

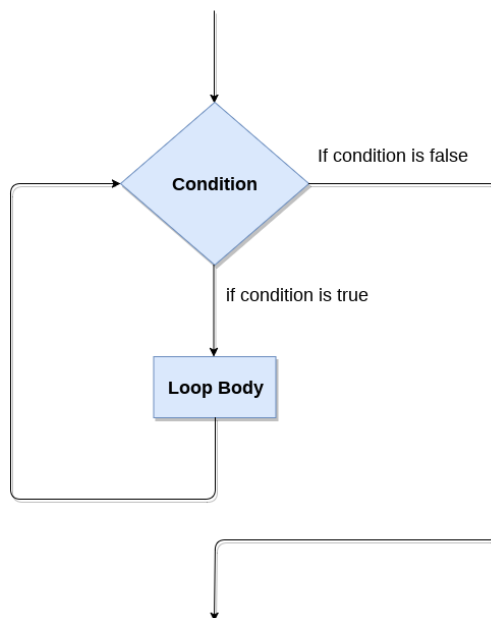


LAB 6 : VÒNG LẶP WHILE - FOR

A. HIỂU VỀ VÒNG LẶP

- **Vòng Lặp**

Vòng lặp cho phép bạn lặp lại việc thực thi một khối mã nguồn cho tới khi một điều kiện nhất định được thỏa mãn. Vòng lặp được sử dụng khá phổ biến trong lập trình. Không giống các ngôn ngữ lập trình khác với các vòng lặp khác nhau như for, while, do...while..., Python chỉ hỗ trợ vòng lặp for và vòng lặp while.



- **WHILE :**

Vòng lặp được sử dụng trong lập trình để lặp lại một đoạn code cụ thể, while cũng là một trong số đó. Trong Python, while được dùng để lặp lại một khối lệnh, đoạn code khi **điều kiện kiểm tra là đúng**. while dùng trong những trường hợp mà chúng ta **không thể dự đoán trước được số lần cần lặp là bao nhiêu**.

- **FOR**

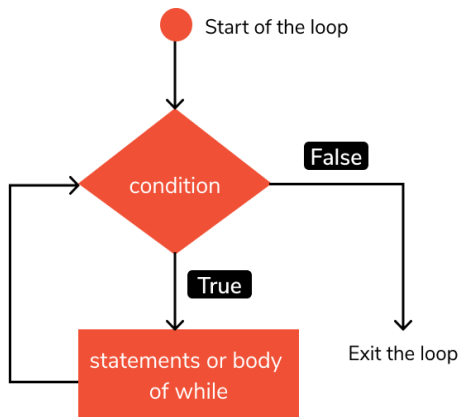
Vòng lặp for được sử dụng để **duyệt qua các phần tử trong một tập hợp**. Nó thường được sử dụng khi bạn có một khối mã nguồn cần được thực thi “n” lần.

B. CẤU TRÚC VÒNG LẶP

While : Vòng lặp while trong python bao gồm từ khóa while, một biểu thức điều kiện và một khối gồm các câu lệnh được mô tả trong khối đó. Các lệnh mô tả trong khối chỉ được xử lý khi biểu thức điều kiện được chỉ định là True (đúng). Chúng ta sử dụng cú pháp câu lệnh while trong python như sau:



while biểu thức điều kiện:
câu lệnh trong khối **while**



```
num = 0
total = 0

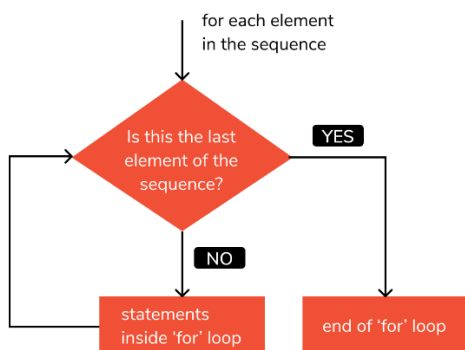
while num < 4:
    print(num)
    num += 1
    total+=num
print("Tổng =",total)
#> 0
#> 1
#> 2
#> 3
#> Tổng = 10
```

```
num = 0
total = 0

while num < 4:
    if num == 2:
        print('!!BREAK!!')
        break
    print(num)
    total+=num
    num += 1
print("Tổng =",total)
```

