</> CodeWithHarry

HTML CSS JS C C++ JAVA PYTHON PHP REACT JS

Decorators in Python | Python Tutorial - Day #59

Overview Q&A Downloads Announcements

Day 59 - Python Decorators

Python decorators are a powerful and versatile tool that allow you to modify the behavior of functions and methods. They are a way to extend the functionality of a function or method without modifying its source code.

A decorator is a function that takes another function as an argument and returns a new function that modifies the behavior of the original function. The new function is often referred to as a "decorated" function. The basic syntax for using a decorator is the following:

```
@decorator_function
def my_function():
    pass
```

The @decorator_function notation is just a shorthand for the following code:

```
def my_function():
    pass
my_function = decorator_function(my_function)
```

Decorators are often used to add functionality to functions and methods, such as logging, memoization, and access control.

Practical use case

One common use of decorators is to add logging to a function. For example, you could use a decorator to log the arguments and return value of a function each time it is called:

```
import logging

def log_function_call(func):
    def decorated(*args, **kwargs):
        logging.info(f"Calling {func.__name__} with args={args}, kwargs={kwargs}")
        result = func(*args, **kwargs)
        logging.info(f"{func.__name__} returned {result}")
        return result
```

```
    return decorated

@log_function_call
def my_function(a, b):
    return a + b
```

In this example, the log_function_call decorator takes a function as an argument and returns a new function that logs the function call before and after the original function is called.

Conclusion

Decorators are a powerful and flexible feature in Python that can be used to add functionality to functions and methods without modifying their source code. They are a great tool for separating concerns, reducing code duplication, and making your code more readable and maintainable.
In conclusion, python decorators are a way to extend the functionality of functions and methods, by modifying its behavior without modifying the source code. They are used for a variety of purposes, such as logging, memoization, access control, and more. They are a powerful tool that can be used to make your code more readable, maintainable, and extendable.

For Next Lesson Please Click On The "Next" Button Below.

Previous Next