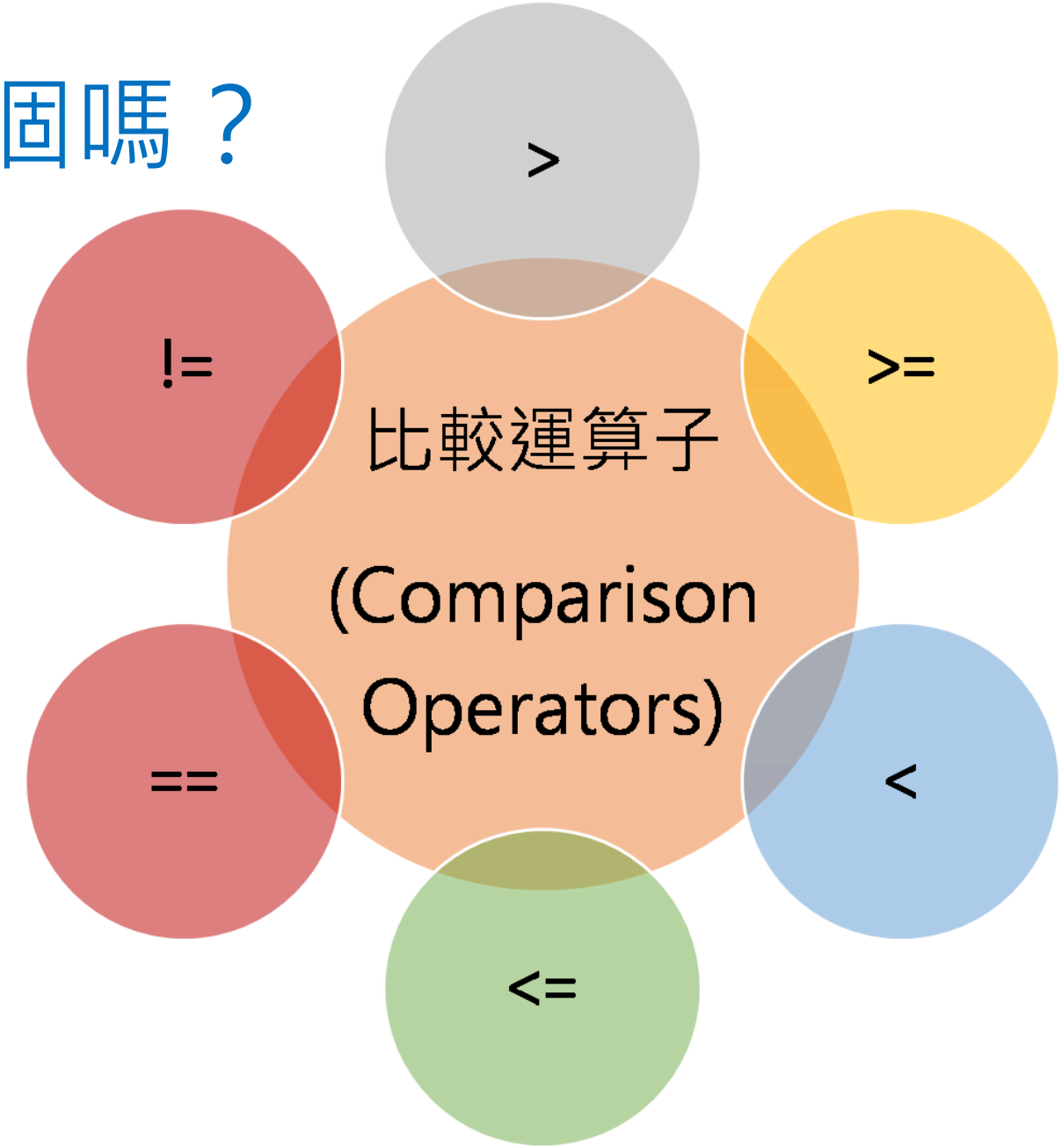
Python

判斷式-if

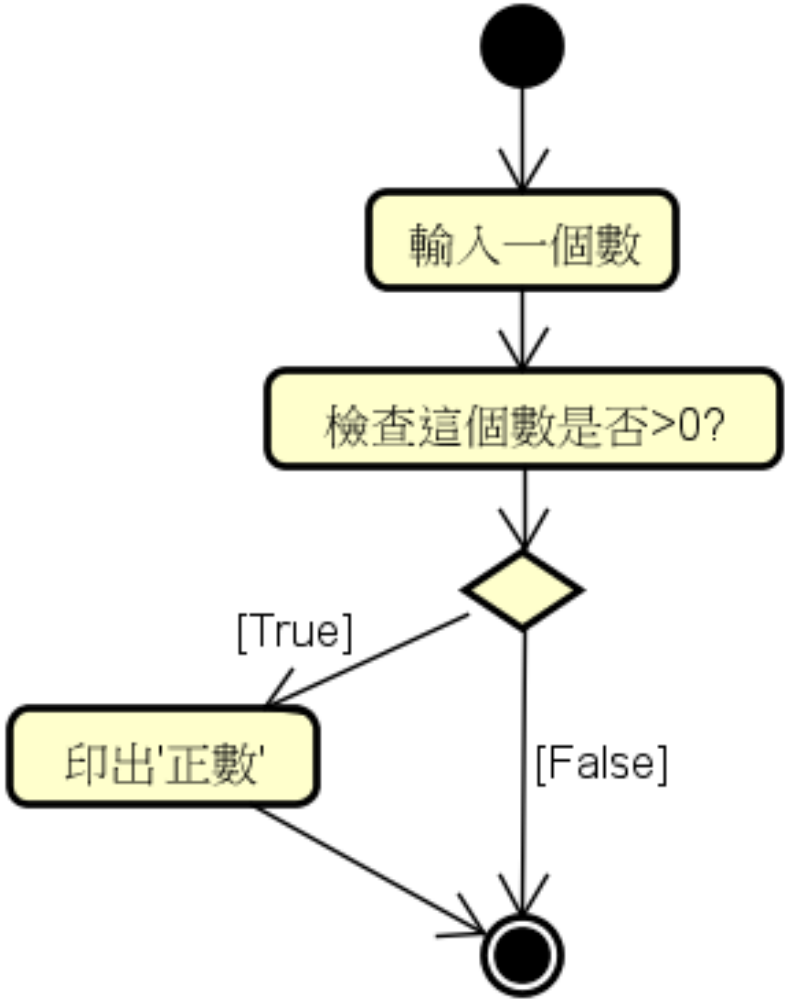

