



LIST SLICING





Python list slicing là một kỹ thuật giúp các bạn có thể truy cập vào 1 khoảng các phần tử trong list thông qua toán tử.

Sử dụng list slicing:

- **Với toán tử:** Bạn có thể xác định chỉ số bắt đầu, chỉ số kết thúc, bước nhảy của các phần tử trong list mà bạn muốn cắt ra.
- **Cú pháp:** List[start, stop, step].
- **Giá trị trả về:** Một list chứa các phần tử trong List bắt đầu từ chỉ số start, kết thúc ở chỉ số stop - 1 với bước nhảy là step. Nếu không chỉ rõ step thì giá trị của step mặc định là 1.

1. Slicing với chỉ số không âm:

Ví dụ:

```
a = [2, 3, 1, 5, 7, 4, 3]
b = a[0 : 4 : 1]
print(b)
c = a[0 : 4]
print(c)
```

OUTPUT

```
[2, 3, 1, 5]
[2, 3, 1, 5]
```

1. Slicing với chỉ số không âm:



Nếu không có tham số stop thì mặc định giá trị của stop sẽ là chỉ số cuối cùng trong mảng, nếu không có tham số start thì giá trị mặc định của start sẽ là 0.

EXAMPLE

```
a = [2, 3, 1, 5, 7, 4, 3]
b = a[3 : ]
print(b)
c = a[ : 4]
print(c)
```

OUTPUT

```
[5, 7, 4, 3]
[2, 3, 1, 5]
```

EXAMPLE

Bước nhảy là 2:

```
a = [2, 3, 1, 5, 7, 4, 3]
b = a[0 : 5 : 2]
print(b)
```

OUTPUT

```
[2, 1, 7]
```

Chú ý:

Nếu start quá nhỏ (nhỏ hơn 0) hoặc stop quá lớn thì chỉ start sẽ bắt đầu từ 0 và stop sẽ kết thúc tại len của list.

2. Slicing với chỉ số âm:

a	2	3	1	5	7	4	3
Negative index	-7	-6	-5	-4	-3	-2	-1

EXAMPLE

Ví dụ :

```
a = [2, 3, 1, 5, 7, 4, 3]
b = a[-4 : -1]
print(b)
```

OUTPUT

[5, 7, 4]

EXAMPLE

Kết hợp chỉ số âm và chỉ số dương:

```
a = [2, 3, 1, 5, 7, 4, 3]
b = a[1 : -2]
print(b)
```

OUTPUT

[3, 1, 5, 7]

3. Lật ngược list:



Để lật ngược list các bạn chỉ cần bỏ trống phần start và stop, phần step có giá trị là -1.

EXAMPLE

```
a = [2, 3, 1, "python", "28tech"]  
b = a[::-1]  
print(b)
```

OUTPUT

```
['28tech', 'python', 1, 3, 2]
```



4. Thay đổi giá trị của nhiều phần tử trong một đoạn:



Bạn có thể thay đổi hoặc xóa 1 loạt các phần tử trong 1 đoạn xác định bằng list slicing. Chú ý các bạn chỉ có thể gán 1 iterable cho slice mà bạn cắt ra.

Ví dụ: Thay đổi các phần tử từ chỉ số 2 tới chỉ số 4 bằng 1 phần tử mới

EXAMPLE

```
a = [2, 3, 1, 5, 1, 2]
a[2:5] = ["28tech"]
print(a)
```

OUTPUT

```
[2, 3, '28tech', 2]
```

EXAMPLE

```
a = [2, 3, 1, 5, 1, 2]
a[2:5] = [100, 200, 300]
print(a)
```

OUTPUT

```
[2, 3, 100, 200, 300, 2]
```

5. Chèn và xóa với list slicing:

Thêm vào đầu list

```
a = ['x', 'y', 'z']  
a[:0] = [10, 20, 30]  
print(a)
```

OUTPUT

```
[10, 20, 30, 'x', 'y', 'z']
```

Thêm vào cuối list

```
a = ['x', 'y', 'z']  
a[len(a):] = [10, 20, 30]  
print(a)
```

OUTPUT

```
['x', 'y', 'z', 10, 20, 30]
```


5. Chèn và xóa với list slicing:

Thêm vào giữa list

EXAMPLE

```
a = ['x', 'y', 'z']  
a[1:1] = [100, 200, 300]  
print(a)
```

OUTPUT

```
['x', 100, 200, 300, 'y', 'z']
```

EXAMPLE

```
a = ['x', 'y', 'z', 't']  
a[1:2] = [100, 200, 300]  
print(a)
```

OUTPUT

```
['x', 100, 200, 300, 'z', 't']
```

5. Chèn và xóa với list slicing:

Xóa nhiều phần tử

Xóa các phần tử từ chỉ số 1 tới chỉ số 3

```
a = ['x', 'y', 'z', 't', 'u', 'v']  
a[1:4] = []  
print(a)
```

OUTPUT

```
['x', 'u', 'v']
```



6. Copy một list (shallow copy):

Ví dụ:

```
a = ['x', 'y', 'z', 't', 'u', 'v']  
b = a[:]  
print(b)  
print(b == a)  
print(b is a)
```

OUTPUT

```
['x', 'y', 'z', 't', 'u', 'v']  
True  
False
```