



Class Methods in Python | Python Tutorial - Day #69



[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

Day 69 – Python Class Methods

Python Class Methods: An Introduction

In Python, classes are a way to define custom data types that can store data and define functions that can manipulate that data. One type of function that can be defined within a class is called a "method." In this blog post, we will explore what Python class methods are, why they are useful, and how to use them.

What are Python Class Methods?

A class method is a type of method that is bound to the class and not the instance of the class. In

other words, it operates on the class as a whole, rather than on a specific instance of the class. Class methods are defined using the "@classmethod" decorator, followed by a function definition. The first argument of the function is always "cls," which represents the class itself.

Why Use Python Class Methods?

Class methods are useful in several situations. For example, you might want to create a factory method that creates instances of your class in a specific way. You could define a class method that creates the instance and returns it to the caller. Another common use case is to provide alternative constructors for your class. This can be useful if you want to create instances of your class in multiple ways, but still have a consistent interface for doing so.

How to Use Python Class Methods

To define a class method, you simply use the "@classmethod" decorator before the method definition. The first argument of the method should always be "cls," which represents the class itself. Here is an example of how to define a class method:

```
class ExampleClass:
    @classmethod
    def factory_method(cls, argument1, argument2):
        return cls(argument1, argument2)
```

In this example, the "factory_method" is a class method that takes two arguments, "argument1" and "argument2." It creates a new instance of the class "ExampleClass" using the "cls" keyword, and returns the new instance to the caller.

It's important to note that class methods cannot modify the class in any way. If you need to modify the

class, you should use a class level variable instead.

Conclusion

Python class methods are a powerful tool for defining functions that operate on the class as a whole, rather than on a specific instance of the class. They are useful for creating factory methods, alternative constructors, and other types of methods that operate at the class level. With the knowledge of how to define and use class methods, you can start writing more complex and organized code in Python.

[Previous](#)[Next](#)**CodeWithHarry**

Copyright © 2023 CodeWithHarry.com

