Static Variables and Methods in Python

Defining static variable and method is a common programming concept and is widely used in C++, Java, Php and many other programming languages for creating class variables and methods that belong to the class and are shared by all the objcts of the class.

In Python, there is no special keyword for creating static variables and methods. Python follows a different but simple approach for defining static variables and methods which we will learn in this tutorial.

***Class or Static Variables in Python***

Class or Static variables are the variables that belong to the class and not to objects. Class or Static variables are shared amongst objects of the class. All variables which are assigned a value in the class declaration are class variables. And variables which are assigned values inside class methods are instance variables.

Let's have an example to understand this:

In the above program, cat is a class variable because it is defined outside of all the class methods and inside the class definition and type is an instance variable as it is defined inside a method.

This is confirmed using the print statement where the cat variable is referenced using the class name Shape while the type variable is referenced using the different object references.

The above example shows a scenario where there are different shape objects each belonging to the same category which is **Geometrical** but are of different types, so each object of the class have the same category which we have made the class variable and the type variable is different for all the objects hence it is an instance variable.

**NOTE:** Python allows providing same variable name for a class/static variable and an instance variable. But we would recommend you to not to provide same name variables to these variables to avoid confusion.

***Static Methods in Python***

Just like static variables, static methods are the methods which are bound to the class rather than an object of the class and hence are called using the class name and not the objects of the class.

As static methods are bound to the class hence they cannot change the state of an object.

To call a static method we don't need any class object it can be directly called using the class name.

In python there are two ways of defining a static method:

1. Using the staticmethod()
2. Using the @staticmethod

Define static method using staticmethod()

Let's take an example to see how this is done:

class Shape:

def info(msg):

# show custom message

print(msg)

print("This class is used for representing different shapes.")

# create info static method

Shape.info = staticmethod(Shape.info)

Shape.info("Welcome to Shape class")

Welcome to Shape class

This class is used for representing different shapes.

In the program above, we declared the info method as static method outside the class using the staticmethod() function approach and after that we were able to call the info() method directly using the class Shape.

Define static method using @staticmethod

Let's take an example to see how this is done:

class Shape:

@staticmethod

def info(msg):

# show custom message

print(msg)

print("This class is used for representing different shapes.")

Shape.info("Welcome to Shape class")

Welcome to Shape class

This class is used for representing different shapes.

Using @staticmethod is a more modern approach of defining static method and we recommend this approach.

Points to Remember about static variable and methods

Following are some important take aways:

1. Static variable and methods are used when we want to define some behaviour or property specific to the class and which is something common for all the class objects.
2. If you look closely, for a static method we don't provide the argument self because static methods don't operate on objects.