

Học xong bài này, em sẽ:

- ✓ Biết được các phép so sánh và các phép tính logic tạo thành biểu thức logic thể hiện điều kiện rẽ nhánh trong chương trình.
- ✓ Viết được câu lệnh rẽ nhánh trong Python.



Cấu trúc rẽ nhánh trong mô tả thuật toán dùng để thể hiện một hành động được thực hiện hay không tùy thuộc vào một điều kiện có được thỏa mãn hay không. Nếu em trình bày cách giải một phương trình bậc hai $ax^2 + bx + c = 0$, em có sử dụng cấu trúc rẽ nhánh hay không?

1) Cấu trúc rẽ nhánh trong mô tả thuật toán

Em đã biết, trong quá trình thực hiện thuật toán, khi phải dựa trên một điều kiện cụ thể nào đó để xác định bước thực hiện tiếp theo thì cần cấu trúc rẽ nhánh (Hình 1a).

Nếu <điều kiện>:
Nhánh đúng
Trái lại:
Nhánh sai
Hết nhánh

Hình 1a. Mẫu cấu trúc rẽ nhánh

Nếu a chia hết cho 2:
In ra màn hình 'số chẵn'
Trái lại:
In ra màn hình 'số lẻ'
Hết nhánh

Hình 1b. Ví dụ thể hiện mẫu cấu trúc rẽ nhánh



1

Em hãy vẽ sơ đồ khối thể hiện cấu trúc rẽ nhánh trong ví dụ ở Hình 1b.

Các ngôn ngữ lập trình bậc cao đều cung cấp các công cụ để mô tả <điều kiện>, tính giá trị <điều kiện> và câu lệnh thể hiện cấu trúc rẽ nhánh dựa trên giá trị tính được của <điều kiện>.

2) Điều kiện rẽ nhánh

Trong mô tả thuật toán, <điều kiện> rẽ nhánh phải là một biểu thức nhận giá trị logic True hoặc False.

Phép so sánh hai giá trị hay so sánh hai biểu thức sẽ cho ta một biểu thức logic. Như vậy, các phép so sánh thường được sử dụng để biểu diễn các <điều kiện>. Bảng 1 mô tả cách viết các phép so sánh trong Python.

Bảng 1. Kí hiệu phép so sánh trong Python

So sánh	Kí hiệu trong Python
Lớn hơn	>
Lớn hơn hoặc bằng	>=
Nhỏ hơn	<
Nhỏ hơn hoặc bằng	<=
Bằng	==
Khác	!=

Ví dụ 1. Bảng 2 minh họa một số <điều kiện> được biểu diễn bằng phép so sánh viết trong Python và giá trị logic tương ứng của nó.

Bảng 2. Ví dụ một số phép toán quan hệ

Điều kiện	Giá trị logic của điều kiện với A=5, B=10
A < B	True
A * A + B * B <= 100	False
A + 5 != B	False
2 * A == B	True

Kết nối các biểu thức logic với nhau bằng các phép tính logic (and – và, or – hoặc, not – phủ định) ta lại nhận được một biểu thức logic (Hình 2).

Phép tính	Biểu thức	Ý nghĩa
and	x and y	Cho kết quả là True khi và chỉ khi x và y đều nhận giá trị True
or	x or y	Cho kết quả là False khi và chỉ khi x và y đều nhận giá trị False
not	not x	Đảo giá trị logic của x

Hình 2. Một số phép toán logic

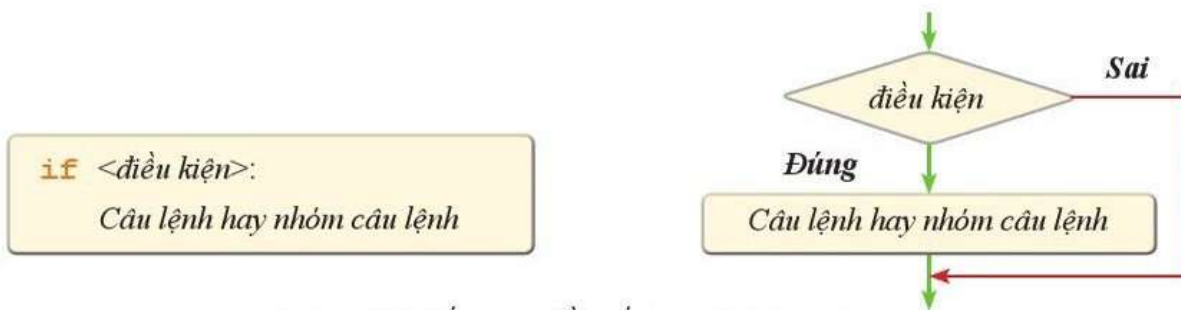
Ví dụ 2. Bảng 3 cho ta một số ví dụ về <điều kiện> được tạo thành do kết nối một vài biểu thức logic lại bằng các phép tính logic.

