



Magic/Dunder Methods in Python | Python Tutorial - Day #73



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

Day 73 – Magic/Dunder Methods in Python

These are special methods that you can define in your classes, and when invoked, they give you a powerful way to manipulate objects and their behaviour.

Magic methods, also known as “dunders” from the double underscores surrounding their names, are powerful tools that allow you to customize the behaviour of your classes. They are used to implement special methods such as the addition, subtraction and comparison operators, as well as some more advanced techniques like descriptors and properties.

Let’s take a look at some of the most commonly used magic methods in Python.

`__init__` method

The **init** method is a special method that is automatically invoked when you create a new instance of a class. This method is responsible for setting up the object's initial state, and it is where you would typically define any instance variables that you need. Also called "constructor", we have discussed this method already

`__str__` and `__repr__` methods

The **str** and **repr** methods are both used to convert an object to a string representation.

The **str** method is used when you want to print out an object, while the **repr** method is used when you want to get a string representation of an object that can be used to recreate the object.

`__len__` method

The **len** method is used to get the length of an object. This is useful when you want to be able to find the size of a data structure, such as a list or dictionary.

`__call__` method

The **call** method is used to make an object callable, meaning that you can pass it as a parameter to a function and it will be executed when the function is called. This is an incredibly powerful tool that allows you to create objects that behave like functions.

These are just a few of the many magic methods available in Python. They are incredibly powerful tools that allow you to customize the behaviour of your objects, and can make your code much cleaner and easier to understand. So if you're looking for a way to take your Python code to the next level, take some time to learn about these magic methods.

[Previous](#)

[Next](#)





CodeWithHarry

Copyright © 2023 CodeWithHarry.com

