Defining a Class in Python ¶

- Like function definitions begin with the keyword def, in Python, we define a class using the keyword class.
- The first string is called docstring and has a brief description about the class. Although not mandatory, this is recommended.

Example:

In [1]:

```
class MyNewClass:
    '''This is a docstring. I have created a new class'''
    pass
```

- A class creates a new local namespace where all its attributes are defined. Attributes may be data or functions.
- As soon as we define a class, a new class object is created with the same name. This class object allows us to access the different attributes as well as to instantiate new objects of that class.

In [6]:

```
class Myclass:
    '''This is demo class'''
    a = 50
    def func(self):
        print ('Hello Students')
print(Myclass.a)
print(Myclass.func)
print(Myclass.__doc__)
```

```
50
<function Myclass.func at 0x7f14486812f0>
This is demo class
```

Creating an Object in Python

The procedure to create an object is similar to a function call.

```
ob = MyClass()
```

- **This will create a new instance object named ob. We can access attributes of objects using the object name prefix.**
- **Attributes may be data or method.**
- **Method of an object are corresponding functions of that class.**
- **Any function object that is a class attribute defines a method for objects of that class.**
    - **This means to say, since MyClass.func is a function object (attribute of class), ob.func will be a method object**

In [12]:

```
class MyClass:
    '''This is demo class'''
    a = 50
    def func(self):
        print('Hello Students')

ob = MyClass()

print(ob.a)

ob.func()
```

```
50
Hello Students
```

- **You may have noticed the self parameter in function definition inside the class but, we called the method simply as ob.func() without any arguments. It still worked.**
- **This is because, whenever an object calls its method, the object itself is passed as the first argument. So, ob.func() translates into MyClass.func(ob).**
- **In general, calling a method with a list of n arguments is equivalent to calling the corresponding function with an argument list that is created by inserting the method's object before the first argument.**
- **For these reasons, the first argument of the function in class must be the object itself. This is conventionally called self. It can be named otherwise but we highly recommend to follow the convention.**

In [ ]:

```
```