

CHỦ ĐỀ 5. GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP  
CỦA MÁY TÍNH

BÀI 18: CÁC CÂU LỆNH VÀO RA ĐƠN GIẢN

A. TÓM TẮT NỘI DUNG

1. CÁC LỆNH VÀO RA ĐƠN GIẢN

- Chức năng: Hàm input() là có chức năng **nhận dữ liệu nhập từ bàn phím** vào Python và trả về kết quả dưới dạng kiểu **xâu string(str)**.

- Cú pháp:

<Tên biến> = **input**(<Dòng thông báo>)

- Trong đó:

- <Tên biến>: Là biến lưu trữ dữ liệu người dùng nhập vào từ bàn phím.
- [Dòng thông báo]: Là một chuỗi thông báo cho người dùng biết nhập dữ liệu cần nhập, chuỗi này có thể là **rỗng**.

- Ví dụ:

| Chương trình                                                         | Kết quả biên dịch                                                   |
|----------------------------------------------------------------------|---------------------------------------------------------------------|
| <pre>name = input("Nhập họ tên em: ") print("Xin chào ", name)</pre> | <pre>Nhập họ tên em: Nguyễn Văn Toán Xin chào Nguyễn Văn Toán</pre> |

🔄 Lưu ý:

- Khi người dùng nhấn phím Enter việc nhập dữ liệu từ bàn phím sẽ kết thúc.
- Giá trị trả về của hàm input() là một xâu, do đó, ta muốn chuyển kiểu từ kiểu xâu sang kiểu dữ liệu khác như (kiểu số nguyên, số thực ...) thì ta phải kết hợp thêm hàm chuyển đổi kiểu dữ liệu

2. CÁC HÀM CHUYỂN ĐỔI KIỂU DỮ LIỆU

- Chúng ta có thể chuyển đổi qua lại giữa các kiểu dữ liệu số và chuỗi thông qua các hàm được Python tích hợp sẵn như **int()**, **float()** và **str()**. Điều này còn được gọi là ép kiểu.

|                     |                                                                         |
|---------------------|-------------------------------------------------------------------------|
| <b>Lệnh int ( )</b> | có chức năng chuyển đổi số thực hoặc xâu chứa số nguyên thành số nguyên |
|---------------------|-------------------------------------------------------------------------|

| Chuyển đổi       | Chương trình                         | Kết quả thông dịch                   |
|------------------|--------------------------------------|--------------------------------------|
| 1. Số thực       | <pre>&gt;&gt;&gt; int(12.6)</pre>    | 12                                   |
| 2. Xâu số nguyên | <pre>&gt;&gt;&gt; int("123")</pre>   | 123 # Chuyển đổi được                |
| 3. Xâu số thực   | <pre>&gt;&gt;&gt; int("10.35")</pre> | # <b>LỖI</b> : Không chuyển đổi được |

Traceback (most recent call last): File "<pyshe11#21>", line 1, in  
<module> int("10.35")  
ValueErrpr : invalid literal for int( ) with base 10: "10.35"

|                       |                                                         |
|-----------------------|---------------------------------------------------------|
| <b>Lệnh float ( )</b> | dùng để chuyển đổi số nguyên và xâu kí tự thành số thực |
|-----------------------|---------------------------------------------------------|

| Chuyển đổi   | Chương trình                             | Kết quả thông dịch |
|--------------|------------------------------------------|--------------------|
| 1. Số nguyên | <code>&gt;&gt;&gt; float(8)</code>       | 8.0                |
| 2. Xâu số    | <code>&gt;&gt;&gt; float("10.23")</code> | 10.23              |

### Lệnh str ()

dùng để chuyển đổi các kiểu dữ liệu khác thành xâu kí tự

| Chuyển đổi   | Chương trình                          | Kết quả thông dịch                    |
|--------------|---------------------------------------|---------------------------------------|
| 1. Biểu thức | <code>&gt;&gt;&gt; str(12+34)</code>  | '46' # Tính tổng rồi chuyển thành xâu |
|              | <code>&gt;&gt;&gt; str(2&gt;3)</code> | Flase                                 |
| 2. Số        | <code>&gt;&gt;&gt; str(12.567)</code> | '12.567'                              |

### Những trường hợp Không chuyển đổi được:

|                                                                   | Chương trình                                                                       |
|-------------------------------------------------------------------|------------------------------------------------------------------------------------|
| 1. Hàm int() không chuyển đổi được xâu số thực                    | <code>&gt;&gt;&gt; int("10.35")</code>                                             |
| 2. Hàm int(), float() không chuyển đổi được xâu là biểu thức toán | <code>&gt;&gt;&gt; int("12+45")</code><br><code>&gt;&gt;&gt; float("12+45")</code> |

### – Chuyển đổi kiểu cho giá trị trả về của hàm input() thành kiểu nguyên, thực

|              |                                                                                        |
|--------------|----------------------------------------------------------------------------------------|
| 1. Số nguyên | <code>&gt;&gt;&gt; n = int(input("Nhập số tự nhiên: "))</code><br>Nhập số tự nhiên: 13 |
| 2. Số thực   | <code>&gt;&gt;&gt; x = float(input("Nhập số thực x: "))</code><br>Nhập số thực x: 7.5  |

### – Nhập nhiều số trên một dòng

|                                                                                                                                                       |                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bước 1: Nhập cả dòng bằng hàm input.<br><code>&gt;&gt;&gt; s = input('Nhập 3 số nguyên: ')</code><br>Nhập 3 số nguyên: 10 20 30                       | Kết quả s là một xâu kí tự<br><code>&gt;&gt;&gt; print(s)</code><br>'10 20 30'                                                                                 |
| Bước 2: Dùng hàm split để tách các số trong xâu input ra.<br><code>&gt;&gt;&gt; a = s.split()</code>                                                  | Kết quả a là một danh sách (kiểu list) gồm các phần tử được tách ra từ xâu s<br><code>&gt;&gt;&gt; print(a)</code><br>['10', '20', '30']                       |
| Bước 3: Sử dụng map để ép các xâu được tách ra trong input sang số int hoặc float tùy theo đầu bài.<br><code>&gt;&gt;&gt; x, y, z = map(int,a)</code> | Kết quả x = 10, y = 20, z = 30<br><code>&gt;&gt;&gt; print(x,y,z)</code><br>10 20 30<br><code>&gt;&gt;&gt; print('Tổng 3 số: ', x+y+z)</code><br>Tổng 3 số: 60 |

Gộp 3 bước trên ta có câu lệnh

```
>>> x, y, z = map(int,input('Nhập 3 số nguyên: ').split())
      Nhập 3 số : 10 20 30
>>> print('Tổng 3 số x, y, z: ', x + y + z)
      Tổng 3 số x, y, z: 60
```

## B. CÂU HỎI TRẮC NGHIỆM

**Câu 1:** `int` là kiểu dữ liệu nào sau đây ?

- A. Kiểu số nguyên      B. Kiểu số thực      C. Kiểu xâu ký tự      D. Kiểu logic

**Câu 2:** `float` là kiểu dữ liệu nào sau đây?

- A. Kiểu số nguyên      B. Kiểu số thực      C. Kiểu xâu ký tự      D. Kiểu logic

**Câu 3:** Kiểu dữ liệu của biểu thức nào sau đây không thuộc kiểu **bool**?

- A.  $67 > 78$       B.  $17 \neq 12 + 3$       C.  $14 == 10 + 2$       D. `'12=10+2'`

**Câu 4:** `str` là kiểu dữ liệu nào sau đây?

- A. Kiểu số nguyên      B. Kiểu số thực      C. Kiểu xâu ký tự      D. Kiểu logic

**Câu 5:** Hàm nào sau đây là hàm kiểm tra kiểu dữ liệu ?

- A. `Style`      B. `Tyle`      C. `Type`      D. `type`

**Câu 6:** Cho biến `s = 'Xin chào'`, Python sẽ cung cấp cho biến `s` kiểu dữ liệu nào?

- A. `string`      B. `bool`      C. `float`      D. `str`

**Câu 7:** Biết `x` có giá trị là **5.6**, vậy `x` thuộc kiểu giá trị nào trong các kiểu sau?

- A. `bool`      B. `int`      C. `float`      D. `str`

**Câu 8:** Cho `x = -49` và `y = 4.9` Python cung cấp kiểu dữ liệu nào cho biến `x` và `y`?

- A. `x` là `int`, `y` là `float`      B. `x` là `float`, `y` là `int`      C. `x`, `y` là `int`      D. `x`, `y` là `float`

**Câu 9:** Hãy cho biết kiểu dữ liệu trả về của biến `a` sau câu lệnh? `a = float(5)`

- A. Kiểu nguyên      B. Kiểu thực      C. Kiểu ký tự      D. Kiểu logic

**Câu 10:** Cho `x = 3.4` và `y = int(x)`. Biến `y` nhận giá trị là:

- A. 3      B. 4      C. 3.4      D. Báo lỗi

## C. CÂU HỎI ĐÚNG/SAI

**Câu 1:** Trong Python, câu lệnh `input()` được sử dụng để nhận dữ liệu từ người dùng, còn câu lệnh `print()` dùng để hiển thị dữ liệu ra màn hình.

- a) Câu lệnh `input()` luôn trả về dữ liệu dưới dạng chuỗi (`str`).
- b) Câu lệnh `print()` có thể hiển thị nhiều giá trị cùng lúc bằng cách sử dụng dấu phẩy “,”.
- c) Hàm `input()` có thể nhận nhiều giá trị cùng lúc từ người dùng mà không cần xử lý thêm.
- d) Có thể sử dụng tham số sep trong `print()` để tùy chỉnh ký tự phân cách giữa các giá trị được in ra.

**Câu 2:** Câu lệnh `input()` có thể được kết hợp với các kiểu dữ liệu khác nhau để xử lý dữ liệu đầu vào.

- a) Dữ liệu từ `input()` luôn cần được chuyển đổi sang kiểu số nếu muốn thực hiện phép toán số học.
- b) Có thể sử dụng `input()` để nhập một danh sách (`list`) trực tiếp từ người dùng mà không cần xử lý thêm.
- c) Dữ liệu nhập vào từ `input()` có thể được ép kiểu sang `int` hoặc `float` bằng cách sử dụng `int(input())` hoặc `float(input())`.
- d) Câu lệnh `input()` có thể nhận tham số để hiển thị lời nhắc nhập dữ liệu.

**Câu 3:** Câu lệnh `print()` có nhiều tùy chọn để định dạng đầu ra.

- a) Có thể sử dụng `print(f'Xin chào, {name}!')` để chèn giá trị của biến `name` vào chuỗi in ra.
- b) Tham số end trong `print()` dùng để kiểm soát ký tự kết thúc dòng in ra, mặc định là dấu xuống dòng (`\n`).

c) print() không thể in một danh sách (list) hoặc từ điển (dictionary).

d) Có thể sử dụng toán tử + để nối nhiều chuỗi khi sử dụng print(), nhưng không thể dùng dấu phẩy “,”.

