

yield nâng cao

yield **nâng cao** – tức là những cách dùng **yield** ở mức cao hơn, như:

1. **yield** như **kênh giao tiếp 2 chiều**
2. **yield from** để **lồng generator**
3. **generator pipeline** – **xử lý dữ liệu từng bước**
4. **send()** và **close()** – điều khiển luồng nâng cao

1. yield như một kênh 2 chiều

Không chỉ dùng để trả về giá trị, **yield** còn có thể nhận giá trị từ bên ngoài bằng cách kết hợp với **send()**.

```
def echo():
    while True:
        x = yield
        print("Nhận được:", x)

g = echo()
next(g)      # Khởi động generator
g.send("Hello") # => Nhận được: Hello
g.send("World") # => Nhận được: World
```

2. yield from - Lồng generator

Nếu bạn có nhiều generator con, bạn có thể "gom" chúng lại bằng **yield from**

```
def numbers():
    yield from [1, 2, 3]

def letters():
    yield from ['a', 'b', 'c']

def combo():
    yield from numbers()
    yield from letters()

for val in combo():
```

```
print(val)
```

Kết quả:

```
1
2
3
a
b
c
```

3. Pipeline generator - Xử lý dữ liệu từng bước

Bạn có thể xâu chuỗi nhiều generator để tạo pipeline xử lý dữ liệu rất gọn gàng:

```
def read_lines(lines):
    for line in lines:
        yield line

def to_upper(lines):
    for line in lines:
        yield line.upper()

def starts_with_a(lines):
    for line in lines:
        if line.startswith('A'):
            yield line

data = ["apple", "banana", "Avocado", "Apricot", "mango"]
pipeline = starts_with_a(to_upper(read_lines(data)))

for line in pipeline:
    print(line)
```

```
APPLE
AVOCADO
APRICOT
```

Pipeline này giống như filter + map + reduce nhưng **tiết kiệm RAM và rất "Pythonic"!**

4. send và close nâng cao

Gửi dữ liệu vào generator

```
def running_total():
    total = 0
    while True:
        x = yield total
        if x is None:
            break
        total += x

gen = running_total()
print(next(gen))    # 0
print(gen.send(5))  # 5
print(gen.send(3))  # 8
gen.close()         # Dừng generator
```

Tổng kết

Tính năng	Mô tả ngắn
yield	Trả về giá trị từng bước
yield from	Kết hợp nhiều generator
send(value)	Gửi dữ liệu vào generator
close()	Đóng generator
Generator pipeline	Kết hợp xử lý dữ liệu theo luồng

Tác giả: **Đỗ Ngọc Tú**
Công Ty Phần Mềm **VHTSoft**