

LẬP TRÌNH PYTHON



Cấu trúc rẽ nhánh và biểu thức logic

Nội dung

- Biến Boolean
- Toán tử logic
- Câu lệnh if
- Câu lệnh if-else
- Câu lệnh if-elif-else
- Cấu trúc quyết định lồng nhau

Biểu thức Boolean

- Biểu thức Boolean: biểu thức được kiểm tra bằng câu lệnh if để xác định xem nó đúng hay sai
- Ví dụ: $a > b$ đúng nếu a lớn hơn b ; sai nếu không
- Toán tử quan hệ: xác định xem có mối quan hệ cụ thể nào tồn tại giữa hai giá trị hay không. Ví dụ: lớn hơn ($>$)

Biểu thức Boolean

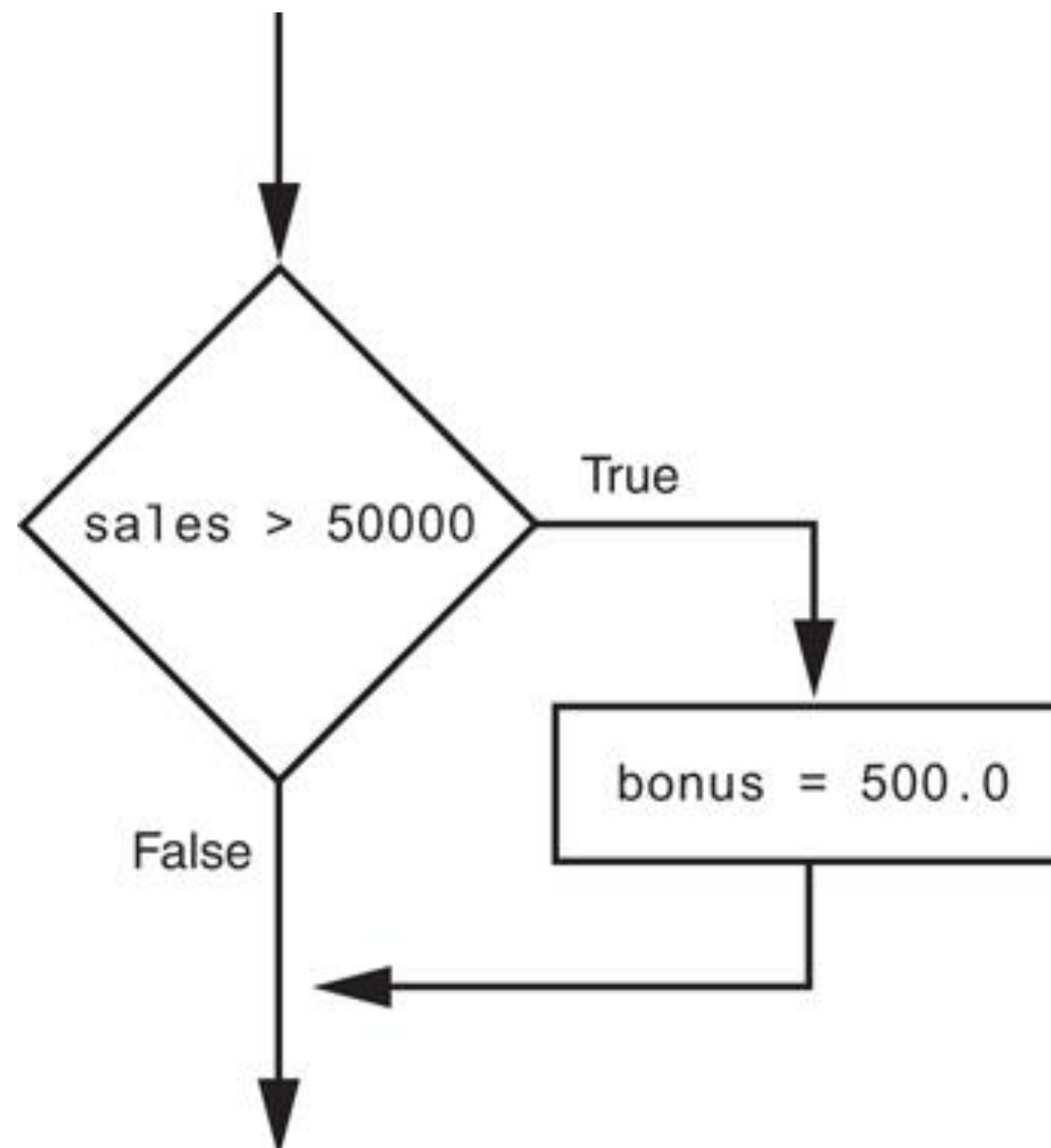
- Toán tử \geq và \leq kiểm tra nhiều hơn một mối quan hệ
- Chỉ cần một trong các mối quan hệ tồn tại là đủ để biểu thức đúng
- Toán tử $==$ xác định xem hai toán hạng có bằng nhau không
- Không nhầm lẫn với toán tử gán ($=$)
- Toán tử $!=$ xác định xem hai toán hạng có không bằng nhau không