## BÀI 19: CÂU LỆNH Rẽ NHÁNH IF

### A. TÓM TẮT NỘI DUNG

#### 1. Rẽ NHÁNH

– Một việc được thực hiện hay không phụ thuộc vào một điều kiện thì ta thể hiện bằng câu mệnh đề có dạng điều kiện:

○ Dạng thiếu: **Nếu ... thì ...**

Ví dụ 1: Nếu a chia hết cho 2 thì a là số chẵn

○ Dạng đủ: **Nếu ... thì ... ngược lại...**

Ví dụ 2: Nếu a chia hết cho 2 thì a là số chẵn ngược lại a là số lẻ

– Trong lập trình, cấu trúc dùng để mô tả các mệnh đề có dạng như trên gọi là: **cấu trúc rẽ nhánh thiếu và đủ**

#### 2. LỆNH IF

##### a) Câu lệnh rẽ nhánh dạng thiếu

– Cú pháp:

**if <điều kiện>:** ← Sau điều kiện cần có dấu hai chấm “:”  
**<Khối lệnh>** ← Nhóm, khối lệnh tiếp theo cần **viết lùi vào** (bằng **1 Tab** hay **4 dấu cách**) và thẳng hàng

– Trong đó:

○ **<Khối lệnh>**: là một câu lệnh (không cần xuống hàng mà viết sau dấu “:”) hoặc nhiều câu lệnh được viết lùi vào (bằng **1 Tab** hay **4 dấu cách**) và thẳng hàng.

– **<Điều kiện>**: là biểu thức quan hệ hoặc biểu thức logic.

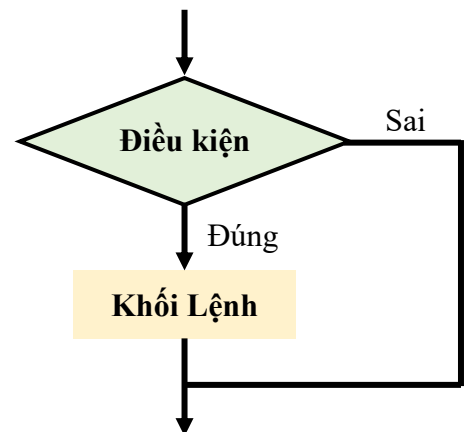
– Thực hiện lệnh:

Python sẽ kiểm tra **<điều kiện>** **nếu đúng thì** thực hiện **<khối lệnh>**, **ngược lại thì bỏ qua** chuyển sang lệnh tiếp theo sau lệnh if.

– Chạy chương trình cho biết kết quả xuất ra màn hình:

Câu 1:

```
x, y = 5, 10
if x > y:
    x+=1
y+=1
print('x =', x, '; y =', y)
```



Hình 19.2 Cấu trúc rẽ nhánh dạng thiếu

A. x = 6 ; y = 10

B. x = 5 ; y = 10

C. x = 5 ; y = 11

D. x = 6 ; y = 11

Câu 2:

```
x, y = 5, 10
if x <= y:
    x+=1
y+=1
print('x =', x, '; y =', y)
```

A. x = 6 ; y = 10

B. x = 5 ; y = 10

C. x = 5 ; y = 11

D. x = 6 ; y = 11

– VD: Dùng cấu trúc rẽ nhánh dạng thiếu để mô tả các mệnh đề rẽ nhánh sau:

|                                               |       |
|-----------------------------------------------|-------|
| (1) Nhập diem                                 | ..... |
| Nếu diem >= 19.5                              | ..... |
| thì xuất ra thông báo “đạt THPT LTK”          | ..... |
| (2) Nhập DTB                                  | ..... |
| Nếu 6.5 <= DTB <8.0                           | ..... |
| thì xuất ra thông báo “học lực khá”           | ..... |
| (3) Nếu a chia hết cho 2 thì a là số chẵn     | ..... |
|                                               | ..... |
| (4) Nếu a không chia hết cho 2 thì a là số lẻ | ..... |
|                                               | ..... |

### b) Câu lệnh rẽ nhánh dạng đủ

– Cú pháp:

```
if <điều kiện>:
    <Khối lệnh 1>
else:
    <Khối lệnh 2>
```

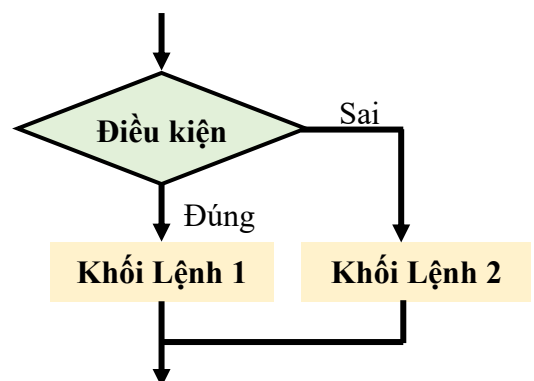
- Từ khóa if và else cần viết thẳng lề trái
- Các khối lệnh 1 và khối lệnh 2 cần **viết lùi vào** (bằng 1 Tab hay 4 dấu cách) và thẳng hàng

- Thực hiện lệnh:  
Python sẽ kiểm tra <điều kiện> **nếu đúng** thì thực hiện <khối lệnh 1>, **ngược lại** thì thực hiện <khối lệnh 2>.

- Chạy chương trình cho biết kết quả xuất ra màn hình:

Câu 1:

```
x, y = 5, 10
if x >= y:
    x+=1
else:
    y+=1
x+=2
```



Hình 19.3 Cấu trúc rẽ nhánh dạng đủ

```
print('x =', x, '; y =', y)
```

A. x = 6 ; y = 10

B. x = 7 ; y = 11

C. x = 8 ; y = 10

D. x = 5 ; y = 11

Câu 2:

```
x, y = 5, 10
```

```
if x > y:
```

```
    x+=1
```

```
else:
```

```
    y+=1
```

```
x+=2
```

```
print('x =', x, '; y =', y)
```

A. x = 6 ; y = 10

B. x = 7 ; y = 11

C. x = 8 ; y = 10

D. x = 5 ; y = 11

– VD: Dùng cấu trúc rẽ nhánh dạng thiếu để mô tả các mệnh đề rẽ nhánh sau:

(1) Nhập a

Nếu a chia hết cho 2

thì a là số chẵn

Ngược lại a là số lẻ

(2) if Delta < 0:

print(' PT vô nghiệm ')

else:(Delta>=0)

print(' PT có nghiệm.

### c) Cấu trúc if lồng nhau

– Cú pháp:

```
if <điều kiện>:
```

```
    <khối lệnh 1>
```

```
elif <điều kiện 2>:
```

```
    <khối lệnh 2>
```

```
else:
```

```
    <khối lệnh 3>
```

**Ví dụ:** Nhập x. Nếu x lớn hơn 0 thì thông báo x là số dương, nếu x < 0 thì thông báo x là số âm, ngược lại thông báo x=0.

```
x = float(input('Hãy nhập số x: '))
```

```
if x > 0:
```

```
    print(x, 'là số dương')
```

```
elif x < 0:
```

```
    print(x, 'là số âm')
```

```
else:
```

```
    print('x = 0')
```

## B. CÂU HỎI TRẮC NGHIỆM

Câu 1: Cho a=15, b=8. Hãy tính giá trị của a, b trong trường hợp sau?

```
if a > b:
```

```
    a=b-a
```

```
b=a+2
```

```
else:
```

```
    a=b-a
```

A. a = 15, b = 8.

B. a = -7, b = -5

C. a = 5, b = -7.

D. a = 5, b = -7.

Câu 2: Phát biểu nào sau đây có thể lấy làm biểu thức điều kiện trong cấu trúc rẽ nhánh?

A. A+B

B. A>B

C. A MOD 100

D. "A nhỏ hơn B"

Câu 3: Cho đoạn chương trình sau:

```

if (a!=0):
    x=9 // a
else:
    x=-1
print ( ' x= ', x + 1)

```

Khi cho  $a=0$ , đoạn chương trình trên sẽ in ra màn hình giá trị  $x=?$

A. 0                                      B. -1  
C. 1                                      D. Không xác định

**Câu 4:** Khối lệnh trong đoạn chương trình:

```

if <điều kiện>:
    <khối lệnh>

```

- A. Luôn thực hiện                                      B. Thực hiện khi điều kiện đúng  
C. Thực hiện khi điều kiện sai                                      D. Điều kiện không tính được.

**Câu 5:** Đoạn chương trình:

```

if <điều kiện>:
    <khối lệnh 1>
else:
    <khối lệnh 2>

```

- A. Thực hiện khối lệnh 1 khi điều kiện đúng  
B. Thực hiện khối lệnh 1 khi điều kiện sai  
C. Thực hiện khối lệnh 2 khi điều kiện đúng  
D. Thực hiện khối lệnh 1 trước rồi đến khối lệnh 2

**Câu 6:** Trong ngôn ngữ lập trình Python, để diễn đạt rẽ nhánh ta dùng câu lệnh **if - else**, sau **if** là **<điều kiện>** vậy **điều kiện** là?

- A. phép toán logic                                      B. Biểu thức số học  
C. Câu lệnh đơn                                      D. Biến kiểu bool, biểu thức so sánh, biểu thức logic

**Câu 7:** Đoạn chương trình:

**Max=a**

**if b>Max:**

**Max=b**

Hãy cho biết đoạn chương trình bên dùng để?

- A. Tính giá trị a                                      B. Tính giá trị b  
C. Tìm GTLN trong 2 số a và b                                      D. Tìm GTNN trong 2 số a và b

**Câu 8:** Câu lệnh ghép được sử dụng khi?

- A. Cần nhiều lệnh đơn thực hiện một công việc                                      B. Ghép nhiều câu lệnh thành một câu lệnh  
C. Tác động của câu lệnh trước đó đến nhiều câu lệnh                                      D. Cả ba trường hợp trên

**Câu 9:** Cho biết giá trị của T sau khi thực hiện đoạn chương trình sau:

```

T,A,B=0,5,7
if A<B:
    T=T+A
else:
    T=T-A
print(T)

```

- A. 2                                      B. 7  
C. 5                                      D. 12

**Câu 10:** Đoạn chương trình:

```

if b>a:
    Max=b
else:
    Max=a

```

Hãy cho biết đoạn chương trình bên dùng để?

- A. Tính giá trị a                                      B. Tính giá trị b  
C. Tìm giá trị lớn nhất trong 2 số a và b                                      D. Cả ba lựa chọn trên

## C. CÂU HỎI ĐÚNG/ SAI

**Câu 1:** Trong Python, câu lệnh rẽ nhánh if được sử dụng để kiểm tra điều kiện và thực thi các khối lệnh khác nhau tùy thuộc vào điều kiện đó, giúp kiểm soát luồng thực thi của chương trình.