較量
才剛開始呢

還記得這個嗎？

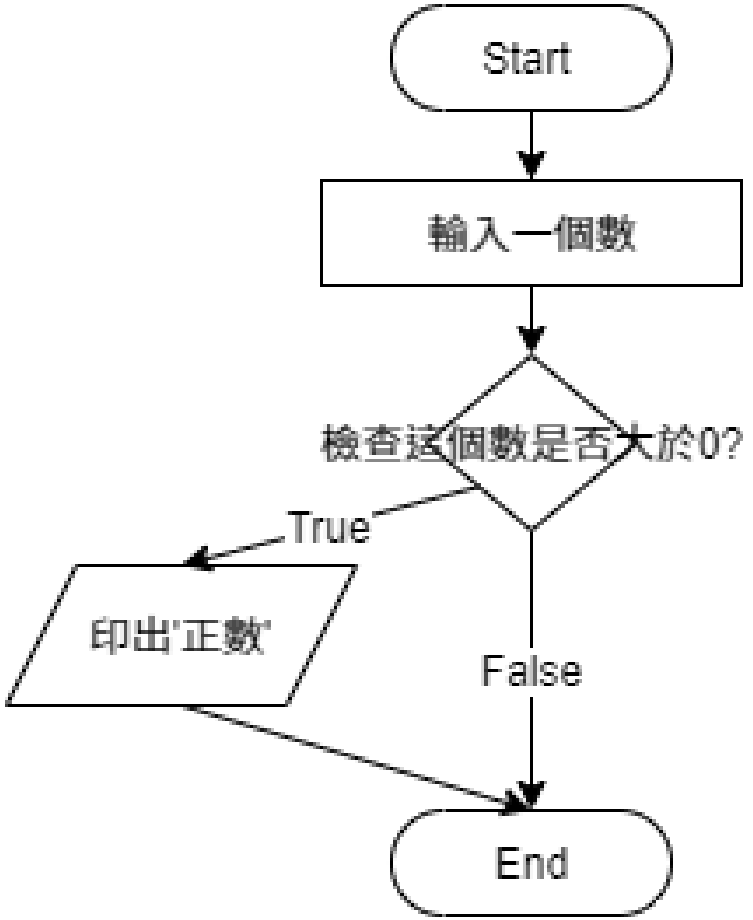


寫一個程式，
判斷一個數是正數嗎?

