

## GIỚI THIỆU VỀ ĐỐI TƯỢNG LIST TRONG PYTHON

```
In [ ]: myList = ["apple", "banana", "cherry"]
        type(myList)
```

```
Out[ ]: list
```

### 1. Đối tượng List

- List được dùng để *chứa nhiều giá trị trong một biến duy nhất*.
- List là một trong 4 kiểu dữ liệu *có sẵn* trong Python, nó được dùng để chứa một tập hợp các dữ liệu. *Ngoài đối tượng List thì Python còn có Tuple, Set, và Dictionary (tất cả đều có chất lượng và cách sử dụng khác nhau)*
- List được tạo bằng cách dùng dấu ngoặc `[]`

```
In [ ]: thisList = ["apple", "banana", "cherry"]
        print(thisList)
```

```
['apple', 'banana', 'cherry']
```

### 2. Phần tử của List

- Phần tử của List được sắp xếp theo thứ tự, có thể thay đổi, và cho phép các giá trị trùng lặp.
- Phần tử của List đều có chỉ mục (index), phần tử đầu tiên có index là 0, phần tử thứ hai có index là 1,...

### 3. Sắp xếp List

- Như đề cập ở nội dung trước thì đối tượng List được sắp xếp theo thứ tự, điều đó có nghĩa là các phần tử của List có một thứ tự nhất định và thứ tự đó không thay đổi.
- Nếu bạn thêm một phần tử mới vào List thì phần tử mới đó sẽ được đặt ở cuối List.
- Tuy nhiên, Python sẽ có một vài phương thức sẽ thay đổi thứ tự, nhưng tổng quát thì thứ tự của các phần tử sẽ không thay đổi.

### 4. Có thể thay đổi

List có thể thay đổi, điều này có nghĩa là sau khi List được tạo thì bạn có thể change, thêm và xóa bỏ phần tử trong List.

### 5. Cho phép giá trị trùng lặp

Bởi vì List có index nên list có thể chứa các phần tử có giá trị giống nhau.

```
In [ ]: thisList = ["apple", "banana", "cherry", "apple", "cherry"]
        print(thisList)
```

```
['apple', 'banana', 'cherry', 'apple', 'cherry']
```

### 6. Độ dài của List

Để xác định có bao nhiêu phần tử mà List có thì bạn có thể dùng hàm `len()`:

```
In [ ]: thislist = ["apple", "banana", "cherry"]
print(len(thislist))
```

3

## 7. Kiểu dữ liệu của các phần tử trong List

- Các phần tử trong List có thể có bất kỳ kiểu dữ liệu nào:

```
In [ ]: list1 = ["apple", "banana", "cherry"]
list2 = [1, 5, 7, 9, 3]
list3 = [True, False, False]

print(list1)
print(list2)
print(list3)
```

```
['apple', 'banana', 'cherry']
[1, 5, 7, 9, 3]
[True, False, False]
```

- List có thể chứa các phần tử có kiểu dữ liệu khác nhau:

```
In [ ]: list1 = ["abc", 34, True, 40, "male"]
print(list1)
```

```
['abc', 34, True, 40, 'male']
```

## 8. Hàm type()

Trong Python, list được định nghĩa như một đối tượng với kiểu dữ liệu "list":

```
In [ ]: mylist = ["apple", "banana", "cherry"]
print(type(mylist))
```

```
<class 'list'>
```

## 9. Hàm khởi tạo list()

Ngoài việc dùng dấu ngoặc [] thì bạn cũng có thể dùng hàm khởi tạo list() để tạo ra đối tượng (object) List.

```
In [ ]: thislist = list(("apple", "banana", "cherry"))
print(thislist)
```

```
['apple', 'banana', 'cherry']
```

## 10. Các kiểu dữ liệu tập hợp trong Python

Python có 4 kiểu dữ liệu tập hợp:

- List** là một tập hợp được sắp xếp và có thể thay đổi và cho phép phần tử trùng lặp.
- Tuple** là một tập hợp được sắp xếp, không thể thay đổi và cho phép phần tử trùng lặp.
- Set** là một tập hợp không được sắp xếp, phần tử trong đối tượng này không thể thay đổi nhưng bạn có thể xóa hoặc thêm phần tử bất cứ khi nào bạn muốn, Set không cho phép phần tử trùng lặp.

- **Dictionary** là một tập hợp được sắp xếp, có thể thay đổi, và không cho phép phần tử trùng lặp.

```
In [ ]: myList = [1,2,3,4,5] #list
myTuple = (1,2,3,4,5) #tuple
mySet = {1,2,3,4,5} #set
myDictionary = {"language":"Python", "age": 24} #dictionary

print(myList)
print(type(myList))
print(myTuple)
print(type(myTuple))
print(mySet)
print(type(mySet))
print(myDictionary)
print(type(myDictionary))
```

```
[1, 2, 3, 4, 5]
<class 'list'>
(1, 2, 3, 4, 5)
<class 'tuple'>
{1, 2, 3, 4, 5}
<class 'set'>
{'language': 'Python', 'age': 24}
<class 'dict'>
```

---

Biên dịch: Nguyễn Ngọc Dương

Facebook: <https://www.facebook.com/danh.kean.353803/>

Email: [ngocduong5642@gmail.com](mailto:ngocduong5642@gmail.com)