- a) Câu lệnh if có thể được sử dụng để kiểm tra một điều kiện và thực thi một khối lệnh nếu điều kiện đó đúng.  
b) Câu lệnh if luôn yêu cầu phải có else đi kèm.

- c) Có thể sử dụng if...elif...else để kiểm tra nhiều điều kiện trong cùng một khối lệnh.
- d) Câu lệnh if chỉ có thể kiểm tra các điều kiện là số nguyên, không thể kiểm tra chuỗi hoặc danh sách.

**Câu 2:** Câu lệnh if có thể sử dụng với các kiểu dữ liệu khác nhau để kiểm tra điều kiện.

- a) Trong Python, các giá trị như 0, None, "" (chuỗi rỗng) và [] (danh sách rỗng) được coi là False trong điều kiện if.
- b) Python cho phép sử dụng if để kiểm tra xem một danh sách có rỗng hay không.
- c) Câu lệnh if không thể kiểm tra trực tiếp kiểu dữ liệu set.
- d) Khi so sánh hai chuỗi bằng if, Python phân biệt chữ hoa và chữ thường.

**Câu 3:** Câu lệnh rẽ nhánh if có thể kết hợp với các toán tử logic (and, or, not) để kiểm tra nhiều điều kiện đồng thời.

- a) Có thể sử dụng if để kiểm tra nhiều điều kiện cùng lúc bằng cách kết hợp toán tử and, or.
- b) Toán tử not có thể được dùng để đảo ngược giá trị của một điều kiện trong if.
- c) Khi sử dụng if A or B:, cả hai điều kiện A và B đều phải đúng thì khối lệnh mới được thực thi.
- d) Python cho phép viết if mà không có điều kiện kiểm tra.

## BÀI 20: CÂU LỆNH LẶP FOR

### A. TÓM TẮT NỘI DUNG

#### 1. LỆNH RANGE

- Hàm range() sẽ sinh ra một dãy số và bạn sẽ sử dụng vòng for để duyệt qua từng số trong dãy .

- Cú pháp:

- Các tham số:

**range (start, stop, step)**

- **start:** Giá trị bắt đầu của dãy số (mặc định là 0).
- **stop:** Giá trị cuối cùng của dãy số (cận này không được lấy – nghĩa là dãy sinh ra sẽ đi từ **start đến stop -1**)
- **step:** Bước nhảy của dãy số (mặc định là 1).

**range (stop)** trả lại vùng giá trị từ 0 đến stop - 1  
**range (start, stop)** trả lại vùng giá trị từ start đến stop - 1

- Ví dụ:

- + **range (100)** là dãy số nguyên liên tục từ **0 đến 99** (dãy số tăng)
- + **range (50)** là dãy số nguyên liên tục từ **0 đến 49**
- + **range (2, 10)** là dãy số nguyên liên tục từ **2 đến 9**
- + **range (5, 20)** là dãy số nguyên liên tục từ ..... **đến** .....
- + **range (2, 10, 2)** là dãy các số nguyên **2, 4, 6, 8**
- + **range (5, 20, 5)** là dãy các số nguyên .....

- + `range(100, 0, -1)` là dãy số nguyên 100, 99, 98, ..., 3, 2, 1 (dãy số giảm)
- + `range(10, 1, -2)` là dãy các số nguyên .....
- + `range(100, 1)` cho vùng rỗng.

## 2. LỆNH LẶP FOR

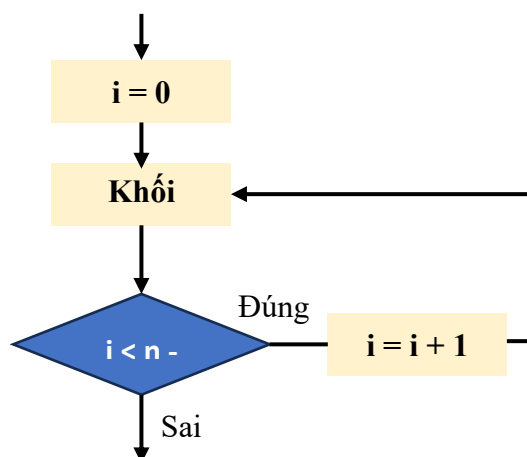
– Cú pháp:

```
for <biến lặp> in <chuỗi lặp>:
    <Khối lệnh>
```

– Trong đó:

- **<chuỗi lặp>** là chuỗi cần lặp có thể là **chuỗi**, **danh sách (kiểu list)** hoặc **dãy số**.
- **<biến lặp>** là **biến nhận giá trị của từng mục bên trong <chuỗi lặp>** trên mỗi lần lặp.  
Vòng lặp sẽ tiếp tục cho đến khi nó lặp tới mục cuối cùng trong **<chuỗi lặp>**
- **<Khối lệnh>** có thể là một câu lệnh (không cần xuống hàng mà viết sau dấu “:”) hoặc nhiều câu lệnh được viết lùi vào (bằng **1 Tab** hay **4 dấu cách**) và thẳng hàng.

```
for i in range(n):
    <Khối lệnh>
```



| Chương trình                                                                                                                                          | Kết quả chạy chương trình                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| <b>&lt;chuỗi lặp&gt; là dãy số.</b>                                                                                                                   |                                                          |
| Vd 1: Xuất ra màn hình dãy số từ 0 đến 4<br><pre>for i in range(5):     print(i, end=', ')</pre>                                                      | 0, 1, 2, 3, 4,                                           |
| Vd 2: Xuất ra màn hình 5 lần câu “Xin chào”<br><pre>for i in range(5):     print('Xin chào')</pre>                                                    | Xin chào<br>Xin chào<br>Xin chào<br>Xin chào<br>Xin chào |
| Vd 3: Xuất ra màn hình dãy số từ 0 đến n-1. Với n nhập từ bàn phím<br><pre>n = int(input("Nhập n: ")) for i in range(n):     print(i, end=', ')</pre> | Nhập n: <b>10</b><br>0, 1, 2, 3, 4, 5, 6, 7, 8, 9,       |
| Vd 4: Tính tổng các số tự nhiên nhỏ hơn n (0+1+2+3+...+n-1). Xuất tổng ra màn hình                                                                    | Nhập n: <b>6</b><br>Tổng các số là 15                    |

| Chương trình                                                                                                                                                                                                                   | Kết quả chạy chương trình                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| <pre>n = int(input ("Nhập n: ")) tong = 0 for i in range(n):     tong = tong + i print('Tổng các số là', tong)</pre>                                                                                                           |                                                             |
| <p>Vd 5: Tính <b>tổng</b> các số tự nhiên <b>chẵn</b> nhỏ hơn n. Xuất tổng ra màn hình</p> <pre>n = int(input ("Nhập n: ")) tong = 0 for i in range(n):     .....     tong = tong + i print('Tổng các số chẵn là', tong)</pre> |                                                             |
| <p>Vd 6: <b>Đếm</b> có bao nhiêu <b>số là bội của 3</b> trong dãy số tự nhiên nhỏ hơn n. Xuất đếm ra màn hình</p> <pre>n = int(input ("Nhập n: ")) dem = 0 for i in range(n):     .....     ..... print(.....)</pre>           |                                                             |
| <b>&lt;chuỗi lặp&gt; là chuỗi</b>                                                                                                                                                                                              |                                                             |
| <p>Vd 7: Lặp từng ký tự trong chuỗi</p> <pre>for kytu in 'tin':     print('Ký tự hiện tại:', kytu)</pre>                                                                                                                       | Ký tự hiện tại: t<br>Ký tự hiện tại: i<br>Ký tự hiện tại: n |
| <b>&lt;chuỗi lặp&gt; là danh sách (list)</b>                                                                                                                                                                                   |                                                             |
| <p>Vd 8: Lặp theo danh sách (list)</p> <pre>ds = ['bố', 'mẹ', 'em'] for i in ds:     print('Tôi yêu', i)</pre>                                                                                                                 | Tôi yêu bố<br>Tôi yêu mẹ<br>Tôi yêu em                      |
| <p>Vd 9: Tính tổng tất cả các số trong danh sách A</p> <pre>A = [1, 7, 10] tong = 0 for i in A:     tong = tong+i print("Tổng các số là", tong)</pre>                                                                          | Tổng các số là: 18                                          |

## B. CÂU HỎI TRẮC NGHIỆM

**Câu 1:** Vòng lặp nào trả về kết quả như cột bên trái dưới đây?

|              |                                               |
|--------------|-----------------------------------------------|
| <b>11111</b> | A. for i in range(1, 6): print(i, i, i, i, i) |
| <b>22222</b> | B. for i in range(1, 5): print(str(i) * 5)    |
| <b>33333</b> | C. for i in range(1, 6): print(str(i) * 5)    |
| <b>44444</b> | D. for i in range(0, 5): print(str(i) * 5)    |
| <b>55555</b> |                                               |

**Câu 2:** Điền phần còn thiếu trong đoạn code để được out như cột bên trái dưới đây:

|       |                                                   |
|-------|---------------------------------------------------|
| 55555 | <code>for i in range(5, 0, ____):</code>          |
| 44444 | <code>print(str(i) * 5)</code>                    |
| 33333 | A. 0                                      B. None |
| 22222 | C. 1                                        D. -1 |
| 11111 |                                                   |

**Câu 4:** Trong NNLT Python, câu lệnh nào sau đây đúng khi in ra màn hình các số tự nhiên từ 0 đến 10 trên 1 dòng?

- A. `for i in range(9): print(i,end=")`                                      B. `for i in range(10): print(i,end=")`  
C. `for i in range(11): print(i,end=")`                                      D. `for i in range(12): print(i,end=")`

**Câu 5:** Trong NNLT Python, câu lệnh nào sau đây sai khi in ra màn hình các số tự nhiên từ 0 đến 10 trên 1 dòng?

- A. `for i in range(0,11,1): print(i,end=")`                                      B. `for i in range(11): print(i,end=")`  
C. `for i in range(0,11): print(i,end=")`                                      D. `for i in range(11,1): print(i,end=")`

**Câu 6:** Trong NNLT Python, câu lệnh nào sau đây đúng khi in ra màn hình các số tự nhiên từ 10 đến 0 trên 1 dòng?

- A. `for i in range(10,0,-1): print(i,end=")`                                      B. `for i in range(10,-1): print(i,end=")`  
C. `for i in range(10,-1,-1): print(i,end=")`                                      D. `for i in range(10,1,-1): print(i,end=")`

**Câu 7:** Cho đoạn chương trình sau:

|                                                                           |                                                                                                                                                               |
|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>s=0</code><br><code>for i in range(5):</code><br><code>s=s+i</code> | <b>Sau khi thực hiện, giá trị của s bằng bao nhiêu?</b><br>A. 5                                      B. 10<br>C. 0                                      D. 15 |
|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Câu 8:** Đoạn chương trình `for i in range(5): print(i,end=")` cho kết quả là:

- A. 1234                                      B. 12345                                      C. 01234                                      D. 012345

**Câu 9:** Đoạn chương trình sau cho kết quả là gì?

|                                                                                             |                                                                                                                  |
|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| <code>for i in range(10):</code><br><code>if i%2==0:</code><br><code>print(i,end='')</code> | A. 02468                                      B. 13579<br>C. 2468                                      D. 246810 |
|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|

**Câu 10:** Đoạn chương trình `for i in range(0,10,2): print(i, end=")` cho kết quả gì?

- A. 2468                                      B. 13579                                      C. 02468                                      D. 246810

## C. CÂU HỎI ĐÚNG/ SAI

**Câu 1:** Trong Python, câu lệnh lặp for được sử dụng để lặp qua một dãy giá trị như danh sách, chuỗi, hoặc phạm vi số, giúp tự động hóa quá trình lặp lại một đoạn mã nhiều lần.

