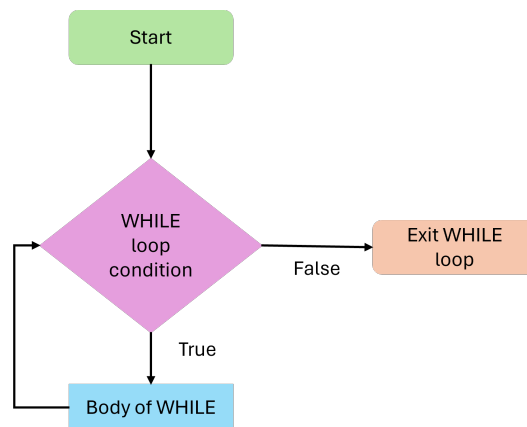


# BASIC PYTHON - WHILE LOOP

*Dinh-Tiem Nguyen và Quang-Vinh Dinh*

## 1 Giới thiệu

Trong hầu hết các ngôn ngữ lập trình, vòng lặp là một cấu trúc cho phép ta thực thi một khối code nhiều lần. Trong Python, có hai loại vòng lặp chính là for và while. Bài viết này sẽ tập trung vào vòng lặp while, giải thích cách hoạt động qua các ví dụ cụ thể để có thể dễ dàng hiểu và áp dụng.



### 1.1 Vòng lặp while là gì?

Vòng lặp while thực thi một khối mã liên tục miễn là điều kiện được chỉ định là đúng. Cú pháp của vòng lặp while như sau:

```
1 while điều_kiện:
2     # Khối code trong while
```

Vòng lặp while bắt đầu bằng việc kiểm tra điều kiện. Nếu điều kiện là True, khối code bên trong vòng lặp sẽ được thực thi. Sau khi khối code kết thúc, điều kiện sẽ được kiểm tra lại. Quá trình này tiếp tục cho đến khi điều kiện trở thành False.

Ví dụ chương trình in ra màn hình các số từ 1 đến 5:

```
1 i = 1
2 while i <= 5:
3     print(i)
4     i += 1
5
6
7 #ai vietnam
```

```
===== Output =====
1
2
3
4
5
=====
```

Một ví dụ khác để hiểu vòng lặp while, hãy tưởng tượng bạn đang chơi một trò chơi bật nhảy với một quả bóng. Bạn sẽ tiếp tục nhảy và đập bóng xuống đất cho đến khi bạn mệt và không thể nhảy nữa. Ở đây, vòng lặp while hoạt động tương tự như việc bạn bật nhảy. Bạn sẽ tiếp tục thực hiện một hành động (nhảy và đập bóng) cho đến khi một điều kiện nào đó không còn đúng nữa (bạn cảm thấy mệt và không thể nhảy tiếp).

```

1 has_energy = True
2 jump_count = 0
3
4 while has_energy:
5     jump_count += 1
6     print(f"Jump {jump_count} time(s)")
7     # Giả sử sau 5 lần nhảy, bạn mệt và dừng lại
8     if jump_count == 5:
9         has_energy = False

```

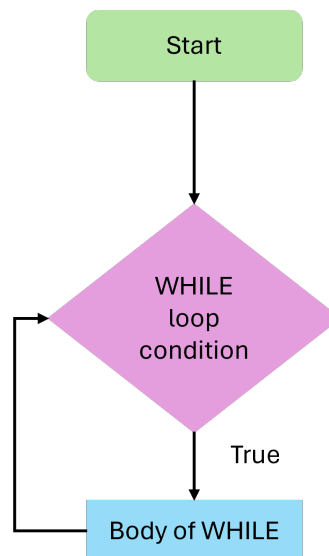
```

===== Output =====
Jump 1 time(s)
Jump 2 time(s)
Jump 3 time(s)
Jump 4 time(s)
Jump 5 time(s)
=====

```

has\_energy là biến điều kiện. Mỗi lần vòng lặp chạy, bạn sẽ nhảy một lần và tăng số lần nhảy jump\_count lên. Khi số lần nhảy đạt đến 5, bạn cảm thấy mệt và đặt has\_energy thành False, vòng lặp kết thúc.

## 2 Vòng lặp vô hạn



Một vòng lặp vô hạn là một vòng lặp tiếp tục thực thi câu lệnh trong vòng lặp đến mãi mãi, vì điều kiện của vòng lặp luôn là True. Mặc dù chúng thường là điều cần tránh, nhưng cũng có những trường hợp chúng ta có thể tạo ra một vòng lặp vô hạn một cách cố ý, chẳng hạn khi xây dựng các chương trình liên tục lắng nghe đầu vào hoặc chạy các tiến trình máy chủ.

Ví dụ Vòng lặp vô hạn:

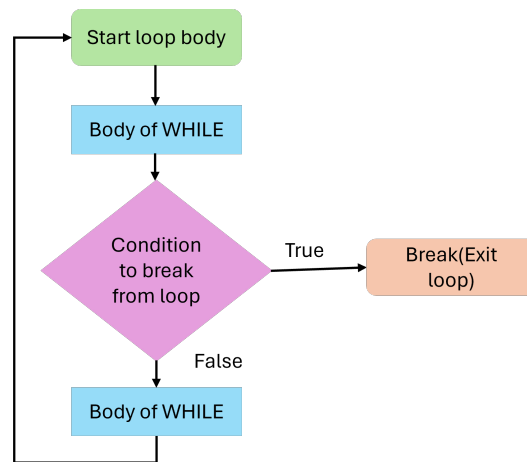
```

1 while True:
2     print("This loop will run forever!")