流程



UML流程圖



傳統流程圖

這是一個句子，
如果num>0印出正數。

```
num = int(input())

if num > 0:
        print("{num}是正數".format(num=num))
```

執行結果

```
22
22是正數
```

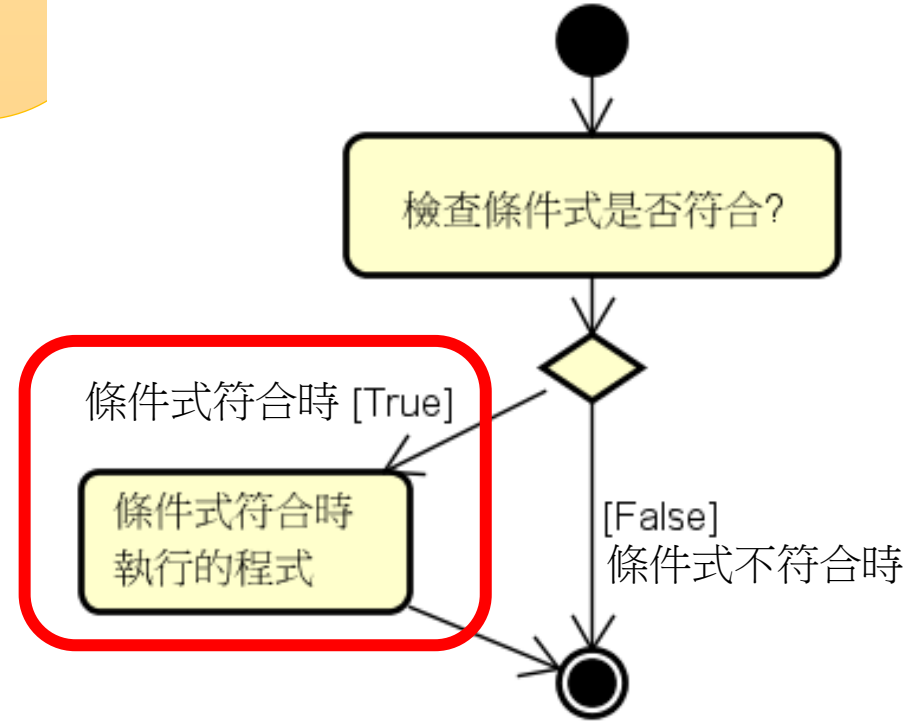
if 條件式

當 if 條件式符合時就
會執行紅框內的程式，
若不符合就不執行

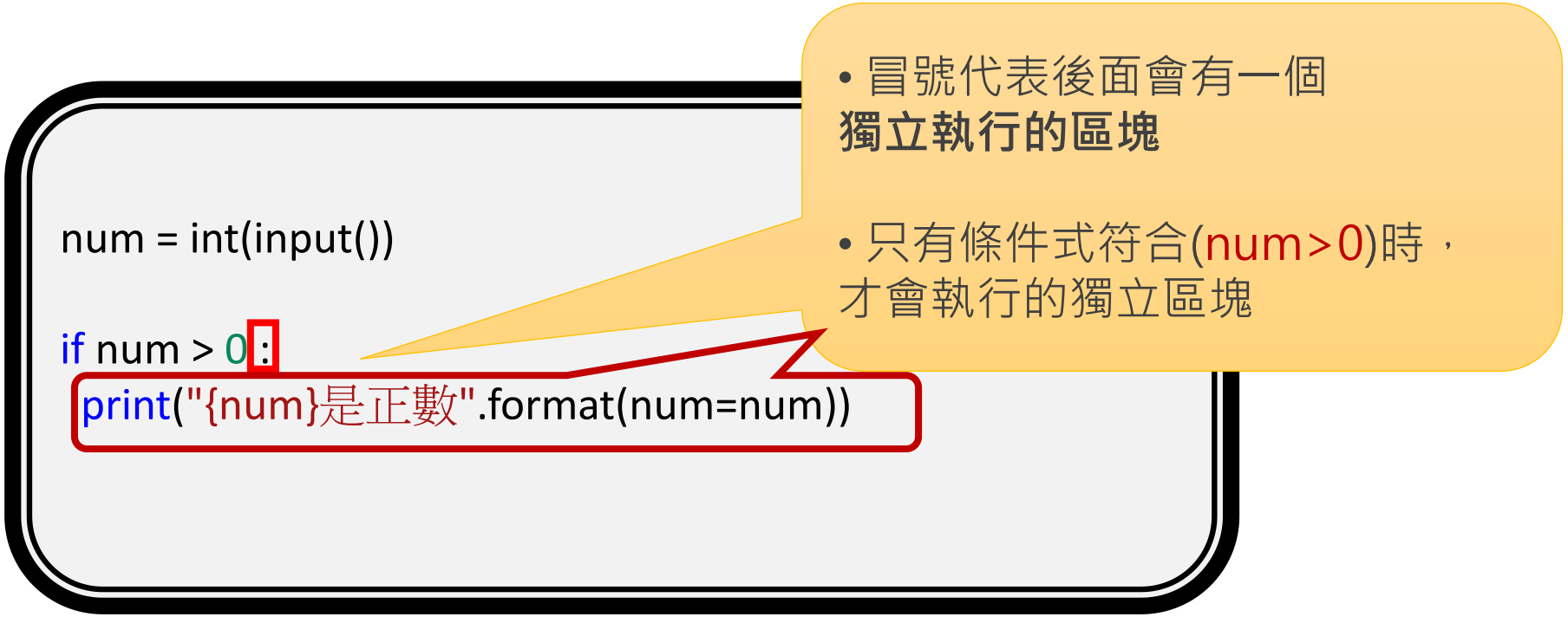
```
num=int(input())
If num>0
    print(“{0}是正數”.format(num))
```