- a) Câu lệnh for có thể được sử dụng để duyệt qua từng phần tử trong một danh sách, bộ, hoặc chuỗi.  
b) Câu lệnh for yêu cầu bắt buộc phải có điều kiện dừng rõ ràng, giống như vòng lặp while.  
c) Sử dụng for kết hợp với range() có thể giúp tạo vòng lặp chạy một số lần nhất định.  
d) Câu lệnh for trong Python không thể lặp qua một từ điển (dictionary).

**Câu 2:** Trong Python, vòng lặp for có thể được sử dụng để duyệt qua nhiều kiểu dữ liệu khác nhau như danh sách, chuỗi, tập hợp và từ điển.

- a) Vòng lặp for có thể duyệt qua từng ký tự trong một chuỗi.  
b) Chỉ có thể sử dụng vòng lặp for với danh sách (list)  
c) Có thể sử dụng for để duyệt qua các cặp khóa-giá trị trong từ điển (dictionary)  
d) Vòng lặp for yêu cầu một biến đếm giống như trong các ngôn ngữ lập trình C hoặc Java.

**Câu 3:** Câu lệnh for có thể được kết hợp với range() để lặp lại một đoạn mã một số lần nhất định.

- a) range(5) sẽ tạo ra một dãy gồm các số từ 0 đến 5.
- b) range(2, 10, 2) tạo ra một dãy số chẵn từ 2 đến 8.
- c) Có thể sử dụng range() trong vòng lặp for để lặp theo bước nhảy âm.
- d) Vòng lặp for trong Python không thể lặp ngược một danh sách.

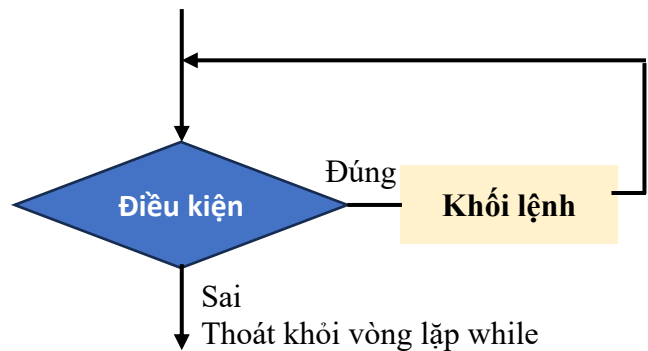
## BÀI 21: CÂU LỆNH LẶP WHILE

### A. TÓM TẮT NỘI DUNG

#### 1. LỆNH WHILE

- Lệnh lặp while thực hiện khối lệnh với **số lần lặp không biết trước**. Khối lệnh lặp được thực hiện cho đến khi **<điều kiện> = False** thì ngưng lặp
- Cú pháp:

```
while <điều kiện>:
    <Khối lệnh>
```

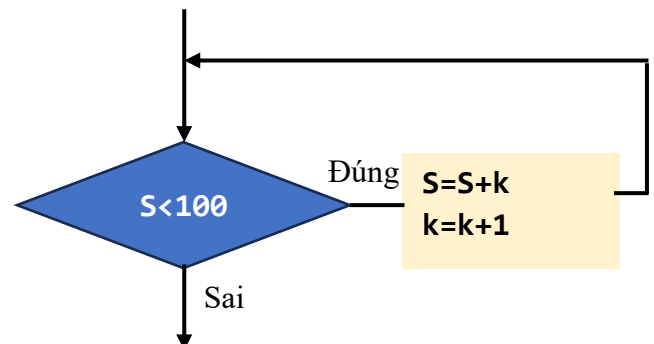


- Trong đó:
  - **<điều kiện>** là biểu thức logic
  - **<Khối lệnh>** có thể là một câu lệnh (không cần xuống hàng mà viết sau dấu “:”) hoặc nhiều câu lệnh được viết lùi vào (bằng 1 Tab hay 4 dấu cách) và thẳng hàng.
- Thực hiện lệnh: Python sẽ **kiểm tra <điều kiện>**, nếu **đúng** thì **thực hiện khối lệnh** lặp, nếu **sai** thì **kết thúc** lệnh while
- Số lần lặp của lệnh while phụ thuộc vào điều kiện của lệnh.

❖ **Lưu ý:** Để thoát khỏi một vòng lặp vô hạn. Khi thực thi chương trình Python ta sử dụng tổ hợp phím **Ctrl + C**.

- **Ví dụ 1:** Tính tổng các số tự nhiên đầu tiên sao cho tổng này nhỏ nhất và lớn hơn 100

```
S=0
k=0
while S<100:
    S=S+k
    k=k+1
print (S)
```



- **Ví dụ 2:** Thực hiện các lệnh sau. Kết quả sẽ in ra những số nào?

```
>>> k = 2
>>> while k < 50:
```

Giải thích: Vòng lặp while sẽ dừng khi k vượt quá 50. Bắt đầu vòng lặp. k = 2. Sau

`print(k, end = " ")` mỗi bước lặp k tăng lên 3 đơn vị. Do vậy, `k = k + 3` kết quả sẽ phải in ra dãy sau:

2 5 8 11 14 17 20 23 26 29 32 35 38 41 44  
47

#### ❖ Lưu ý:

1. Vì lệnh `while` không biết trước số lần lặp, mà phụ thuộc vào điều kiện. Do đó, cần chú ý đến điều kiện của lệnh `while` để tránh bị lặp vô hạn.
2. Trong trường hợp nếu muốn dừng và thoát ngay khỏi vòng lặp `while` hoặc `for` có thể dùng lệnh **break**

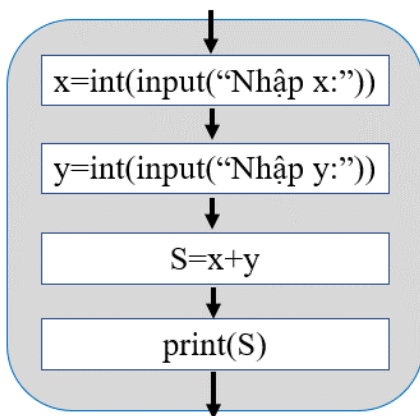
```
>>> for k in range(10):
      print(k, end = " ")
      if k == 5: break
```

Kết quả chương trình:  
0 1 2 3 4 5

## 2. CẤU TRÚC LẬP TRÌNH

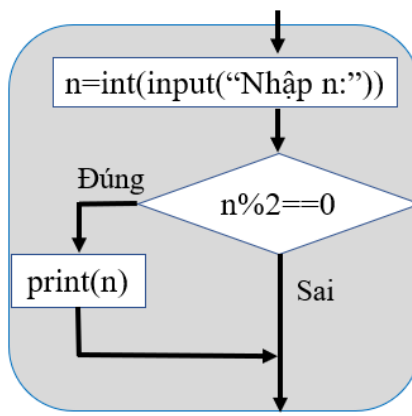
- Ba cấu trúc lập trình cơ bản của các ngôn ngữ lập trình bậc cao gồm: cấu trúc tuần tự, cấu trúc rẽ nhánh, cấu trúc lặp.

```
x=int(input("Nhập x:"))
y=int(input("Nhập y:"))
S=x+y
print(S)
```



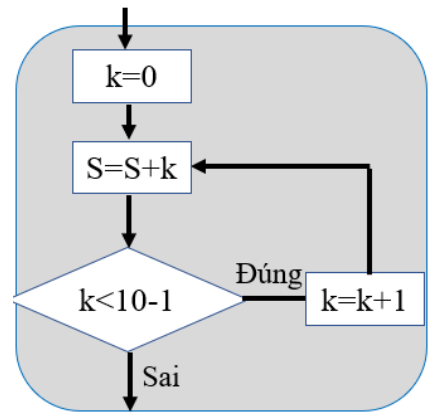
Cấu trúc tuần tự

```
n=int(input("Nhập n:"))
if n%2==0:
    print(n)
```



Cấu trúc rẽ nhánh

```
S=0
for k in range(10):
    S=S+k
```



Cấu trúc lặp

## B. CÂU HỎI TRẮC NGHIỆM

**Câu 1:** Trong câu lệnh **while** thì dãy câu lệnh nào chỉ được thực hiện khi điều kiện còn?

- A. True                      B. True hoặc False                      C. False                      D. Khác 0

**Câu 2:** Trong câu lệnh **while**, ngay sau điều kiện sẽ là dấu nào?

- A. ;                      B. ?                      C. =                      D. :

**Câu 3:** Cú pháp của câu lệnh lặp với số lần chưa biết trước là?

|                                      |                                         |                                       |                                         |
|--------------------------------------|-----------------------------------------|---------------------------------------|-----------------------------------------|
| A. while <điều kiện>:<br><Khối lệnh> | B. while <điều kiện> do<br>< Khối lệnh> | C. while < Khối lệnh>:<br><điều kiện> | D. while < Khối lệnh> do<br><điều kiện> |
|--------------------------------------|-----------------------------------------|---------------------------------------|-----------------------------------------|

**Câu 4: Cho đoạn chương trình:**

|                                                                                         |                                                                                                |
|-----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <pre>1 s=0 2 i=10 3 while i &gt;= 1: 4     s = s + 2*i 5     i = i - 2 6 print(s)</pre> | Kết quả của s sau khi thực hiện đoạn chương trình trên là:<br>A. 60<br>B. 15<br>C. 14<br>D. 28 |
|-----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|

**Câu 5: Cho đoạn chương trình sau:**

|                                                                                                                 |                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>1 a = 12 2 b = 16 3 while a!=b: 4     if a&gt;b: 5         a = a - b 6     else: 7         b = b - a</pre> | Sau khi thực hiện đoạn chương trình trên, giá trị của <b>a</b> và <b>b</b> lần lượt là:<br>A. 4 và 48<br>B. 4 và 4<br>C. 16 và 12<br>D. 12 và 16 |
|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|

**Câu 6: Tính tổng  $S = 1 + 2 + 3 + \dots + n + \dots$  cho đến khi  $S > 109$ . Điều kiện nào sau đây cho vòng lặp **while** là đúng:**

- A. while  $S \geq 109$ :      B. while  $S = 109$ :      C. while  $S \leq 109$ :      D. while  $S \neq 109$ :

**Câu 7: Cho đoạn chương trình sau:**

|                                                                 |                                                                                              |
|-----------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| <pre>1 s=0 2 i=1 3 while i&lt;=5: 4     s=s+1 5     i=i+1</pre> | Sau khi thực hiện đoạn chương trình trên giá trị của s là:<br>A. 9<br>B. 15<br>C. 5<br>D. 10 |
|-----------------------------------------------------------------|----------------------------------------------------------------------------------------------|

## C. CÂU HỎI ĐÚNG/ SAI

**Câu 1:** Trong Python, câu lệnh lặp **while** được sử dụng để lặp lại một đoạn mã khi một điều kiện nhất định còn đúng, giúp kiểm soát luồng thực thi dựa trên điều kiện logic.

