

Monkey patch

Monkey patch là kỹ thuật thay đổi hoặc mở rộng hành vi của **một hàm, method, class, hoặc module tại thời điểm runtime** (lúc chương trình đang chạy) **mà không sửa mã gốc**.

Nó được dùng khi:

- Không thể/chưa thể sửa mã nguồn gốc
- Muốn mở rộng thư viện bên thứ 3
- Override tạm thời một behavior

Nói đơn giản:

“ **Bạn thay thế một hàm hoặc class trong thư viện gốc bằng hàm của bạn**
– ngay lúc chương trình đang chạy!

Cấu trúc cơ bản

```
# Module gốc (giả sử không thể sửa)
class Animal:
    def speak(self):
        return "Gâu gâu"

# Monkey patch hàm speak
def meow(self):
    return "Meo meo"

Animal.speak = meow

# Test
a = Animal()
print(a.speak()) # [] Meo meo
```

Ví dụ với module tiêu chuẩn Python

Giả sử bạn muốn override `math.sqrt` để log mỗi lần gọi:

```
import math

original_sqrt = math.sqrt

def logged_sqrt(x):
    print(f"√{x} was called")
    return original_sqrt(x)

math.sqrt = logged_sqrt

print(math.sqrt(9)) # √ log + 3.0
```

Monkey patch method của class

```
class User:
    def greet(self):
        return "Hello"

def custom_greet(self):
    return "Xin chào!"

User.greet = custom_greet

u = User()
print(u.greet()) # Xin chào!
```

Monkey patch trong unit test

Rất phổ biến khi test cần thay thế tạm hàm gọi API, DB, hoặc logic tốn thời gian.

```
import myapp.utils

original = myapp.utils.get_current_time

def fake_time():
    return "2025-01-01"

myapp.utils.get_current_time = fake_time
```

Cách hiện đại (với `unittest.mock`)

```
from unittest.mock import patch

with patch('myapp.utils.get_current_time', return_value="2025-01-01"):
    print(myapp.utils.get_current_time()) # 2025-01-01
```

Kỹ thuật monkey patch nâng cao

1. Patch theo điều kiện

```
if settings.DEBUG:
    module.func = debug_version
else:
    module.func = prod_version
```

2. Chỉ patch một instance (không toàn bộ class)

```
import types

class A:
    def say(self): return "Hello"

a = A()
a.say = types.MethodType(lambda self: "Xin chào", a)

print(a.say()) # Xin chào
```

Monkey patch và thư viện bên thứ 3

Ví dụ override hàm trong thư viện requests:

```
import requests

_original_get = requests.get

def my_get(*args, **kwargs):
    print(f"GET called: {args[0]}")
    return _original_get(*args, **kwargs)
```

```
requests.get = my_get
```

Tác giả: Đỗ Ngọc Tú
Công Ty Phần Mềm VHTSoft

Phiên bản #4

Được tạo 15 tháng 4 2025 16:25:13 bởi Đỗ Ngọc Tú

Được cập nhật 15 tháng 4 2025 17:01:55 bởi Đỗ Ngọc Tú