# VÒNG LẶP

## Câu lệnh if else trong python

Trong bài này chúng ta sẽ học cách sử dụng câu **lệnh if else trong Python**, đây là lệnh giúp rẻ nhánh chương trình và được sử dụng rất nhiều trong lúc lập trình với Python.

Nói đến lệnh if else thì mình xin nhắc lại một chút kiến thức về toán học, đó là **mệnh đề**.

Như ta biết thì mệnh đề thường sẽ trả lời các câu hỏi như "nếu đúng ... thì làm ... ", nghĩa là nếu thỏa mãn điều kiện thì làm công việc gì đó. Nếu như bạn đã học qua một số ngôn ngữ khác như [PHP](https://freetuts.net/hoc-php), [C++](https://freetuts.net/hoc-c/hoc-c++-can-ban-va-nang-cao), [C#](https://freetuts.net/hoc-csharp) thì chúng ta có cú pháp là:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | if (expression){      //statement  }  else {      // statement  } |

Nhưng với Python thì cú pháp của lệnh if else hoàn toàn khác, và khác như thế nào thì chúng ta bắt đầu tìm hiểu nhé.

**Mục lục**

* [1. Cú pháp lệnh if else trong Python](https://freetuts.net/if-else-trong-python-197.html#goto-h2-0)
  + [Cú pháp if else](https://freetuts.net/if-else-trong-python-197.html#goto-h3-0)
  + [Cú pháp if else lồng nhau](https://freetuts.net/if-else-trong-python-197.html#goto-h3-1)
  + [Cú pháp elif](https://freetuts.net/if-else-trong-python-197.html#goto-h3-2)
* [2. Cách xác định dòng kết thúc của lệnh if else trong Python](https://freetuts.net/if-else-trong-python-197.html#goto-h2-1)
* [3. Một vài bài tập thực hành lệnh if else trong Python](https://freetuts.net/if-else-trong-python-197.html#goto-h2-2)
* [4. Kết luận](https://freetuts.net/if-else-trong-python-197.html#goto-h2-3)

### Cú pháp lệnh if else trong Python

Vì trong Python không dùng dấu ngoặc nhọn {} để tạo ra các block nên cú pháp nó hơi khác biệt một chút.

Ta dùng dấu hai chấm : để ngăn cách lệnh if, các đoạn code phía dưới sẽ được thụt vào một tab và nó tạo thành một khối.

#### Cú pháp if else

**Cú pháp**

|  |  |
| --- | --- |
| 1  2  3  4 | if expression:      #statement  else :      #statement |

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5 | bien= 100;  if bien == 100:      print("Gia tri cua bien = 100");  else :      print("Gia tri cua bien khac 100"); |

#### Cú pháp if else lồng nhau

**Cú pháp**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | if expression1:      if expression2:          #statment      #statment  else :      #statment |

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | bien= 100;  #Cau if thu nhat  if bien == 100:      ten = "thehalfheart";        # Cau if thu hai      if ten == "thehalfheart":          print("Gia tri cua ten la thehalfehart");      print("Gia tri cua bien = 100");  else :      print("Gia tri cua bien khac 100"); |

#### Cú pháp elif

Nghe **elif** có vẻ hơi khác thường nhỉcool, thông thường trong các ngôn ngữ PHP, C, C++ thì **else if**chứ ko giống Python (**elif**)

**Cú pháp**

|  |  |
| --- | --- |
| 1  2  3  4 | if expression1:      #statement  elif expression2 :      #statement |

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5 | bien = 101;  if bien == 100:      print("Gia tri bien = 101");  elif bien == 101 :      print("Gia tri cua bien khac 101"); |

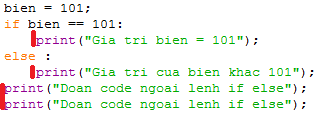
Như vậy trong Python ta không nhất thiết phải dùng cặp dấu ngoặc () để bao quanh điều kiện trong lệnh if, và ta cũng không dùng cặp dấu {} để bao quanh khối lệnh, mà thay vào đó ta dùng dấu hai chấm (:) ở đầu để khai báo bắt đầu đoạn code và kết thúc không có gì cả.

**\* Lưu ý**: Trong Python các đoạn code được hiểu là một khối nếu nó cùng một tab (thẳng hàng từ trên xuống) cho tới khi gặp tab ngắn hơn, nên nếu bạn tự ý tab vào thì trình biên dịch sẽ báo lỗi ngay.

