

# Chapter 5

---

[IF – ELSE] 判斷式

# 縮排

---

**因為 Python 是直譯的程式，所以要開始學習縮排，否則程式有可能不會動**

如何縮排 很簡單

1. 按下鍵盤的 Tab 鍵
2. 按下鍵盤的 空白鍵



# 條件分支

---

程式會按照當時判斷式的答案，來分別執行不同的動作

舉例來說

當你去 7-11 要買花雕雞麵或是阿 Q 桶麵時，發現你身上只有 50 元

所以在買不起花雕雞麵的情況下，只能選擇阿 Q 桶麵

# 判斷

---

## 單向 if

if 是程式語言中最簡單的判斷式

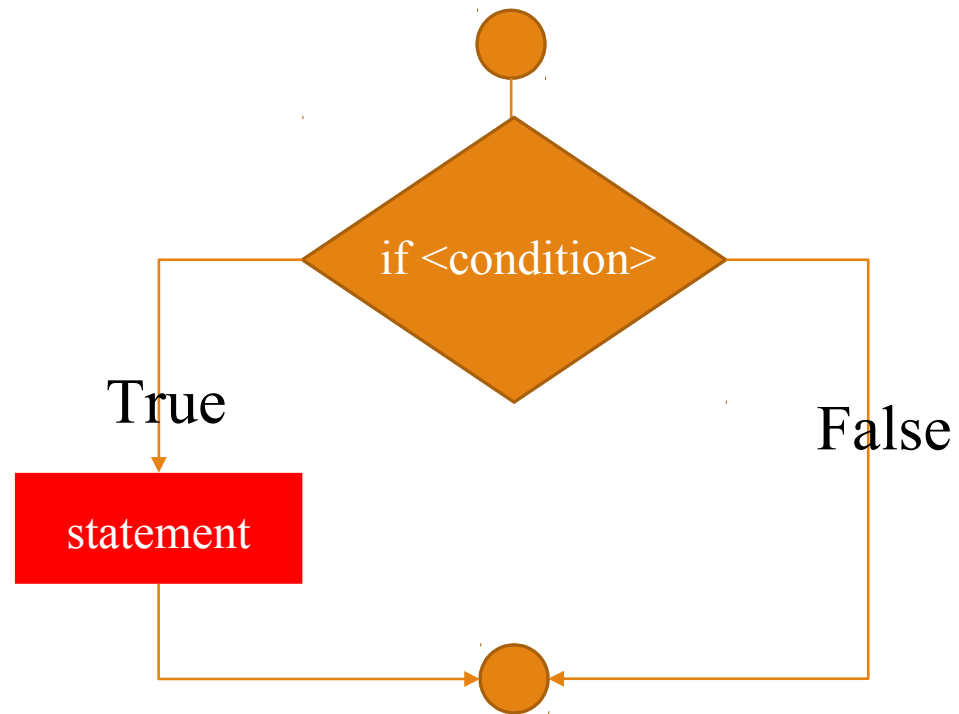
先以單向判斷來當成第一個例子 (記得縮排)

```
if <condition> :  
    <statement>
```