Biểu thức Boolean

Expression	Meaning
$x > y$	Is x greater than y?
$x < y$	Is x less than y?
$x \geq y$	Is x greater than or equal to y?
$x \leq y$	Is x less than or equal to y?
$x == y$	Is x equal to y?
$x != y$	Is x not equal to y?

Biểu thức Boolean

- Sử dụng biểu thức Boolean với toán tử quan hệ lớn hơn >



Biểu thức Boolean

Ví dụ:

```
a = True
```

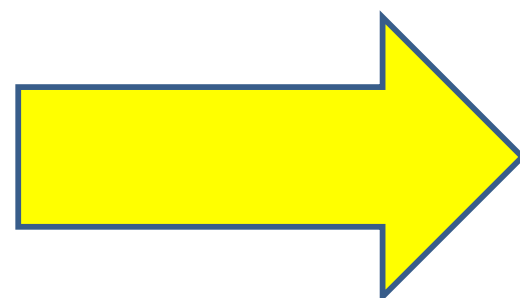
```
b = False
```

```
print('a =', a, ' b =', b)
```

```
# gán lại kết quả cho a
```

```
a = False
```

```
print('a =', a, ' b =', b)
```



```
a = True b = False
```

```
a = False b = False
```

Biểu thức Boolean

Bảng tổng quát:

Biểu thức	Ý nghĩa
$x == y$	True nếu $x=y$, False nếu x khác y
$x < y$	True nếu $x < y$, False nếu $x \geq y$
$x \leq y$	True nếu $x \leq y$, False nếu $x > y$
$x > y$	True nếu $x > y$, False nếu $x \leq y$
$x \geq y$	True nếu $x \geq y$, False nếu $x < y$
$x \neq y$	True nếu x khác y , False nếu $x=y$

Toán tử logic

- Toán tử logic: toán tử có thể được sử dụng để tạo ra các biểu thức Boolean phức tạp
 - toán tử và **and**, toán tử hoặc **or**: toán tử nhị phân, kết nối hai biểu thức Boolean thành một biểu thức Boolean hợp chất
 - toán tử không **not**: toán tử đơn, đảo ngược giá trị đúng của toán hạng Boolean của nó

Toán tử **and**

Expression	Value of the Expression
false and false	false
false and true	false
true and false	false
true and true	true

Toán tử **or**

Expression	Value of the Expression
false and false	false
false and true	true
true and false	true
true and true	true

