

Dữ liệu kiểu Tuple

TS. Nguyễn Tuấn Anh

Trường Đại học Công nghệ Thông tin & Truyền thông

Khoa Công nghệ Thông tin

Tháng 2 năm 2023



Khái niệm về Tuple

Kiểu dữ liệu Tuple tương tự như danh sách, sự khác biệt giữa chúng là ta không thể thay đổi các phần tử của một tuple.

Chúng ta thường dùng Tuple cho các kiểu dữ liệu không đồng nhất (các phần tử có kiểu dữ liệu khác nhau) và List cho các kiểu dữ liệu đồng nhất (các phần tử cùng kiểu dữ liệu).

Vì tuple là bất biến, nên việc lặp qua tuple nhanh hơn so với List. Nếu dữ liệu ít thay đổi, việc triển khai dữ liệu dưới dạng Tuple sẽ đảm bảo không bị thay đổi.



Khởi tạo tuple trong Python

Để khởi tạo một tuple, chúng ta sử dụng cú pháp:

```
mytuple = (element1, element2, ...)
```

Ví dụ:

```
myTuple = ("oto", "xe may", "may bay")
```

Tạo một Tuple chỉ có một phần tử phải bổ sung thêm dấu phẩy ở cuối:

```
mytuple = ("hello", )
```



Duyệt các phần tử của tuple

Để duyệt các phần tử của một tuple, chúng ta sử dụng toán tử `[]` và index là một số nguyên.

Đối với một tuple thuần thì các chỉ số được đánh từ $0, 1, 2, \dots, n-1$ theo chiều từ trái sang phải và được đánh từ $-1, -2, \dots, -n$ theo chiều từ phải sang trái.

Tuple lồng nhau được truy cập bằng cách sử dụng lập chỉ mục lồng nhau.

Ví dụ lấy phần tử thứ 2 của `myTuple`:

```
print(myTuple[1])
```



Một số thao tác với dữ liệu tuple

Sử dụng vòng lặp for với index

```
for i in range(len(myTuple)):
    print(myTuple[i], end=' ')
```

Sử dụng vòng lặp for để in ra từng phần tử một tuple

```
for x in myTuple:
    print(x, end=' ')
```



Một số thao tác với dữ liệu tuple

Lấy các phần tử liên tiếp trong tuple, sử dụng tuple[a:b]

```
for x in myTuple[1:3]:  
    print(x)
```



Một số thao tác với dữ liệu tuple

Nối các tuple: `tuple3 = tuple1 + tuple2`

```
a = (1, 2, 3) + (4, 5, 6)
```

Lặp các bộ sử dụng `n*tuple`

```
b = ("ABC", ) * 3
```

b có kết quả là: `("ABC", "ABC", "ABC")`



Một số thao tác với dữ liệu tuple

Một số hàm thao tác với tuple:

- `cmp(tuple1, tuple2)`: So sánh các phần tử của 2 tuple
- `len(tuple)`: Trả về chiều dài của tuple
- `max(tuple)`: Trả về phần tử có giá trị lớn nhất trong tuple
- `min(tuple)`: Trả về phần tử có giá trị nhỏ nhất trong tuple
- `tuple(seq)`: Chuyển đổi một list thành tuple



Một số thao tác với dữ liệu tuple

Ví dụ

```
mytuple = (4, 9, 6, 5, 2, 8, 7)
print("Sum = " + str(sum(mytuple)))
print("Min = " + str(min(mytuple)))
print("Max = " + str(max(mytuple)))
print("Length = " + str(len(mytuple)))
```



Kiểm tra phần tử có thuộc Tuple

Từ khóa **"in"** để kiểm tra một phần tử có thuộc Tuple hay không:

```
myTuple = ("oto", "xe may", "may bay")  
if "may bay" in myTuple:  
    print("Phan tu thuoc tuple")
```



Kiểm tra phần tử có thuộc Tuple

Từ khóa "**not in**" để kiểm tra một phần tử **KHÔNG** thuộc Tuple hay không:

```
myTuple = ("oto", "xe may", "may bay")  
if "tau hoa" in myTuple:  
    print("Phan tu KHONG thuoc tuple")
```



Thêm phần tử vào Tuple

Chuyển đổi kiểu Tuple sang kiểu List sau đó sử dụng hàm `append()` để thêm phần tử như sau:

```
myTuple = ("oto", "xe may", "may bay")  
listA = list(myTuple)  
listA.append("xe hoi")  
myTuple = tuple(listA)
```



Xóa phần tử thuộc Tuple

Tuple là không thể thay đổi, vì vậy bạn không thể xóa các phần tử khỏi nó. Giải pháp đó là, ta cũng có thể chuyển đổi kiểu Tuple sang kiểu List sau đó sử dụng hàm `remove()` để xóa phần tử như sau:

```
tupleA = ("oto", "xe may", "may bay")  
listA = list(tupleA)  
listA.remove("may bay")  
tupleA = tuple(listA)
```



Xóa phần tử thuộc Tuple

Xóa toàn bộ phần tử có trong Tuple, ta chỉ cần sử dụng từ khóa del:

```
myTuple = ("oto", "xe may", "may bay")  
del tupleA
```

