

Câu lệnh Rẽ nhánh If

Nội dung trình bày

- ❑ Kiểm tra có điều kiện
- ❑ Kiểm tra đẳng thức
- ❑ Ví dụ câu lệnh If

Kiểm tra có điều kiện

Ở trung tâm của mọi câu lệnh if là biểu thức mà có thể được đánh giá là True hoặc False và nó được gọi là kiểm tra có điều kiện.

Python sử dụng các giá trị True và False để quyết định xem mã trong câu lệnh if có được thực thi hay không.

Nếu một kiểm tra có điều kiện đánh giá là True, Python thực thi mã sau câu lệnh if. Nếu kiểm tra đánh giá là False, Python sẽ bỏ qua mã sau câu lệnh if.

Kiểm tra có điều kiện

Cú pháp câu lệnh if

- Cú pháp:

```
if condition:  
    # one or more statements
```

Trong đó:

- `condition`: là biểu thức trả về giá trị kiểu boolean
- `statements`: Các câu lệnh sẽ được thực thi nếu điều kiện trả về **true**

Kiểm tra có điều kiện

- **Ví dụ:**

```
i = 10
if (i > 15) :
    print ("10 is less than 15")
print ("I am Not in if")
```

- **Kết quả:** I am Not in if

Điều kiện trong câu lệnh if là False nên khối lệnh bên dưới câu lệnh if không được thực thi

Kiểm tra có điều kiện

- Đoạn mã sau xác định một số là số chẵn hay là số lẻ:

```
if (number % 2 == 0) :  
    print (number + " is even.")  
else:  
    print (number + " is odd.")
```

Kiểm tra đẳng thức

Hầu hết các kiểm tra có điều kiện đều so sánh giá trị hiện tại của một biến với một giá trị cụ thể.

Câu kiểm tra có điều kiện đơn giản nhất là kiểm tra giá trị của một biến có bằng một giá trị cụ thể nào không.

```
>>> car = 'bmw'  
>>> car == 'bmw'
```

True

```
>>> car = 'audi'  
>>> car == 'bmw'
```

False

Trong ví dụ trên, dấu bằng '=' chính là một câu, thực hiện lệnh gán. Trong khi đó, dấu bằng kép '==' lại là khiến nó trở thành câu hỏi: “Liệu giá trị của biến car có bằng 'bmw' hay không?”.

Bỏ qua chữ viết hoa khi kiểm tra đẳng thức

Kiểm tra đẳng thức là trường hợp phân biệt chữ hoa và chữ thường trong Python. Ví dụ, hai giá trị với chữ viết hoa khác nhau không được coi là bằng nhau

Nếu chỉ muốn kiểm tra giá trị của một biến, ta có thể chuyển đổi giá trị của biến thành chữ thường trước khi thực hiện so sánh:

Hàm `lower()` không thay đổi giá trị ban đầu được lưu trong biến `car`, do đó chúng ta có thể thực hiện loại so sánh này mà không làm ảnh hưởng tới giá trị ban đầu của biến.

```
>>> car = 'Audi'
>>> car == 'audi'
False
```

```
>>> car = 'Audi'
>>> car.lower() == 'audi'
True
```

```
>>> car = 'Audi'
>>> car.lower() == 'audi'
True
>>> car
'Audi'
```


Kiểm tra bất đẳng thức

Khi muốn kiểm tra hai giá trị không bằng nhau, kết hợp dấu chấm than với dấu bằng (!=).

Dấu chấm than có nghĩa là không(not) và được sử dụng trong nhiều ngôn ngữ lập trình

```
requested_topping = 'mushrooms'  
if requested_topping != 'anchovies':  
    print("Hold the anchovies!")
```

Hold the anchovies!

