



巨匠直播教學

# APCS Python語法基礎班

## 流程控制

[www.pcschoolonline.com.tw](http://www.pcschoolonline.com.tw)

# 本堂教學重點

1. 關係/邏輯運算
2. 分支結構
3. 迴圈結構

# 課程內容

## 1. 關係與邏輯運算

1-1. 關係運算

1-2. 邏輯運算

## 2. 分支結構

2-1. 分支結構

2-2. 單行分支語法

## 3. 迴圈結構

3-1. Range 範圍

3-2. for 迴圈

3-3. while 迴圈

# 課程內容

## 1. 關係與邏輯運算

1-1. 關係運算

1-2. 邏輯運算

## 2. 分支結構

2-1. 分支結構

2-2. 單行分支語法

## 3. 迴圈結構

3-1. Range 範圍

3-2. for 迴圈

3-3. while 迴圈

# Python 分支及迴圈結構

## ◆ Python 分支及迴圈結構

◆ Python 不使用大括號限定區塊(block)範圍

◆ 使用縮排(空白)來顯示區塊範圍

- : 標示區塊(Block)開始
- 相同區塊範圍的程式碼使用相同縮排
  - ▶ 縮排方式必須一致
  - ▶ 使用四個空白字元的縮排與使用一個Tab 縮排不同

# 條件分析

◆ 程式中依資料或條件不同，採取不同的行動流程

◆ Boolean(bool) 布林值

- True
- False

◆ 關係運算

- 單一關係比較結果

◆ 邏輯運算

- 多個關係運算結果邏輯加總

# 關係運算符號

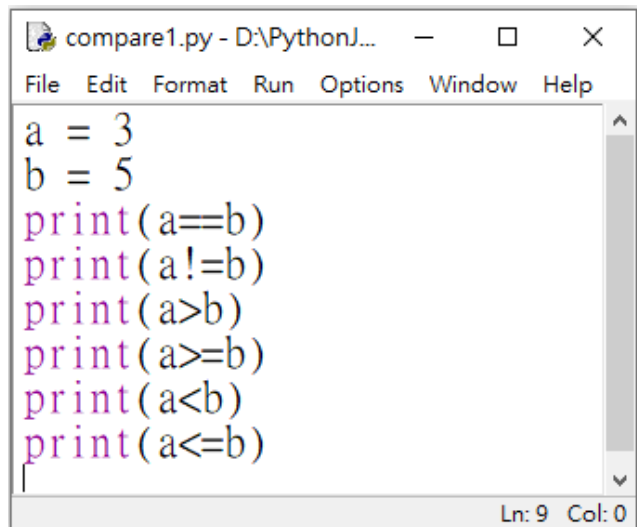
◆ 兩個資料進行關係比較分析的運算符號。

數值關係運算	意義
<	小於
<=	小於等於
>	大於
>=	大於等於
==	兩邊是否等於
!=	兩邊是否不等於

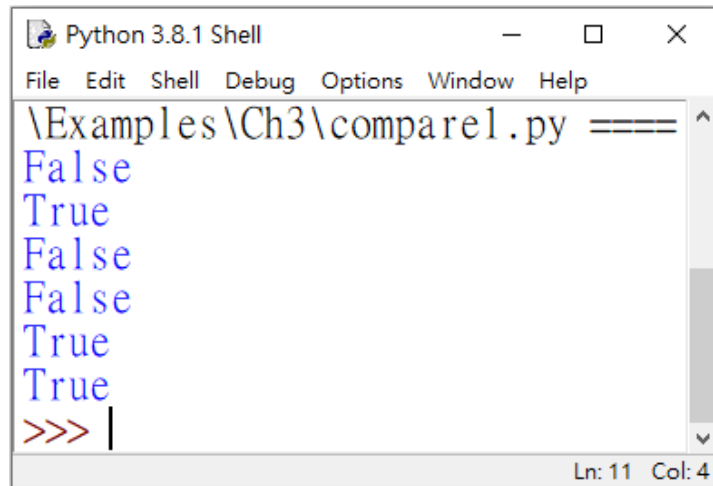
集合關係運算	意義
in	包含
not in	不包含

物件類別關係運算	意義
is	是否為同一個物件
is not	是否非同一個物件

# 程式範例



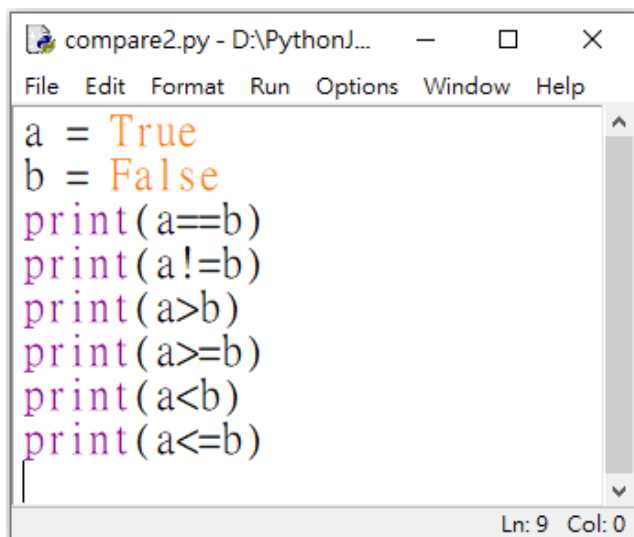
```
compare1.py - D:\PythonJ...  
File Edit Format Run Options Window Help  
a = 3  
b = 5  
print(a==b)  
print(a!=b)  
print(a>b)  
print(a>=b)  
print(a<b)  
print(a<=b)  
Ln: 9 Col: 0
```



