Meeting 3 – object oriented programming

Dan

- Motivation: layer of abstraction and organization
- Classes are a *template* to create objects of that type
- Objects are *instances* of the class
- Syntax

```
class <name>:
    <suite>
```

- Example:

```
class Human:
    planet = "Earth"

    def __init__(self, name):
        self.name = name

    def talk(self):
        print("Hello")

ann = Human()
ann.talk()
```

- Instance attributes: a property of an object, specific to an instance
- Class attributes: a property of an object, shared by all instances of a class
- Methods: an action (function) that all instances of a class may perform; they ought to get self as the first argument
- Dot notation: `<expression>.<name>`
- The expression has to evaluate to a class or instance
- Secretly, we have already worked with objects – in fact, everything is an object in Python!
- For example, lists, strings, numbers, functions:

```
xs = [1, 2, 3]; xs.append(4)
(3).__add__(5)
```

- We can list all attributes of an instance using `dir`
- Note the methods starting and ending with double under-scores: magic methods – they are called when special syntax is performed Examples: `__add__`, `__len__`, `__call__`, `__str__`
- The `__init__` method of a class is automatically invoked whenever an object is constructed.
- The `__str__` method is invoked automatically when printing.
- The `__repr__` is invoked in an interactive session to display values; `repr` gives us all the information

Discussion on `repr` vs. `str`

```
class Dog:
    def __init__(self, color, weight):
        self.color = color
        self.weight = weight
    def __str__(self):
        return 'Using __str__'
    def __repr__(self):
        return 'Using __repr__'

puppy = Dog('white', 10)
```

And in the interpreter:

```
>>> print(puppy)
Using __str__
>>> puppy
Using __repr__
>>> new_s = 'I have a ' + str(puppy)
>>> print(new_s)
I have a Using __str__
>>> new_s
'I have a Using __str__'
>>> new_s = 'I have a ' + puppy
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be str, not Dog
```

Exercises:

- Lab 06: Q2 WWPD
- Hw 06: Q2 Vending Machine

- Hw 06: Q1 Next Fibonacci Object
- Lab 07: Q2 WWPD Linked Lists
- Lab 07: Q4 Link to List
- Lab 07: Q5 Store Digits

Homework:

- Project 3: Ants vs. SomeBees