### Cách xác định dòng kết thúc của lệnh if else trong Python

Trong Python, để báo cho trình biên dịch biết đoạn code nằm ngoài một khối lệnh nào đó thì ta sẽ tap thụt ngoài 1 ô, nghĩa là các lệnh được bao khối bởi cấu trúc cây (cây nằm sâu bên trong là con của cây ngoài).

Xem hình minh họa dưới đây:

  
Các đường màu đỏ mô phỏng các đoạn code cùng cấp trong python

**Ví dụ 1**: Thụt vào 1 tab nên bị hiểu nhầm là nằm trong khối else

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | bien = 101;  if bien == 101:      print("Gia tri bien = 101");  else :      print("Gia tri cua bien khac 101");      print("Doan code ngoai lenh if else"); |

Kết quả:

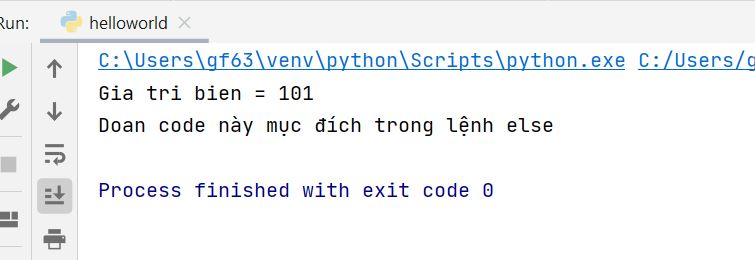


Mục đích mình muốn cái dòng cuối cùng là nằm ngoài lệnh else, nhưng vì mình cho nó cùng cấp với khối else nên trình biên dịch hiểu nhầm nó là code nằm trong lệnh else.

**Ví dụ 2**: Không thụt vào 1 tab nên trình biên dịch hiểu dòng cuối cùng không nằm trong khối if else.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | bien = 101;  if bien == 101:      print("Gia tri bien = 101");  else :      print("Gia tri cua bien khac 101");  print("Doan code này mục đích trong lệnh else"); |

Kết quả:



Tương tự, dòng code cuối cùng mục đích mình muốn nó nằm trong khối else, nhưng vì mình không thụt vào một tab nên trình biên dịch đã hiểu lầm nó nằm ngoài khối else nên được chạy.

### Một vài bài tập thực hành lệnh if else trong Python

Tiếp theo ta sẽ làm một vài bài tập nhé, thông qua những bài tập này bạn sẽ hiểu được cách sử dụng lệnh if else.

**Chương trình kiểm tra số chẵn hay lẻ trong Python**:

|  |  |
| --- | --- |
| 1  2  3  4  5 | number = 20;  if (number % 2 == 0):      print (str(number) + " là số chẵn")  else :      print (str(number) + " là số lẻ") |

**Chương trình tìm số lớn nhất trong 3 số bằng Python**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | number1 = 20  number2 = 300  number3 = 70    # Giả sử số number1 là lớn nhất  max = number1    # Kiểm tra xem số number2 có lớn hơn số max không  # Nếu có thì phải đổi số lớn nhất thành number2  if (max < number2) :      max = number2    # Tương tự, ta sẽ kiểm tra cho số thứ 3  if (max < number3) :      max = number3    print ("Số lớn nhất là " + str(max)) |

**Chương trình in ra các ngày trong tháng bằng Python**:

Hãy viết chương trình in ra các ngày trong tuần bằng lệnh if else.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | # Bạn hãy thay đổi số này để test nhé  dayofweek = 2    if dayofweek == 2 :      print ("Thứ hai")  elif dayofweek == 3 :      print ("Thứ ba")  elif dayofweek == 4 :      print("Thứ 4")  elif dayofweek == 5 :      print("Thứ 5")  elif dayofweek == 6 :      print("Thứ 6")  elif dayofweek == 7 :      print ("Thứ 7")  else :      print("Chủ nhật") |

### Kết luận

Tới bài này mình cảm thấy Python bắt đầu thú vị rồi đấy, các cú pháp code của nó không giống như C hay C++ mà theo một cách riêng biệt.

Nếu bạn mới học lập trình thì không sao chứ bạn nào mà học qua các ngôn ngữ kia rồi thì sẽ cảm thấy Python khó chịu vì khó quản lý code. Tuy nhiên bản thân mình thấy nó rất khắt khe trong việc code, bạn không thể tự mình thụt vào (tab) mà phải tuân theo quy luật của nó.

Qua bày này hy vọng bạn đã biết cách sử dụng lệnh if else trong Python. Hẹn gặp lại bạn ở bài tiếp theo nhé.