For : Vòng lặp for trong python bao gồm từ khóa for, một đối tượng có nhiều phần tử, một biến dùng để gán phần tử và một khối gồm các câu lệnh được mô tả trong khối đó. Đối tượng có nhiều phần tử là các iterable trong python như list, range, dictionary v.v..

for biến in đối tượng chứa các phần tử:
câu lệnh trong khối **for**



```
member_list = ['Hoa', 'Thảo', 'Mai']

for name in member_list:
    print(name)

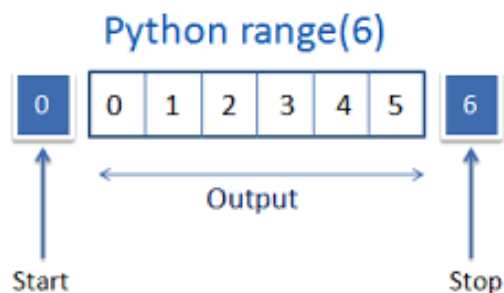
print("\nEnd")
#> Hoa
#> Thảo
#> Mai
#> End
```

```
member_list = ['Hoa', 'Thảo', 'Mai']
member_count = 0

for name in member_list:
    if name == 'Thảo':
        continue
    print(name)
    member_count += 1

print("Số thành viên :",member_count)
#> Hoa
#> Mai
#> Số thành viên : 2
```

Hàm 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. Hiểu đơn giản, hàm nhận một số nguyên và trả về một range object (kiểu iterable).



```

1 # Xuất ra giá trị i chạy từ 1 đến 5
2 # Vì theo luật 6 - 1 = 5
3 for i in range(1,6,1):
4     print(i,end='\t')
5
6 # Cách xuất giá trị i khác
7 print('')
8 for i in range(6):
9     print(i,end='\t')

```

1	2	3	4	5	
0	1	2	3	4	5

[Finished in 36ms]

`range(start=0, stop, step_size=1)`

- **start**: giá trị bắt đầu sinh.
- **stop**: giá trị kết thúc sinh, có nghĩa là sinh đến giá trị bé hơn stop. Không có mặc định.
- **step_size**: bước nhảy để sinh các giá trị tiếp theo từ start (nhận cả giá trị âm và dương).

C. VÍ DỤ

1. Ví dụ hàm for và range :

- In ra các giá trị nguyên chẵn có một chữ số

```

# i sẽ chạy từ 0 đến 10 với bước nhảy là 2
for i in range(0, 10, 2):
    print(i, end=" ") # xuất ra màn hình kết quả của i
# Kết quả xuất ra 0 2 4 6 8

```

- Nhập vào 2 số nguyên a, b. In ra các số nguyên nằm giữa a và b trên cùng 1 dòng.

```

a, b = int(input("Nhập a = ")), int(input("Nhập b = "))
if a > b: # kiểm tra điều kiện nếu a > b thì đổi vị trí
    a, b = b, a # đổi vị trí để i chạy từ số nhỏ đến số to
for i in range(a, b + 1): # b+1 để chạy hết dãy số
    print(i, end=" ") # xuất ra màn hình dãy số i

```

- Nhập vào số tự nhiên n, rồi tính giai thừa của n

```

n = int(input("Nhập n = ")) # nhập n bất kỳ
factorial = 1 # gán giá trị giai thừa = 1 khởi điểm
for i in range(1, n + 1): # i chạy lần lượt 1 đến n + 1
    factorial *= i # thực hiện tính giai thừa qua mỗi vòng lặp for
print('Giai thừa của "{}"! = {}".format(n, factorial))
# in giá trị giai thừa factorial với định dạng "{}! = {}" <=> n và factorial

