Python_L_Static_Method_[Day_18](DATA_MINDS)

September 18, 2023

STATIC METHOD -: FOR REPETEADLY CREATING INSTANT OF OBJECT FUNCTION WE USE TO CREATE A “STATIC FUNCTION” THAT DIRECTLY BIND WITH THE CLASS

IMPORTANT - STATIC METHOD ALSO HELP TO ACHIEVE THE MEMORY OPTIMIZATION IN A CLASS

[2]:

```python
class data_science:
    def studenet_details(self,name ,mail_id , mon_no):
        print(name ,mail_id , mon_no)
```

[8]:

```python
ds=data_science()
```

[11]:

```python
ds.studenet_details("Virat Tiwari ,","Virat@gmail.com ,",846971036458)
```

```
Virat Tiwari , Virat@gmail.com , 846971036458
```

[20]:

```python
class data_science1:
    def studenet_details(self,name ,mail_id , mon_no):
        print(name ,mail_id , mon_no)
    @staticmethod
    def mentor_class(list_mentor):
        print(list_mentor)
    def mentor(self,mentor_list):
        print(mentor_list)
```

[21]:

```python
data_science1.mentor_class(["Virat Tiwari","Virat@gmail.com"])
```

```
['Virat Tiwari', 'Virat@gmail.com']
```

[22]:

```python
stu1=data_science1()
```

[23]:

```python
stu2=data_science1
```

[24]:

```python
stu3=data_science1
```

[28]:

```python
stu1.mentor(["Virat Tiwari","Yash Verma"])
```

```
['Virat Tiwari', 'Yash Verma']
```

[39]:

```
class data_science2:
    def studenet_details(self,name ,mail_id , mon_no):
        print(name ,mail_id , mon_no)
    @staticmethod
    def mentor_class(list_mentor):
        print(list_mentor)
    @classmethod
    def class_name(cls):
        cls.mentor_class(["Virat Tiwari","Yash Verma"])
    def mentor(self,mentor_list):
        print(mentor_list)
```

[40]:

```
data_science2.class_name()
```

```
['Virat Tiwari', 'Yash Verma']
```

[69]:

```
class data_science2:
    def studenet_details(self,name ,mail_id , mon_no):
        print(name ,mail_id , mon_no)

    @staticmethod
    def mentor_mail_id(mail_id_mentor):
        print(mail_id_mentor)

    @staticmethod
    def mentor_class(list_mentor):
        data_science2.mentor_mail_id(["Virat@gmail.com","Yash@gmail.com"])
        print(list_mentor)

    @classmethod
#(cls) - it refers to the object
    def class_name(cls):
        cls.mentor_class(["Virat Tiwari","Yash"])

    def mentor(self,mentor_list):
        print(mentor_list)
        self.mentor_class(["Virat Tiwari","Yash"])
```

[70]:

```
data_science2.mentor_class(["Virat Tiwari","Yash"])
```

```
['Virat@gmail.com', 'Yash@gmail.com']
['Virat Tiwari', 'Yash']
```

[71]:

```
data_science2.class_name()
```

```
['Virat@gmail.com', 'Yash@gmail.com']
['Virat Tiwari', 'Yash']
```

[76]:

```
ds=data_science2()
```

[78]:

```
ds.mentor(["Virat Tiwari","Yash"])
```

```
['Virat Tiwari', 'Yash']
['Virat@gmail.com', 'Yash@gmail.com']
['Virat Tiwari', 'Yash']
```

NOTE - FOR WRITING FUCTION FOR FILE STORAGE AND SOME OTHER THING WE DON'T NEED TO WRITE IT AGAIN AND AGAIN WE SHOULD USE STATIC FUCNTION( ) THAT WILL WRITE ONE TIME AND WE CALL IT AND USE IT AGAIN AGAIN WITHOUT WRITING NEW FUNCTION

IMPORTANT - STATIC METHOD ALSO HELP TO ACHIEVE THE MEMORY OPTIMIZATION IN A CLASS