- a) Vòng lặp **while** tiếp tục chạy khi điều kiện được xác định trong nó vẫn đúng.
- b) Vòng lặp **while** luôn thực thi ít nhất một lần, ngay cả khi điều kiện ban đầu là **False**.
- c) Có thể sử dụng từ khóa **break** để kết thúc vòng lặp **while** ngay lập tức.
- d) Không thể sử dụng từ khóa **else** với vòng lặp **while** trong Python.

**Câu 2:** Trong Python, vòng lặp **while** có thể kết hợp với các cấu trúc dữ liệu khác để tạo ra các thuật toán linh hoạt.

- a) Vòng lặp **while** có thể duyệt qua danh sách bằng cách sử dụng điều kiện dựa trên độ dài của danh sách.
- b) Khi sử dụng **while** với danh sách, việc xóa phần tử khỏi danh sách trong vòng lặp có thể gây ra lỗi ngoài ý muốn.
- c) Vòng lặp **while** không thể kết hợp với các cấu trúc dữ liệu như **stack** hoặc **queue**.
- d) Sử dụng **while** với **input()** có thể giúp tạo ra các chương trình tương tác liên tục cho đến khi người dùng nhập một giá trị cụ thể.

**Câu 3:** Các vấn đề liên quan đến hiệu suất và tối ưu hóa khi sử dụng vòng lặp **while**.

- a) Vòng lặp while thường chậm hơn for vì phải kiểm tra điều kiện mỗi lần lặp.
- b) Sử dụng vòng lặp while thay vì for có thể làm mã nguồn khó đọc nếu không được kiểm soát tốt.
- c) Vòng lặp while luôn là lựa chọn tốt nhất khi làm việc với danh sách có số lượng phần tử xác định trước.
- d) Để tránh vòng lặp vô hạn, cần đảm bảo điều kiện while sẽ trở thành False tại một thời điểm nào đó.

## BÀI 22: KIỂU DỮ LIỆU DANH SÁCH

### A. NỘI DUNG

#### 1. KIỂU DỮ LIỆU DANH SÁCH

##### a) *Khái niệm kiểu danh sách – list*

- List là kiểu dữ liệu **danh sách (dãy, mảng)** trong Python. Các phần tử của danh sách có thể có các kiểu dữ liệu **khác nhau**.
- **VD 1:** Danh sách chữ cái, danh sách môn học, danh sách bài hát yêu thích,.....

##### b) *Cách khai báo*

- Cú pháp:

**<Tên list> = [<phần tử 0>, <phần tử 1>, ..., <phần tử n-1>]**

- Trong đó:
  - <Tên list>: do người lập trình đặt
  - <Các phần tử>: được nằm trong dấu **ngoặc vuông []** và phân cách nhau bởi **dấu phẩy (,)**.  
Giá trị các phần tử có thể có kiểu dữ liệu khác nhau (int, float, str, bool... và cả kiểu dữ liệu list)

- **VD 2:** Câu lệnh tạo danh sách **A = [1.5, 2, "Python", "List", 0, True]**

| Tên list | Các phần tử trong list A |   |        |      |   |      |
|----------|--------------------------|---|--------|------|---|------|
| A        | 1.5                      | 2 | Python | List | 0 | True |

##### c) *Cách truy cập các phần tử trong danh sách:*

- Cú pháp:

**<Tên list>[chỉ số]**

- Trong đó:
  - Chỉ số các phần tử được đánh số bắt đầu từ: 0, 1, 2... (Từ trái sang phải)
  - Hoặc chỉ số các phần tử được đánh số âm bắt đầu từ: -1, -2, -3... (Từ phải sang trái)

- **VD 4:** A = [1.5, 2, "Python", "List", 0, True]

print(A[1])                      2

|                 |                       |
|-----------------|-----------------------|
| print(A[-1])    | True                  |
| print(A[2:5])   | ['Python', 'List', 0] |
| print(A[-3:-1]) | ['List', 0]           |

#### d) Các phép toán trên danh sách

##### ✚ Phép nối danh sách: Sử dụng dấu “+”

< tên list > = < tên list 1> + < tên list 2> + .....< tên list n>

**VD 5:** L1 = [1, 2, 3,] # Kết quả chương trình:  
 L2 = ['a', 'b', 'c']  
 L3 = L1+ L2 .....  
 print(L3)

##### ✚ Phép lặp danh sách: Sử dụng dấu “\*”

**VD 6:** L1 = [1]\*2 L2 = [1, 2, 3]\*3  
 print(L1) print(L2)  
 # Kết quả:..... # Kết quả:.....

## 2. CÁC THAO TÁC TRÊN DANH SÁCH

### a) Nhập một danh sách từ bàn phím:

#### ❖ Cách 1 : Thêm phần tử vào danh sách bằng phép toán nối danh sách

| Chương trình nhập một dãy số nguyên               | Ý nghĩa |
|---------------------------------------------------|---------|
| 1. a = []                                         | .....   |
| 2. n = int(input('Nhập tổng số phần tử list: '))  | .....   |
| 3. for i in range(n):                             | .....   |
| 4. num=int(input('Nhập số thứ '+ str(i+1)+ ': ')) | .....   |
| 5. a = a + [num]                                  | .....   |
| 6. print(a)                                       | .....   |

#### ❖ Cách 2 : Thêm phần tử vào danh sách bằng phương thức append()

- Lệnh thêm phần tử vào cuối danh sách là

**<danh sách>.append()**

| Chương trình nhập một dãy số nguyên               | Ý nghĩa |
|---------------------------------------------------|---------|
| 1. a = []                                         | .....   |
| 2. n = int(input('Nhập tổng số phần tử list: '))  | .....   |
| 3. for i in range(n):                             | .....   |
| 4. num=int(input('Nhập số thứ '+ str(i+1)+ ': ')) | .....   |
| 5. a.append(num)                                  | .....   |

|             |       |
|-------------|-------|
| 6. print(a) | ..... |
|-------------|-------|

**b) Xuất các phần tử của danh sách:**

| Chương trình                                                                                                                                                             | Ý nghĩa                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|
| <b>Cách 1: Xuất toàn bộ danh sách</b><br>a = [2, 4, 3, 1, 5, 7, 8]<br>print("Danh sách a:",a)                                                                            | .....<br>.....                   |
| <b>Cách 2: Xuất từng phần tử trong danh sách</b><br>a = [2, 4, 3, 1, 5, 7, 8]<br>print("Danh sách a:")<br>for i in a:<br>print(i, end= " ")                              | .....<br>.....<br>.....<br>..... |
| <b>Cách 3: Xuất phần tử theo chỉ số trong danh sách</b><br>a = [2, 4, 3, 1, 5, 7, 8]<br>n=len(a)<br>print("Danh sách a:")<br>for i in range(n):<br>print(a[i], end= " ") | .....<br>.....<br>.....          |

## B. CÂU HỎI TRẮC NGHIỆM

**Câu 1:** Trong NNLT Python, đối tượng **L = [1, 23, "Hello", 9]** thuộc kiểu dữ liệu nào?

- A. List                      B. Dictionary                      C. Tuple                      D. Array

**Câu 2:** Đây là giá trị của **color[2]** biết **color = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo']**

- A. 'orange'                      B. 'indigo'                      C. 'blue'                      D. 'yellow'

**Câu 3:** Kết quả sau khi chạy chương trình dưới đây là:

```
my_list = [1, 5, 9, int('0')]
print (sum(my_list))
```

- A. 16                      B. 15                      C. 63                      D. False

**Câu 4:** Kết quả sau khi chạy chương trình dưới đây là:

```
list1=[1, 3]
list2 = list1
list1[0] = 4
print (list2)
```

- A. [4, 3]                      B. [1, 3]                      C. [1, 4]                      D. [1, 3, 4]

**Câu 5:** Kết quả của câu lệnh **num = '5' \* '5'** là gì?

- A. 333                      B. 27                      C. 25                      D. TypeError

**Câu 6:** Kết quả sau khi chạy chương trình dưới đây là:

```
string ="mynameisx"
for i in string:
    print (i, end=',')
```

- A. x,s,i,e,m,a,n,y,m                      C. m,y,n,a,m,e,i,s  
B. m,y,n,a,m,e,i,s,x                      D. m,,y,,n,,a,,m,,e,,i,,s,,x

**Câu 7:** Kết quả sau khi chạy chương trình dưới đây là:

```
all=([2,4,0,6])
print(all)
```

- A. 2,4,0,6                      B. True                      C. [2,4,0,6]                      D. (2,4,0,6)

**Câu 8:** Kết quả sau khi chạy chương trình dưới đây là:

```
any=([2>8,4>2,1>2])
```

```
print (any)
```

A. [False, True, False]

B. False

C. (False, True, False)

C. Error

**Câu 9:** Muốn tham chiếu đến 1 phần tử nào đó trong danh sách, ta cần phải có thông tin nào?

A. Tên danh sách, chỉ số

C. Chỉ số của kí tự trong danh sách

B. Độ dài danh sách

D. Phải biết đó là kí tự gì trong danh sách

**Câu 10:** Chọn câu phát biểu đúng về dữ liệu kiểu danh sách:

A. Các phần tử trong danh sách được phân cách nhau bởi dấu phẩy (,).

B. Các phần tử trong danh sách được phân cách nhau bởi dấu chấm phẩy (;).

C. Các phần tử trong danh sách được phân cách nhau bởi dấu hai chấm (:).

D. Các phần tử trong danh sách được phân cách nhau bởi dấu cách.

### C. CÂU HỎI ĐÚNG/SAI

**Câu 1:** Trong Python, danh sách (list) là một kiểu dữ liệu có thể lưu trữ nhiều giá trị cùng lúc:

a) Danh sách trong Python được đặt trong cặp dấu ngoặc vuông [ ].

b) Các phần tử trong danh sách được phân tách nhau bằng dấu phẩy ,.

c) Hàm len() có thể dùng để đếm số phần tử trong danh sách.

d) Phần tử cuối cùng trong danh sách có chỉ số là 0.

