Kiểu dữ liệu Danh sách (List)

TS. Nguyễn Tuấn Anh

Trường Đại học Công nghệ Thông tin & Truyền thông

Khoa Công nghệ Thông tin

Tháng 2 năm 2023

Kiểu dữ liệu List trong python

Có bốn kiểu dữ liệu tập hợp ngôn ngữ lập trình Python:

- List: là một collection có thứ tự, có thể thay đổi. Cho phép chứa dữ liệu trùng lặp.
- Tuple: là một collection có thứ tự, không thể thay đổi. Cho phép chứa dữ liệu trùng lặp.
- Set: là một collection không có thứ tự, không có chỉ mục. Không cho phép chứa dữ liệu trùng lặp.
- Dictionary: là một collection không có thứ tự, có thể thay đổi và lập chỉ mục. Không cho phép chứa dữ liệu trùng lặp.

Kiểu dữ liệu List trong python

Liểu dữ liệu List dùng để lưu trữ một danh sách các phần tử. Kiểu dữ liệu List là một trong 4 kiểu được python hỗ trợ: Tuple, Set, và Dictionary.

```
myList = ["apple", "banana", "cherry"]
print(myList)
```

List cho phép sắp xếp, thay đổi giá trị và các phần tử trong list có thể trùng nhau. Truy cập các phần tử trong List sử dụng chỉ số đặt trong cặp ngoặc []:

```
myList[0], myList[1]
```

Kiểu dữ liệu List trong python

Hàm len() trả về số phần tử trong List

```
myList = ["apple", "banana", "cherry"]
```

List có thể chứa các kiểu dữ liệu khác nhau:

```
list1 = ["abc", 34, True, 40, "male"]
```

Hàm khởi tạo list()

```
myList = list(("apple", "banana", "cherry"))
```

Truy cập các phần tử trong List

Sử dụng chỉ số để truy cập các phần tử trong List

```
myList = ["apple", "banana", "cherry"]
print(myList[1])
```

Chỉ số âm để đánh chỉ số tử cuối List, chỉ số -1 chỉ phần tử cuối cùng, -2 chỉ phần tử thứ 2 tính từ cuối.

```
myList = ["apple", "banana", "cherry"]
print(myList[-1])
```

Sử dụng chỉ số là một đoạn [a, b]

```
myList = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(myList[2:5])
```

Truy cập các phần tử trong List

Khoảng chỉ số nhưng không có chỉ số bắt đầu.

```
myList = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(myList[:4]) # trả về các phần tử có chỉ số từ 0->3
```

Khoảng chỉ số nhưng không có chỉ số kết thúc.

```
myList = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(myList[2:]) # trả về các phần tử có chỉ số từ 2-> hết
```

Khoảng chỉ số có chỉ số âm

```
myList = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(myList[-4:-1]) # trả về ['orange', 'kiwi', 'melon']
```

Truy cập các phần tử trong List

Kiểm tra tồn tại của phần tử trong List

```
myList = ["apple", "banana", "cherry"]
if "apple" in myList:
  print("Yes, 'apple' is in the fruits list")
```

Thay đổi giá trị các phần tử trong List

Thay đổi giá trị của phần tử sử dụng chỉ số:

```
myList = ["apple", "banana", "cherry"]
myList[1] = "blackcurrant"
```

Thay đổi giá trị sử dụng khoảng chỉ số:

```
myList = ["apple", "banana", "cherry", "orange", "kiwi", "mango"]
myList[1:3] = ["blackcurrant", "watermelon"]
```

Thêm phần tử vào danh sách

Phương thức append(): Thêm một phần tử vào List:

```
myList = ["apple", "banana", "cherry"]
myList.append("orange")
```

Phương thức insert(): Thêm một phần tử vào vị trí nào đó trong List:

```
myList = ["apple", "banana", "cherry"]

# Thêm "orange" vào vị trí có chỉ số 1
myList.insert(1, "orange")
```

Thêm phần tử vào danh sách

Phương thức extend(): Thêm một danh sách vào cuối List

```
myList = ["apple", "banana", "cherry"]
tropical = ["mango", "pineapple", "papaya"]
# Thêm danh sách tropical vào cuối danh sách myList
myList.extend(tropical)
```