Toán tử **not**

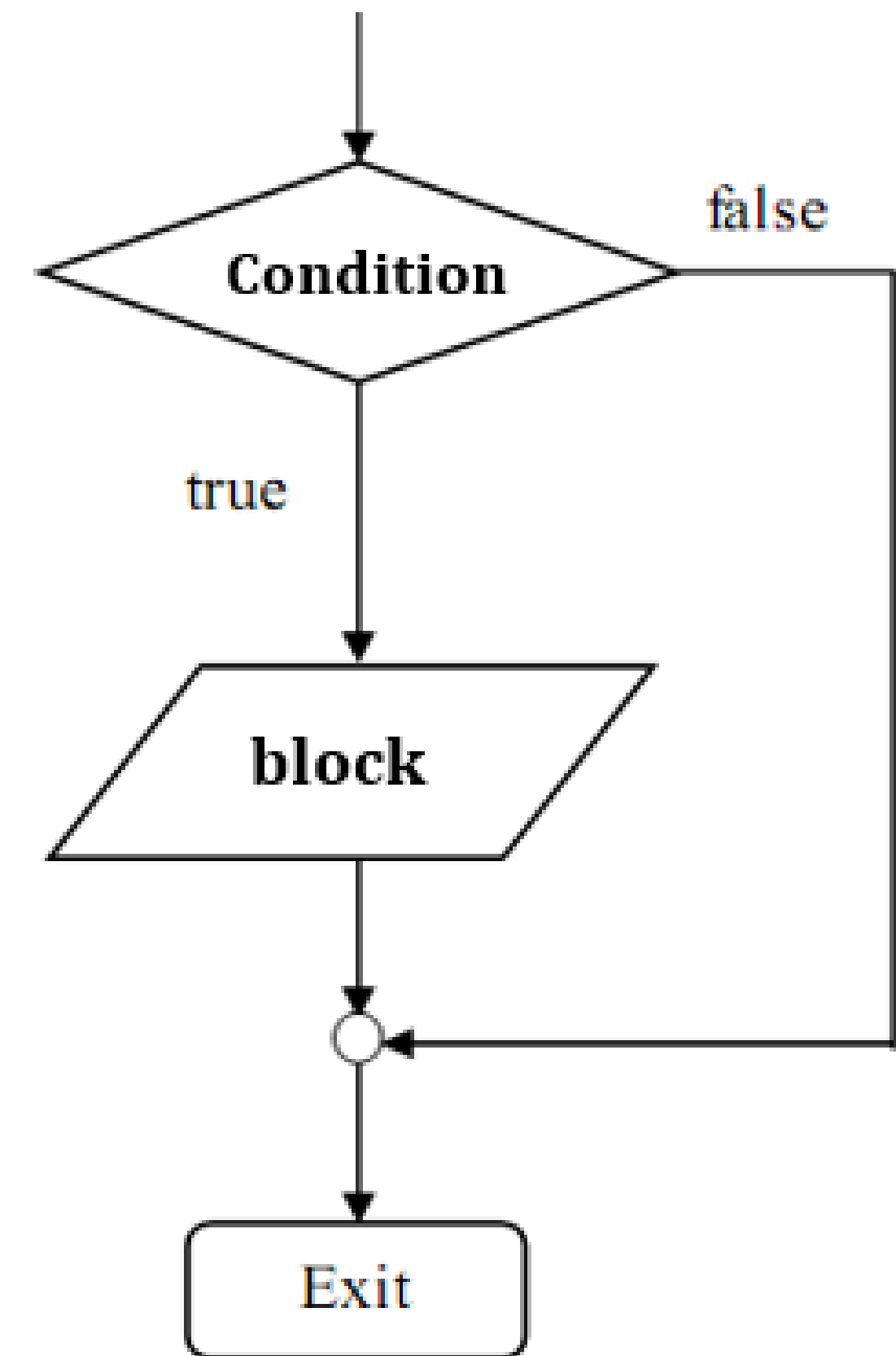
Expression	Value of the Expression
true	false
false	true

Kiểm tra If

Biểu thức if là một biểu thức điều kiện rất quan trọng và phổ biến trong Python. Biểu thức if đứng một mình chỉ quan tâm tới điều kiện đúng (True). Khi điều kiện đúng thì khối lệnh bên trong if sẽ được thực thi.

Cú pháp:

if *condition* :
 block



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

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
```

Biểu thức if ... else

Biểu thức if...else là một biểu thức điều kiện rất quan trọng và phổ biến trong Python. Biểu thức này quan tâm điều kiện đúng(True) và sai(False). Nó phổ biến hơn biểu thức if.

Cú pháp:

if

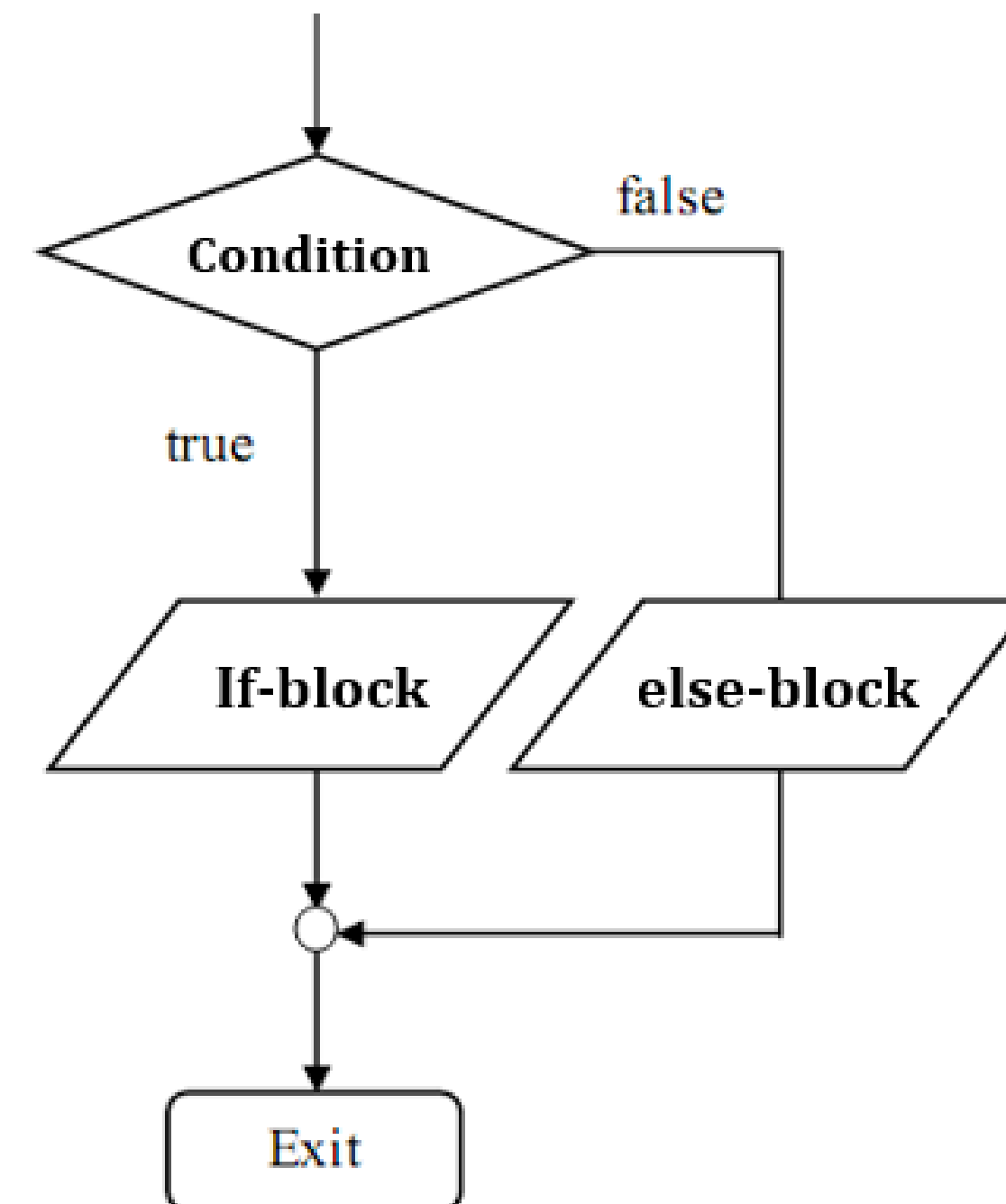
condition

:

if-block

else:

else-block



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.")
```

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

- Câu lệnh if-elif-else: phiên bản đặc biệt của cấu trúc quyết định
- Làm cho logic của các cấu trúc quyết định lồng nhau dễ viết hơn
- Có thể bao gồm nhiều câu lệnh elif

```
if condition_1:  
    statement(s)  
elif condition_2:  
    statement(s)  
elif condition_3:  
    statement(s)  
else  
    statement(s)
```