If條件式

條件符合
時要執行



if條件式後面，為什麼有冒號?



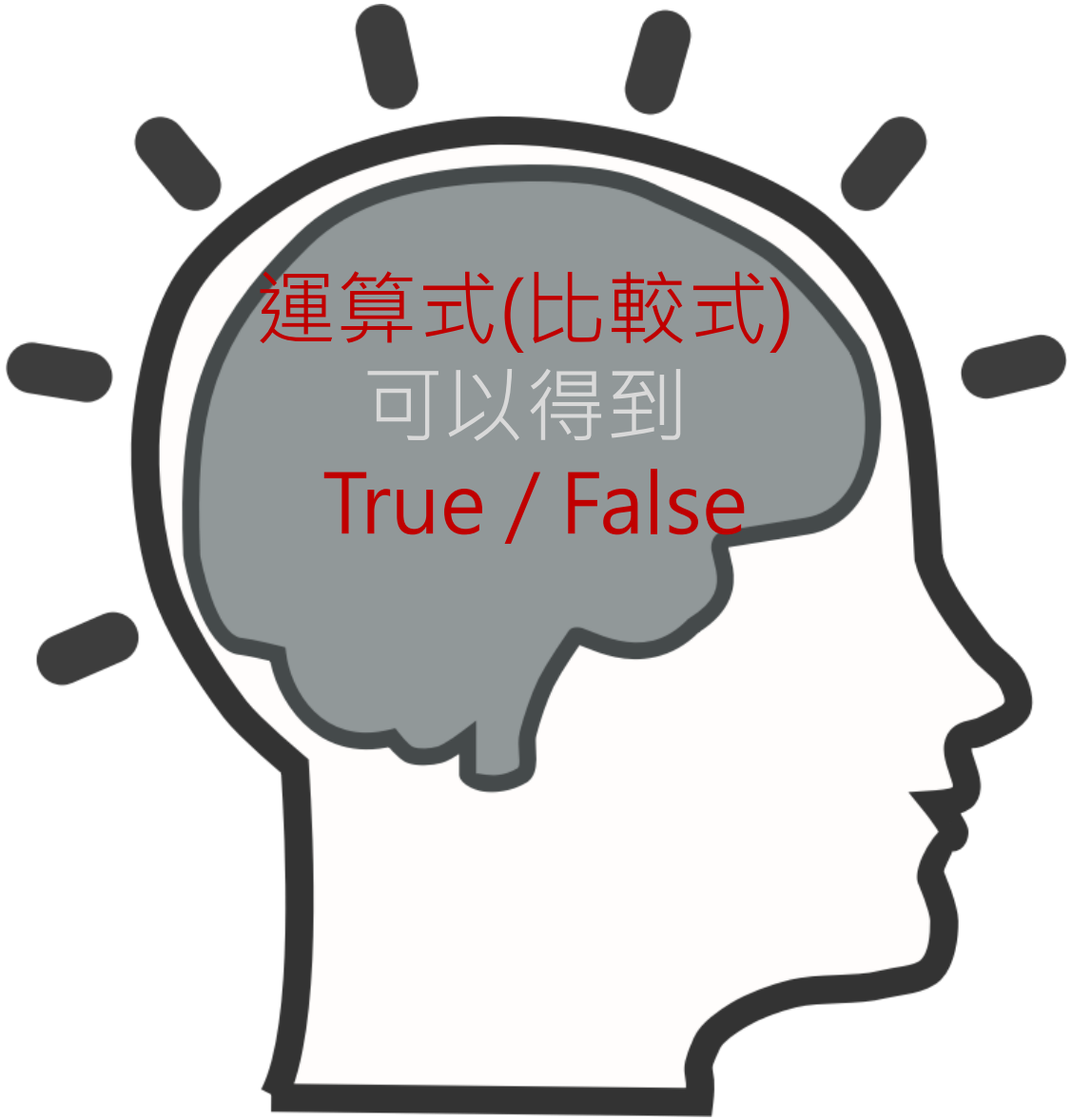
if所包含的敘述式，為什麼要縮排?

```
1  num = int(input())
2
3  if num > 0:
4      print("{0}是正數".format(num))
5      print("正數區塊")
6
```