**Câu 2:** Trong Python, danh sách là một cấu trúc dữ liệu cho phép lưu trữ nhiều giá trị:

a) Một danh sách có thể rỗng, tức là không chứa phần tử nào.

b) Danh sách có thể chứa các giá trị có kiểu dữ liệu khác nhau (vd: số, chuỗi, danh sách...).

c) Dùng cú pháp my\_list[index] để truy cập phần tử tại vị trí index.

d) Các phần tử trong danh sách được phân cách bằng dấu chấm phẩy ;.

**Câu 3:** Xét về cú pháp và cách sử dụng danh sách (list) trong Python:

a) Có thể tạo danh sách bằng cách sử dụng hàm list().

b) Câu lệnh my\_list = [1, 2, 3] tạo ra một danh sách gồm 3 phần tử.

c) Phần tử đầu tiên trong danh sách có chỉ số là 1.

d) Có thể gán danh sách cho một biến để sử dụng về sau.

## BÀI 23: MỘT SỐ LỆNH LÀM VIỆC VỚI DỮ LIỆU DANH SÁCH

### A. NỘI DUNG

#### 1. DUYỆT DANH SÁCH VỚI TOÁN TỬ IN

a) *Kiểm tra sự tồn tại của một giá trị trong danh sách:*

Cú pháp:

```
<giá trị> in <danh sách>
```

=> Trả về giá trị True nếu <Giá trị> thuộc <Tên list>, ngược lại trả về giá trị False

VD1:

```
traicay= ["tao", "le", "man"]
print ("tao" in traicay) # Kết quả: True
print ("oi" in traicay) # Kết quả: False
```

### b) Duyệt từng phần tử của danh sách:

Phân tích và cho biết giá trị của k trong các đoạn lệnh sau:

|                                                                            |              |      |      |      |      |      |      |
|----------------------------------------------------------------------------|--------------|------|------|------|------|------|------|
| A = [10, 11, 12, 13, 14, 15]<br>for k in range(6):<br>print(A[k], end=" ") | A            | 10   | 11   | 12   | 13   | 14   | 15   |
|                                                                            | k            | 0    | 1    | 2    | 3    | 4    | 5    |
|                                                                            | Phần tử A[k] | A[0] | A[1] | A[2] | A[3] | A[4] | A[5] |
| A = [10, 11, 12, 13, 14, 15]<br>for k in A:<br>print(k, end=" ")           | A            | 10   | 11   | 12   | 13   | 14   | 15   |
|                                                                            | k            | 10   | 11   | 12   | 13   | 14   | 15   |

– Cú pháp duyệt từng phần tử của danh sách:

**for k in <danh sách>:  
    <khối lệnh lặp>**

## 2. MỘT SỐ LỆNH LÀM VIỆC VỚI DANH SÁCH:

| STT | Hàm                                 | Ý nghĩa                                                                                                                | Ví dụ                                                                                                              |
|-----|-------------------------------------|------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| 1.  | <b>len(&lt;tên list &gt;)</b>       | Trả về độ dài của một danh sách                                                                                        | L4= [1, 'hello', 3.5]<br>print(len(L4)) → 3                                                                        |
| 2.  | <b>max (&lt;tên list &gt;)</b>      | Trả về giá trị của phần tử lớn nhất trong danh sách.cùng kiểu                                                          | L1=[456, 700, 200]<br>print(max(L1)) → 700.                                                                        |
| 3.  | <b>min(&lt;tên list &gt;)</b>       | Trả về giá trị của phần tử nhỏ nhất trong danh sách cùng kiểu<br>Ví dụ : L=[1, 2, 5, 'ABC']<br>print(min(L)) → báo lỗi | L2=['ABC', 'B', 'C']<br>print(min(L2))→ ABC                                                                        |
| 4.  | <b>sum (&lt;tên list&gt;)</b>       | Trả về tổng các phần tử trong danh sách (Nếu là danh sách kiểu số)                                                     | L1=[100, 600, 200]<br>print (sum(L1))→ 900                                                                         |
| 5.  | <b>del &lt;tên list&gt;[chỉ số]</b> | Thực hiện xóa phần tử có chỉ số trong danh sách.<br><br>Có thể xóa 1 đoạn ví dụ:                                       | L=[1, 2, 3, 1, 4]<br>del L[2]<br>→L1=[1,2,1,4]<br>L1=[1, 2, 3, 5, 9, 8, 10]<br>del L1[1:3]<br>→L1=[1, 5, 9, 8, 10] |
| 6.  | <b>list.append()</b>                | Thêm 1 phần tử vào cuối danh sách.                                                                                     | sole = [1,3,5]<br>sole.append(7)<br>print(sole) → [1,3,5,7]                                                        |

| STT | Hàm                             | Ý nghĩa                                                     | Ví dụ                                                                                        |
|-----|---------------------------------|-------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| 7.  | <b><i>list.extend()</i></b>     | Thêm nhiều phần tử vào cuối danh sách.                      | sole = [1,5,6]<br>sole.extend([3,8,6])<br>print(sole) → [1,5,6,3,8,6]                        |
| 8.  | <b><i>list.count()</i></b>      | Trả về số lần xuất hiện của chuỗi con trong danh sách.      | my = [3,8,1,6,0,8,4]<br>print (my.count(8))<br>→ 2                                           |
| 9.  | <b><i>list.sort()</i></b>       | Sắp xếp các đối tượng trong danh sách theo thứ tự tăng dần. | list1 = [3,7,8,2,5]<br>list1.sort()<br>print(list1) → [2,3,5,7,8]                            |
| 10. | <b><i>list.reverse()</i></b>    | Đảo ngược các phần tử trong danh sách.                      | list2 = [8,7,5,9,2,1]<br>list2.reverse()<br>print (list2) → [1,2,9,5,7,8]                    |
| 11. | <b><i>list.index()</i></b>      | Trả về vị trí xuất hiện phần tử trong danh sách.            | my_list = [3,8,1,6,5,8,4]<br>print(my_list.index(6))<br>→ 3                                  |
| 12. | <b><i>list.clear()</i></b>      | Xóa hết phần tử trong danh sách.                            | your_list = [4,5,6,2,8,2]<br>your_list.clear()<br>print (your_list)<br>→[ ]                  |
| 13. | <b><i>list.remove()</i></b>     | Xóa phần tử có giá trị cho trước trong danh sách            | A = [1,4,7,8,2,6]<br>A.remove(8) print (A)<br>→ [1,4,7,2,6]                                  |
| 14. | <b><i>list.insert(vt,n)</i></b> | Thêm phần tử n mới vào danh sách ở vị trí thứ vt            | vowel = ['a','e','i','u']<br>vowel.insert(3, 'o')<br>print(vowel)<br>→ ['a','e','i','o','u'] |

## B. CÂU HỎI TRẮC NGHIỆM

**Câu 1:** Cho danh sách: **I = [2, 78, 5, 21]**, Nếu muốn danh sách này đảo ngược thứ tự ta sử dụng phương pháp nào?

- A. I.reverse()                      B. reverse(I)                      C. reversed(I)                      D. I(reverse[ ])

**Câu 2:** Cho biết hàm remove trong đoạn chương trình sau dùng để làm gì?

```
A = ['A', 'B', 'C']
A.remove('A')
print(A)
```

- A. Hoán đổi vị trí phần tử trong list                      C. Sao chép phần tử trong list  
B. Xóa 1 phần tử khỏi list                      D. Di chuyển phần tử trong list đến 1 vị trí khác

**Câu 3:** Để kiểm tra một giá trị có nằm trong danh sách hay không, ta dùng cú pháp nào sau đây?

- A. <giá trị> in <danh sách>                      C. <giá trị> on <danh sách>  
B. <danh sách> in <giá trị>                      D. <giá trị> insert <danh sách>

**Câu 4:** Lệnh **A.append(x)** dùng để làm gì?

- A. Bỏ sung phần tử x vào cuối danh sách.
- B. Bỏ sung phần tử x vào đầu danh sách.
- C. Bỏ sung phần tử x vào giữa danh sách.
- D. Xoá phần tử x từ danh sách A.

**Câu 5:** Để xoá toàn bộ dữ liệu của danh sách A, dùng lệnh nào sau đây?

- A. A . clear()
- B. A . remove()
- C. A . del()
- D. A . append()

**Câu 6:** Lệnh **A.insert(k,x)** dùng để làm gì?

- A. Chèn phần tử x vào vị trí k của danh sách A
- B. Chèn phần tử k vào vị trí x của danh sách A
- C. Chèn phần tử k và x vào danh sách A
- D. Xoá phần tử k và x của danh sách A

**Câu 7:** Để xoá phần tử x từ danh sách A, câu lệnh nào sau đây đúng?

- A. A . remove(x)
- B. A . clear(x)
- C. A . insert(x)
- D. remove.A(x)

**Câu 8:** Các dòng lệnh sau đây cho kết quả là gì?

```
>>> A=[]  
>>> A.insert(0,1)  
>>> A
```

- A. [1]
- B. [0]
- C. [0,1]
- D. [1,0]

**Câu 9:** Các dòng lệnh sau cho kết quả là dãy số nào?

```
>>> A=[1,4,10,0]  
>>> A.insert(3,5)  
>>> A
```

- A. [1, 4, 10, 5, 0]
- B. [1, 4, 10, 3, 0]
- C. [1, 4, 10, 0, 5]
- D. [1, 4, 10, 0, 3]

### C. CÂU HỎI ĐÚNG/SAI

**Câu 1:** Trong Python, có nhiều lệnh và phương thức giúp thao tác với dữ liệu kiểu danh sách (list)

- a) Phương thức append() dùng để thêm một phần tử vào cuối danh sách.
- b) Phương thức remove() dùng để xóa phần tử đầu tiên có giá trị được chỉ định.
- c) Phương thức insert() dùng để thay thế phần tử tại một vị trí trong danh sách.
- d) Hàm len() được dùng để đếm số phần tử trong danh sách.

**Câu 2:** Trong Python, có thể sử dụng nhiều lệnh để thay đổi hoặc truy cập dữ liệu trong danh sách.

- a) Phương thức pop() có thể dùng để xóa phần tử ở vị trí chỉ định và trả về giá trị đó.
- b) Phương thức sort() dùng để sắp xếp danh sách theo thứ tự tăng dần.
- c) Phương thức clear() dùng để xóa toàn bộ các phần tử trong danh sách.
- d) Phương thức extend() tạo bản sao của danh sách hiện tại.