## Vòng lặp For trong Python

Trong bài này chúng ta sẽ học cách sử dụng vòng lặp for trong Python, đây là vòng lặp đơn giản và dễ hiểu nhất nên được học đầu tiên trong series học Python cơ bản này.

Nói về vòng lặp thì bạn phải một chút về khái niệm: Vòng lặp là những hành động được lặp đi lặp lại theo một chu kì cho tới khi điều kiện lặp bị sai. Python có hai vòng lặp chủ yếu đó là vòng lặp while và vòng lặp for, tuy nhiên trong bài này ta sẽ tìm hiểu vòng lặp for trước nhé.

**Mục lục**

* [1. Cú pháp vòng lặp for trong Python](https://freetuts.net/vong-lap-for-trong-python-1689.html#goto-h2-0)
* [2. Kết hợp hàm range trong vòng lặp for Python](https://freetuts.net/vong-lap-for-trong-python-1689.html#goto-h2-1)
  + [Trường hợp có một tham số](https://freetuts.net/vong-lap-for-trong-python-1689.html#goto-h3-0)
  + [Trường hợp có hai tham số](https://freetuts.net/vong-lap-for-trong-python-1689.html#goto-h3-1)
  + [Trường hợp có ba tham số](https://freetuts.net/vong-lap-for-trong-python-1689.html#goto-h3-2)
* [3. Ví dụ khác với vòng lặp for trong Python](https://freetuts.net/vong-lap-for-trong-python-1689.html#goto-h2-2)
* [4. Vòng lặp for lồng nhau trong Python](https://freetuts.net/vong-lap-for-trong-python-1689.html#goto-h2-3)
* [5. Sử dụng lệnh if else trong vòng lặp for](https://freetuts.net/vong-lap-for-trong-python-1689.html#goto-h2-4)
* [6. Lời kết](https://freetuts.net/vong-lap-for-trong-python-1689.html#goto-h2-5)

### Cú pháp vòng lặp for trong Python

Nếu vòng lặp while thì sẽ không biết trước được tổng số lần lặp thì vòng lặp for lại khác, chúng ta có thể biết được tổng số lần lặp dựa vào cú pháp của nó như sau:

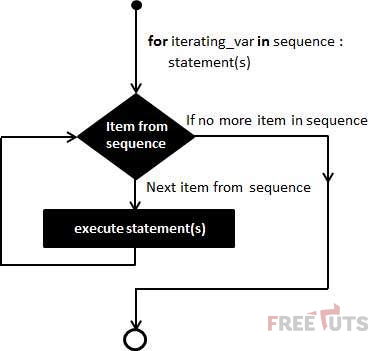
|  |  |
| --- | --- |
| 1  2 | for iterating\_var in sequence:     statements(s) |

Trong đó:

*Bài viết này được đăng tại [free tuts .net]*

* iterating\_var là biến dùng để lưu giá trị của các phần tử trong mỗi vòng lặp
* sequence la tập hợp các giá trị cần lặp
* statements(s) la các dòng lệnh xử lý trong thân vòng lặp

Hãy xem lược đồ của vòng lặp for dưới đây.



**Ví dụ**: Lặp và in ra từng ký tự của chữ freetuts.

|  |  |
| --- | --- |
| 1  2 | for letter in 'Freetuts':     print ('Ký tự hiện tại :', letter) |

Kết quả sẽ như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | Ký tự hiện tại : F  Ký tự hiện tại : r  Ký tự hiện tại : e  Ký tự hiện tại : e  Ký tự hiện tại : t  Ký tự hiện tại : u  Ký tự hiện tại : t  Ký tự hiện tại : s |

Như bạn thấy, vòng lặp này sẽ lặp qua từng phần tử và in chúng ra màn hình.

**Ví dụ**: Lặp qua từng phần tử của mảng

|  |  |
| --- | --- |
| 1  2  3 | fruits = ['banana', 'apple',  'mango']  for fruit in fruits:     print ('Fruit :', fruit) |

Kết quả như sau:

|  |  |
| --- | --- |
| 1  2  3 | Fruit : banana  Fruit : apple  Fruit : mango |

Trong Python thì mảng hay còn gọi là List, vì vậy bạn đừng hiểu lần giữa hai khá niệm này nhé, tuy hai nhưng lại là một.

**\* Lưu ý**: Phạm vi bên trong vòng lặp phải được tab vào nhé các bạn, nếu bạn để cùng cấp với vòng lặp thì Python sẽ hiểu lầm là code ở bên ngoài.