# 判斷

## 單向 if

if <condition> :  
    <statement>



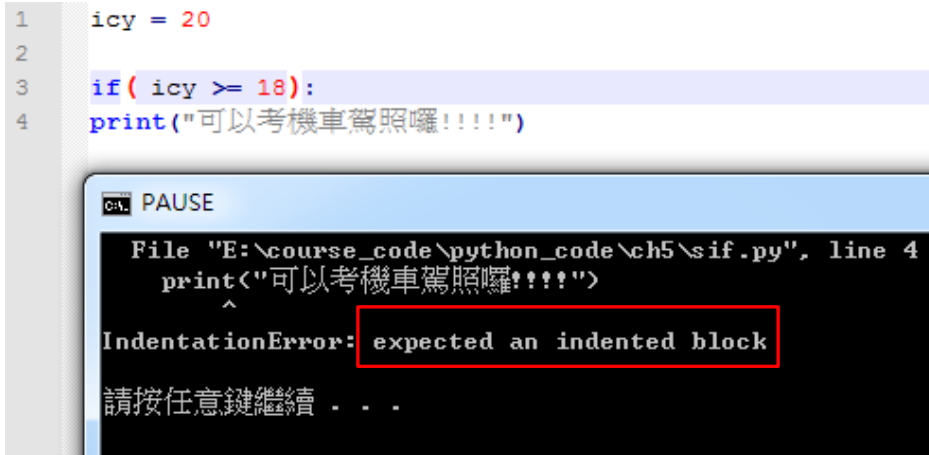
# 判斷

## 單向 if

if 在 <condition> 後面別忘了一定要加上 **:**

下一行的 <statement> 一定要**記得縮排**，不然會沒辦法跑哦

```
1 icy = 20
2
3 if( icy >= 18):
4 print("可以考機車駕照囉!!!!")
```



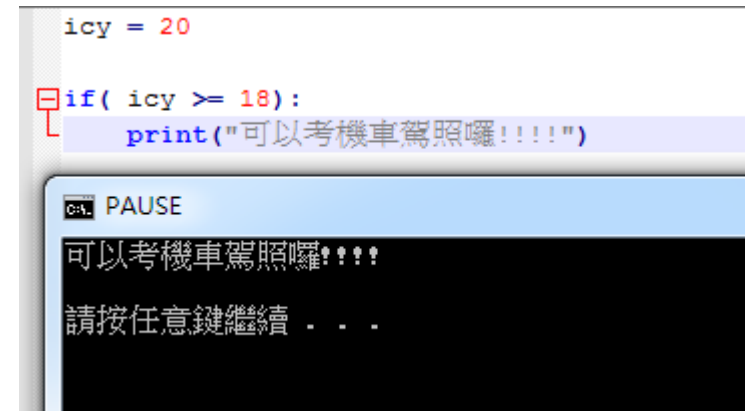
PAUSE

File "E:\course\_code\python\_code\ch5\sif.py", line 4  
print("可以考機車駕照囉!!!!")  
^  
IndentationError: expected an indented block

請按任意鍵繼續 . . .

```
icy = 20

if( icy >= 18):
    print("可以考機車駕照囉!!!!")
```



PAUSE

可以考機車駕照囉!!!!

請按任意鍵繼續 . . .

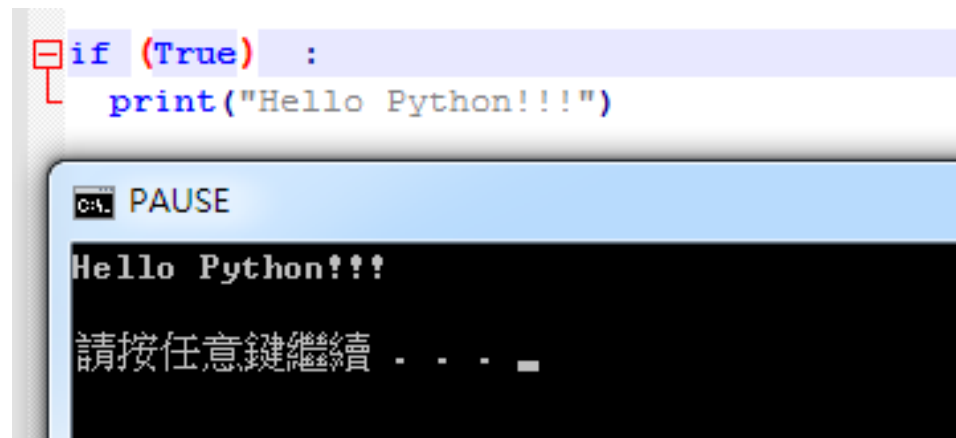
# 判斷

---

## 單向 if

if <condition =True>:

<statement> 一定會執行



The screenshot shows a Python IDE with a code editor and a console window. In the code editor, the following code is written:

```
if (True) :  
    print("Hello Python!!!")
```

The console window, titled "PAUSE", displays the output of the code:

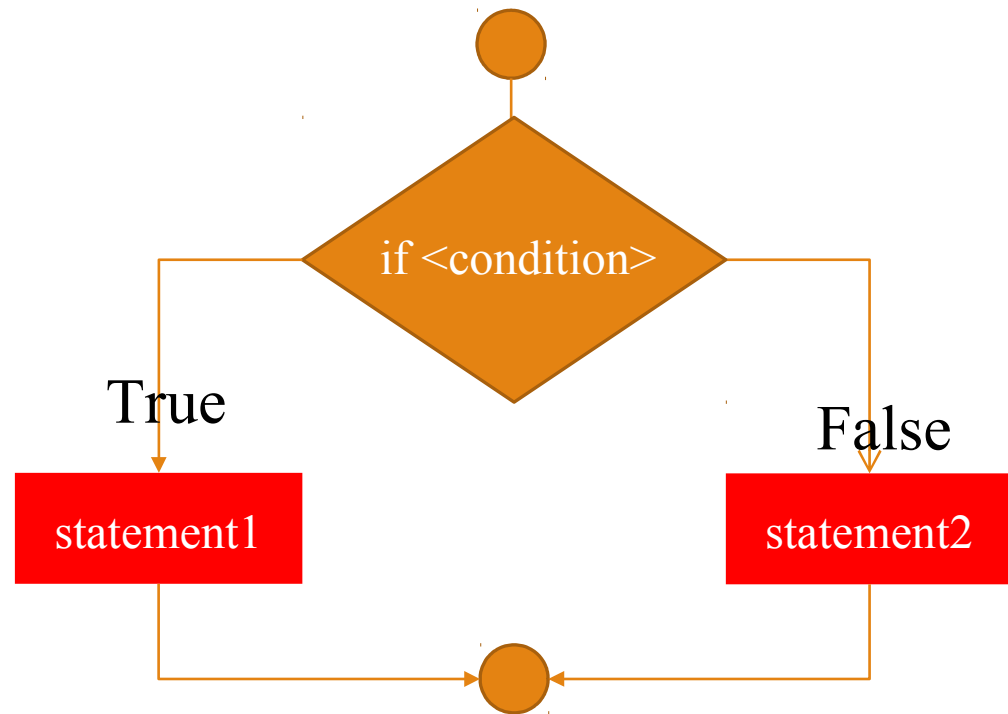
```
Hello Python!!!  
請按任意鍵繼續 . . .
```

# 判斷

if ... else ...

```
if <condition> :  
    <statement1>  
else :  
    <statement2>
```

if 判斷式的結果如是 false  
程式就會執行 else 後面的  
Statement2





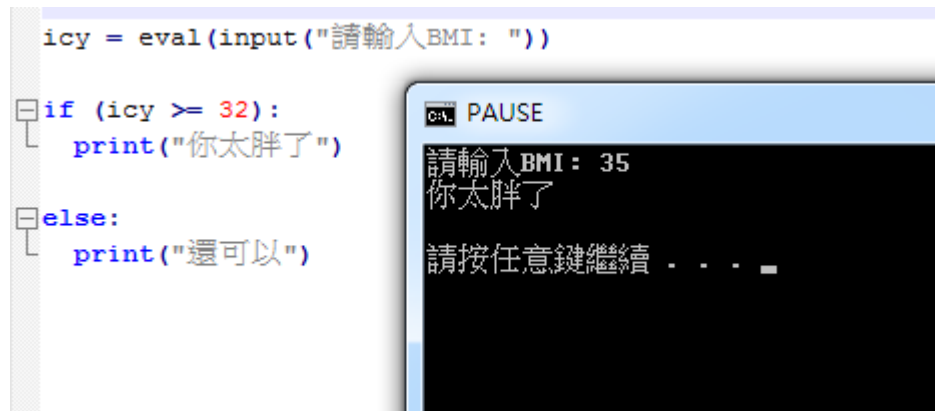
# 判斷

---

if ... else ...

else 後面別忘了一定要加上 :

<statement> 一定要記得縮排，不然會沒辦法跑哦



```
icy = eval(input("請輸入BMI: "))  
  
if (icy >= 32):  
    print("你太胖了")  
  
else:  
    print("還可以")
```

The screenshot shows a Python script in an IDE. The script prompts the user to input a BMI value. If the input is greater than or equal to 32, it prints "你太胖了" (You are too fat). Otherwise, it prints "還可以" (It's okay). The execution output shows the user inputting 35, and the program correctly prints "你太胖了".