Phương thức extend(): Thêm vào một tuples, sets, dictionaries vào List

```
myList = ["apple", "banana", "cherry"]
thistuple = ("kiwi", "orange")
#Thêm thistuple vào cuối danh sách myList
myList.extend(thistuple)
```

Xóa phần tử vào danh sách

Phương thức remove(): Xóa một phần tử khỏi List

```
myList = ["apple", "banana", "cherry"]
myList.remove("banana")
```

Phương thức pop(): Xóa một phần tử khỏi List dựa vào chỉ số

```
myList = ["apple", "banana", "cherry"]

myList.pop(1) # Xóa phần tử thứ 2 (có chỉ số là 1)

myList.pop() # Xóa phần tử cuối danh sách
```

Phương thức clear(): Xóa toàn bộ List,

```
myList = ["apple", "banana", "cherry"]
myList.clear()
```

Duyệt danh sách

Dùng vòng lặp for duyệt List:

```
myList = ["apple", "banana", "cherry"]
for x in myList:
  print(x)
```

Dùng vòng lặp for duyệt List sử dụng chỉ số:

```
for i in range(len(myList)):
  print(myList[i])
```

Duyệt danh sách

Dùng vòng lặp while duyệt List:

```
i = 0
while i < len(myList):
  print(myList[i])
  i = i + 1
```

List Comprehension:

```
myList = ["apple", "banana", "cherry"]
[print(x) for x in myList]
```

List Comprehension

Dựa trên danh sách các loại trái cây, bạn muốn có một danh sách mới, chỉ chứa các loại trái cây có chữ cái "a" trong tên:

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = []

for x in fruits:
  if "a" in x:
    newlist.append(x)
```

Nếu sử dụng Comprehension thì code ngắn hơn

```
myList = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = [x for x in myList if "a" in x]
```

List Comprehension

Cú pháp

```
newlist = [expression for item in iterable if condition == True]
```

Tạo List mới với điều kiện là: gồm những hoa quả không phải là "apple"

```
newlist = [x for x in fruits if x != "apple"]
```

Tạo List mới không cần điều kiện

```
newlist = [x for x in fruits]
```

Tạo List mới từ danh sách đã có, viết hoa các ký tự trong List

```
newlist = [x.upper() for x in fruits]
```

Sắp xếp List

Phương thức sort(): Sắp xếp List theo Alphanumerically

```
myList = ["orange", "mango", "kiwi", "pineapple", "banana"]
myList.sort()
```

Sắp xếp theo thứ tự giảm dần: reverse = True

```
myList = ["orange", "mango", "kiwi", "pineapple", "banana"]
myList.sort(reverse = True)
```

Tùy biến hàm sort(): Sắp xếp List dựa trên mức độ gần số 50

```
def myfunc(n):
  return abs(n - 50)

myList = [100, 50, 65, 82, 23]
myList.sort(key = myfunc)
```

Sắp xếp danh sách

Theo mặc định phương thức sort() phân biệt chữ hoa, chữ thường, để sắp xếp không phân biết chữ hoa, chữ thường sử dụng hàm lower()

```
myList = ["orange", "mango", "kiwi", "pineapple", "banana"]
myList.sort(key = str.lower)
```

Phương thức reverse(): Đảo ngược danh sách

```
myList = ["orange", "mango", "kiwi", "pineapple", "banana"]
myList.reverse()
```

Copy danh sách

Để sao chép danh sách, chúng ta không thể sử dụng phép gán: list2 = list1, bởi vì: list2 sẽ chỉ là một tham chiếu đến list1 và những thay đổi được thực hiện trong list1 cũng sẽ tự động được thực hiện trong list2.

Phương thức copy(): Sao chép danh sách

```
myList = ["apple", "banana", "cherry"]
mylist = myList.copy()
print(mylist)
```

Phương thức list():

```
myList = ["apple", "banana", "cherry"]
mylist = list(myList)
```

Nối danh sách

Nối hai danh sách:

```
list1 = ["a", "b", "c"]
list2 = [1, 2, 3]

list3 = list1 + list2
```

Thêm các phần tử trong list2 vào list1

```
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

for x in list2:
  list1.append(x)
```

Phương thức extend() thêm các phần tử list2 vào list1

```
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

list1.extend(list2)
```