

# Callable

Trong Python, `Callable` là một kiểu dữ liệu đặc biệt dùng để đại diện cho **một hàm (function)** hoặc **bất kỳ thứ gì có thể được gọi**, ví dụ: hàm, phương thức, hoặc class có định nghĩa `__call__`.

```
from typing import Callable

def run(callback: Callable[[int, str], bool]):
    ...
```

## Khi nào dùng `Callable`?

- Khi bạn truyền **hàm làm đối số** cho một hàm khác
- Khi bạn muốn đảm bảo rằng đầu vào là **một callable đúng định dạng**
- Giúp **IDE** và trình kiểm tra type hiểu đúng hàm bạn đang xử lý

```
Callable[[ArgType1, ArgType2, ...], ReturnType]
```

### Ví dụ cụ thể

Viết một hàm `run_operation` nhận vào một hàm (callable) và hai số, sau đó gọi hàm đó với hai số đó.

```
from typing import Callable

# Hàm chính nhận vào một Callable
def run_operation(operation: Callable[[int, int], int], a: int, b: int) -> int:
    return operation(a, b)

# Một số hàm cụ thể
def add(x: int, y: int) -> int:
    return x + y

def multiply(x: int, y: int) -> int:
    return x * y

# Gọi thử
```

```
print(run_operation(add, 3, 5))    # 8
print(run_operation(multiply, 3, 5)) # 15
```

Giải thích

Thành phần	Ý nghĩa
<code>Callable[[int, int], int]</code>	Đây là một kiểu dữ liệu đại diện cho một <b>hàm nhận 2 số nguyên và trả về một số nguyên</b>
<code>operation(a, b)</code>	Hàm được truyền vào sẽ được gọi như bình thường
<code>add</code> & <code>multiply</code>	Là những hàm cụ thể bạn truyền vào

Mở rộng

Bạn có thể dùng `Callable` với các hàm lambda, hoặc class có `__call__`:

```
class Subtract:
    def __call__(self, x: int, y: int) -> int:
        return x - y

sub = Subtract()
print(run_operation(sub, 10, 3)) # 7
```

Tổng kết

- `Callable` dùng để **type-hint cho function** hoặc **object có thể gọi**
- Cú pháp là `Callable[[arg1_type, arg2_type], return_type]`
- Hữu ích trong lập trình hướng hàm (functional programming), LangGraph nodes, và callback logic