為了區分冒號後面的程式碼與前面(包含 if 那行)式不同的執行區塊，必須設定縮排！

冒號下面是當條件式符合時執行的獨立區塊

Python 以縮排作為區塊依據，同一個區塊的程式碼，必須有相同的縮排(同樣的空格或tab數目)


運算式(比較式)
可以得到
True / False

其實 if 的語法是
if True / if False

忘記布爾(Bool)型態的可以複習一下<**比較是真假**>喔！

if 是根據後面的布爾(Bool)值決定是否執行。
- if 後面若是 True，就會執行接著的程式
- if 後面若是 False，就不會執行

```
if True:
        print('is True')

if False:
        print('is False')

print('is End')
```

執行結果

```
is True
is End
```

但 if True / if False 不太實用阿！

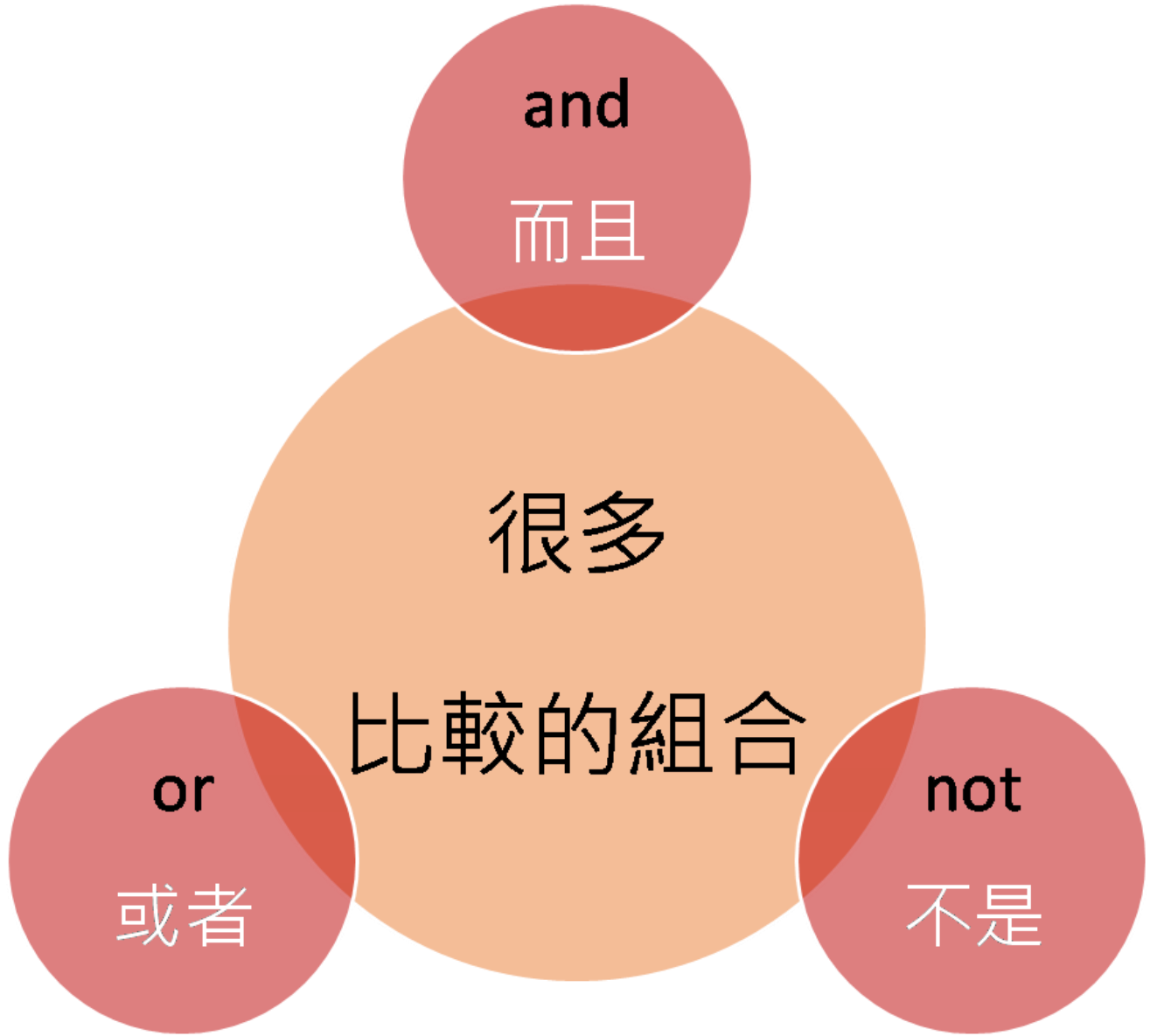
既然 True 跟 False 都已經是完全確定的值，那就沒有必要使用 if ，就可以直接簡化程式阿！

```
if True:
  print('is True')

if False:
  print('is False')

print('is End')
```