### Kết hợp hàm range trong vòng lặp for Python

Mình sẽ không nói rõ hàm range trong bài này mà chỉ giải thích sơ lược và ứng dụng của nó trong vòng lặp thôi nhé.

Hàm range sẽ trả về một mảng trong đó tổng số phần tử sẽ phụ thuộc vào các tham số truyền vào. Cú pháp như sau:

|  |  |
| --- | --- |
| 1 | range(start, end, step) |

Trong đó:

* start: là giá trị bắt đầu
* end là giá trị kết thúc
* step là khoảng cách giữa các phần tử, hay còn gọi la bước nhảy

#### Trường hợp có một tham số

Nếu bạn chỉ truyền một tham số n thì nó sẽ tạo một mảng từ 0 -> n - 1.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2 | for i in range(5):      print(i, end=', ') |

Kết quả sẽ tạo một mảng gồm 5 phần tử có giá trị lần lượt từ 0 -> 5.

|  |  |
| --- | --- |
| 1 | 0, 1, 2, 3, 4, |

#### Trường hợp có hai tham số

Nếu bạn truyền 2 tham số thì sẽ tạo một mảng với bước nhảy là 1, phần tử đầu của mảng là start, phần tử cuối cùng của mảng là end - 1.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2 | for i in range(5, 10):      print(i, end=', ') |

Kết quả sẽ tạo một mảng gồm 5 phần tử có giá trị lần lượt là 5 -> 9

|  |  |
| --- | --- |
| 1 | 5, 6, 7, 8, 9, |

#### Trường hợp có ba tham số

Trường hợp này sẽ tạo một mảng như trường hợp 2 nhưng vì bước nhảy là step nên tổng số phần tử sẽ nhỏ hơn.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2 | for i in range(1, 10, 2):      print(i, end=', ') |

Kết quả trả về một mảng 5 phần tử có giá lần lượt là 1, 3, 5, 7, 9 vì bước nhảy là 2.

|  |  |
| --- | --- |
| 1 | 1, 3, 5, 7, 9, |

### Ví dụ khác với vòng lặp for trong Python

Bây giờ ta sẽ thực hành với một vài ví dụ khác với vòng lặp for nhé.

**Ví dụ**: Kết hợp hàm len để in ra các phần tử của mảng dựa vào chỉ số index.

|  |  |
| --- | --- |
| 1  2  3 | fruits = ['banana', 'apple',  'mango']  for index in range(len(fruits)):     print 'Current fruit :', fruits[index] |

Kết quả:

|  |  |
| --- | --- |
| 1  2  3 | Current fruit : banana  Current fruit : apple  Current fruit : mango |

**Ví dụ**: Kết hợp với else xử lý lần lặp **bị từ chối** đầu tiên, tức là kết thúc vòng lặp thì chạy lệnh trong else.

|  |  |
| --- | --- |
| 1  2  3  4 | for i in range(5):      print(i, end=', ')  else:      print ('Gia tri cua i la', (i)) |

Kết quả:

|  |  |
| --- | --- |
| 1 | 0, 1, 2, 3, 4, Gia tri cua i la 4 |

### Vòng lặp for lồng nhau trong Python

Không khác gì [lệnh if else trong Python](https://freetuts.net/if-else-trong-python-197.html), bạn có thể sử dụng vòng lặp for để tạo ra những đoạn code lồng nhau, nghĩa là vòng lặp này sẽ nằm bên trong một vòng lặp khác.

**Ví dụ**: Viết chương trình in ra bảng cửu chương bằng vòng lặp for.

|  |  |
| --- | --- |
| 1  2  3 | for i in range(2, 10):      for j in range (2, 10):          print(i, " x ", j, ' = ', i \* j) |

Kết quả:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | 2  x  2  =  4  2  x  3  =  6  2  x  4  =  8  2  x  5  =  10  2  x  6  =  12  2  x  7  =  14  2  x  8  =  16  2  x  9  =  18  3  x  2  =  6  3  x  3  =  9  3  x  4  =  12  3  x  5  =  15  3  x  6  =  18  3  x  7  =  21  3  x  8  =  24  ... |

### Sử dụng lệnh if else trong vòng lặp for

Bạn có thể kết hợp sử dụng lệnh if else trong vòng lặp for để giải quyết bài toán, đây không phải là thao tác chỉ có ở mỗi Python mà hầu như ngôn ngữ nào cũng có.

