Bài 11. Các hàm xử lý mảng(list) trong Python

Đăng ký kênh Youtube: Song Lâm Tech để xem lại bài giảng
https://www.youtube.com/c/SongLâmTech

Giáo viên: Dương Thành Long
Email: mrlong07.11@c3vietducdaklak.edu.vn
SĐT/Zalo: 0835.100.699
Website: http://songlamtech.com

SLT
SONG LÂM TECH
KÊNH GIÁO DỤC TRỰC TUYẾN

NỘI DUNG

- Giới thiệu cách sử dụng một số hàm xủ lý mảng trong Python

Hàm list

Hàm này có tác dụng chuyển đổi kiểu dữ liệu của một biến sang dạng list.

Cú pháp: **list(data)**

Trong đó, data là biến chứa tuple bạn cần chuyển đổi.

Hàm list

VÍ DỤ:

```
string = "Tin hoc"
print(list(string))
# Ket Qua: [‘T', ‘i', ‘n', ‘ ', 'h', ‘o', ‘c']

tup = ('A', 'B', 'C')
print(list(tup))
# Ket Qua: ['A', 'B', 'C']
```

Hàm len()

Hàm này trả về số lượng phần tử có trong mảng.

Cú pháp: `len(list)`

Trong đó: list là mảng mà các bạn cần đếm.

VÍ DỤ:

```
list = ['A', 'B', 'C']

print(len(list))
#Kết quả: 3
```

Hàm max()

Hàm này sẽ trả về phần tử có giá trị lớn nhất trong mảng. Nếu là chuỗi thì nó sẽ trả về phần tử có độ dài chuỗi dài nhất, nếu là số thì nó sẽ trả về phần tử có số lớn nhất.

Cú pháp: **`max(list)`**

Trong đó, list là mảng cần kiểm tra.

VÍ DỤ:

```
list = ['A', 'B', 'C']
print(max(list))
#Kết quả: C
list = ['1', '3', '2']
print(max(list))
#Kết quả: 3
```

Hàm min()

Hàm này sẽ trả về phần tử có giá trị nhỏ nhất trong mảng. Nếu là chuỗi thì nó sẽ trả về phần tử có độ dài chuỗi ngắn nhất, nếu là số thì nó sẽ trả về phần tử có số nhỏ nhất.

Cú pháp: **`min(list)`**

Trong đó, list là mảng cần kiểm tra.

VÍ DỤ:

```
list = ['A', 'B', 'C']
print(min(list))
#Kết quả: A
list = ['1', '3', '2']
print(min(list))
#Kết quả: 1
```

Hàm append()

Hàm này có tác dụng thêm phần tử vào cuối của một mảng.

Cú pháp: `mylist.append(obj)`

Trong đó:

mylist là mảng mà các bạn cần thêm phần tử.

obj là phần tử mà bạn muốn thêm vào mylist.

VÍ DỤ:

```
list = ['A', 'B', 'C']
list.append('D')
print(list)
# Kết quả: ['A', 'B', 'C', 'D']

list.append(('E', 'F'))
print(list)
# Kết quả: ['A', 'B', 'C', 'D', ('E', 'F')]
```

Hàm extend()

Hàm này có tác dụng kế thừa lại các phần tử của list2 và thêm vào trong list1.

Cú pháp: `list1.extend(list2)`

Trong đó:

list1 là mảng mà bạn muốn kế thừa từ một list khác (ở đây là list2).

list2 là mảng được sử dụng để cho list khác kết thừa (ở đây là list1).

Hàm extend()

VÍ DỤ:

```
list = ['A', 'B', 'C']
list.extend('D')
print(list)
# Kết quả: ['A', 'B', 'C', 'D']

list.extend(('E', 'F'))
print(list)
# Kết quả: ['A', 'B', 'C', 'D', 'E', 'F']
```

Hàm count()

Hàm này có tác dụng đếm số lần xuất hiện của một thành phần trong mảng!

Cú pháp: `mylist.count(val)`

Trong đó:

mylist là mảng mà các bạn cần kiểm tra.

val là phần tử mà bạn muốn tìm và đếm trong mảng mylist.

VÍ DỤ:

```
list = ['A', 'B', 'C']
print(list.count('A'))
# Kết quả: 1
```

Hàm index()

Hàm này có tác dụng trả về vị trí xuất hiện đầu tiên của phần tử muốn tìm và nếu như không tìm thấy thì chương trình báo lỗi.

Cú Pháp: **mylist.index(val)**

Trong đó:

mylist là mảng mà các bạn cần kiểm tra.

val là phần tử mà bạn muốn tìm trong mảng mylist.