**Câu 3:** Các phương thức và cú pháp sau đây thường được dùng khi thao tác với danh sách trong Python.

- a) Phương thức count(x) trả về số lần xuất hiện của phần tử x trong danh sách.
- b) my\_list[0] là cách truy cập phần tử đầu tiên của danh sách.
- c) Dùng del my\_list[2] để xóa phần tử thứ ba trong danh sách.
- d) Phương thức reverse() đảo ngược thứ tự phần tử trong danh sách và trả về danh sách mới.

## BÀI 24: XÂU KÍ TỰ

### A. NỘI DUNG

#### 1. XÂU LÀ MỘT DÃY CÁC KÍ TỰ

| XÂU KÍ TỰ                                                                                                                                          | DANH SÁCH                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <pre>&gt;&gt;&gt; S="THPT Lý Thường Kiệt" &gt;&gt;&gt; len(S) 19 &gt;&gt;&gt; S[0] 'T' &gt;&gt;&gt; S[9] 'T' &gt;&gt;&gt; S[5]="P" → Báo Lỗi</pre> | <pre>&gt;&gt;&gt; A=[3,4,5,"b","d"] &gt;&gt;&gt; len(A) 5 &gt;&gt;&gt; A[0] 3 &gt;&gt;&gt; A[4] 'd' &gt;&gt;&gt; A[3]="B"</pre> |

- Xâu kí tự là **dãy các kí tự Unicode**
- Độ dài của xâu kí tự chính là **số lượng các kí tự trong xâu**
- Xâu có độ dài bằng 1 chỉ **gồm 1 kí tự**
- Xâu rỗng là xâu không có **độ dài bằng 0** (xâu không có kí tự nào)
- Truy cập từng kí tự của xâu qua **chỉ số**, chỉ số từ **0** đến **độ dài len() -1**
- Không thể *thay đổi từng kí tự* của xâu

#### 2. LỆNH DUYỆT TỪNG KÍ TỰ CỦA XÂU:

|                                                                              |                                                                                                                                                                                                                                                                                                                                                   |      |      |      |      |      |      |      |   |   |    |   |   |   |  |   |   |   |   |    |      |      |      |      |      |      |      |      |
|------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|------|------|------|------|---|---|----|---|---|---|--|---|---|---|---|----|------|------|------|------|------|------|------|------|
| Duyệt theo chỉ số của xâu:<br>for i in range(len(s)):<br>print(s[i],end=" ") | <table><tr><td>i</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>S:</td><td>X</td><td>I</td><td>N</td><td></td><td>C</td><td>H</td><td>A</td><td>O</td></tr><tr><td>PT</td><td>S[0]</td><td>S[1]</td><td>S[2]</td><td>S[3]</td><td>S[4]</td><td>S[5]</td><td>S[6]</td><td>S[7]</td></tr></table> | i    | 0    | 1    | 2    | 3    | 4    | 5    | 6 | 7 | S: | X | I | N |  | C | H | A | O | PT | S[0] | S[1] | S[2] | S[3] | S[4] | S[5] | S[6] | S[7] |
| i                                                                            | 0                                                                                                                                                                                                                                                                                                                                                 | 1    | 2    | 3    | 4    | 5    | 6    | 7    |   |   |    |   |   |   |  |   |   |   |   |    |      |      |      |      |      |      |      |      |
| S:                                                                           | X                                                                                                                                                                                                                                                                                                                                                 | I    | N    |      | C    | H    | A    | O    |   |   |    |   |   |   |  |   |   |   |   |    |      |      |      |      |      |      |      |      |
| PT                                                                           | S[0]                                                                                                                                                                                                                                                                                                                                              | S[1] | S[2] | S[3] | S[4] | S[5] | S[6] | S[7] |   |   |    |   |   |   |  |   |   |   |   |    |      |      |      |      |      |      |      |      |
| Duyệt theo kí tự của xâu:<br>for ch in s:<br>print(ch,end=" ")               | <table><tr><td>i</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>S:</td><td>X</td><td>I</td><td>N</td><td></td><td>C</td><td>H</td><td>A</td><td>O</td></tr><tr><td>ch</td><td>X</td><td>I</td><td>N</td><td></td><td>C</td><td>H</td><td>A</td><td>O</td></tr></table>                                  | i    |      |      |      |      |      |      |   |   | S: | X | I | N |  | C | H | A | O | ch | X    | I    | N    |      | C    | H    | A    | O    |
| i                                                                            |                                                                                                                                                                                                                                                                                                                                                   |      |      |      |      |      |      |      |   |   |    |   |   |   |  |   |   |   |   |    |      |      |      |      |      |      |      |      |
| S:                                                                           | X                                                                                                                                                                                                                                                                                                                                                 | I    | N    |      | C    | H    | A    | O    |   |   |    |   |   |   |  |   |   |   |   |    |      |      |      |      |      |      |      |      |
| ch                                                                           | X                                                                                                                                                                                                                                                                                                                                                 | I    | N    |      | C    | H    | A    | O    |   |   |    |   |   |   |  |   |   |   |   |    |      |      |      |      |      |      |      |      |

### B. CÂU HỎI TRẮC NGHIỆM

**Câu 1:** Biểu thức `print(" " in "0123")` trả lại giá trị gì?

- A. True                      B. False                      C. Báo lỗi                      D. Số 1

**Câu 2:** Biểu thức logic `"010" in "001100"` trả kết quả nào?

- A. True                      B. False                      C. 4                      D. -1

**Câu 3:** Xâu rỗng là xâu có độ dài bằng:

- A. 0                      B. 1                      C. 2                      D. 3

**Câu 4:** Để biết độ dài của xâu s, ta dùng hàm:

- A. str(s)                      B. len(s)                      C. length(s)                      D. s.len( )

**Câu 5:** Cho xâu `s1 = 'ab'`, xâu `s2 = 'a'`. Câu lệnh `s1 in s2` cho kết quả là:

A. true                      B. True                      C. false                      D. False

**Câu 6:** Cho chuỗi `st = 'abc'`. Chuỗi `st` có độ dài là:

A. 1                      B. 2                      C. 3                      D. 4

**Câu 7:** Để khởi tạo chuỗi rỗng `s`, ta dùng lệnh:

A. `s = '0'`                      B. `s = ''`                      C. `s = []`                      D. `s = 0`

**Câu 8:** Trong NNL Python, chuỗi ký tự là:

- A. Dãy các ký tự trong bảng mã ASCII.
- B. Dãy các ký tự Unicode
- C. Mảng các ký tự.
- D. Tập hợp các chữ cái và các chữ số trong bảng chữ cái tiếng Anh.

**Câu 9:** Trong NNL Python, phần tử đầu tiên của chuỗi ký tự mang chỉ số là?

A. 0                      B. 1                      C. Không có chỉ số                      D. Do người lập trình khai báo

**Câu 10:** Trong NNL Python, chuỗi không có ký tự nào được gọi là?

A. Chuỗi rỗng                      B. Chuỗi trắng                      C. Chuỗi không                      D. Không phải chuỗi

### C. CÂU HỎI ĐÚNG/SAI

**Câu 1:** Trong Python, kiểu dữ liệu chuỗi ký tự (string) được sử dụng để lưu trữ và xử lý văn bản:

- a) Chuỗi trong Python được đặt trong dấu ngoặc đơn `' '` hoặc dấu ngoặc kép `" "`.
- b) Hàm `len()` trả về số ký tự có trong một chuỗi.
- c) Toán tử `+` có thể dùng để nối hai chuỗi lại với nhau.
- d) Có thể thay đổi một ký tự bất kỳ trong chuỗi bằng cách gán trực tiếp qua chỉ số (index).

**Câu 2:** Trong Python, chuỗi (string) là một kiểu dữ liệu bất biến (immutable)

- a) Phương thức `upper()` dùng để chuyển tất cả các ký tự trong chuỗi thành chữ hoa.
- b) Câu lệnh `my_string[2] = "a"` sẽ thay đổi ký tự ở vị trí chỉ số 2 trong chuỗi.
- c) Phương thức `replace()` có thể thay thế một chuỗi con bằng chuỗi con khác trong chuỗi gốc.
- d) Dùng phương thức `strip()` để loại bỏ ký tự khoảng trắng ở đầu và cuối chuỗi.

**Câu 3:** Các phương thức sau đây giúp thao tác với chuỗi trong Python:

- a) Phương thức `split()` giúp chia một chuỗi thành danh sách các chuỗi con dựa trên ký tự phân cách.
- b) Phương thức `find()` trả về vị trí đầu tiên của chuỗi con trong chuỗi gốc.
- c) Phương thức `join()` dùng để nối các ký tự trong chuỗi thành một chuỗi mới.
- d) Dùng cú pháp `my_string[-1]` để lấy ký tự cuối cùng trong chuỗi.