**Ví dụ**: In ra các số chẵn và số lẻ từ 1 đến 100.

|  |  |
| --- | --- |
| 1  2  3  4  5 | for i in range(1, 100):      if i % 2 == 0:          print(i, ' là số chẵn')      else :          print (i, ' là số lẻ'); |

Kết quả:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | 1  là số lẻ  2  là số chẵn  3  là số lẻ  4  là số chẵn  5  là số lẻ  6  là số chẵn  7  là số lẻ  8  là số chẵn  9  là số lẻ  10  là số chẵn  ... |

### Lời kết

Như vậy là bạn đã học xong cú pháp và cách sử dụng cơ bản của vòng lặp for trong Python. Vì đây là kiến thức cơ bản nên mình không giải thích quá nhiều mà thay vào đó là những ví dụ trực quan giúp bạn dễ học hơn. Chào tạm biệt và hẹn gặp lại các bạn ở bài tiếp theo đó là bài tìm hiểu number trong Python.

## Vòng lặp While trong Python

Trong bài này chúng ta sẽ tìm hiểu về **vòng lặp while trong Python**, đây là vòng lặp được sử dụng khá nhiều khi bạn làm các ứng dụng trong thực tế, không chỉ riêng ở Python mà ở các ngôn ngữ khác đều vậy.

Khác với vòng lặp for, vòng lặp while sẽ lặp trong những trường hợp bạn không xác định được tổng số lần lặp là bao nhiêu, nó phụ thuộc vào điều kiện dừng vòng lặp nằm trong lệnh while.

**Mục lục**

* [1. Cú pháp vòng lặp While trong Python](https://freetuts.net/vong-lap-while-trong-python-685.html#goto-h2-0)
* [2. Ví dụ với vòng lặp wihle trong Python](https://freetuts.net/vong-lap-while-trong-python-685.html#goto-h2-1)
* [3. Cẩn thận vòng lặp While vô hạn](https://freetuts.net/vong-lap-while-trong-python-685.html#goto-h2-2)
* [4. Sử dụng else trong while](https://freetuts.net/vong-lap-while-trong-python-685.html#goto-h2-3)
* [5. Chuyển đổi từ vòng lặp for sang vòng lặp while](https://freetuts.net/vong-lap-while-trong-python-685.html#goto-h2-4)
* [6. Lời kết](https://freetuts.net/vong-lap-while-trong-python-685.html#goto-h2-5)

### Cú pháp vòng lặp While trong Python

Python là ngôn ngữ đơn giản nên cú pháp của nó cũng đơn giản. Sau đây là cú pháp chung của vòng lặp while.

|  |  |
| --- | --- |
| 1  2 | while expression:     statement(s) |

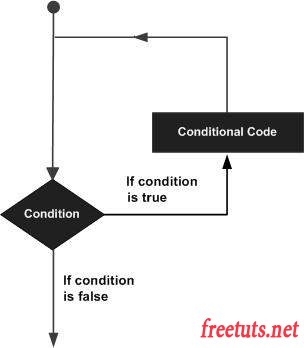
Trong đó:

* **statement(s)** là một lệnh đơn hoặc một tập lênh gồm nhiều lênh đơn dùng để xử lý chương trình.
* **expression**có thể là một biến hoặc một biểu thức, nhưng bắt buộc giá trị của nó phải là True hoặc False.

Như mình đã giới thiệu ở các bài trước, trong python sử dụng khoảng trắng để xác định các khối lệnh, điều này khác với các ngôn ngữ khác như PHP, C, C++ thì sử dụng cặp đóng mở {}. Nếu giá trị của expression là True thì những đoạn code trong khối lệnh của nó sẽ được thực thi, ngược lại nó sẽ bỏ qua và chạy các lệnh ở bên ngoài.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | while expression:     // Lệnh trong vòng lặp     // Lệnh trong vòng lặp     // ...    // Các lệnh ở ngoài vòng lặp |

Hãy xem sơ đồ sau để hiểu rõ hơn.