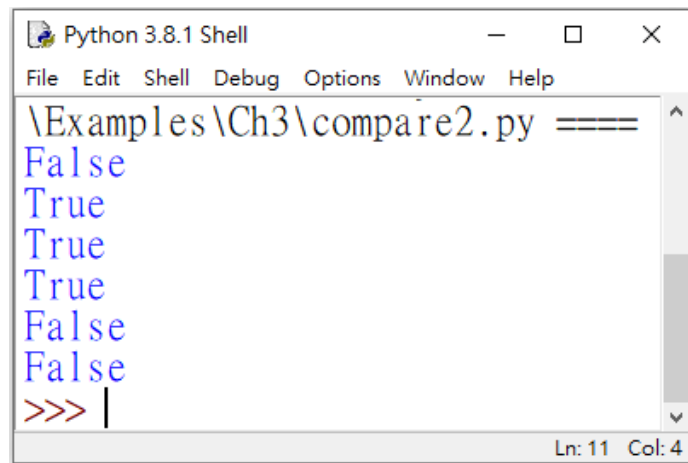
```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
\\Examples\\Ch3\\compare1.py ===  
False  
True  
False  
False  
True  
True  
>>> |  
Ln: 11 Col: 4
```



# 程式範例

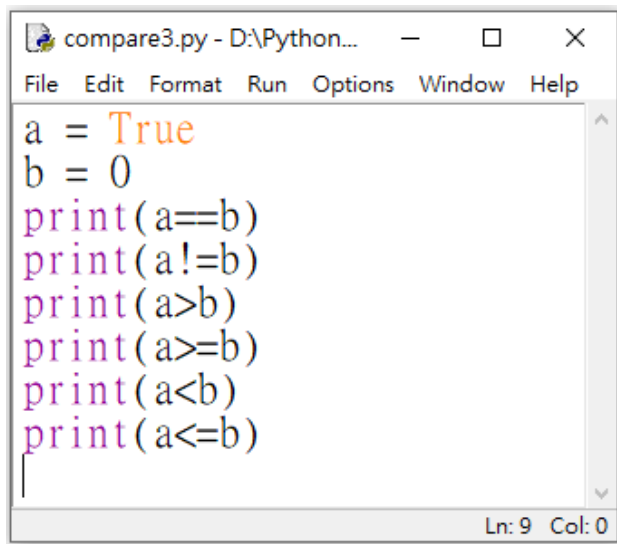


```
compare2.py - D:\PythonJ...  
File Edit Format Run Options Window Help  
a = True  
b = False  
print(a==b)  
print(a!=b)  
print(a>b)  
print(a>=b)  
print(a<b)  
print(a<=b)  
Ln: 9 Col: 0
```

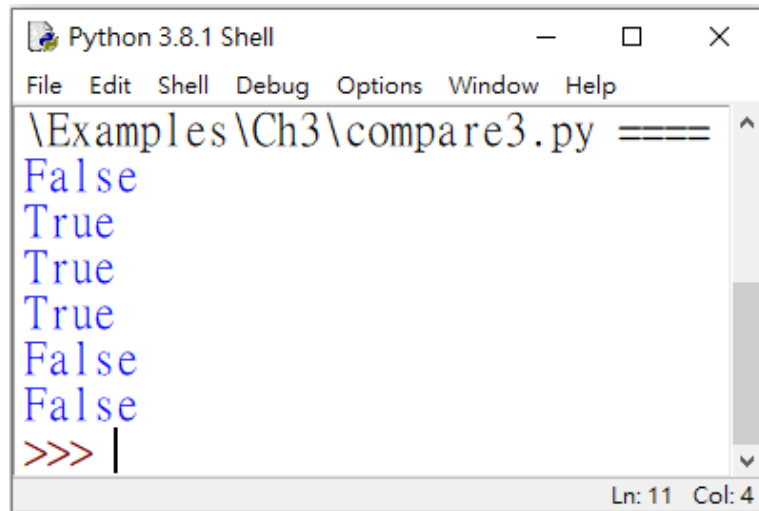


```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
\\Examples\\Ch3\\compare2.py ===  
False  
True  
True  
True  
False  
False  
>>> |  
Ln: 11 Col: 4
```

# 程式範例

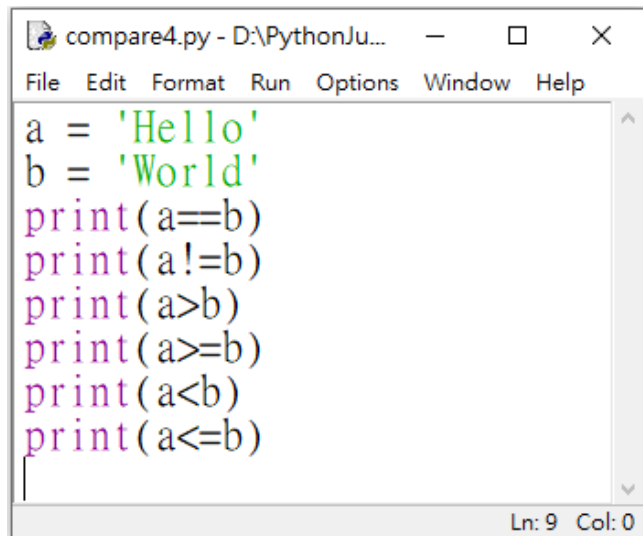


```
compare3.py - D:\Python...  
File Edit Format Run Options Window Help  
a = True  
b = 0  
print(a==b)  
print(a!=b)  
print(a>b)  
print(a>=b)  
print(a<b)  
print(a<=b)  
Ln: 9 Col: 0
```

