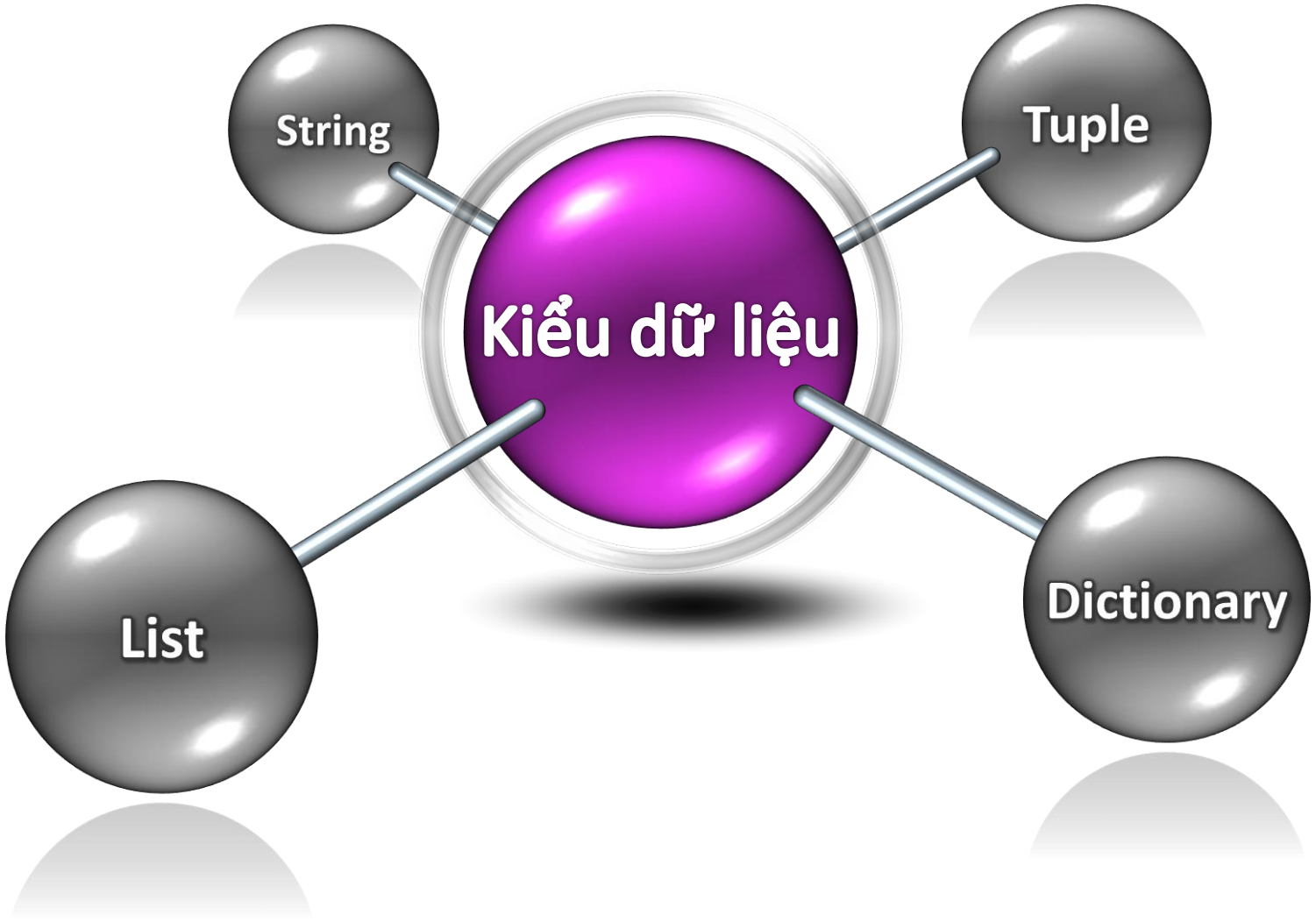
String, List
(Python Day 6)

ducnc@PTCC

Kiến thức đã học

Kiểu dữ liệu trong Python



List

List là một dạng Collection trong Python

Collection cho phép lưu nhiều giá trị trong một biến

```
>>> balo = ['quan', 'ao', 'mu']
```