# 判斷

---

if...elif...else

多向判斷 (if...elif...else) :

elif = else if

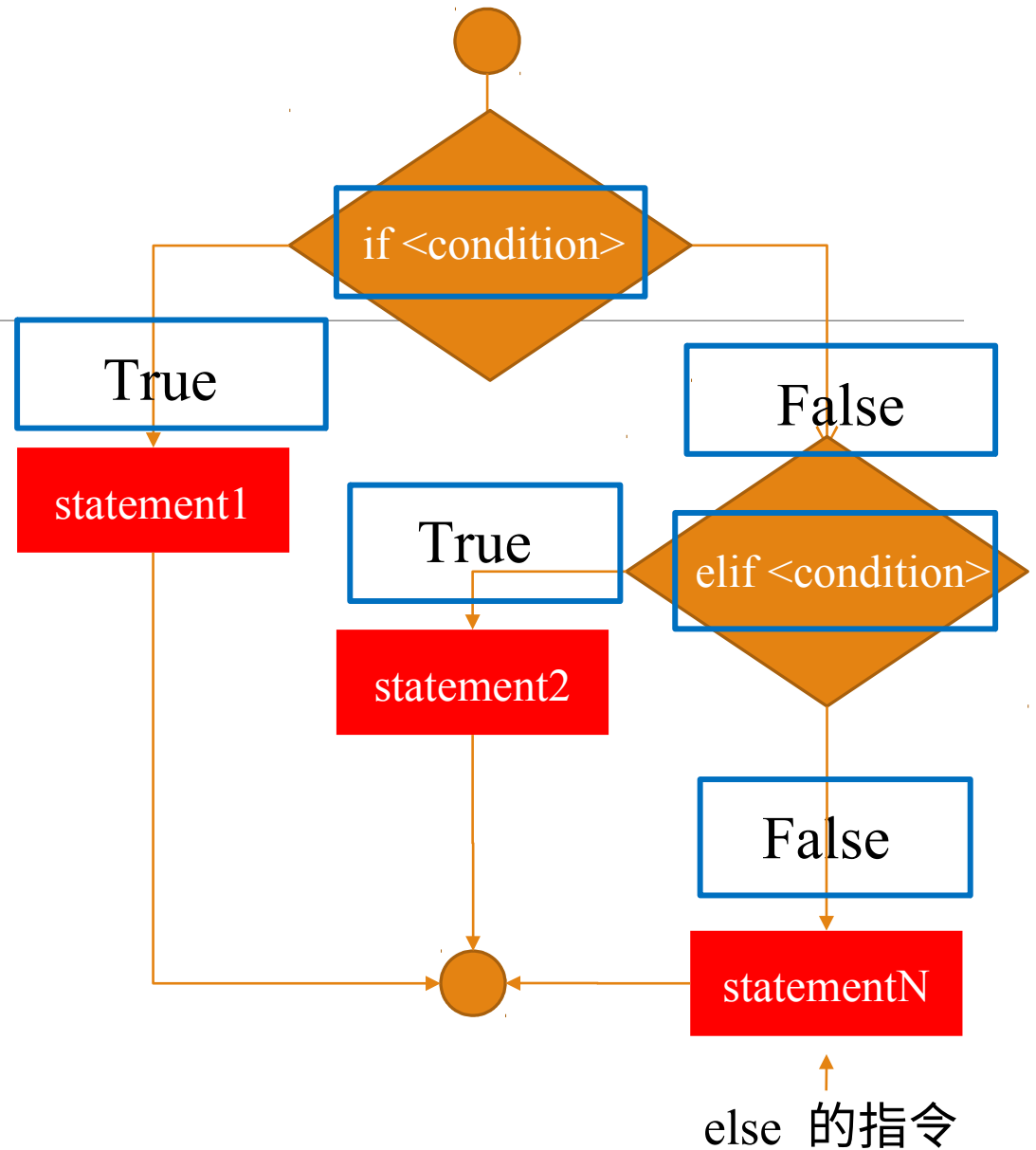
```
if <condition> :  
    <statement1>  
elif <condition> :  
    <statement2>  
elif <condition> :  
    <statementN-1>  
else:  
    <statementN>
```