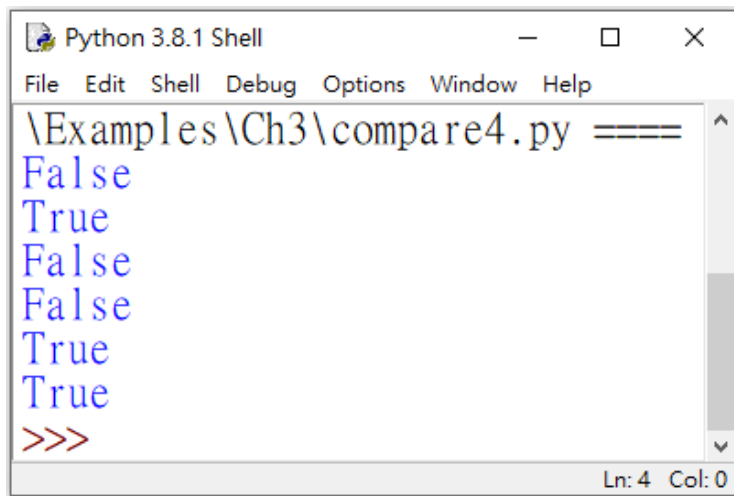


```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
\\Examples\\Ch3\\compare3.py ===  
False  
True  
True  
True  
False  
False  
>>> |  
Ln: 11 Col: 4
```

# 程式範例

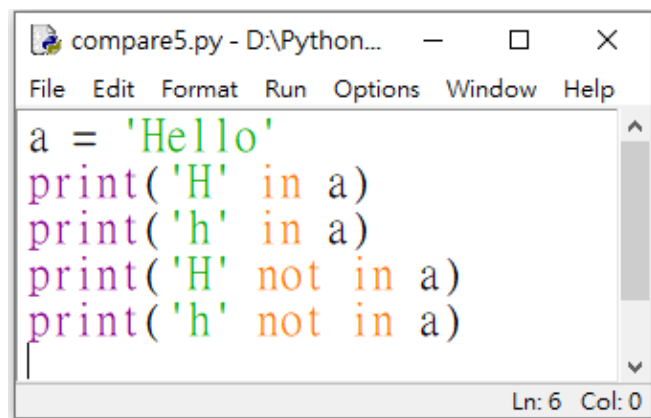


```
compare4.py - D:\PythonJu...  
File Edit Format Run Options Window Help  
a = 'Hello'  
b = 'World'  
print(a==b)  
print(a!=b)  
print(a>b)  
print(a>=b)  
print(a<b)  
print(a<=b)  
Ln: 9 Col: 0
```

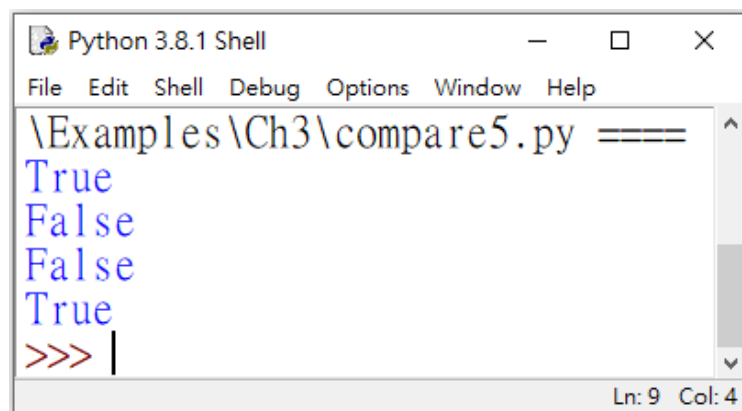


```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
\\Examples\\Ch3\\compare4.py ===  
False  
True  
False  
False  
True  
True  
>>>  
Ln: 4 Col: 0
```

# 程式範例

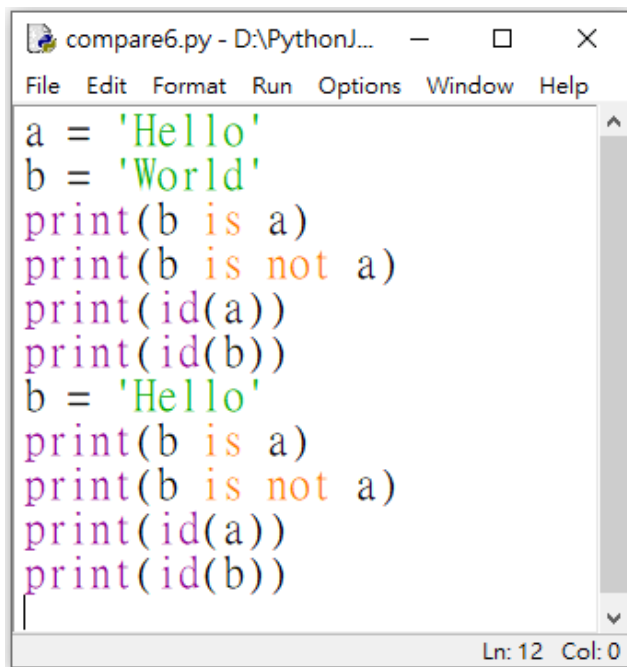


```
compare5.py - D:\Python...  
File Edit Format Run Options Window Help  
a = 'Hello'  
print('H' in a)  
print('h' in a)  
print('H' not in a)  
print('h' not in a)  
Ln: 6 Col: 0
```