A collection

```
>>>  x = 'quan'
```

Not a collection

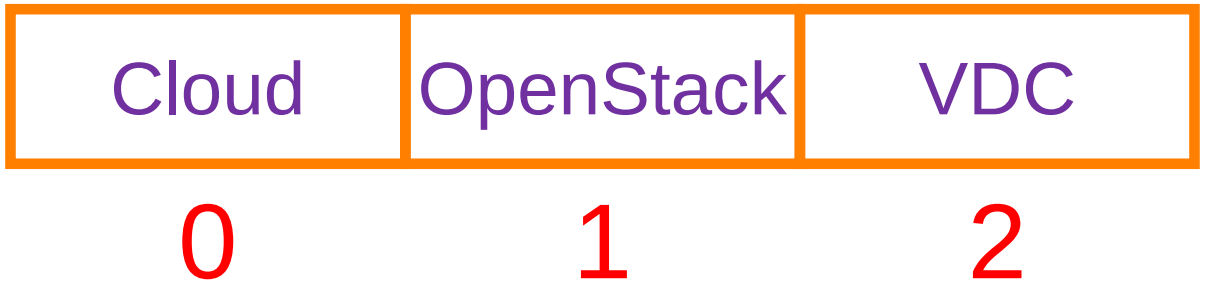
List

- Định nghĩa một list bằng dấu []
- Nội dung trong một list có thể chứa bất cứ đối tượng nào kể cả một list khác
- Có thể là một list rỗng

Ví dụ:

```
>>> [1, 2, 10, 11]
[1, 2, 10, 11]
>>> ['Cloud', 'OpenStack', 'VDC']
['Cloud', 'OpenStack', 'VDC']
>>> [1, 2, [2, 5], 'Duc']
[1, 2, [2, 5], 'Duc']
>>> []
[]
```

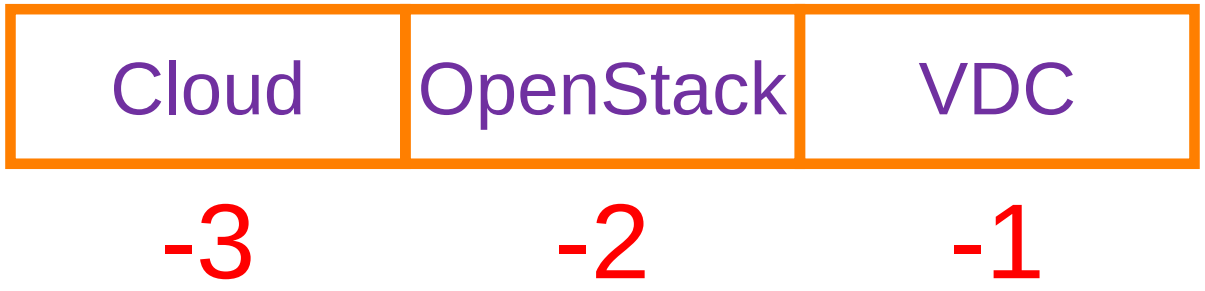
- Đánh số các phần tử trong List từ phần tử đầu tiên và bắt đầu từ 0



- Truy xuất dữ liệu bằng cách gọi index

```
>>> myteam = ['Cloud', 'OpenStack', 'VDC']
>>> myteam[0]
'Cloud'
>>> myteam[2]
'VDC'
```

- Cách đánh số ngược từ phần tử cuối cùng với index = -1



- Truy xuất dữ liệu

```
>>> myteam = ['Cloud', 'OpenStack', 'VDC']
>>> myteam[-3]
'Cloud'
>>> myteam[-1]
'VDC'
```