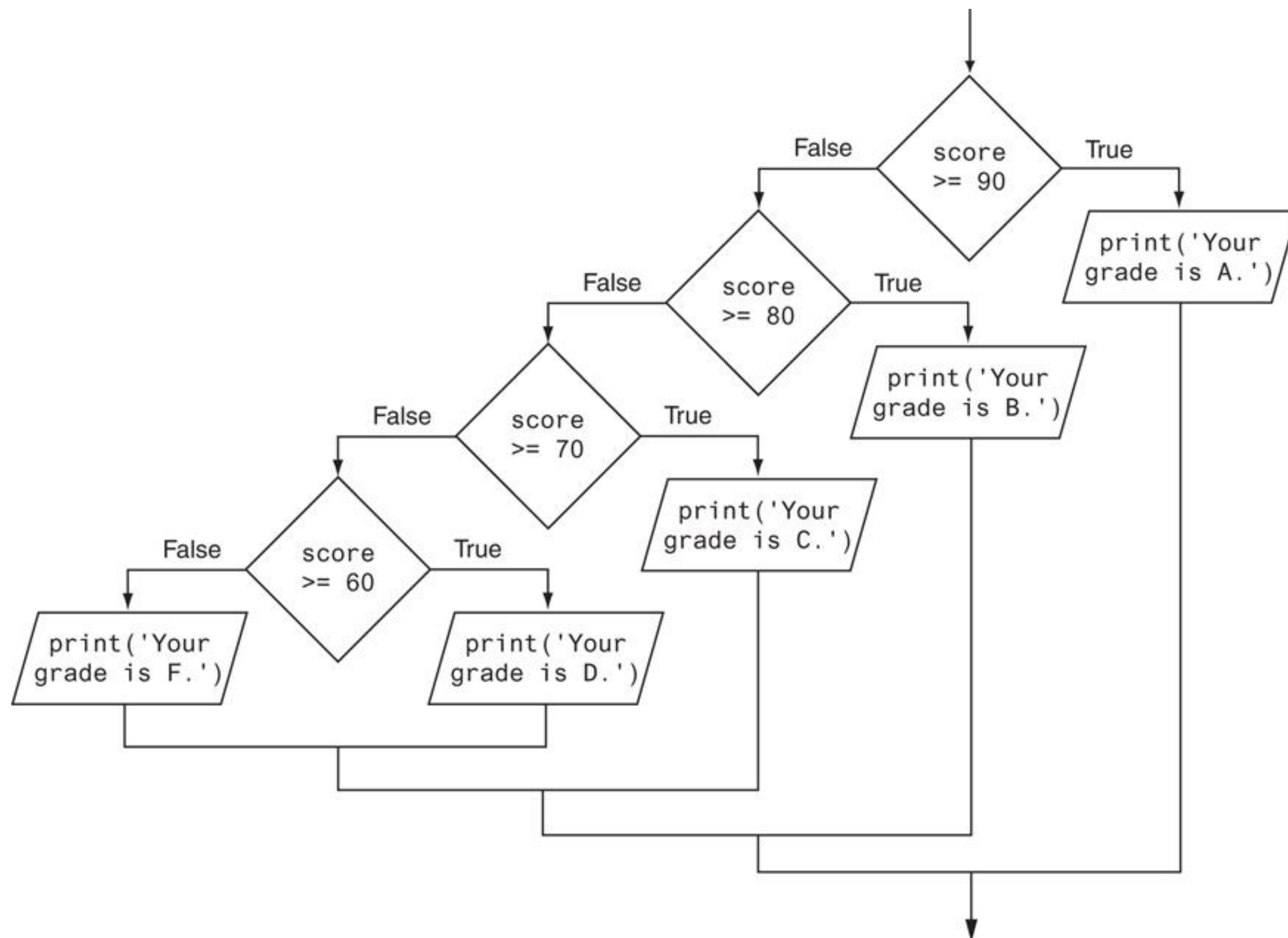
Insert as many `elif` clauses
as necessary.

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 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}.")
```


Chuỗi if-elif-else (t)



Câu lệnh pass

Câu lệnh **pass** dùng để dành chỗ lập trình. Ví dụ biết chỗ đó phải viết rất nhiều coding, nhưng tại thời điểm này chưa kịp làm. Ta sẽ dùng **pass** để đánh dấu vị trí đó.

Lỗi, Python không cho để trống như thế này

```
a=float(input("Nhập hệ số a:"))
b=float(input("Nhập hệ số b:"))
if a==0:
else:
    x=-b/a;
    print("{0}x+{1}=0".format(a,b))
    print("có nghiệm x=",x)
```



```
a=float(input("Nhập hệ số a:"))
b=float(input("Nhập hệ số b:"))
if a==0:
    pass
else:
    x=-b/a;
    print("{0}x+{1}=0".format(a,b))
    print("có nghiệm x=",x)
```

Sử dụng if/else như phép gán

Đôi khi việc thực hiện if else trong một biểu thức quá đơn giản sẽ làm cho if else phức tạp không cần thiết:

```
a=5  
b=7  
if a != b:  
    c = 113  
else:  
    c = 115  
print(c)
```

→ `c = 113 if a != b else 115`

expression-1

if

condition

else

expression-2

Tổng kết

- Biến Boolean và giá trị logic
- Sử dụng toán tử logic và/hoặc/phủ định
- Cấu trúc câu lệnh quyết định dạng if
- Cấu trúc câu lệnh quyết định dạng if-else
- Cấu trúc câu lệnh quyết định dạng if-elif-else
- Cấu trúc quyết định lồng nhau

HỎI ĐÁP





Cảm ơn đã lắng nghe!