# 判斷

if...elif...else

```
if <condition> :  
    <statement1>  
elif <condition> :  
    <statement2>  
elif <condition> :  
    <statementN-1>  
else:  
    <statementN>
```

elif 可以有很多個



# 判斷

if...elif...else

縮排縮排縮排

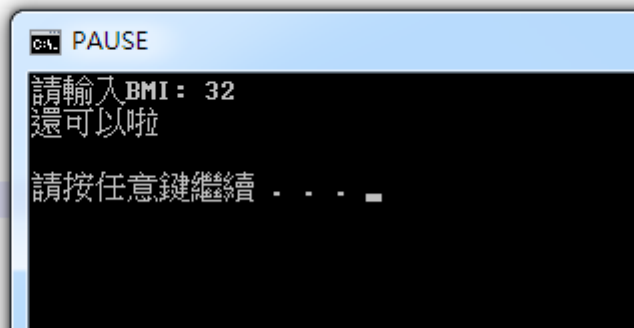
if <condition> :  
    <statement1>

elif <condition> :  
    <statement2>

elif <condition> :  
    <statementN-1>

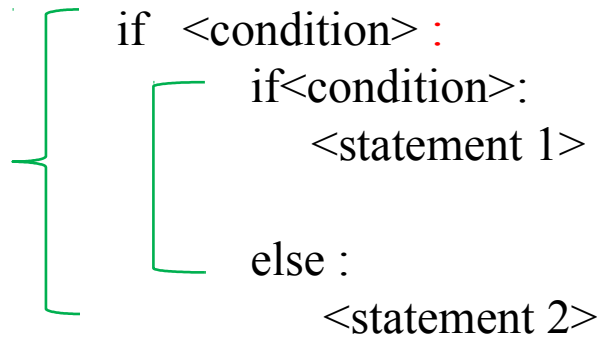
else:  
    <statementN>

```
icy = eval(input("請輸入BMI: "))  
  
if (icy > 32):  
    print("你太胖了")  
  
elif (25 <= icy <= 32 ) :  
    print("還可以啦")  
  
else :  
    print("太瘦了哦 多吃一點")
```



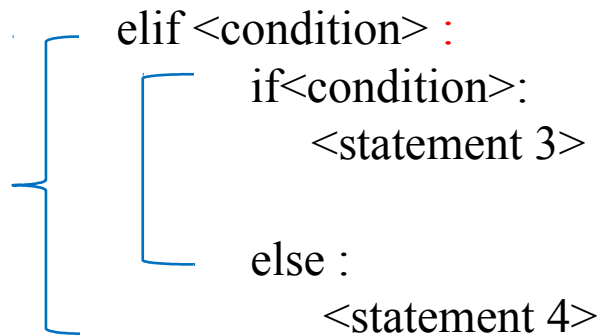
# 判斷

頂級版巢狀判斷 (if...elif...else) :



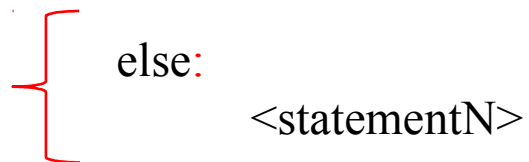
```
if <condition> :  
    if<condition>:  
        <statement 1>  
    else :  
        <statement 2>
```

A diagram showing an if-else statement structure. A large green bracket on the left groups the entire block. Inside, the text is: if <condition> : followed by an indented if<condition>: block containing <statement 1>, and an else : block containing <statement 2>.



```
elif <condition> :  
    if<condition>:  
        <statement 3>  
    else :  
        <statement 4>
```