List

- List là mutable (có thể thay đổi được)
- Gán giá trị cho phần tử thông qua index

```
>>> myteam = ['Cloud', 'OpenStack', 'VDC']
>>> myteam[0] = 'Dien toan dam may'
>>> myteam
['Dien toan dam may', 'OpenStack', 'VDC']
```

- Xác định độ dài list: sử dụng hàm len()

```
>>> len(myteam)
3
```

List

- Cộng nhiều list

```
>>> a = [1, 2, 3, 4]
>>> b = [3, 4, 5, 6]
>>> a + b
[1, 2, 3, 4, 3, 4, 5, 6]
```

- Hàm range(x): trả về một list từ 0 đến x-1

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

- Kiểm tra phần tử trong list: dùng in hoặc not in

```
>>> a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> 8 in a
True
```

List

- Cắt list: sử dụng dấu ":"
  [a:b]: lấy list từ phần tử thứ a đến phần tử b-1

```
>>> a = [1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> a[1:4]
[2, 3, 4]
>>> a[1:]
[2, 3, 4, 5, 6, 7, 8, 9]
>>> a[:2]
[1, 2]
>>> a[:-3]
[1, 2, 3, 4, 5, 6]
```

List

- Hàm append(): thêm một phần tử vào cuối list

```
>>> a = [1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> a.append('Python')
>>> a
[1, 2, 3, 4, 5, 6, 7, 8, 9, 'Python']
```

- Hàm sort(): sắp xếp list thành list mới với các phần tử tăng dần

```
>>> myteam = ['Cloud', 'OpenStack', 'VDC', 1,
'abc']
>>> myteam.sort()
>>> myteam
[1, 'Cloud', 'OpenStack', 'VDC', 'abc']
```

- Một số hàm built-in

```
len(myteam)
max(myteam)
min(myteam)
sum(myteam)
```

- Bài tập: viết lại chương trình quản lý máy ảo ở buổi trước có sử dụng list để lưu thông tin. Trong bài phải sử dụng function, built-in, docstring.

Dictionary

- Một loại Collection
- Kiểu dữ liệu mạnh nhất của Python
- Không sử dụng index như list
- Mỗi phần tử là một cặp key – value
- Giống như một danh bạ

Dictionary

Cú pháp:

d = {key1: value1, key2: value2, key3:value3,...}

Ví dụ:

>>> d = {1: 'ducnc', 2: 'congtt', 'a': 'longlq'}

key không được là một biến

Dictionary

Có thể khởi tạo một từ điển rỗng

Gán giá trị cho từ điển: D[key] = value

Ví dụ:

```
>>> D = {}
>>> D[1] = 2
>>> D['xxx'] = 'PTCC'
>>> D
{1: 2, 'xxx': 'PTCC'}
```

Dictionary

Truy xuất dữ liệu:

```
d[key]
```

Ví dụ:

```
>>> d[1]
>>> d['a']
```

Kiểm tra xem key đã có trong dictionary chưa:

```
key in d
>>> 1 in d
>>> True
```

Dictionary

Ứng dụng:
Viết chương trình đếm số lần xuất hiện của các từ trong 10 lần nhập của người dùng

Dictionary

Sử dụng dict trong vòng lặp:

```
for key in dict:
    print key, d[key]
```

Hàm items(): trả về phần tử của dict dưới dạng một list các tuple.

```
>>> D.items()
[(1, 2), ('xxx', 'PTCC')]
```

Dictionary

Sử dụng cả *key, value* trong vòng lặp:

```
for key,value in d.items():
    print key, value
```

Dict không thể sử dụng phương thức sort() như list nhưng vẫn có thể sắp xếp thứ tự các item bằng cách chuyển sang kiểu dữ liệu Tuple

Buổi sau

Bài tập:
Viết lại chương trình quản lý máy ảo dict để lưu thông tin. Trong bài phải sử dụng function, built-in, docstring.

Hỏi đáp