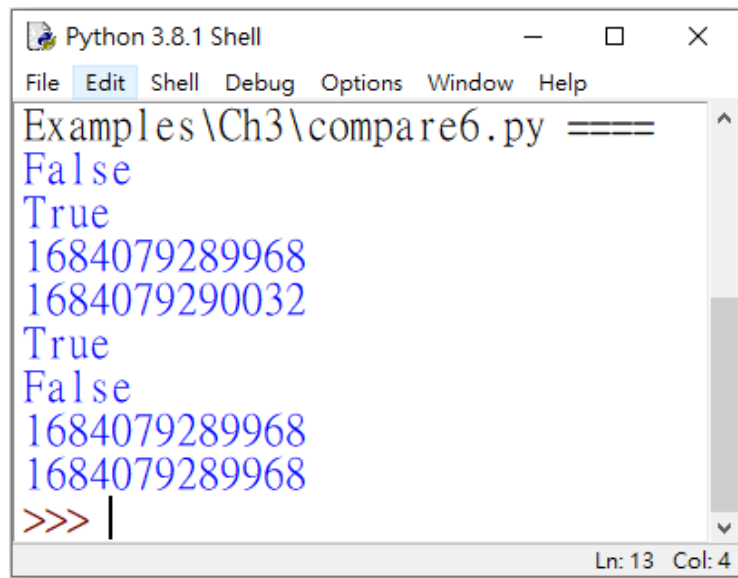


```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
\Examples\Ch3\compare5.py ===  
True  
False  
False  
True  
>>> |  
Ln: 9 Col: 4
```

# 程式範例



```
compare6.py - D:\PythonJ...  
File Edit Format Run Options Window Help  
a = 'Hello'  
b = 'World'  
print(b is a)  
print(b is not a)  
print(id(a))  
print(id(b))  
b = 'Hello'  
print(b is a)  
print(b is not a)  
print(id(a))  
print(id(b))  
Ln: 12 Col: 0
```



```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
Examples\Ch3\compare6.py ===  
False  
True  
1684079289968  
1684079290032  
True  
False  
1684079289968  
1684079289968  
>>> |  
Ln: 13 Col: 4
```

# 邏輯運算符號

◆ 兩個 Boolean 以特定邏輯規則加總的結果

AND	true	false
true	true	false
false	false	false

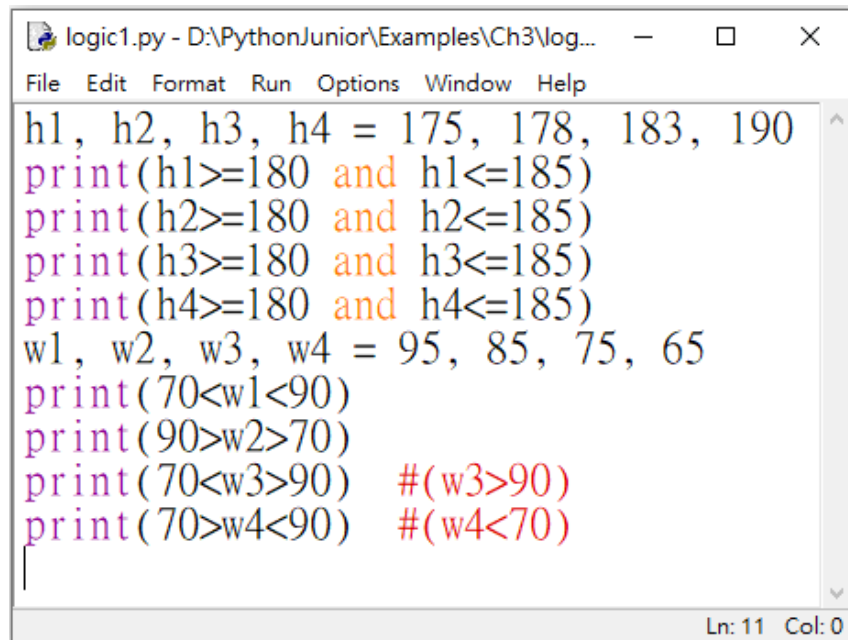
OR	true	false
true	true	true
false	true	false