So sánh số

Mã nguồn sau đây kiểm tra xem một người đã đủ 18 tuổi hay chưa

```
>>> age = 18
>>> age == 18
True
```

có thể kiểm tra nếu hai số không bằng nhau

```
answer = 17
if answer != 42:
    print("That is not the correct answer. Please try again!")
That is not the correct answer. Please try again!
```

So sánh 2 số

Có thể bao gồm các phép so sánh toán học khác nhau trong các câu lệnh, chẳng hạn như nhỏ hơn, nhỏ hơn hoặc bằng, lớn hơn, và lớn hơn hoặc bằng

```
>>> age = 19
```

```
>>> age < 21
```

```
True
```

```
>>> age <= 21
```

```
True
```

```
>>> age > 21
```

```
False
```

```
>>> age >= 21
```

```
False
```

Kiểm tra nhiều điều kiện

SỬ DỤNG AND

Kiểm tra xem hai điều kiện có đồng thời là True hay không, hãy sử dụng từ khóa **and** để kết hợp hai kiểm tra điều kiện

```
>>> age_0 = 22
>>> age_1 = 18
>>> age_0 >= 21 and age_1 >= 21
False
>>> age_1 = 22
>>> age_0 >= 21 and age_1 >= 21
True
```

Có thể sử dụng dấu ngoặc đơn xung quanh biểu thức kiểm tra, nhưng chúng không bắt buộc

```
(age_0 >= 21) and (age_1 >= 21)
```

Kiểm tra nhiều điều kiện – Dùng OR

Từ khóa Or cho phép chúng ta kiểm tra nhiều điều kiện, nhưng nó vượt qua khi một trong hai hoặc cả hai bài kiểm tra riêng lẻ vượt qua. Một biểu thức Or chỉ thất bại khi cả hai bài kiểm tra riêng lẻ đều thất bại.

① `>>> age_0 = 22`

`>>> age_1 = 18`

② `>>> age_0 >= 21 or age_1 >= 21`

True

③ `>>> age_0 = 18`

`>>> age_0 >= 21 or age_1 >= 21`

False

Kiểm tra xem một giá trị có trong danh sách hay không

Để tìm hiểu xem một giá trị cụ thể đã có trong danh sách hay chưa, hãy sử dụng từ khóa **in**.

```
>>> requested_toppings = ['mushrooms', 'onions', 'pineapple']
>>> 'mushrooms' in requested_toppings
True
>>> 'pepperoni' in requested_toppings
False
```

Từ khóa **in** yêu cầu Python kiểm tra sự tồn tại của 'mushrooms' và 'pepperoni' trong danh sách `requested_toppings`. Kỹ thuật này là khá mạnh mẽ vì chúng ta có thể tạo danh sách các giá trị thiết yếu và sau đó dễ dàng kiểm tra xem giá trị ta đang kiểm tra có khớp với một trong các giá trị trong danh sách hay không.

Kiểm tra một giá trị không nằm trong danh sách

sử dụng từ khóa *not in* kiểm tra một phần tử không nằm trong danh sách

```
banned_users = ['andrew', 'carolina', 'david']
user = 'marie'
if user not in banned_users:
    print(f"{user.title()}, you can post a response if you wish.")
```

Marie, you can post a response if you wish.

4.3. Câu lệnh If

Khi hiểu các câu kiểm tra điều kiện, ta có thể bắt đầu viết câu lệnh if. Một số loại câu lệnh if khác nhau tồn tại và sự lựa chọn sử dụng phụ thuộc vào số lượng điều kiện ta cần kiểm tra.

Câu lệnh if đơn giản

Loại câu lệnh if đơn giản nhất có một kiểm tra và một hành động

```
if conditional_test:  
    do something
```

```
age = 19
```

①

```
if age >= 18:
```

②

```
    print("You are old enough to vote!")
```

`You are old enough to vote!`

Chúng ta có thể có bao nhiêu dòng mã tùy thích trong khối theo sau câu lệnh if.

```
age = 19
```

```
if age >= 18:
```

```
    print("You are old enough to vote!")
```

```
    print("Have you registered to vote yet?")
```

`You are old enough to vote!`

`Have you registered to vote yet?`

Câu lệnh if-else

Một khối if-else tương tự như một câu lệnh if đơn giản, nhưng câu lệnh else cho phép ta xác định một hành động hoặc một tập hợp các hành động được thực thi khi kiểm tra điều kiện không thành công.

```
age = 17
```

① `if age >= 18:`

```
    print("You are old enough to vote!")
```

```
    print("Have you registered to vote yet?")
```

② `else:`

```
    print("Sorry, you are too young to vote.")
```

```
    print("Please register to vote as soon as you turn 18!")
```

Sorry, you are too young to vote.