### Ví dụ với vòng lặp wihle trong Python

Hãy thực hành một ví dụ nhỏ dưới đây để hiểu hơn về vòng lặp này: In ra các số từ 0 đến 9.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | count = 0  while (count <= 9):      print ('Lượt đếm:', count)      count = count + 1    print ("Good bye!") |

Kết quả sẽ như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | Lượt đếm: 0  Lượt đếm: 1  Lượt đếm: 2  Lượt đếm: 3  Lượt đếm: 4  Lượt đếm: 5  Lượt đếm: 6  Lượt đếm: 7  Lượt đếm: 8  Lượt đếm: 9  Good bye! |

Như các bạn thấy, dòng Good bye! không bị lặp bởi vì nó nằm ngoài vòng lặp, còn lượt đếm sẽ bị lặp 10 lân biến count có giá trị ban đầu là 0 (count = 0), sau mỗi vòng lặp nó tăng lên 1 đơn vị (count = count + 1) và điều kiện dừng vòng lặp là count bé hơn hoặc bằng 9 (count <= 9).

### Cẩn thận vòng lặp While vô hạn

Vòng lặp while sẽ không biết chính xác tổng số lần lặp là bao nhiêu nên điều này rất nguy hiểm, vì nó có thể gây ra lặp vô hạn và dẫn đến ứng dụng bị chết, nên bạn phải tính toán thật kĩ điều kiện dừng vòng lặp.

Như ở ví dụ ở phần 2 thì điều kiện dừng là count <= 9, mà biến count có giá trị ban đầu là 0 nên sau mỗi vòng lặp mình khong tăng biến count lên 1 đơn vị thì sẽ dẫn đế lặp vô hạn.

Sau đây là ví dụ chương trình bị lặp vô hạn vì **expression**luôn trả về **True**.

**Bị lặp vô hạn**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | # Chương trình Python return true  # Lặp vô hạn nên không được chạy test nhé  count = 0  while (count <= 9):     print ('Lượt đếm:', count)    print ("Good bye!") |

Hoặc bạn có thể tham khảo ví dụ dưới đây, mình yêu cầu người dùng nhập dữ liệu vô hạn vì điều kiện var == 1 luôn luôn đúng.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var = 1  while var == 1 :      num = input("Nhập số của bạn  :")      print ("Bạn đã nhập số: ", num)    print ("Good bye!") |

Ví dụ này cũng bị lặp vô hạn bởi vì điều kiện dừng luôn luôn đúng. Nếu bạn chạy mà không thoát được thì hãy nhấn CTR + C để thoát ra khỏi chương trình nhé.

### Sử dụng else trong while

Có một sự mới mẻ đối với vòng lặp while trong Python đó là bạn có thể kết hợp thêm từ khóa else để xử lý cho lần lặp đầu tiên **không được thực hiện**khi điều kiện lặp false.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | count = 0  while count < 5:     print (count, " nhỏ hơn 5")     count = count + 1  else:     print (count, " lớn hơn 5") |

Chạy ví dụ này kết quả sẽ như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | 0  nhỏ hơn 5  1  nhỏ hơn 5  2  nhỏ hơn 5  3  nhỏ hơn 5  4  nhỏ hơn 5  5  lớn hơn 5 |

Các bạn thấy đó, lần lặp thứ 6 sẽ không xảy ra nên code trong lệnh else sẽ được thực hiện.

### Chuyển đổi từ vòng lặp for sang vòng lặp while

Trong Python, bạn có thể chuyển đổi chương trình sử dụng vòng lặp for sang sử dụng vòng lặp while cực kì đơn giản.

Mình sẽ lấy một ví dụ mà ta đã thực hành ở bài trước nhé, đó là in ra bảng cửu chương.

**Đối với vòng lặp for:**

|  |  |
| --- | --- |
| 1  2  3 | for i in range(2, 10):      for j in range (2, 10):          print(i, " x ", j, ' = ', i \* j) |

**Chuyển sang sử dụng vòng lặp while:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | i = 2;  j = 1;    while i <= 9:      while j <= 9:          print(i, " x ", j, ' = ', i \* j)          j += 1      i += 1      j = 1 # Thiết lập lại giá trị cho j cho bảng cửu chương tiếp theo |

Tuy hơi dài một chút nhưng chương trình chạy vẫn cho kết quả giôgns nhau.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | 2  x  1  =  2  2  x  2  =  4  2  x  3  =  6  2  x  4  =  8  2  x  5  =  10  2  x  6  =  12  2  x  7  =  14  2  x  8  =  16  2  x  9  =  18  3  x  1  =  3  3  x  2  =  6  3  x  3  =  9  3  x  4  =  12  3  x  5  =  15  3  x  6  =  18  3  x  7  =  21  ... |

### Lời kết

Như vậy là bạn đã hiểu được cách sử dụng vòng lặp while trong python rồi phải không nào, qua bài học này hy vọng sẽ giúp ích được cho bạn khi tiếp cận với ngôn ngữ Python.

