



LIST





List tương tự như cấu trúc dữ liệu mảng ở các ngôn ngữ lập trình khác nhưng có phần linh hoạt và mạnh mẽ hơn.

Các tính chất của list

- **Lists are ordered:** Các phần tử trong list là có thứ tự.
- **Accessed by index:** Truy cập các phần tử trong list thông qua chỉ số.
- **Lists can contain any sort of object:** List có thể chứa các object thuộc kiểu dữ liệu khác nhau như int, str, float, thậm chí là một list khác.
- **Lists are changeable (mutable):** Các phần tử trong list có thể thay đổi giá trị, các thao tác thêm, xóa các phần tử cũng được hỗ trợ.

1. Tạo list:



Có nhiều cách để tạo một list, đơn giản nhất là tạo 1 list bằng cách đưa các giá trị của list vào bên trong dấu đóng mở ngoặc vuông, các giá trị phân cách nhau bằng dấu phẩy.

EXAMPLE

```
#List các số nguyên
```

```
a = [1, 2, 3, 4]
```

```
#List các chuỗi kí tự
```

```
b = ["28tech", "python", "ML"]
```

```
#List các số nguyên, chuỗi kí tự, số phức...
```

```
c = [1, 2, 3, 4, "Javascript", 3 + 5j]
```

```
#List rỗng
```

```
d = []
```

2. List constructor:



Bạn có thể sử dụng hàm khởi tạo **list()** để biến đổi các object khác thành list.

EXAMPLE

```
s = "28tech"  
a = list(s)  
print(a)
```

OUTPUT

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

3. Hàm len():



Để biết số lượng phần tử trong list ta sử dụng hàm **len()**.

EXAMPLE

```
a = [1, 2, 3, 4, "python", "28tech"]  
print(len(a))
```

OUTPUT

```
6
```

4. Truy cập phần tử thông qua chỉ số:



Chỉ số các phần tử trong list được đánh số từ 0 tính từ trái qua phải, ngoài ra Python list hỗ trợ cả chỉ số âm.



Chú ý: Nếu duyệt một chỉ số không hợp lệ sẽ gây ra lỗi `IndexError`.

Index	0	1	2	3	4
a	2	7	6	3	1
Negative index	-5	-4	-3	-2	-1

EXAMPLE

```
a = [1, 2, 3, 4, "python", "28tech"]
```

```
print(a[0])
```

```
print(a[-1])
```

OUTPUT

1

28tech

5. Duyệt list:

Duyệt thông qua chỉ số

```
a = [1, 2, 3, 4, "python", "28tech"]

for i in range(len(a)):
    print(a[i], end = ' ')
print()
for i in range(-1, len(a) * -1 - 1, -1):
    print(a[i], end = ' ')
```

OUTPUT

```
1 2 3 4 python 28tech
28tech python 4 3 2 1
```

Duyệt bằng for each

```
a = [1, 2, 3, 4, "python", "28tech"]

for item in a :
    print(item, end = ' ')
```

OUTPUT

```
1 2 3 4 python 28tech
```

6. Thay đổi giá trị phần tử:



Bạn có thể thay đổi giá trị các phần tử của list thông qua chỉ số tương ứng của nó.

EXAMPLE

```
a = [1, 2, 3, 4, "python", "28tech"]
```

```
a[-1] = "Java"
```

```
a[-2] = "C++"
```

```
print(a)
```

OUTPUT

```
[1, 2, 3, 4, 'C++', 'Java']
```



7. Thêm một phần tử vào trong list:



Để thêm 1 phần tử vào list bạn có thể sử dụng hàm `append()`, hàm này sẽ thêm mới phần tử vào cuối list.

EXAMPLE

```
a = [1, "python", "28tech"]  
  
a.append("C++")  
a.append("100")  
print(a)
```

OUTPUT

```
[1, 'python', '28tech', 'C++', '100']
```



Nếu bạn muốn thêm 1 phần tử vào 1 vị trí bất kì thì bạn sử dụng hàm `insert()`.

EXAMPLE

```
a = ["apple", "python", "28tech"]  
  
a.insert(1, "facebook")  
print(a)
```

OUTPUT

```
['apple', 'facebook', 'python', '28tech']
```


8. Xóa phần tử khỏi list:



Xóa phần tử khỏi list thông qua chỉ số bằng hàm pop(). Nếu không chỉ rõ chỉ số cho hàm pop, hàm này sẽ xóa phần tử cuối cùng trong list.