XOR	true	false
true	false	true
false	true	false

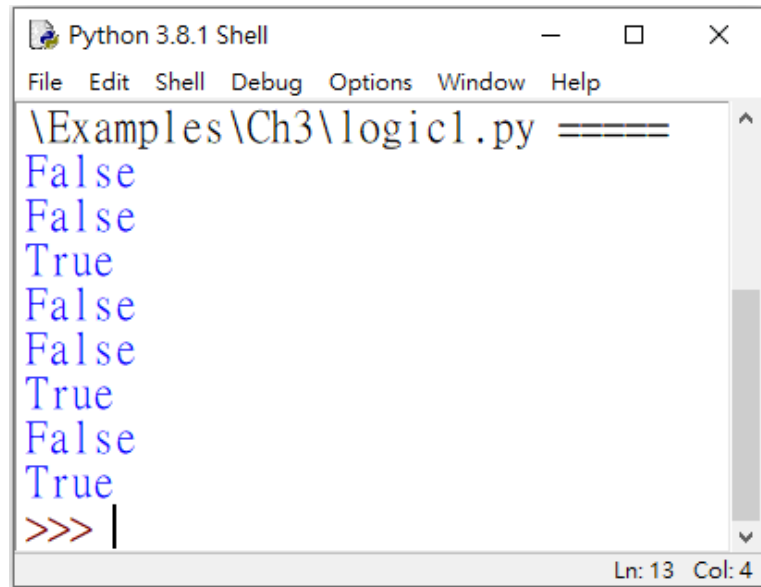
NOT	
true	false
false	true

邏輯關係運算	意義	運算子類別
and    &	且	二元運算
or	或	二元運算
^	XOR 互斥	二元運算
not	非	一元運算

# 程式範例

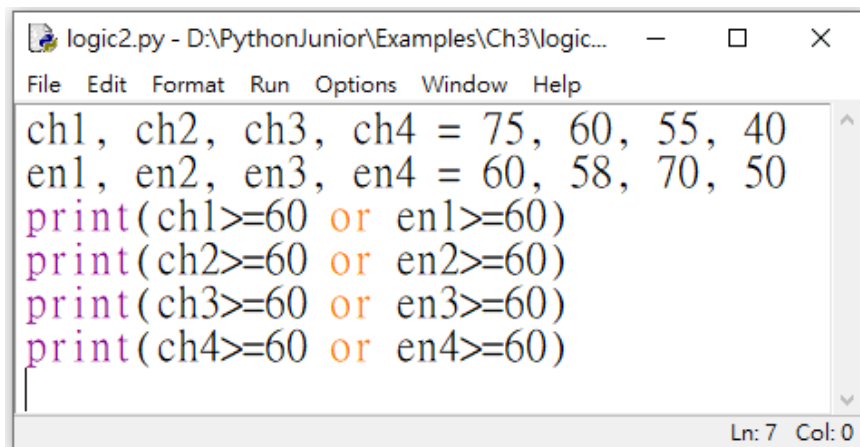


```
logic1.py - D:\PythonJunior\Examples\Ch3\log...
File Edit Format Run Options Window Help
h1, h2, h3, h4 = 175, 178, 183, 190
print(h1>=180 and h1<=185)
print(h2>=180 and h2<=185)
print(h3>=180 and h3<=185)
print(h4>=180 and h4<=185)
w1, w2, w3, w4 = 95, 85, 75, 65
print(70<w1<90)
print(90>w2>70)
print(70<w3>90)    #(w3>90)
print(70>w4<90)    #(w4<70)
Ln: 11 Col: 0
```

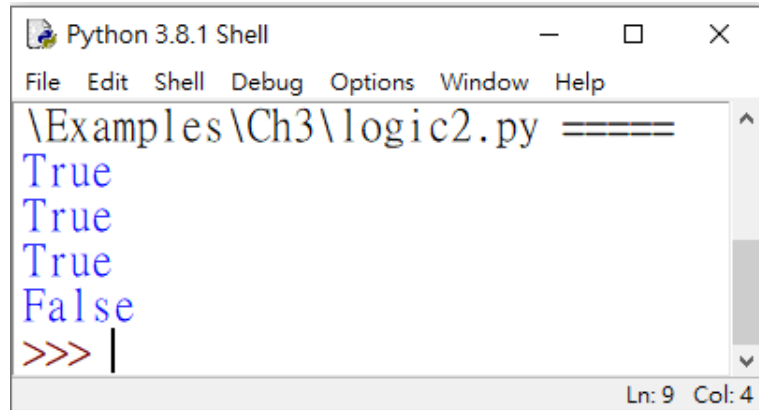


```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
\\Examples\\Ch3\\logic1.py =====
False
False
True
False
False
True
False
True
>>> |
Ln: 13 Col: 4
```

# 程式範例



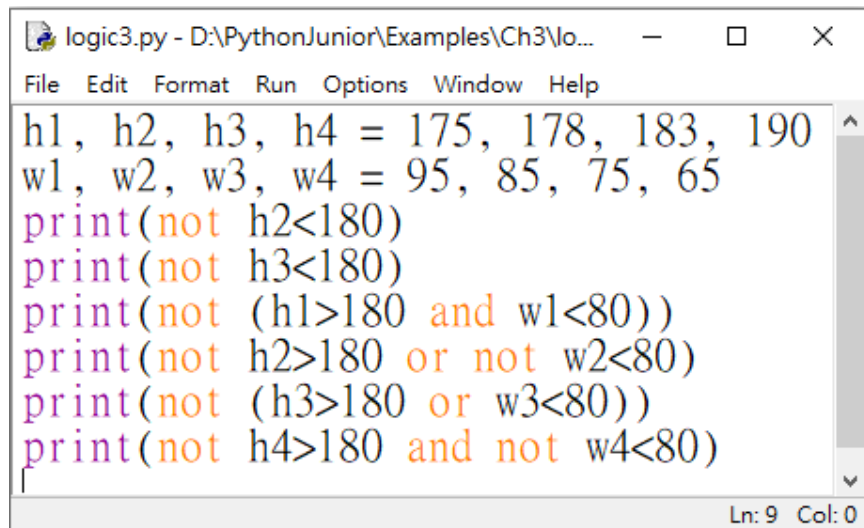
```
logic2.py - D:\PythonJunior\Examples\Ch3\logic...  
File Edit Format Run Options Window Help  
ch1, ch2, ch3, ch4 = 75, 60, 55, 40  
en1, en2, en3, en4 = 60, 58, 70, 50  
print(ch1>=60 or en1>=60)  
print(ch2>=60 or en2>=60)  
print(ch3>=60 or en3>=60)  
print(ch4>=60 or en4>=60)  
Ln: 7 Col: 0
```



```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
\\Examples\\Ch3\\logic2.py =====  
True  
True  
True  
False  
>>> |  
Ln: 9 Col: 4
```

