[Hindi] Static Methods In Python oops | Object Oriented Programming In Python Tutorial #6

Static Methods In Python oops

In the previous article, we have discussed how to work with class methods. In this article, we are going to discuss the static method. Take a look at the code given below to remember the work that we've done in the previous tutorial:

```
class Employee:
    increment = 1.5
    no_of_employe= 0

    def __init__(self,fname,lname,salary):
        self.fname=fname
        self.lname=lname
        self.salary=salary
        self.increment=1.4
        Employee.no_of_employe +=1

    @classmethod
    def from_str(cls,emp_string):
```

```
        fname,lname,salary= emp_string.split("-")
        return cls(fname,lname,salary)

harry = Employee("harry","jackson",4400)
lovish = Employee.from_str("lovish-jackson-7600")
rohan = Employee("rohan","das",4400)
```

In the above code, we have created class Methods to deal with class variables. The class variable is used when we don't have any requirements for the instance variable. Now a question must have popped in your mind: Which method should be used if we don't want to work with both the class method and the instance method? To deal with this scenario, the *static method* is taken into consideration.

What is static method?

- It is one type of decorator used when you don't want any class and instance variable parameters. This is just a simple function.
- Static methods deal with the parameters, and they know nothing about the class.

Syntax:

```
@staticmethod
def method_name(argument):
    pass
```

Working with our Program:

1. We are creating a static method with the name *isopen* with the argument that will be a string in datatype.
2. The approach of the program will be, if we pass the string "sunday" the function should return False else should return True.

Method:

```
@staticmethod
def isopne(day):
       if day=="sunday":
            return False
        else:
            return True
```

3. Assume, this function will return false when the company is closed and will return True in the working days.

4. Remember that, this method is neither taking class variable nor instance variable this is just a simple python function.

Getting the output:

1. With class:

```
print(Employee.isopen("Monday"))
print(Employee.isopen("sunday"))
print(Employee.isopen("tuesday"))
```

Output:

```
True
False
```

2. With Instance variable:

```
print(harry.isopen("sunday"))
print(harry.isopen("Monday"))
print(harry.isopen("tuesday"))
```

Output:

Copy

```
False
True
True
```

This tutorial ends here. I hope this tutorial was valuable for you. In the next article, we will study about inheritance. Till then, keep coding!

Previous Next