Bảng 3. Ví dụ kết quả tính biểu thức logic

Điều kiện	Giá trị của biểu thức logic điều kiện A=5, B=10
(A < B) and (A + 5 != B)	False
(3 * A > B) or (2 * A == B)	True
not (A * A + B * B <= 100)	True

3 Câu lệnh rẽ nhánh trong chương trình Python

Tương ứng với hai loại cấu trúc rẽ nhánh trong thuật toán, Python cung cấp hai câu lệnh rẽ nhánh. Hình 3 cho thấy cách viết câu lệnh rẽ nhánh dạng **if** (bên trái) và sơ đồ khối tương ứng của cấu trúc này (bên phải).



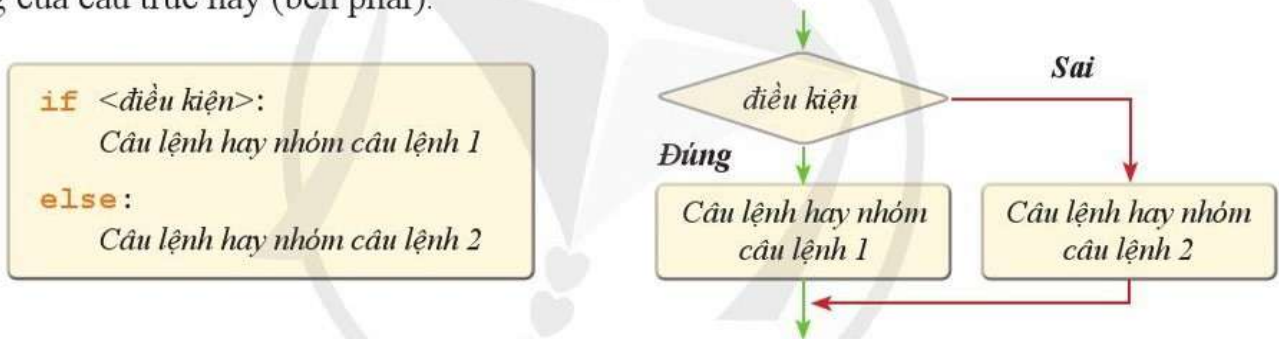
Hình 3. Cách viết và sơ đồ khối của câu lệnh `if`

Ví dụ 3. Hình 4 minh họa một chương trình sử dụng câu lệnh `if` trong Python.

```
File Edit Shell Debug Options Window Help
>>> t=9
>>> if t<10:
    print(t,"không phải là số nguyên dương có hai chữ số")
9 không phải là số nguyên dương có hai chữ số
>>>
```

Hình 4. Chương trình kiểm tra số nguyên dương có hai chữ số

Hình 5 minh họa cách viết câu lệnh rẽ nhánh `if-else` (bên trái) và sơ đồ khối tương ứng của cấu trúc này (bên phải).



Hình 5. Cách viết và sơ đồ khối của câu lệnh `if-else`

Câu lệnh hoặc các câu lệnh trong cùng nhóm phải được viết lùi vào trong một số vị trí so với dòng chứa điều kiện và viết thẳng hàng với nhau (Hình 6). Một nhóm các câu lệnh như vậy còn gọi là một khối lệnh.

Khối lệnh sau `if` phải lùi vào trong so với `if`

Khối lệnh sau `else` phải lùi vào trong so với `else`

File Edit Format Run Options Window Help

```
A = int(input("Nhập vào một số nguyên: "))
if A%2 == 0:
    print(A, " là số chẵn.")
else:
    print(A, " là số lẻ.")
```

Chương trình

File Edit Shell Debug Options Window Help

```
Nhập vào một số nguyên: 15
15 là số lẻ.
>>>
```

Kết quả thực hiện

Hình 6. Cách viết các câu lệnh

Lưu ý: Cách viết các câu lệnh trong Python:

- Các câu lệnh ở khối trong viết lùi các đầu dòng nhiều hơn các câu lệnh khối ngoài.
- Các câu lệnh cùng một khối: có khoảng cách tới đầu dòng như nhau.