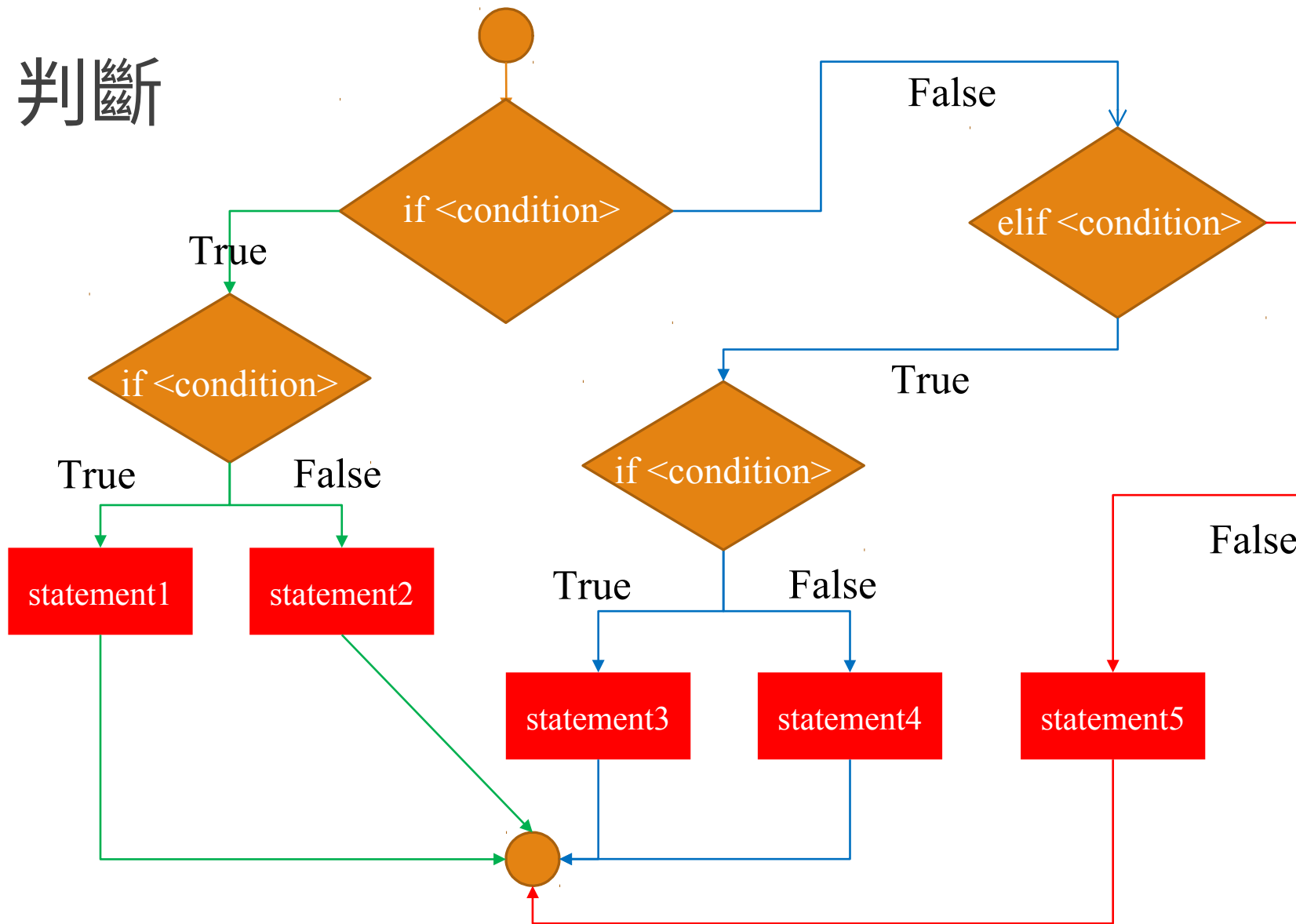
A diagram showing an elif statement structure. A large blue bracket on the left groups the entire block. Inside, the text is: elif <condition> : followed by an indented if<condition>: block containing <statement 3>, and an else : block containing <statement 4>.



```
else:  
    <statementN>
```

A diagram showing an else statement structure. A large red bracket on the left groups the entire block. Inside, the text is: else: followed by <statementN>.

