

# Декоратори

У мовам програмування, декоратори — це конструкція, яка дозволяє модифікувати функціональність функцій або класів без зміни їхнього вихідного коду. Це дозволяє вставляти або змінювати логіку до, під час або після виклику функції.

Основна ідея декораторів полягає в тому, щоб функція (або клас) приймала іншу функцію як аргумент, модифікувала її або виконала якісь додаткові дії та повернула нову функцію (або об'єкт, якщо це клас).

Декоратори є важливим елементом багатьох мов програмування, і вони мають різні синтаксичні конструкції залежно від мови. Однак загальний принцип залишається той же: вони дозволяють розширити функціональність без прямого втручання в оригінальний код.

```
def my_decorator(func):
    def wrapper(*args, **kwargs):
        print("Before function call")
        result = func(*args, **kwargs)
        print("After function call")
        return result
    return wrapper

@my_decorator
def my_function():
    print("Function is called")

my_function()
```

Принцип роботи декоратора полягає в тому, що він додає або змінює функціональність іншої функції, не змінюючи її вихідний код. Основна ідея - це прийом функції як аргументу і повернення іншої функції.

Ось кроки, які відбуваються при використанні декоратора:

1. **Створення декоратора:** Ви створюєте функцію, яка приймає іншу функцію як аргумент. Ця функція зазвичай внутрішньо містить додатковий код, який додає новий функціонал до переданої функції.
2. **Створення функції-обгортки (wrapper function):** Усередині декоратора ви визначаєте нову функцію, яку називають "обгорткою" (wrapper). Ця функція

викликається замість оригінальної функції.

3. **Додавання додаткової логіки:** У функції-обгортці ви можете додавати будь-яку додаткову логіку перед та/або після виклику оригінальної функції.
4. **Виклик оригінальної функції:** Ви викликаєте оригінальну функцію всередині функції-обгортки, можливо, з певними аргументами, які передались у функцію-обгортку.
5. **Повернення результату:** Якщо потрібно, функція-обгортка може повертати результат виклику оригінальної функції.
6. **Повернення декорованої функції:** Декоратор зазвичай повертає функцію-обгортку, щоб можна було використовувати декоровану функцію.

Отже, коли ви застосовуєте декоратор до функції, ви фактично замінюєте посилання на оригінальну функцію на посилання на обгорнуту функцію. При виклику декорованої функції виконується функція-обгортка, яка може додати свою власну логіку перед і після виклику оригінальної функції.

Цей механізм дозволяє легко додавати нові функції до існуючого коду без необхідності вносити зміни в оригінальний код.