```

Trong ví dụ trên, điều kiện của vòng lặp while được đặt là True, có nghĩa là vòng lặp sẽ chạy mãi mãi vì điều kiện luôn đúng. Bên trong vòng lặp, câu lệnh print("This loop will run forever!") được thực thi lặp đi lặp lại. Để thoát khỏi vòng lặp, chúng ta cần phải dừng chương trình bằng cách kết thúc quá trình thực thi của nó. Trong hầu hết các môi trường phát triển, chúng ta có thể làm điều này bằng cách nhấn Ctrl + C nếu chúng ta chạy chương trình bằng dòng lệnh hoặc nhấn nút stop trên công cụ lập trình.

### 3 Sử dụng break trong while



Chúng ta có thể sử dụng lệnh break để thoát khỏi vòng lặp ngay lập tức, bất kể điều kiện của vòng lặp là gì. Sử dụng break khi chúng ta muốn dừng vòng lặp dựa trên một điều kiện khác xảy ra trong quá trình thực thi.

Ví dụ sử dụng break để thoát khỏi vòng lặp:

```

1 #ai vietnam
2 i = 1
3 while i <= 10:
4     print(i)
5     if i == 5:
6         break
7     i += 1
  
```

```

===== Output =====
1
2
3
4
5
=====
  
```

Trong ví dụ này vòng lặp while sẽ chạy miễn là  $i$  nhỏ hơn hoặc bằng 10. Bên trong vòng lặp, giá trị của  $i$  được in ra và sau đó được tăng lên một đơn vị. Khi  $i$  đạt giá trị 5, lệnh `if i == 5: break` sẽ được thực thi, làm cho vòng lặp dừng lại ngay lập tức. Các giá trị từ 1 đến 5 sẽ được in ra, sau đó vòng lặp kết thúc khi  $i$  bằng 5.

### 4 Sử dụng continue trong while

Chúng ta có thể sử dụng lệnh continue để bỏ qua các lệnh còn lại của khối code trong vòng lặp và bắt đầu lần lặp tiếp theo. Continue thường được sử dụng khi chúng ta muốn bỏ qua một số bước trong vòng lặp dựa trên một điều kiện nào đó.

Ví dụ sử dụng continue để bỏ qua lần lặp của những trường hợp  $i$  chia hết cho 2:

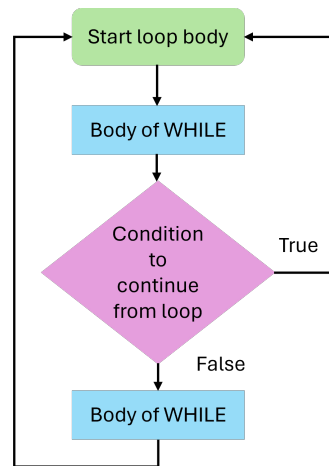
```

1 #ai vietnam
2 i = 0
3 while i < 10:
4     i += 1
5     if i % 2 == 0:
6         continue
7     print(i)
  
```

```

===== Output =====
1
3
5
7
9
=====
  
```

Trong ví dụ này vòng lặp while sẽ chạy miễn là  $i$  nhỏ hơn 10. Mỗi lần vòng lặp chạy, giá trị của  $i$  được tăng lên một đơn vị. Nếu  $i$  là số chẵn ( $i \% 2 == 0$ ), lệnh continue sẽ được thực thi, bỏ qua câu lệnh `print(i)` và bắt đầu lần lặp tiếp theo. Kết quả chỉ gồm các số lẻ từ 1 đến 9 được hiển thị ra màn hình.



## 5 Sử dụng while với list

Chúng ta cũng có thể sử dụng vòng lặp while để duyệt qua các phần tử của một danh sách. Cách này thường dùng khi chúng ta cần thực hiện các thao tác lặp lại trên các phần tử của danh sách mà không sử dụng vòng lặp for.

Ví dụ sử dụng while để duyệt qua danh sách

```

1 #ai vietnam
2 fruits = ["apple", "banana", "cherry"]
3 i = 0
4 while i < len(fruits):
5     print(fruits[i])
6     i += 1
  
```

```

===== Output =====
apple
banana
cherry
=====
  
```

Trong ví dụ trên, chúng ta có một danh sách `fruits` chứa các phần tử là tên các loại trái cây. Vòng lặp while sẽ chạy khi chỉ số  $i$  nhỏ hơn độ dài của danh sách `fruits`. Trong mỗi lần lặp, phần tử tại vị trí  $i$  trong danh sách `fruits` sẽ được in ra. Chỉ số  $i$  sau đó được tăng lên một đơn vị, giúp duyệt qua các phần tử tiếp theo của danh sách. Khi  $i$  bằng độ dài của danh sách, vòng lặp sẽ kết thúc. Ngoài ra các bạn có thể thử với dictionary, tuple...

## 6 Kết Luận

Vòng lặp while là một công cụ mạnh mẽ trong Python, cho phép ta thực thi một khối code nhiều lần dựa trên điều kiện nào đó. Tuy nhiên, chúng ta cần cẩn thận để tránh vòng lặp vô hạn và sử dụng các lệnh `break` và `continue` một cách hợp lý để kiểm soát luồng của vòng lặp.

Hy vọng rằng qua bài viết này, các bạn đã hiểu rõ hơn về vòng lặp while và cách sử dụng nó trong Python. Hãy thử áp dụng những gì các bạn đã học vào các bài tập và project trên lớp để nắm vững kiến thức hơn!