Python Foundations

Python

Python3

- should be mapped to just `python`
- very common scripting language
  - quick to write and test
  - slower to run compared to compiled languages
- easy to learn!
  - English-like syntax
- hard to learn!
  - English-like syntax
  - overloaded operations by default

Convenient

- precoded packages and libraries with added functionality

Inconvenient

- multiple ways to do the same thing

Building Blocks of Code

Variables

Arithmetic Operators

Conditional Logic

Iteration

Functions

How to fire up Python

Terminal

- start interpreter
- write single lines
- run through interpreter

Script

- write in IDE (.py file)
- run in terminal

Notebook (specifically Colab)

- write in browser (.ipynb file)
- run in browser
- mix between terminal and script

More on the Terminal

cd

ls

cd ..

cd ~/Documents/courses/cmpsc101/lab1

gatorgrade --config config/gatorgrade.yml

python script-example-name.py

python square/square.py

git commands in terminal

git status

git add .

don't forget the `.`

git status

git commit -m "Write what you did as reminder to yourself"

git push origin main

Code example 1

```
sum = 0
count = 0
data = [1,5,8,2,0,9,10,4]
for number in data:
  sum += number
  count += 1
print(sum/count)
```

Code example 2

```
sum = 0
count = 0
file = open("data")
for number in file:
  n = int(number)
  sum += n
  count += 1
print(sum/count)
```

Code example 3

```
def average(filename):
    sum = 0
    count = 0
    file = open(filename)
    for number in file:
      n = int(number)
      sum += n
      count += 1
    return(sum/count)
```

Code example 4

```
def average(filename: Path):
    sum = 0
    count = 0
    file = open(filename)
    for number in file:
      n = int(number)
      sum += n
      count += 1
    return(sum/count)
```