

**Week 4**

Concepts

* RegEx
* Web Scraping
  + Make a window pop up with the weather and your favorite news

**Week 5**

* Snake (?)
* Democracy!!

**Week 6**

How to learn more

-- Stack Overflow

-- How to read documentation

-- GitHub

-- Modules

-- Data Science

* Numpy
* Pandas
* Matplotlib
* SciPy

-- GUI

--Django (Web development)

-- Languages used for specific tasks

-- web development (JS, HTML, CSS)

-- Robotics (C, C++)

-- Machine Learning and scientific computing (Python, C++)]\

--SQL (For Relational Databases)

-- Useless (Java, Ruby, Perl, Lisp )

**Week 7 (Hackathon!!!)**