Và sau đây mình xin rút ra kết luận như sau:

* Vòng lặp while được dùng trong trường hợp không biết trước tổng số lần lặp, còn vòng lặp for thì biết trước.
* Vòng lặp while có điều kiện lặp là trả về True hoặc False nên sử dụng trong những bài toán phức tạp, còn vòng lặp while dùng trong trường hợp đơn giản.

Hẹn gặp lại bạn ở bài tiếp theo đó là bài vòng lặp For.

## Lệnh break và continue trong Python

Trong bài viết này ta sẽ học cách sử dụng hai lệnh **break**và **continue**trong Python, công dụng của hai lệnh này trong Python là thay đổi luồng xử lý của vòng lặp.

Nếu bạn muốn kết thúc vòng lặp một cách đột ngột, hoặc bỏ qua một lần lặp nào đó thì có thể sử dụng hai lệnh này. Và cũng lưu ý là nó được sử dụng rất nhiều khi làm việc với Python, vì vậy bạn cần nắm vững nó nhé.

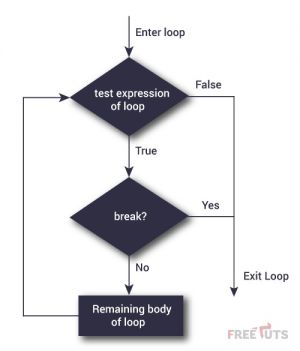
**Mục lục**

* [1. Lệnh break trong Python](https://freetuts.net/lenh-break-va-continue-trong-python-1703.html#goto-h2-0)
* [2. Lệnh continue trong Python](https://freetuts.net/lenh-break-va-continue-trong-python-1703.html#goto-h2-1)
* [3. Lời kết](https://freetuts.net/lenh-break-va-continue-trong-python-1703.html#goto-h2-2)

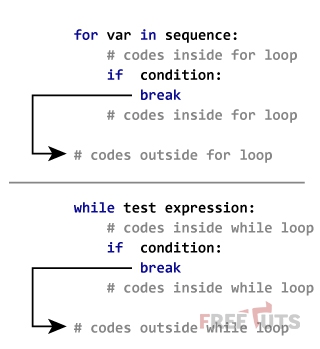
### Lệnh break trong Python

Lệnh break có công dụng là chấm dứt vòng lặp ngay lập tức, nó không quan tâm đến điều kiện dừng của vòng lặp như thế nào, miễn gặp lệnh này là vòng lặp sẽ kết thúc.

Trường hợp bạn sử dụng nhiều vòng lặp lồng nhau thì phạm vi tác dụng của break là vòng lặp gần nó nhất, tức là vòng lặp trong cùng. Hãy tham khảo lược đồ sau:



Và dưới đây là ví dụ sơ đồ hoạt động của nó ở trong hai vòng lặp for và while.



**Ví dụ**: Python break

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | # Use of break statement inside loop    for val in "string":      if val == "i":          break      print(val)    print("The end") |

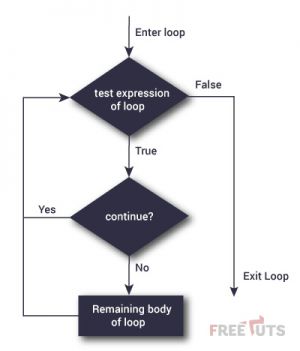
Kết quả sẽ như sau:

|  |  |
| --- | --- |
| 1  2  3  4 | s  t  r  The end |

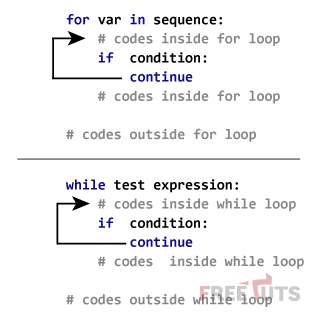
Chương trình này đã lặp qua lặp qua từng ký tự trong chuỗi "string", sau đó mỗi lần lặp sẽ kiểm tả đã đến kí tự "i" chưa, nếu đến rồi thì thoát khỏi vòng lặp, chưa thì tiếp tục lặp. Đấy là lý do tại sao nó không in đủ các ký tự trong chuỗi "string".

### Lệnh continue trong Python

Câu lệnh continue được sử dụng để bỏ qua phần còn lại của code bên trong thân vòng lặp và chỉ có tác dụng cho lần lặp hiện tại. Vòng lặp không kết thúc mà tiếp tục với lần lặp tiếp theo.