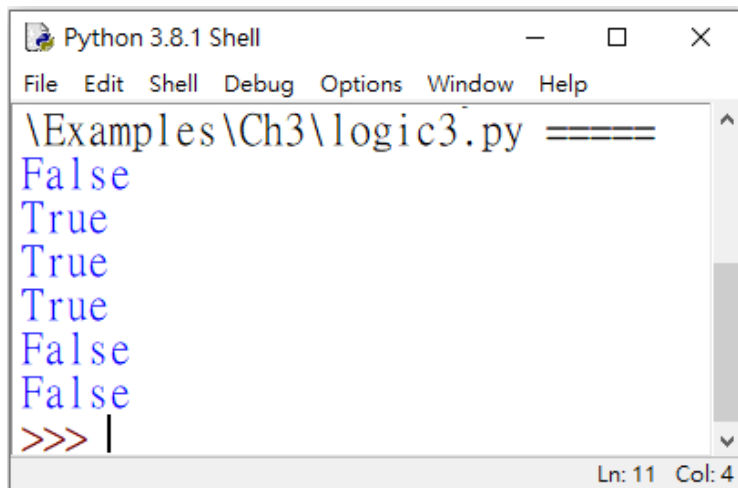


# 程式範例



```
h1, h2, h3, h4 = 175, 178, 183, 190
w1, w2, w3, w4 = 95, 85, 75, 65
print(not h2<180)
print(not h3<180)
print(not (h1>180 and w1<80))
print(not h2>180 or not w2<80)
print(not (h3>180 or w3<80))
print(not h4>180 and not w4<80)
```

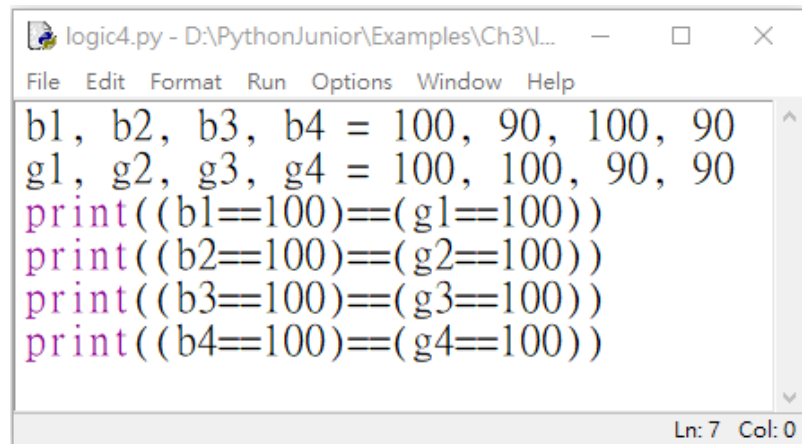
Ln: 9 Col: 0



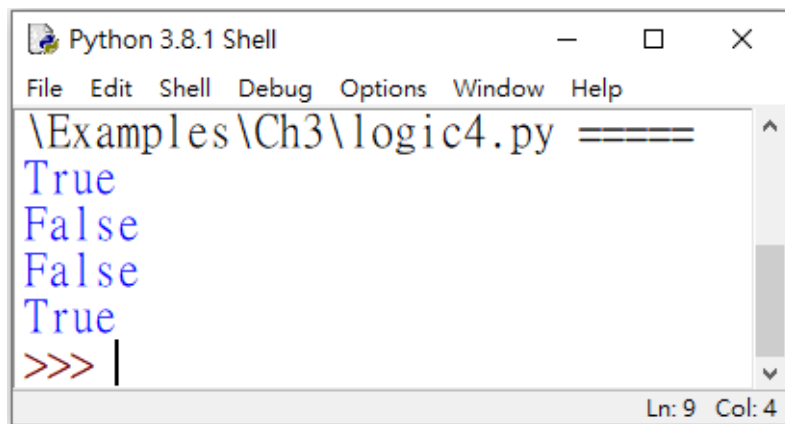
```
\Examples\Ch3\logic3.py =====
False
True
True
True
False
False
>>> |
```

Ln: 11 Col: 4

# 程式範例



```
logic4.py - D:\PythonJunior\Examples\Ch3\...  
File Edit Format Run Options Window Help  
b1, b2, b3, b4 = 100, 90, 100, 90  
g1, g2, g3, g4 = 100, 100, 90, 90  
print((b1==100)==(g1==100))  
print((b2==100)==(g2==100))  
print((b3==100)==(g3==100))  
print((b4==100)==(g4==100))  
Ln: 7 Col: 0
```



```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
\\Examples\\Ch3\\logic4.py =====  
True  
False  
False  
True  
>>> |  
Ln: 9 Col: 4
```