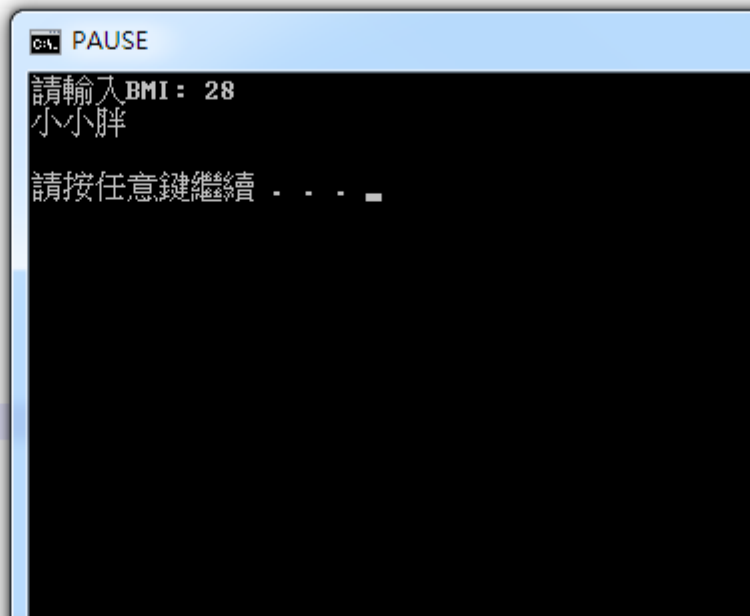
# 判断



# 判斷

```
icy = eval(input("請輸入BMI: "))
```

```
if (icy > 32):  
    if(icy > 40 ):  
        print("大胖")  
    else:  
        print("中胖")  
elif (28 <= icy <= 32 ) :  
    if(30 <= icy <= 32) :  
        print("小胖")  
    else :  
        print("小小胖")  
else :  
    print("可以多吃一點沒關係")
```



if <condition> :  
 if<condition>:  
 <statement 1>

else :  
 <statement 2>

elif <condition> :  
 if<condition>:  
 <statement 3>

else :  
 <statement 4>

else:  
 <statementN>

# 隨堂練習

---

輸入以下數字，並判斷是否為 2, 3, 5, 7 的倍數或此數為質數

輸入	輸出
6 :	2 和 3 的倍數
4 :	2 的倍數
10:	2 和 5 的倍數
30:	2 、 3 以及 5 的倍數
210:	2 、 3 、 5 或 7 的倍數
17:	質數

