Types Of Methods In Python - Instance Method, Class Method, and Static Method

Before moving on with the topic, we have to know some key concepts.

**Class Variable:**A class variable is nothing but a variable that is defined outside the constructor. A class variable is also called as a **static variable**.

**Accessor(Getters):** If you want to fetch the value from an instance variable we call them accessors.

**Mutator(Setters):**If you want to modify the value we call them mutators.

## **Instance Method**

This is a very basic and easy method that we use regularly when we create [classes in python](https://www.studytonight.com/python/class-in-python). If we want to print an instance variable or instance method, we must create an object of that required class.

If we are using self as a function parameter or in front of a variable, that is nothing but the calling instance itself.

As we are working with instance variables we use self-keyword.

**Class method**

The **classmethod()** is an inbuilt function in Python, which returns a class method for a given function.

***Syntax:****classmethod(function)*

***Parameter :****This function accepts the function name as a parameter.*

***Return Type:****This function returns the converted class method.*

A class method is a method that is bound to a class rather than its object. It doesn't require creation of a class instance, much like static method.

The difference between a static method and a class method is:

* Static method knows nothing about the class and just deals with the parameters
* Class method works with the class since its parameter is always the class itself.
* def classMethod(cls, args...)

**The class method can be called both by the class and its object**.

**class C(object):**

**@classmethod**

**def fun(cls, arg1, arg2, ...):**

....

**fun:** the function that needs to be converted into a class method

**returns:** a class method for function.

* A class method is a method which is bound to the class and not the object of the class.
* They have the access to the state of the class as it takes a class parameter that points to the class and not the object instance.
* It can modify a class state that would apply across all the instances of the class. For example, it can modify a class variable that would be applicable to all the instances.

In the below example we use a staticmethod() and classmethod() to check if a person is an adult or not.