=

```
print('is True')


print('is End')
```

所以，一般使用時 if 後面接的會是**條件式**，當條件符合(True)時就執行，條件不符合(False)就不執行。


and
而且
很多
比較的組合
or
或者
not
不是

是的!
條件式大多是各種運算或是運算的組合

符號	數學意義	比較運算子
＝	等於	==
≠	不等於	!=
＜	小於	<
≦	小於等於	<=
＞	大於	>
≧	大於等於	>=

邏輯運算	意義	例子
and	前後兩者都要符合才是 True	[False] 3 > 5 and 5 < 10 [True] 3 < 5 and 5 < 10
or	前後兩者只要有一個符合就是 True	[True] 3 > 5 or 5 < 10 [True] 3 < 5 or 5 < 10
not	包住的條件不符合時就是 True	[True] not(3 > 5 and 5 < 10) [False] not(3 < 5 and 5 < 10)

忘記這幾個表格的，記得複習
一下<**比較-真假**>喔！

隸屬運算	意義	例子
is	判斷前後兩者是否是相同的東西	[False] 3 is 5 [True] 5 is 5
is not	判斷前後兩者是否是不相同的東西	[False] 5 is not 5 [True] 3 is not 5

身分運算	意義	例子
in	判斷前者是否包含在後者裡面	[True] 'str' in 'string' [False] 'str' in 'stiring'
not in	判斷前者是否不包含在後者裡面	[False] 'str' not in 'string' [True] 'str' not in 'stiring'