## BÀI 25: MỘT SỐ LỆNH LÀM VIỆC VỚI XÂU KÍ TỰ

### A. NỘI DUNG

#### 1. XÂU LÀ MỘT DÃY CÁC KÍ TỰ

| Phép toán và lệnh                                                                                                                                                                                                                                                                                                                                                                                                                         | Ví dụ                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>* Phép ghép chuỗi:</b><br/> <b>Cú pháp:</b>            &lt;chuỗi 1&gt; + &lt;chuỗi 2&gt; + .... &lt;chuỗi n&gt;<br/> Chuỗi được tạo thành bằng cách ghép các chuỗi lại với nhau</p>                                                                                                                                                                                                                                                 | <pre>&gt;&gt;&gt; S1="Lập trình" &gt;&gt;&gt; S2="Python" &gt;&gt;&gt; print( S1 + S2) 'Lập trình Python'</pre>                                                  |
| <p><b>* Phép lặp chuỗi:</b><br/> <b>Cú pháp:</b>            &lt;chuỗi&gt;*&lt;số lần ghép&gt;<br/> Hoặc:                    &lt;số lần ghép&gt;*&lt;chuỗi&gt;</p>                                                                                                                                                                                                                                                                         | <pre>&gt;&gt;&gt; S="Python" &gt;&gt;&gt; print (S*3) 'PythonPythonPython'</pre>                                                                                 |
| <p><b>* So sánh chuỗi:</b><br/> Các phép so sánh chuỗi: bằng (==), khác (!=), lớn hơn (&gt;), bé hơn (&lt;), bé hơn hoặc bằng (&lt;=), lớn hơn hoặc bằng (&gt;=).<br/> <b>TH1:</b> Chuỗi A == B nếu chúng giống nhau hoàn toàn.<br/> <b>TH2:</b> Chuỗi A &gt;= B nếu ký tự đầu tiên khác nhau giữa chúng kể từ trái sang trong chuỗi A có điểm mã UNICODE lớn hơn chuỗi B.<br/> Để kiểm tra điểm mã, sử dụng hàm ord ( ) trong Python</p> | <pre>&gt;&gt;&gt; "TIN HOC" == "TIN HOC" True &gt;&gt;&gt; print (ord("b")) 98 &gt;&gt;&gt; print (ord("c")) → 99 &gt;&gt;&gt; print ("b" &lt; "c") → True</pre> |
| <p><b>* Toán tử in:</b><br/> <b>Cú pháp:</b>            &lt;chuỗi 1&gt; in &lt;chuỗi 2&gt;<br/> Kiểm tra chuỗi 1 nằm trong chuỗi 2. Nếu đúng trả giá trị True, nếu sai trả giá trị False</p>                                                                                                                                                                                                                                              | <pre>&gt;&gt;&gt; "abc" in "abc123" True &gt;&gt;&gt; "123" in "abc" False</pre>                                                                                 |
| <p><b>* Lệnh len():</b><br/> <b>Cú pháp:</b>            len(&lt;chuỗi&gt;)<br/> Cho biết độ dài của chuỗi</p>                                                                                                                                                                                                                                                                                                                             | <pre>&gt;&gt;&gt; a="Python" &gt;&gt;&gt; print (len(a)) 6</pre>                                                                                                 |
| <p><b>* Lệnh find():</b><br/> <b>Cú pháp đơn:</b>        &lt;chuỗi mẹ&gt;.find(&lt;chuỗi con&gt;)<br/> Tìm vị trí xuất hiện đầu tiên của chuỗi con trong chuỗi mẹ.<br/> Nếu không tìm thấy thì trả về -1<br/> <b>Cú pháp đầy đủ:</b> &lt;chuỗi mẹ&gt;.find(&lt;chuỗi con&gt;, start)<br/> Tìm vị trí chuỗi con bắt đầu từ vị trí start</p>                                                                                                | <pre>&gt;&gt;&gt; y = "Cái xác xinh xinh" &gt;&gt;&gt; x = "xinh" &gt;&gt;&gt; z = "bé" &gt;&gt;&gt; print (y.find(x)) 8 &gt;&gt;&gt; print (y.find(z)) -1</pre> |
| <p><b>* Lệnh split():</b><br/> <b>Cú pháp:</b>            &lt;chuỗi mẹ&gt;.split(&lt;ký tự tách&gt;)<br/> Tách một chuỗi thành các từ và đưa vào một danh sách. Ký tự tách có thể là dấu cách hoặc ký tự khác</p>                                                                                                                                                                                                                         | <pre>&gt;&gt;&gt; a="Lập trình Python" &gt;&gt;&gt; print(a.split()) ['Lập','trình','Python']</pre>                                                              |

| Phép toán và lệnh                                                                                                                                                                                                                                                                                                                      | Ví dụ                                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>* Lệnh join():</b></p> <p>Cú pháp: <code>"kí tự nối".join(&lt;danh sách&gt;)</code></p> <p>Nối các phần tử (là chuỗi) của một danh sách thành một chuỗi</p>                                                                                                                                                                      | <pre>&gt;&gt;&gt; a=["Lập","trình","Python"] &gt;&gt;&gt; print(" ".join(a)) 'Lập trình Python'</pre>                                                                                  |
| <p><b>* Lấy chuỗi con:</b></p> <p>Cú pháp: <code>&lt;chuỗi&gt;[m : n]</code></p> <p>Lấy chuỗi con gồm các kí tự liên tiếp từ chỉ số m đến chỉ số n-1 (m &lt; n)</p> <p><b>Đảo ngược thứ tự chuỗi:</b></p> <p>Cú pháp: <code>&lt;chuỗi&gt;[: : - 1]</code></p>                                                                          | <pre>&gt;&gt;&gt; a="Lập trình Python" &gt;&gt;&gt; a[0:9] 'Lập trình' &gt;&gt;&gt; a="123abc" &gt;&gt;&gt; a[::-1] 'cba321'</pre>                                                     |
| <p><b>* Hàm upper():</b></p> <p>Cú pháp: <code>&lt;chuỗi&gt;.upper( )</code></p> <p>Tạo chuỗi mới là chuỗi in hoa của chuỗi cũ</p>                                                                                                                                                                                                     | <pre>&gt;&gt;&gt;a="Lập trình Python" &gt;&gt;&gt;print(a.upper()) LẬP TRÌNH PYTHON</pre>                                                                                              |
| <p><b>* Hàm lower():</b></p> <p>Cú pháp: <code>&lt;chuỗi&gt;.lower()</code></p> <p>Tạo chuỗi mới là chuỗi in thường của chuỗi cũ</p>                                                                                                                                                                                                   | <pre>&gt;&gt;&gt;a="LẬP TRÌNH PYTHON" &gt;&gt;&gt;print(a.lower()) lập trình python</pre>                                                                                              |
| <p><b>* Hàm count():</b></p> <p>Cú pháp:</p> <p style="text-align: center;"><code>&lt; y &gt;.count(&lt; x &gt;) hoặc</code></p> <p style="text-align: center;"><code>&lt; y &gt;.count(&lt; x &gt;, &lt;chỉ số bắt đầu&gt;, &lt;chỉ số kết thúc&gt;)</code></p> <p>Đếm số lần xuất hiện không giao nhau của chuỗi x trong chuỗi y</p> | <pre>&gt;&gt;&gt;y = "aaa" &gt;&gt;&gt;x = "aa" &gt;&gt;&gt;print (y.count(x)) 1 &gt;&gt;&gt; a='LẬP TRÌNH PYTHON' &gt;&gt;&gt; b="P" &gt;&gt;&gt; print (a.count(b,0,len(a))) 2</pre> |
| <p><b>* Hàm replace():</b></p> <p>Cú pháp:</p> <p style="text-align: center;"><code>y.replace (x1, x2)</code></p> <p>Tạo chuỗi mới từ chuỗi y bằng cách thay thế chuỗi con x1 của y bằng chuỗi x2. Tất cả các chuỗi con bằng x1 và không giao nhau của y đều được thay bằng chuỗi x2</p>                                               | <pre>&gt;&gt;&gt;y = "Trúc xinh trúc mọc sân đình" &gt;&gt;&gt;x1 = "sân đình" &gt;&gt;&gt;x2 = "bờ ao" &gt;&gt;&gt;print(y.replace(x1,x2)) 'Trúc xinh trúc mọc bờ ao'</pre>           |

## B. CÂU HỎI TRẮC NGHIỆM

**Câu 1:** Lệnh `"abcde".find("")` trả lại giá trị gì?

- A. -1                                      B. 0                                      C. 1                                      D. Báo lỗi

**Câu 2:** Lệnh `"0123456789".find("012abc")` trả lại giá trị gì?

- A. -1                                      B. 0                                      C. 1                                      D. Báo lỗi

**Câu 3:** Lệnh `len(" Hà Nội Việt Nam ".split())` trả lại giá trị gì?

- A. 0                                      B. 4                                      C. 5                                      D. Báo lỗi

**Câu 4:** Lệnh `"Trường Sơn".find("Sơn",4)` trả lại giá trị gì?

- A. 5                                      B. 6                                      C. 7                                      D. 8

**Câu 5:** Lệnh `"Trường Sơn".find("Sơn",8)` trả lại giá trị gì?

- A. 5                                      B. 6                                      C. 7                                      D. -1

**Câu 6:** Để thay thế kí tự 'a' trong chuỗi s bằng một chuỗi mới rỗng ta dùng lệnh:

- A. `s = s.replace('a','')`                                      C. `s = replace(a,"")`  
B. `s = s.replace('a')`                                      D. `s = s.replace( )`

**Câu 7:** Trong NNLT Python, để tìm vị trí xuất hiện đầu tiên của chuỗi A trong chuỗi B ta viết:

- A. `<chuỗi A>.find(<chuỗi B>)`                                      B. `<chuỗi B>.find(<chuỗi A>)`  
C. `find (<chuỗi A>,<chuỗi B>)`                                      D. `find (<chuỗi B>,<chuỗi A>)`

**Câu 8:** Trong NNLT Python, hàm `y[m:n]` thực hiện việc gì trong các việc sau đây?

- A. Xác định chuỗi con của chuỗi n từ vị trí y đến trước vị trí m.  
B. Xác định chuỗi con của chuỗi y từ vị trí m đến trước vị trí n.  
C. Xác định chuỗi con của chuỗi y từ vị trí n đến trước vị trí m-1.  
D. Xác định chuỗi con của chuỗi y từ vị trí m đến trước vị trí n-1.

**Câu 9:** Cú pháp của lệnh `split()` là:

- A. `<chuỗi mẹ> . split(<kí tự tách>)`                                      B. `<chuỗi mẹ> . split(<chuỗi con>)`  
C. `<chuỗi mẹ> : split(<kí tự tách>)`                                      D. `<chuỗi mẹ> : split(<chuỗi con>)`

**Câu 10:** Cú pháp của lệnh `join()` là:

- A. `'kí tự nối' . join(<danh sách>)`                                      B. `'kí tự nối' : join(<danh sách>)`  
C. `'kí tự nối' . join(<chuỗi mẹ>)`                                      D. `'kí tự nối' . join(<chuỗi con>)`

### C. CÂU HỎI ĐÚNG/SAI

**Câu 1:** Trong Python, kiểu dữ liệu chuỗi ký tự (string) hỗ trợ nhiều phương thức để xử lý dữ liệu văn bản:

- a) Phương thức `lower()` dùng để chuyển toàn bộ ký tự trong chuỗi thành chữ thường.  
b) Phương thức `upper()` dùng để chuyển chuỗi thành chữ hoa.  
c) Phương thức `split()` dùng để gộp các chuỗi thành một chuỗi duy nhất.  
d) Phương thức `replace()` dùng để thay thế một chuỗi con bằng một chuỗi con khác.

**Câu 2:** Trong Python, có thể sử dụng nhiều phương thức để thao tác với chuỗi (chuỗi ký tự):

- a) Phương thức `find()` trả về vị trí đầu tiên xuất hiện của chuỗi con trong chuỗi ban đầu.  
b) Phương thức `len()` có thể dùng để đếm số ký tự trong một chuỗi.  
c) Toán tử `+` dùng để nối hai chuỗi với nhau.  
d) Phương thức `split()` dùng để gộp các chuỗi thành một chuỗi duy nhất.

**Câu 3:** Python cung cấp nhiều phương thức giúp xử lý chuỗi hiệu quả:

- a) Chuỗi trong Python là kiểu dữ liệu không thay đổi được (immutable).  
b) Cú pháp `"Hello"[1]` sẽ trả về ký tự e.  
c) Toán tử `*` dùng để nối hai chuỗi với nhau.  
d) Dấu ngoặc vuông `[]` có thể dùng để truy cập từng ký tự trong chuỗi.

**HẾT**