```

2. Ví dụ While

- Nhập vào một số tự nhiên n rồi tính tổng các số tự nhiên đến n



```
n = int(input("Nhập n = ")) # nhập giá trị n bất kỳ
i = 0 # biến chạy
sum_n = 0 # gán giá trị tổng khởi tạo = 0
while i <= n: # nếu i nhỏ hơn n thì thỏa điều kiện
    sum_n = sum_n + i # tính tổng các số
    i += 1 # tăng biến đếm lên 1 qua mỗi vòng lặp while
print("Tổng các số tự nhiên đến {} là: {}".format(n, sum_n))
```

D. BÀI TẬP

Bài 1: Viết chương trình thực hiện các công việc sau :

Nhập 1 số nguyên dương n bất kỳ ($n < 1000$).

Yêu cầu kiểm tra dữ liệu đầu vào, nếu sai yêu cầu nhập lại.

Tính tổng các chữ số của số đó.

Hiển thị kết quả ra màn hình

Bài 2: Viết chương trình thực hiện các công việc sau:

Nhập 3 số a, b, c bất kỳ

Hãy kiểm tra ba số đó có phải là độ dài của các cạnh của một tam giác hay không?

Nếu đúng là tam giác thì xác định là tam giác vuông hay không?

Bài 3: Tính tổng từ 1 cho đến N

Gợi ý:

Khai báo một biến sum = 0.

Dùng vòng lặp cho chạy từ i đến N rồi cộng dồn giá trị của i vào biến sum

Xuất kết quả biến sum ra màn hình

Bài 4: Tính tổng các số CHẴN nằm trong đoạn từ 0 đến N

Gợi ý:

Khai báo một biến tongChan = 0.

Dùng vòng lặp cho chạy từ i đến N

Kiểm tra điều kiện xem i có thỏa điều kiện chia hết cho 2 hay không, nếu thỏa điều kiện thì cộng dồn giá trị của i vào biến tongChan.

Xuất kết quả biến tongChan ra màn hình



Bài 5: Tính tổng các số Lẻ nằm trong đoạn từ 0 đến N

Gợi ý:

Khai báo một biến tong_le = 0.

Dùng vòng lặp cho chạy từ i đến N

Kiểm tra điều kiện xem i có thỏa điều kiện không chia hết cho 2 hay không, nếu thỏa điều kiện thì cộng dồn giá trị của i vào biến tong_le.

Xuất kết quả biến tong_le ra màn hình

Bài 6: Tính trung bình cộng các số trong mảng

Gợi ý:

Khai báo một biến tong = 0.

Dùng vòng lặp cho chạy từ i đến N

Cộng dồn giá trị của i vào biến tong

Xuất kết quả trung bình các số là: tong/20 ra màn hình

Bài 7: Tính tổng giá trị từ 1 đến N, nếu chạy đến số 13 thì không chạy nữa và xuất kết quả

Gợi ý:

Khai báo một biến ketQua = 0.

Dùng vòng lặp cho chạy từ i đến N

Kiểm tra điều kiện xem i có bằng 13 hay không, nếu i chưa bằng thì cộng dồn giá trị của i vào biến ketQua. Ngược lại, i = 13 thì dùng lệnh break để thoát khỏi vòng lặp

Xuất kết quả biến ketQua ra màn hình

Bài 8: Guess the number

In this workbook, the user has to keep guessing a number until they get it right.

▪ **Task 1**

Store a "secret" number between 1 and 5. Then, prompt the user to enter a guess.

▪ **Task 2**

Set up a loop that keeps the user guessing until they get it right.

In other words, the loop should keep running while the guess is different from the secret.

▪ **Task 3**

After they guess the number, print:



You got it!

Bài 9: Nhập một số nguyên dương n . Tính:

a. $S = 1 + 2 + \dots + n$

b. $S = 1^2 + 2^2 + \dots + n^2$

c. $S = 1 + 1/2 + \dots + 1/n$

d. $S = 1 * 2 * \dots * n = n!$

e. $S = 1! + 2! + \dots + n!$