Basic Python 5 - Classes and Object

December 4, 2020

1 Python Classes and Objects

Python is an object oriented programming language. Almost everything in Python is an object, with its properties and methods. A Class is like an object constructor, or a "blueprint" for creating objects.

1.1 Create a class

[ ]:
```
class MyClass:
    x = 5
```

1.2 Create an object

The examples below are not really useful in real life applications.

[ ]:
```
p1 = MyClass()
print(p1.x)
```

1.3 ___init___() function

All classes have a function called ___init___(), which is always executed when the class is being initiated. Use the ___init___() function to assign values to object properties, or other operations that are necessary to do when the object is being created:

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

[ ]:
```
p1 = Person("John", 36)

print(p1.name)
print(p1.age)
```

1.4 Object methods

Objects can also contain methods. Methods in objects are functions that belong to the object. Let us create a method in the Person class:

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

    def myfunc(self):
        print("Hello my name is " + self.name)
```

[ ]:
```python
p1 = Person("John", 36)
p1.myfunc()
```

1.5 Example

[ ]:
```python
class Car:
    def __init__(self,brand,horsePower,color):
        self.brand = brand
        self.horsePower = horsePower
        self.color = color
        self.currentPosition = 0

    def start(self):
        print("start the car")
    def moveForward(self):
        self.currentPosition = self.currentPosition + self.horsePower
    def moveBackward(self):
        self.currentPosition = self.currentPosition - self.horsePower
```

[ ]:
```python
car_ricky = Car("honda",200,"red")
car_tao = Car("toyota",300,"blue")
```

[ ]:
```python
print(car_tao.currentPosition)
car_tao.start()
car_tao.moveForward()
print(car_tao.currentPosition)
```

1.6 Importing class definition

myclass.py

```python
class CarNew:
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year

    def get_descriptive_name(self):
        long_name = str(self.year) + " " + self.make + " " + self.model
        return long_name.title()
```

[ ]:
```
from myclass import CarNew
```

[ ]:
```
my_new_car = CarNew('audi', 'a4', 2016)
my_new_car.get_descriptive_name()
```

[ ]:
```
import myclass
my_new_car = myclass.CarNew('tesla', 'model s', 2020)
my_new_car.get_descriptive_name()
```

[ ]:
```
import myclass as mc
my_new_car = mc.CarNew('toyota', 'Camry', 2008)
my_new_car.get_descriptive_name()
```