

LAB 3: String, List trong Python

a. Kiểu dữ liệu chuỗi Python

Chuỗi là một chuỗi các ký tự. Python hỗ trợ các ký tự Unicode. Nói chung, các chuỗi được biểu thị bằng dấu ngoặc đơn hoặc dấu ngoặc kép.

```
a = "string in a double quote"
b = 'string in a single quote'
print(a)
print(b)
# # sử dụng ',' để nối hai hoặc nhiều chuỗi
print(a,"concatenated with",b)
#sử dụng '+' để nối hai hoặc nhiều chuỗi
print(a+" concated with "+b)
```

b. List

Danh sách là một kiểu dữ liệu linh hoạt độc quyền trong Python. Theo một nghĩa nào đó, nó giống như mảng trong C/C++. Nhưng danh sách trong Python là nó có thể chứa đồng thời nhiều loại dữ liệu khác nhau.

Danh sách chính thức là một chuỗi có thứ tự của một số dữ liệu được viết bằng dấu ngoặc vuông ([]) và dấu phẩy (,).

```
#list chỉ có số nguyên
a= [1,2,3,4,5,6]
print(a)
#list có chuỗi
b=["hello", "john", "reese"]
print(b)
#list có số nguyên và chuỗi
c= ["hey", "you", 1,2,3, "go"]
print(c)
#index dựa trên 0. → sẽ in một ký tự
print(c[1]) #in [1] trong list c
```

c. Tuple

Tuple là một kiểu dữ liệu khác, là một chuỗi dữ liệu tương tự như một danh sách. Nhưng nó là **bất biến**. Điều đó có nghĩa là dữ liệu trong một bộ dữ liệu được bảo vệ chống ghi. Dữ liệu trong một tuple được viết bằng dấu ngoặc đơn và dấu phẩy.

```
#tuple có số nguyên trong data.  
A =(1,2,3,4)  
print(a)  
#tuple có nhiều kiểu trong data.  
B =("hello", 1,2,3, "go")  
print(b)  
#index tuple cũng dựa trên 0.  
print(b[4]) #in một phần tử trong một tuple, trong trường hợp  
này là "go"
```

d. range

Hàm **range()** được tích hợp sẵn trong Python được sử dụng để tạo một chuỗi các số bắt đầu từ 0 theo mặc định, tăng thêm 1 (theo mặc định) và kết thúc tại một số được chỉ định.

Hàm range() trong Python có hai dạng cú pháp:

- **range(stop)**
- **range(start, stop[, step])**

Các tham số của hàm **range()**

- **start**: số nguyên bắt đầu, chuỗi sẽ bắt đầu với tham số này. Giá trị mặc định là 0.
- **stop**: số nguyên kết thúc, chuỗi sẽ kết thúc với tham số này.
- **step**: số nguyên xác định khoảng cách giữa các số bên trong chuỗi. Giá trị mặc định là 1.

Ví dụ:

```
print(list(range(0)))  
# sử dụng range(stop)  
print(list(range(10)))  
# sử dụng range(start, stop)  
print(list(range(1, 10)))
```

Thực hành:

Câu 1. Nhập từ bàn phím liên tiếp các từ tiếng Anh viết tách nhau bởi dấu cách. Hãy tách thành các từ sau đó in ra màn hình các từ đó theo thứ tự từ điển.

Câu 2. Nhập từ bàn phím chuỗi các số nhị phân viết liên tiếp được nối nhau bởi dấu phẩy, sau đó in ra những giá trị được nhập.

Câu 3. Chuyển kiểu dữ liệu thành chuỗi trong danh sách cho trước.

Câu 4. Viết chương trình tính và hiển thị ra màn hình số Fibonacci thứ n. Với n là số tự nhiên nhập từ bàn phím

- Dãy fibonacci là dãy số nguyên được định nghĩa một cách đệ quy như sau:

$$f(0)=0, f(1) = 1, f(1 < n) = f(n-1) + f(n-2)$$

Giải:

```
3  n = int(input("Nhập số n : "))
4
5  #Sử dụng cấu trúc re nhanh xử lý trường hợp n < 0
6  if n <= 0:
7      print("Vui lòng nhập số nguyên dương!")
8  elif n == 1 or n == 2:
9      print(1)
10 else:
11     firstNum, secondNum = 1, 1
12     for i in range(n-2):
13         #Dãy fibonacci là dãy số nguyên được định nghĩa: f(0)=0, f(1) = 1, f(1 < n) = f(n-1) + f(n-2)
14         firstNum, secondNum = secondNum, firstNum + secondNum
15     print(0, 1, secondNum)
```

Câu 5. Viết một chương trình tính giá trị của $a+aa+aaa+aaaa$ với a là số được nhập vào bởi người dùng.(Giả sử a được nhập vào là 2 thì đầu ra sẽ là: 2468)

Câu 6. Sử dụng hàm len() để so sánh chiều dài của hai chuỗi nhập vào.