## Q : Python使用哪種方式表示程式區塊？

- a) 大括號 { }
- b) 冒號( : ) 及縮排
- c) 分號( ; )
- d) 井字號( # )

**Q：**下列何者為Python不等於關係運算子？

a)  $\geq$

b)  $<>$

c)  $\leq$

d)  $\neq$

**Q：**下列Python關係運算何者為正確？

- a) 等於關係使用 " = "
- b) And 關係使用 " && "
- c)  $(10 < 8)$  or  $(10 < 20)$  結果為True
- d)  $(10 > 8)$  and  $(10 > 20)$  結果為True

# 課程內容

## 1. 關係與邏輯運算

1-1. 關係運算

1-2. 邏輯運算

## 2. 分支結構

2-1. 分支結構

2-2. 單行分支語法

## 3. 迴圈結構

3-1. Range 範圍

3-2. for 迴圈

3-3. while 迴圈

# Python 分支結構

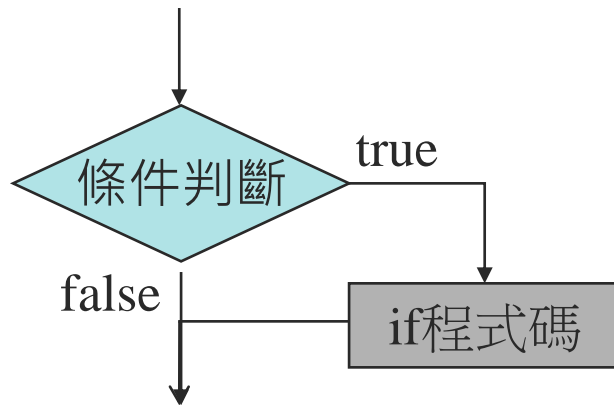
## ◆ Python 分支結構

### ◆ 單一選擇 if 結構

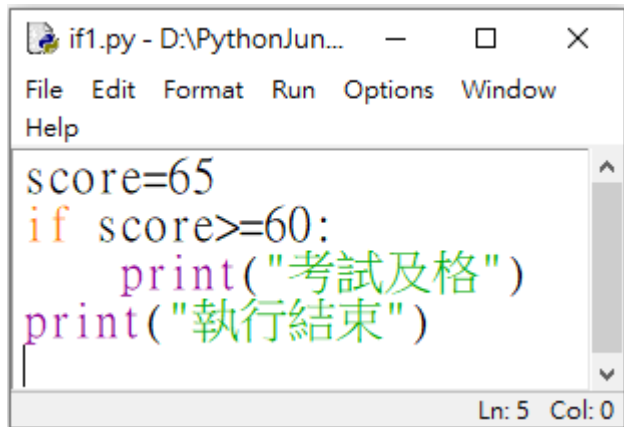
if 條件判斷式：

指令敘述

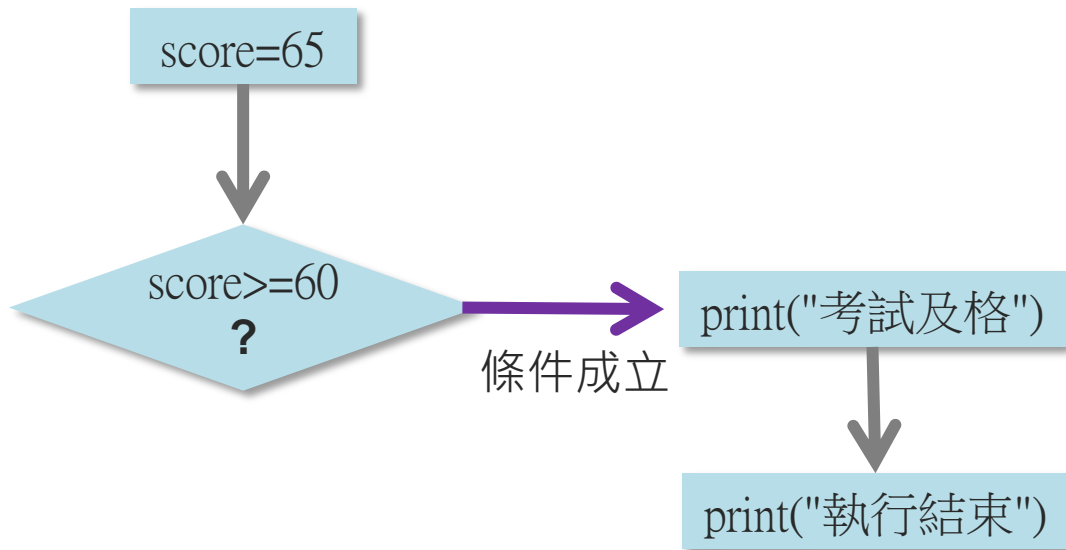
指令敘述



# 程式範例

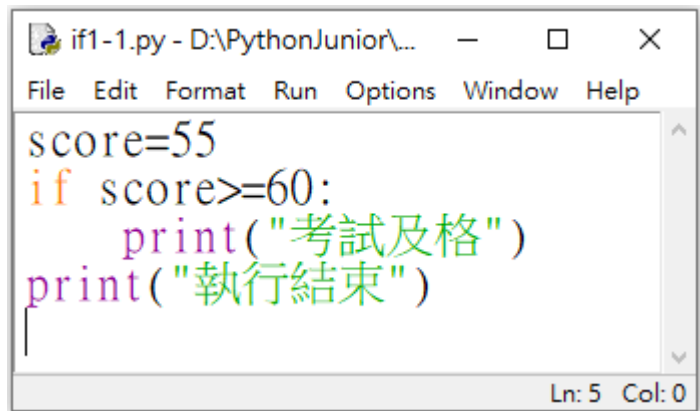


```
if1.py - D:\PythonJun...  -  □  X
File Edit Format Run Options Window Help
score=65
if score>=60:
    print("考試及格")
print("執行結束")
Ln: 5 Col: 0
```