Ví dụ 4. Tây Nguyên sản xuất hai loại cà phê là Robusta và Arabica. Trung bình hằng năm lượng cà phê Arabica chiếm 10% tổng sản lượng và giá bán trung bình gấp 2,5 lần so với cà phê Robusta. Những năm Arabica được mùa (chiếm từ 10% tổng sản lượng trở lên), giá bán chỉ gấp 2 lần, còn khi mất mùa thì giá bán gấp 3 lần.

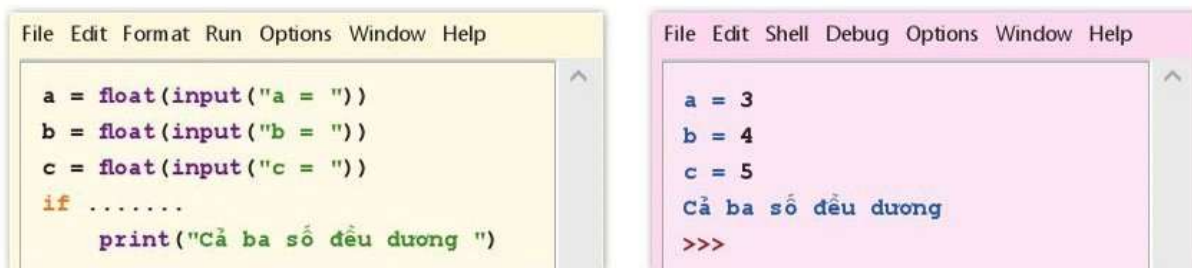
Chương trình ở *Hình 7* cho phép nhập vào tổng sản lượng cà phê và sản lượng cà phê Arabica. Chương trình sẽ đưa ra thông báo “Arabica được mùa” hoặc “Arabica mất mùa” cùng tỉ lệ giá bán tương ứng của Arabica.



Hình 7. Chương trình đánh giá sản lượng cà phê ở Tây Nguyên



Bài 1. Hoàn thiện câu lệnh `if` trong chương trình ở *Hình 8a* để có được chương trình nhập từ bàn phím ba số thực a, b, c và đưa ra màn hình thông báo “Cả ba số đều dương” nếu ba số nhập vào đều dương. *Hình 8b* minh họa một kết quả chạy chương trình.



Hình 8a. Chương trình

Hình 8b. Ví dụ chạy chương trình với $a = 3, b = 4$ và $c = 5$

Bài 2. Viết chương trình để nhập từ bàn phím hai số nguyên a và b, đưa ra màn hình thông báo “Positive” nếu $a + b > 0$, “Negative” nếu $a + b < 0$ và “Zero” nếu $a + b = 0$.

Ví dụ:

INPUT	OUTPUT
$a = 4$ $b = -10$	Negative



Năm nhuận

Năm nhuận là những năm chia hết cho 400 hoặc là những năm chia hết cho 4 nhưng không chia hết cho 100. Đặc biệt, những năm chia hết cho 3 328 được đề xuất là năm nhuận kép. Với số nguyên dương n nhập vào từ bàn phím, em hãy đưa ra màn hình thông báo: “Không là năm nhuận” nếu n không phải là năm nhuận; “Năm nhuận” nếu n là năm nhuận và “Năm nhuận kép” nếu n là năm nhuận kép.



Trong các câu sau đây, những câu nào đúng?

- 1) Trong câu lệnh rẽ nhánh của ngôn ngữ lập trình bậc cao phải có một biểu thức logic thể hiện điều kiện rẽ nhánh.
- 2) Biểu thức logic chỉ được lấy làm điều kiện rẽ nhánh nếu chưa chạy chương trình đã xác định được giá trị của biểu thức đó đúng hay sai.
- 3) Có thể kết nối các biểu thức logic với nhau bằng các phép tính logic để được một điều kiện rẽ nhánh.
- 4) Trong Python câu lệnh rẽ nhánh có dạng: **if** <điều kiện> **else** <các câu lệnh>.

Tóm tắt bài học

- ✓ Các ngôn ngữ lập trình bậc cao đều có câu lệnh thể hiện cấu trúc rẽ nhánh.
- ✓ Điều kiện trong câu lệnh rẽ nhánh là một biểu thức logic, nhận giá trị logic True hoặc False.

Câu lệnh rẽ nhánh trong Python có hai dạng cơ bản là:

if <điều kiện>:
 Câu lệnh hay nhóm câu lệnh

if <điều kiện>:
 Câu lệnh hay nhóm câu lệnh 1
else:
 Câu lệnh hay nhóm câu lệnh 2