Agenda for Week 2

This week, I start with some parameters from last week: students have a *wide* range of coding experiences and needs. So tonight let's experiment with a longer breakout room/activity with the goal of people of all levels collaborating on the activities.

Keep up with your domain's trends: https://www.forbes.com/sites/shourjyasanyal/2019/05/01/5-reasons-why-doctors-should-learn-data-science/#7531205d2b85

Main topics: We review some fundamentals and ensure your computing environment is ready to go.

While we review some fundamentals, I direct you also to consider the *syntax* of python (and indeed any programming language). My pov is

- there's one way to do it (perhaps clunky but the logic is there)
- there's a better way to do it (using the optimal approaches in the programming languages) [Weeks 1-8; Big-O]
- a still-better way (using optimized libraries of code, such as Numpy, Pandas, etc.) [Week 8-14]

Agenda for Week 2

Syntax and OOP: the syntax tells us if a datum is an object (notice the period, indicating a class: e.g., .methodName(). [Copying a var to a new location with the new data; deleting the first version; pointing to the new version. *Why?*]

Example: `x = 1 + 1` *while* `x += 1` *is better . . . and then* `for/while` *loops and then* `list comprehensions`*!*

1. Control-of-flow statements
2. Variables
3. Name versus Object Space in Python
   a. "Scope and visibility" of variables
   b. `x == y` versus `x is y`
   c. Best practices
4. Python's "Type" command
   a. what's the purpose?
   b. how can you use it in practice?
5. Activities

Agenda for Week 2

Define a bicycle object prototype

attributes:

- speed
- gear

behaviours:

- speed up
- apply brake
- change gear

```
class bicycle:
    ''' properties'''
    # Class variables.
    gear = 1
    speed = 0

    def __init__(self, gear, speed):
        self.gear = gear
        self.speed = speed

    def speedUp(self, increase):
        self.speed += increase

    def changeGear(self,newGear):
        self.gear = newGear

    def applyBrake(self, decrease):
        self.speed -= decrease
```

https://www.bournetocode.com/projects/AQA_A_Theory/pages/OOP.html

https://www.geeksforgeeks.org/difference-operator-python/