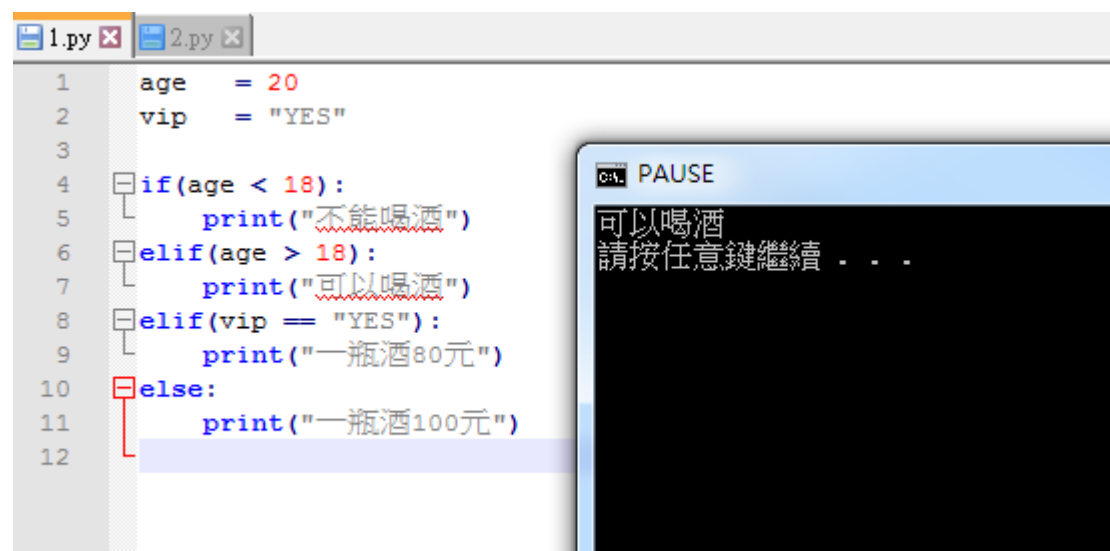


# 判斷

## 多重 if 判斷句

下列範例的問題是它只會執行其中一個判斷式

無法同時輸出多個判斷



```
1  age  = 20
2  vip  = "YES"
3
4  if (age < 18):
5      print("不能喝酒")
6  elif (age > 18):
7      print("可以喝酒")
8  elif (vip == "YES"):
9      print("一瓶酒80元")
10 else:
11     print("一瓶酒100元")
12
```

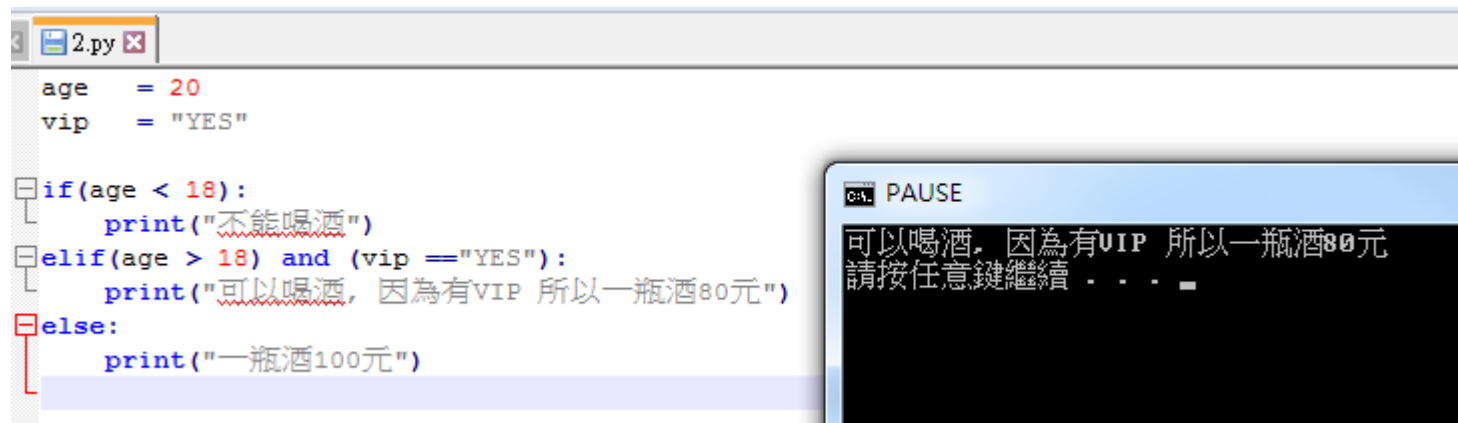
PAUSE

可以喝酒  
請按任意鍵繼續 . . .

# 判斷

## 多重 if 判斷句

所以我們可以用 `and` / `or` 來將多項條件彙整成單一條件：



The image shows a code editor window with a file named '2.py' containing the following Python code:

```
age = 20
vip = "YES"

if (age < 18):
    print("不能喝酒")
elif (age > 18) and (vip == "YES"):
    print("可以喝酒, 因為有VIP 所以一瓶酒80元")
else:
    print("一瓶酒100元")
```

Below the code editor, a terminal window displays the output of the script. The terminal shows a 'PAUSE' prompt, followed by the message '可以喝酒, 因為有VIP 所以一瓶酒80元' and a prompt to press any key to continue.

Any Questions !?