Lược đồ hoạt động ở trong hai vòng lặp while và for.



**Ví dụ**: Python continue

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | # Program to show the use of continue statement inside loops    for val in "string":      if val == "i":          continue      print(val)    print("The end") |

Chạy lên kết quả sẽ như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | s  t  r  n  g  The end |

Giải thích tương tự như phần continue, chỉ có điểm khác là lần lặp cho ký tự "i" sẽ không chạy đoạn code print(val) ở phía dưới mà nhảy đến lần lặp tiếp theo nên sẽ không in ra giá trị này.

### Lời kết

Trên là cách sử dụng lệnh break và continue trong Python, đây là hai lệnh sử dụng khá nhiều nên rất quan trọng. Với lệnh break thì nó có nhiệm vụ là dừng mọi hoạt động của vòng lặp, thoát một cách đột ngột. Còn với lệnh continue thì nó sẽ bỏ qua lần lập hiện tại và nhảy đến lần lặp tiếp theo.

## Bài tập Python về lưu đồ thuật toán cơ bản

Trong bài này mình sẽ giải một số bài tập Python cơ bản về phần lưu đồ thuật toán, đây là những bài thực hành giúp bạn nắm vững cấu trúc lệnh cơ bản của Python.

Danh sách các bài tập này được tổng hợp từ internet và mình sẽ sử dụng ngôn ngữ Python để giải nó. Mỗi bài tập mình sẽ đăng trên một bài riêng biệt nên bạn có thể click vào từng link và xem nhé.

**Danh sách bài tập Python cơ bản**

* [Kiểm tra dữ liệu là number hay string bằng Python](https://freetuts.net/ref/kiem-tra-du-lieu-la-number-hay-string-bang-python-394.html)
* [Tính tổng S(n) = 1 + 2 + 3 + … + n bằng Python](https://freetuts.net/ref/bai-tap-python-tinh-tong-sn-1-+-2-+-3-+-+-n-393.html)
* [Tính tổng S(n) = 1^2 + 2^2 + … + n^2 bằng Python](https://freetuts.net/ref/tinh-tong-sn-12-+-22-+-+-n2-bang-python-395.html)
* [Tính tổng S(n) = 1 + ½ + 1/3 + … + 1/n bằng Python](https://freetuts.net/ref/tinh-s-n-1-+-+-1-3-+-+-1-n-bang-python-396.html)
* [Tính tổng S(n) = ½ + ¼ + … + 1/2n bằng Python](https://freetuts.net/ref/tinh-s-n-+-+-12-n-bang-python-397.html)
* [Tính tổng S(n) = 1/3 + 1/5 + … + 1/(2n + 1)](https://freetuts.net/ref/tinh-tong-sn-1-+-1-3-+-15-+-+-1-2n-+-1-398.html)
* [Liệt kê tất cả các ước số của số nguyên dương n bằng Python](https://freetuts.net/ref/liet-ke-uoc-so-cua-so-nguyen-duong-n-bang-python-399.html)
* [Tính tổng tất cả các ước số của số nguyên dương N bằng Python](https://freetuts.net/ref/tinh-tong-tat-ca-cac-uoc-so-bang-python-400.html)
* [Tìm ước số lẻ lớn nhất của một số bằng Python](https://freetuts.net/ref/tim-uoc-so-le-lon-nhat-bang-python-401.html)
* [Kiểm tra một số có phải là số hoàn hảo bằng Python](https://freetuts.net/ref/kiem-tra-so-hoan-hao-bang-python-402.html)
* [Kiểm tra số chính phương bằng Python](https://freetuts.net/ref/kiem-tra-so-chinh-phuong-bang-python-403.html)
* [Kiểm tra số nguyên tố bằng Python](https://freetuts.net/ref/kiem-tra-so-nguyen-to-bang-python-404.html)
* [Đảo ngược một số bằng Python](https://freetuts.net/ref/dao-nguoc-mot-so-bang-python-406.html)
* [In ra từng ký tự của một số bằng Python](https://freetuts.net/ref/in-ra-tung-ky-tu-cua-mot-so-bang-python-405.html)
* [Giải phương trình bậc nhất một ẩn bằng Python (ax + b = 0)](https://freetuts.net/ref/giai-phuong-trinh-bac-nhat-mot-an-bang-python-407.html)
* [Giải phương trình bậc hai một ẩn bằng Python (ax^2 + bx + c = 0)](https://freetuts.net/ref/giai-phuong-trinh-bac-hai-mot-an-bang-python-408.html)