**Code Formatter**

* PEP8 is a code formatter guide.
* Vs code use autopep8 package for this.
* PEP8 is a community guide how to format python code.

**OOP**

* Everything in python is object.
* Everything built in class.
* We can create own methods and data types.
* OOP is a paradigm to structure and organize code.
* In object we first use clans to create a model which is blue print.
* We can define methods and actions in it.
* We create object to instantiate the class.
* Classes store in memory.

**Class**

class PlayerCharacter:

members = True

def \_\_init\_\_(*self*, *name*, *age*):

*self*.name = name # Attributes

*self*.age = age

def check(*self*):

print(f'my name is {*self*.name}')

return True

player1 = PlayerCharacter("Raheel", 30)

player1.check()

* \_\_init\_\_ is a special method. It is a constructor method. It runs when create object.
* Self refers to class.
* It will use in constructor methods.

**Attribute and Methods**

* Attributes are pieces of data and they are dynamic.
* Class has some CLASS OBJECT ATTRIBUTE which written outside constructor and they are static.
* COA are static it will same for all instances.
* To access attributes self is define before attribute.
* We can access COA with self and class name.
* We can not access construction attributes using class name.

**\_\_init\_\_**

* we can give default parameters in constructor.
* We can use safeguards with conditional.

**@classmethod and @staticmethod**

@classmethod

* For using those methods we don't have to initiate class.
* We can access them using class name.
* We can use class function to instantiate object in the class method an give the value of constructor attribute.

@staticmethod

* Same as @classmethod but not access cls parameter

**Pillars of OOP**

* Encapsulation
* Abstraction
* Inheritance
* Polymorphism

**Encapsulation**

* Binding of data and functions that manipulate that data.
* Data are attributes and Functions.
* Combining things making a package with meaning.

**Abstraction**

* Hiding of information and give access to what is necessary.
* It can be seen in class function or built-in function user not to worry what is the function user just use it.

**Inheritance**

* Inherit classes.
* Allows new object to take up the properties of existing classes.
* \_\_inti\_\_ if we don’t have any attribute we don’t use it.
* It keeps code DRY and Abstracting the the put away we don’t need.
* isinstance(subclass, derived class) it shows where the class derived from.
* Everything in python inherit base python object.

**Polymorphism**

* Poly means many
* morphism means forms.
* The way in which classes can share same method name but those method names can act differently based on what object calls them.
* Polymorphism will override parent methods