



# MAP VÀ FILTER



# 1. Map():



**Map()** là một hàm trong Python có chức năng apply một hàm có sẵn cho mọi phần tử trong 1 iterable (list, tuple, str...)

## CÚ PHÁP

```
map(function, iterable...)
```

- **Function:** Hàm sẽ được sử dụng để apply với các phần tử trong iterable.
- **Iterable:** Các iterable ví dụ như list, tuple, str...
- **Giá trị trả về:** map trả về một đối tượng thuộc lớp map. Các bạn nên ép sang list để sử dụng.

# 1. Map():

## VÍ DỤ

### EXAMPLE

**Apply một hàm tự định nghĩa:**

```
def add(n):  
    return n + 100  
  
if __name__ == '__main__':  
    a = range(5)  
    b = list(map(add, a))  
    print(b)
```

### OUTPUT

```
[100, 101, 102, 103, 104]
```

### EXAMPLE

**Apply một hàm built-in:**

```
if __name__ == '__main__':  
    s = "28tech"  
    b = list(map(ord, s))  
    print(b)
```

### OUTPUT

```
[50, 56, 116, 101, 99, 104]
```

# 1. Map():

## VÍ DỤ

### EXAMPLE

**Đọc 1 mảng số nguyên từ input và chuyển vào list:**

```
if __name__ == '__main__':  
    a = list(map(int, input().split()))  
    print(a)
```

**INPUT**

1 2 3 5

**OUTPUT**

[1, 2, 3, 5]

### EXAMPLE

**Trong trường hợp hàm apply trong map ngắn gọn bạn có thể thay thế bằng lambda**

```
def square(n):  
    return n ** 2  
  
if __name__ == '__main__':  
    a = [1, 2, 3, 5]  
    b = list(map(square, a))  
    print(b)  
    c = list(map(lambda x : x ** 2, a))  
    print(c)
```

**OUTPUT**

[1, 4, 9, 25]

[1, 4, 9, 25]

# 1. Map():

## VÍ DỤ

### EXAMPLE

#### Áp dụng map với nhiều iterable:

```
def add(a, b):  
    return a + b
```

```
if __name__ == '__main__':  
    a = [1, 2, 3, 4]  
    b = [2, 3, 5, 4]  
    c = list(map(add, a, b))  
    print(c)
```

**OUTPUT**`[3, 5, 8, 8]`

### EXAMPLE

#### Áp dụng map với nhiều iterable sử dụng trong lambda:

```
if __name__ == '__main__':  
    a = [1, 2, 3, 4]  
    b = [2, 3, 5, 4]  
    c = list(map(lambda x, y : x + y, a, b))  
    print(c)
```

**OUTPUT**`[3, 5, 8, 8]`

## 2. Filter():



**Filter()** được sử dụng để trích xuất các phần tử trong một iterable khi apply một hàm nào đó với phần tử đó mà hàm trả về giá trị là True.

### CÚ PHÁP

```
filter(function, iterable)
```

- **Function:** Hàm sẽ được sử dụng để apply với các phần tử trong iterable.
- **Iterable:** Các iterable ví dụ như list, tuple, str...
- **Giá trị trả về:** filter trả về một đối tượng của lớp filter, bạn nên ép nó sang list hoặc tuple.

## 2. Filter():

### VÍ DỤ

#### EXAMPLE

**Lọc ra các số chẵn trong list:**

```
def even(n):  
    return n % 2 == 0  
  
if __name__ == '__main__':  
    a = [1, 2, 3, 4, 5]  
    b = list(filter(even, a))  
    print(b)
```

**OUTPUT**

[2, 4]

#### EXAMPLE

**Lọc ra các số nguyên tố trong list:**

```
import math  
def prime(n):  
    for i in range(2, math.isqrt(n) + 1):  
        if n % i == 0:  
            return False  
    return n > 1  
  
if __name__ == '__main__':  
    a = [20, 10, 17, 7, 29, 19]  
    b = list(filter(prime, a))  
    print(b)
```

**OUTPUT**

[17, 7, 29, 19]

## 2. Filter():

### VÍ DỤ

#### EXAMPLE

#### Áp dụng filter với lambda

```
if __name__ == '__main__':  
    a = [20, 10, 17, 7, 29, 19]  
    b = list(filter(lambda x : x >= 15, a))  
    print(b)
```

#### OUTPUT

```
[20, 17, 29, 19]
```

