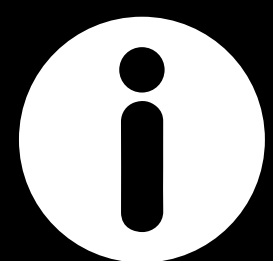


# decorators

Decorators are Python design patterns used to give new functionality to objects (functions), modifying their behavior without altering their structure: they are functions that modify functions.



*Functions in Python support operations such as being assigned to a variable, passed as an argument, and being returned by another function as a result.*

*Also, it is possible to define functions within functions, not available outside of the function within which they were defined.*

*Decorators allow a function to be modified in certain scenarios, without repeating code.*

```
def show_information(function):  
  
    def interior():  
        print(f'Running the function {function.__name__}')  
        function()  
        print('Execution finished')  
  
    return interior  
  
def some_message():  
  
    print("Hello World")
```

```
decorated_function = show_information(some_message)
```

```
decorated_function ()
```

```
>>Running the function some_message  
>>Hello World  
>>Execution finished
```

> Function as parameter of a function

> Definition of a function within another

> Function that returns another function as a result

> Assign a function to a variable

> Execution of decorated function

> Extended functionality