PYTHON

延伸的概念

概念 1:偷懶的比較

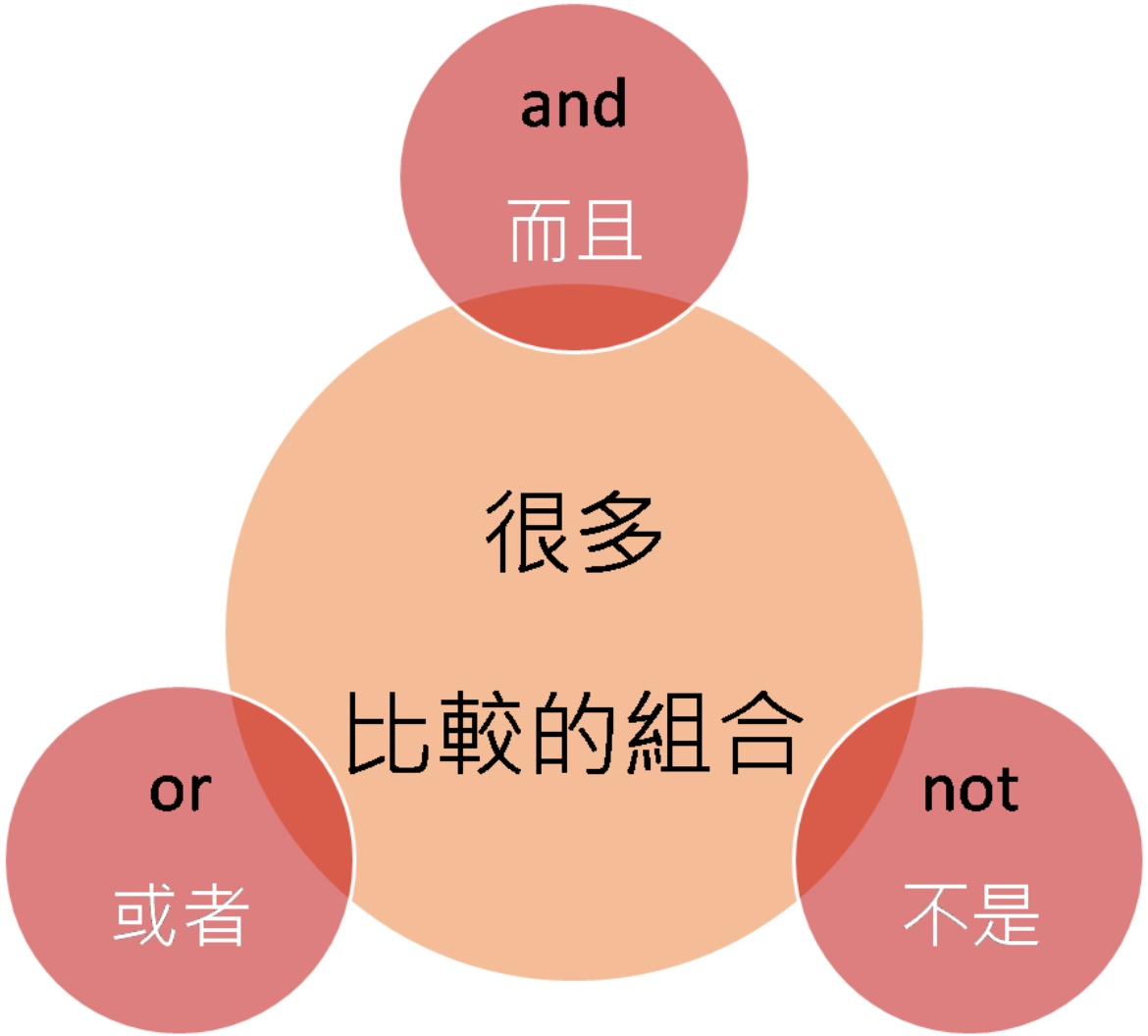
```
num1 = 30
num2 = 20

if num1 > num2:
  print("num1 is greater than
num2")
```

=

```
num1 = 30
num2 = 20

if num1 > num2: print("num1 is greater than
num2")
```

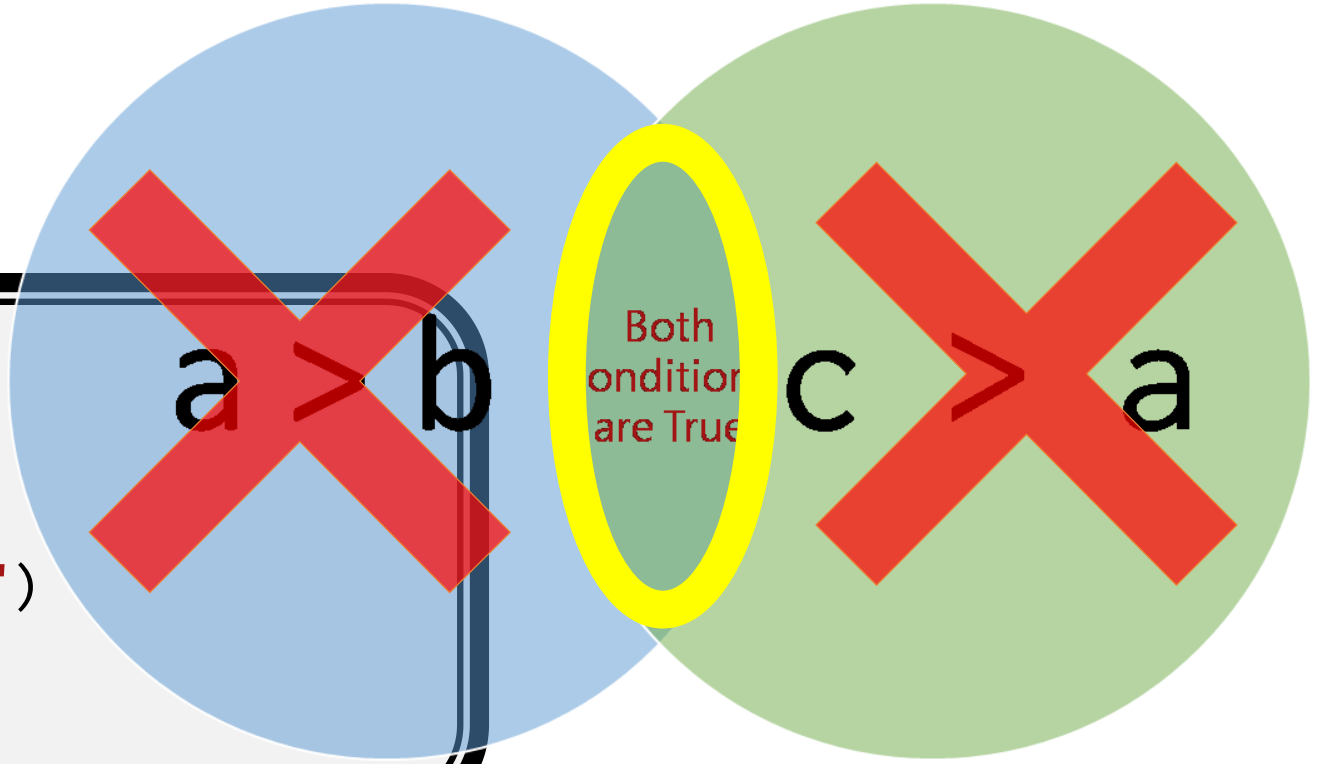
當 if 符合時要執行的程式碼只有一行時，可直接放到 if 後面，但若 if 後面要執行的有多行時，就不建議放在同一行，會讓程式碼比較難讀！