Please register to vote as soon as you turn 18!

Chuỗi if-elif-else

hãy xem xét một công viên giải trí tính phí các mức giá khác nhau cho các nhóm tuổi khác nhau

```
age = 12  
① if age < 4:  
    print("Your admission cost is $0.")  
② elif age < 18:  
    print("Your admission cost is $25.")  
③ else:  
    print("Your admission cost is $40.")
```

Your admission cost is \$25

- Miễn phí vé vào cửa cho bất kỳ ai dưới 4 tuổi.
- Vé vào cửa cho bất kỳ ai trong độ tuổi từ 4 đến 18 là \$ 25.
- Vé vào cửa cho bất kỳ ai từ 18 tuổi trở lên là \$ 40.

Chuỗi if-elif-else (t)

Thay vì in giá vào cửa trong khối if-elif-else, sẽ ngắn gọn hơn nếu chỉ đặt giá bên trong chuỗi if-elif-else và sau đó có một lệnh gọi print() đơn giản chạy sau khi chuỗi đã được đã đánh giá

```
age = 12
if age < 4:
    ① price = 0
elif age < 18:
    ② price = 25
else:
    ③ price = 40
    ④ print(f"Your admission cost is ${price}.")
```

Sử dụng nhiều khối elif

Có thể sử dụng bao nhiêu khối elif trong mã tùy thích

Giả sử rằng bất kỳ ai 65 tuổi trở lên trả một nửa số tiền vào cửa thông thường, hoặc \$ 20:

```
age = 12
if age < 4:
    price = 0
elif age < 18:
    price = 25
① elif age < 65:
    price = 40
② else:
    price = 20
print(f"Your admission cost is ${price}.")
```

Bỏ qua khối else

Python không yêu cầu một khối else ở cuối chuỗi if-elif.

Đôi khi một khối else hữu ích; đôi khi việc sử dụng một khối elif khác khiến nó rõ ràng hơn

```
age = 12
if age < 4:
    price = 0
elif age < 18:
    price = 25
elif age < 65:
    price = 40
① elif age >= 65:
    price = 20
print(f"Your admission cost is ${price}.")
```

Khối elif bỏ sung ở ① ấn định giá 20 đô la khi người đó 65 tuổi hoặc già hơn, rõ ràng hơn một chút so với khối else. Với sự thay đổi này, mọi khối mã phải vượt qua một bài kiểm tra riêng biệt để được thực thi.

Bài tập

TRY IT YOURSELF

5-3. Alien Colors #1: Imagine an alien was just shot down in a game. Create a variable called `alien_color` and assign it a value of 'green', 'yellow', or 'red'.

- Write an `if` statement to test whether the alien's color is green. If it is, print a message that the player just earned 5 points.
- Write one version of this program that passes the `if` test and another that fails. (The version that fails will have no output.)

5-4. Alien Colors #2: Choose a color for an alien as you did in Exercise 5-3, and write an `if-else` chain.

- If the alien's color is green, print a statement that the player just earned 5 points for shooting the alien.
- If the alien's color isn't green, print a statement that the player just earned 10 points.
- Write one version of this program that runs the `if` block and another that runs the `else` block.

Bài tập

5-5. Alien Colors #3: Turn your `if-else` chain from Exercise 5-4 into an `if-elif-else` chain.

- If the alien is green, print a message that the player earned 5 points.
- If the alien is yellow, print a message that the player earned 10 points.
- If the alien is red, print a message that the player earned 15 points.
- Write three versions of this program, making sure each message is printed for the appropriate color alien.

Bài tập

5-6. Stages of Life: Write an if-elif-else chain that determines a person's stage of life. Set a value for the variable age, and then:

- If the person is less than 2 years old, print a message that the person is a baby.
- If the person is at least 2 years old but less than 4, print a message that the person is a toddler.
- If the person is at least 4 years old but less than 13, print a message that the person is a kid.
- If the person is at least 13 years old but less than 20, print a message that the person is a teenager.
- If the person is at least 20 years old but less than 65, print a message that the person is an adult.
- If the person is age 65 or older, print a message that the person is an elder.

XIN CẢM ƠN!