EXAMPLE

```
a = [1, 5, 10, 20, 30]
a.pop(2)
print(a)
a.pop()
print(a)
```

OUTPUT

```
[1, 5, 20, 30]
[1, 5, 20]
```



Hàm pop sẽ trả về phần tử mà bạn xóa khỏi list, bạn cũng có thể sử dụng hàm del nếu không muốn lấy phần tử bị xóa đó.

EXAMPLE

```
a = [1, 5, 10, 20, 30]
del a[1]
print(a)
```

OUTPUT

```
[1, 10, 20, 30]
```

8. Xóa phần tử khỏi list:



Xóa phần tử thông qua giá trị bằng **hàm remove()**, nếu trong list có nhiều phần tử giống phần tử bạn cần xóa thì hàm này chỉ xóa đi phần tử đầu tiên.



Nếu muốn dùng remove bạn phải đảm bảo giá trị bạn cần xóa có tồn tại trong list. Xóa phần tử không tồn tại trong list bằng remove sẽ gây ra lỗi.

EXAMPLE

```
a = [1, 5, 10, 20, 30, 5]
a.remove(5)
print(a)
```

OUTPUT

```
[1, 10, 20, 30, 5]
```



Xóa mọi phần tử trong list bằng **hàm clear()**.

EXAMPLE

```
a = [1, 5, 10, 20, 30, 5]
a.clear()
print(a)
```

OUTPUT

```
[]
```

9. Sao chép list:



Sao chép list có thể giúp các bạn nhân bản nội dung của 1 list ban đầu.

EXAMPLE

```
a = [1, 2, 3]
b = a * 2
print(b)
```

OUTPUT

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



Tạo 1 list có chứa 1000 phần tử là số 0.

EXAMPLE

```
a = [0] * 1000
print(a)
```



10. Tìm kiếm phần tử trong list:



Để tìm kiếm phần tử trong list bạn có thể dùng toán tử in hoặc thuật toán tìm kiếm tuyến tính.

EXAMPLE

```
a = ["apple", "facebook", "google"]  
if "apple" in a:  
    print('YES')
```

OUTPUT

YES

11. Combine list:



Bạn có thể sử dụng hàm extend() để thêm các phần tử của 1 list khác vào list hiện tại hoặc sử dụng toán tử +

EXAMPLE

```
a = [1, 2, 3]  
b = [4, 5, 6]  
a.extend(b)  
print(a)  
a += b  
print(a)
```

OUTPUT

```
[1, 2, 3, 4, 5, 6]  
[1, 2, 3, 4, 5, 6, 4, 5, 6]
```

12. Các phương thức list:



Hàm `copy()`: Trả về một list mới có nội dung tương tự như list ban đầu nhưng không phải là list ban đầu.

EXAMPLE

```
a = [1, 2, 3]
b = a
print(a is b)
c = a.copy()
print(c)
print(c == a)
print(c is a)
```

OUTPUT

```
True
[1, 2, 3]
True
False
```

12. Các phương thức list:



Hàm count(): Trả về số lần xuất hiện của một phần tử bất kì trong list.

EXAMPLE

```
a = [1, 2, 3, 1, 2, 1, 1]
print(a.count(1))
```

OUTPUT

4



Hàm index(): Trả về chỉ số đầu tiên của 1 phần tử trong list.

EXAMPLE

```
a = [4, 3, 1, 2, 1, 1]
print(a.index(1))
#Chú ý câu lệnh này gây lỗi
#print(a.index(100))
```

OUTPUT

2

12. Các phương thức list:



Hàm reverse(): Lật ngược một list.

EXAMPLE

```
a = [1, 2, 3, 4]
a.reverse()
print(a)
```

OUTPUT

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



Hàm sort(): Sắp xếp các phần tử trong list.

EXAMPLE

```
a = [1, 5, 4, 3, 2, 1]
a.sort()
print(a)
```

OUTPUT

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



13. Các hàm built-in sử dụng với list:

Hàm	Chức năng
all()	Trả về True nếu mọi phần tử trong list đều là True
any()	Trả về True nếu có ít nhất 1 phần tử trong list là True
max()	Trả về phần tử lớn nhất trong list
min()	Trả về phần tử nhỏ nhất trong list
sorted()	Trả về list đã sắp xếp của list ban đầu
sum()	Trả về tổng các phần tử trong list