概念 2: 組合兩個以上的條件(1/4)



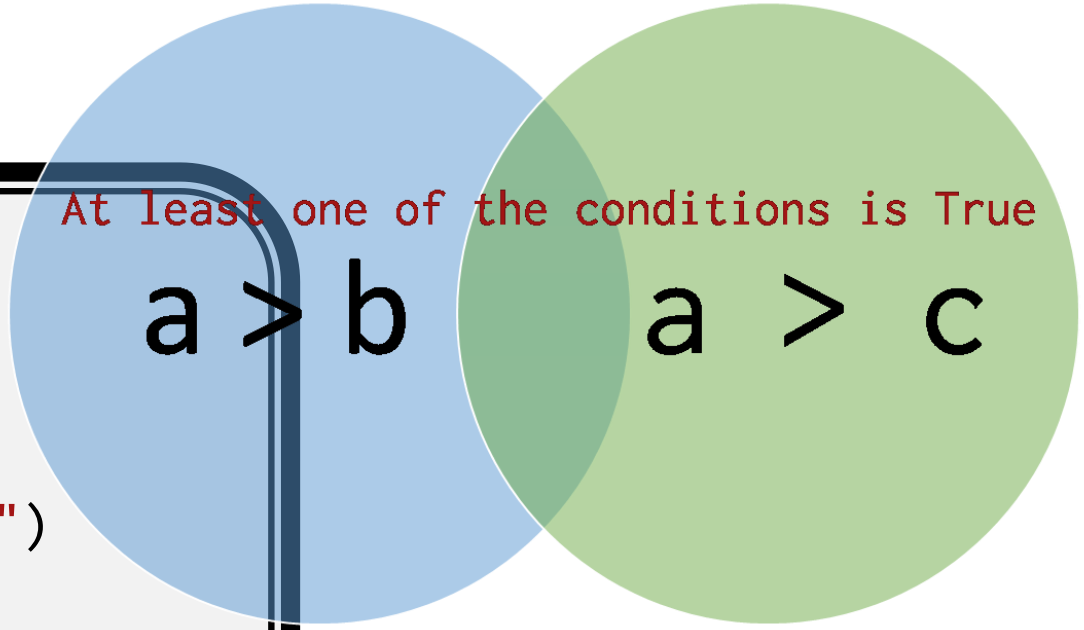
概念 2: 使用and組合兩個條件

```
if a > b and c > a:
 print("Both conditions are True")
```



概念 2:使用or組合兩個條件

```
if a > b or a > c:
 print("At least one of the conditions is True")
```



概念 2: not

```
1 a=int(input())
2 b=int(input())
3 print(not(a>b))
```

not(3>7)

```
楊靜怡[Python_10_5]$
3
7
True
楊靜怡[Python_10_5]$
```

```
1 a=int(input())
2 b=int(input())
3 print(not(a>b))
```

not(7>3)

```
楊靜怡[Python_10_5]
7
3
False
楊靜怡[Python_10_5]
```

概念 2: 玩玩not與or

```
if not(a > c) or a > b:
 print("非a>c或a>b其中一個條件要成立")
```