Object Oriented Programming

Intro: What is Object Oriented Programming?

- Until now, you have been writing your code in a step-by-step fashion.
- OOP is a design pattern that organizes your system into **classes** or **objects**. (We will use those terms interchangeably for now)
  - Ex: You can define (or code) a class called Animal.

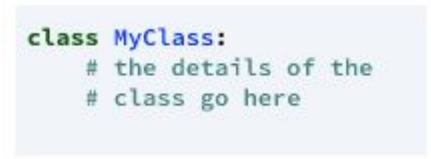
Intro: What is Object Oriented Programming? (cont)

- Those objects define attributes and methods.
- **Attributes** are variables that belong to your class.
  - Ex: Animal class has an attribute for its skin_color
- **Methods** are functions that an object can perform
  - Ex: Animal class has a method make_noise()

Intro: Why use Object Oriented Programming?

- Helps you write organized, quality code
- Reusable components
- Can clearly see how pieces interact with each other
- Hides pieces of functionality that are not necessary to see when outside of the class
- Secure - can limit access or implement validation easily

Syntax: Defining a class



```python
class MyClass:
    # the details of the
    # class go here
```

- Convention: classes are named with CamelCase
- (variable convention is to name with snake_case)

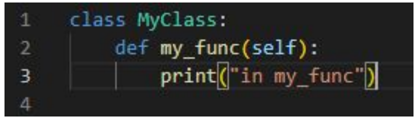
Syntax: defining a method

- Remember, a method is a function that the class performs

```
class ExampleClass:
    def greet():
        return "hello"
```

Syntax: The "self" argument

- Any method that is defined in a class needs to have "self" as its first argument. It is a way to reference the current instance of the object.
- This argument can really be called anything that you want, but we will stick to always naming it self
- It is an argument for the method, but when you are calling (invoking) the method, it does not get passed in.

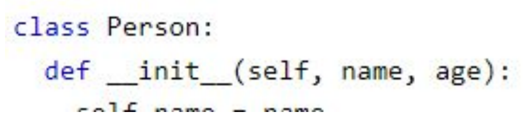


```python
class MyClass:
    def my_func(self):
        print("in my_func")

```

Syntax: The __init__() Function

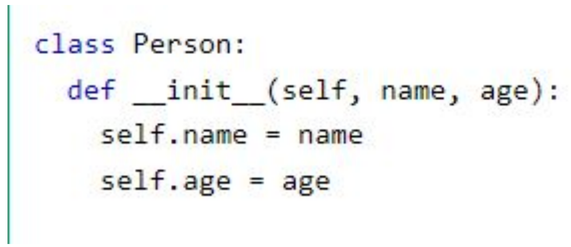
- The __init__() function is a built in function that gets called every time you use the class.
- Inside the __init__() function is where you would initialize any basic information that you want your object to start out with.



```
class Person:
  def __init__(self, name, age):
    self.name = name
```

Syntax: Attributes

- When you are coding a class, you will probably want it to have attributes or variables
    - For example, the Animal class will have an attribute for favorite_food
- The way that you define an attribute is by assigning it a value in the __init__() function



```python
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age
```

Assignment

- Write a class that will organize your house for you
- The name of the class is HomeOrganizer
- Include an __init__ method
- Include an attribute for num_shirts_to_fold
- Include 3 additional methods: (you can just add a print() statement as the body of each of the methods. Bonus points if you add something more complex).
  - sweep()
  - fold_laundry() - use the attribute num_shirts_to_fold inside this method
  - Your own creative method

**For a clear review of today's lesson, see https://www.w3schools.com/python/python_classes.asp