VÍ DỤ:

```
list = ['A', 'B', 'C']
print(list.index('B'))
# Kết quả: 1
print(list.index('D'))
# Kết quả: ValueError: 'D' is not in list
```

Hàm insert()

Hàm có tác dụng thêm phần tử vào vị trí cần chèn của mảng, và các phần tử sau vị trí đó sẽ được đẩy về phía sau.

Cú pháp: `mylist.insert(index, val)`

Trong đó:

mylist là mảng cần thêm phần tử.

index là vị trí thêm phần tử val vào.

val là phần tử mà bạn muốn thêm vào trong mảng mylist.

VÍ DỤ:

```
list = ['A', 'B', 'C']
list.insert(0, 'Z')
print(list)
# Kết quả: ['Z', 'A', 'B', 'C']
list.insert(2, 'D')
print(list)
# Kết quả: ['Z', 'A', 'D', 'B', 'C']
```

Hàm reverse()

Hàm này có tác dụng đảo ngược vị trí của các phần tử trong mảng.

Cú pháp: `mylist.reverse()`

Trong đó: mylist là mảng mà các bạn muốn đảo ngược.

VÍ DỤ:

```
list = ['A', 'B', 'C']
list.reverse()
print(list)
# Kết quả: ['C', 'B', 'A']
```

Hàm remove()

Hàm này có tác dụng xóa phần tử khỏi mảng.

Cú Pháp: `mylist.remove(val)`

Trong đó:

mylist là mảng cần xóa phần tử.

val là phần tử muốn xóa trong mảng mylist.

VÍ DỤ:

```
list = ['A', 'B', 'C']
list.remove('C')
print(list)
# Kết quả: ['A', 'B']
```

Hàm pop()

Hàm này có tác dụng xóa bỏ phần tử trong mảng dựa trên vị trí của nó.

Cú pháp: `mylist.pop(index)`

Trong đó:

mylist là mảng mà các bạn cần xóa phần tử.

index là vị trí của phần tử mà bạn muốn muốn xóa trong mảng mylist. Mặc định thì index = mylist[-1] (phần tử cuối cùng trong list).

Hàm pop()

VÍ DỤ:

```
list = ['A', 'B', 'C', 'D', 'E']
list.pop()
print(list)
# Kết quả: ['A', 'B', 'C', 'D']

list.pop(2)
print(list)
# Kết quả: ['A', 'B', 'D']
```

Hàm sort()

Hàm này có tác dụng sắp xếp lại các phần tử trong mảng theo một thứ tự xác định.

Cú pháp: `mylist.sort(reverse, key)`

Trong đó:

mylist là mảng muốn sắp xếp.

reverse là một boolean cấu hình kiểu sắp xếp. Nếu reverse = True thì mảng sẽ được sắp xếp từ lớn đến bé, nếu reverse = False thì mảng sẽ được sắp xếp theo thứ tự từ bé đến lớn. Mặc định thì reverse = False.

key là gọi lại hàm để xử lý mảng hoặc là một lamda function (thường được dùng để sắp xếp các list tuple hoặc dictionary).

Hàm sort()

VÍ DỤ:

```
list = ['A', 'C', 'B', 'E', 'D']
list.sort()
print(list)
# Kết quả: ['A', 'B', 'C', 'D', 'E']
list.sort(reverse=True)
print(list)
# Kết quả: ['E', 'D', 'C', 'B', 'A']
def custom_sort(elem):
    return elem[1]
list = [(1, 2), (5, 7), (7, 100), (4, 4)]
list.sort(key=custom_sort)
print(list)
# Kết quả: [(1, 2), (4, 4), (5, 7), (7, 100)]
```

Hàm clear()

Hàm này có tác dụng xóa bỏ hết tất cả các phần tử trong mảng.

Cú pháp: **`mylist.clear()`**

Trong đó: mylist là mảng mà bạn muốn xóa bỏ hết phần tử.

VÍ DỤ:

```
list = ['A', 'C', 'B', 'E', 'D']
list.clear()
print(list)
# Kết quả: []
```

Thank You!!!

Đăng ký kênh Youtube: Song Lâm Tech để xem lại bài giảng
https://www.youtube.com/c/SongLâmTech

SONG LÂM TECH
KÊNH GIÁO DỤC TRỰC TUYẾN

Giáo viên: Dương Thành Long
Email: mrlong07.11@c3vietducdaklak.edu.vn
SĐT/Zalo: 0835.100.699
Website: http://songlamtech.com