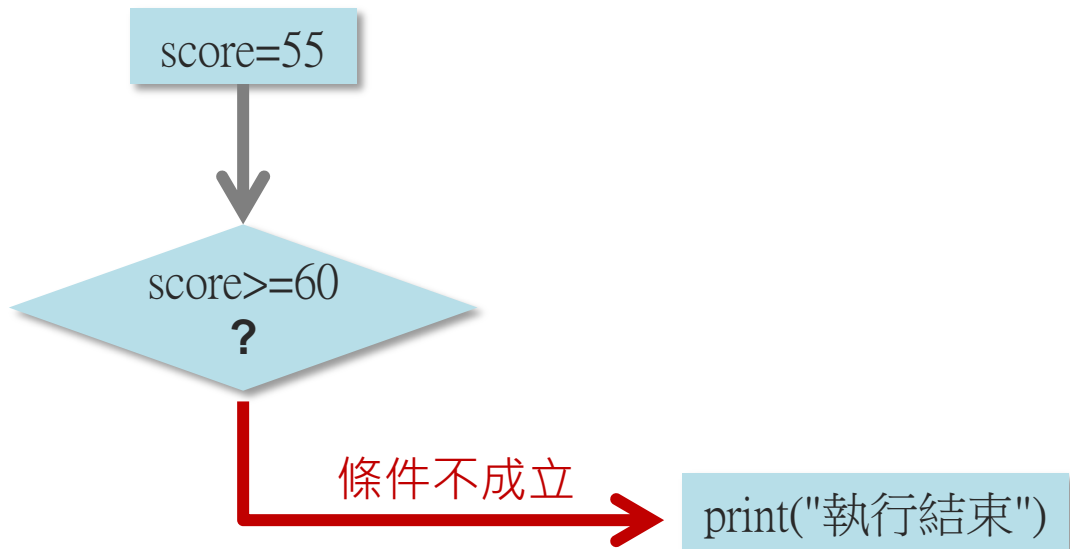




# 程式範例



```
if1-1.py - D:\PythonJunior\...  
File Edit Format Run Options Window Help  
score=55  
if score>=60:  
    print("考試及格")  
print("執行結束")  
Ln: 5 Col: 0
```



# Python 分支結構

## ◆ Python 分支結構

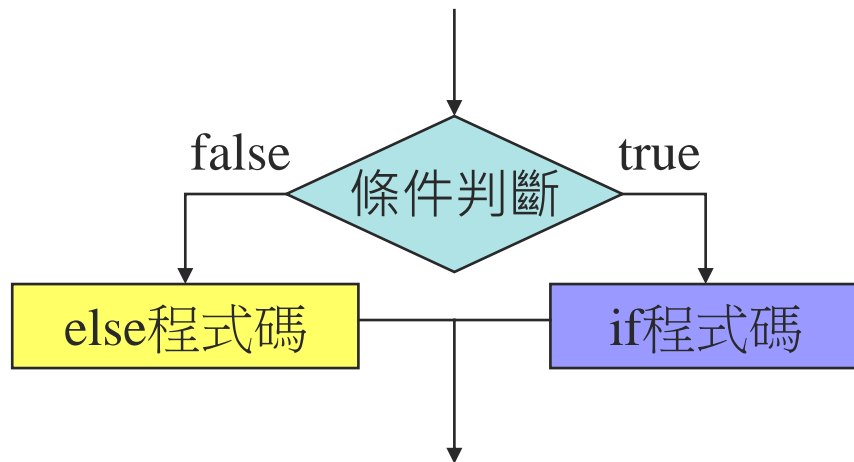
### ◆ 雙向選擇 if-else 結構

if 條件判斷式：

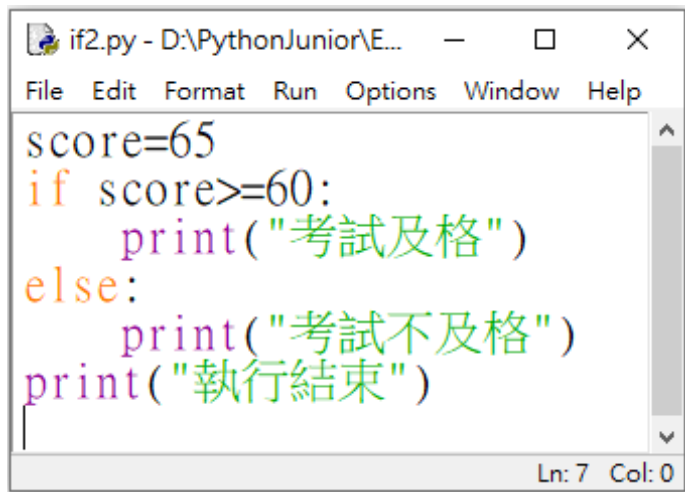
指令敘述

else：

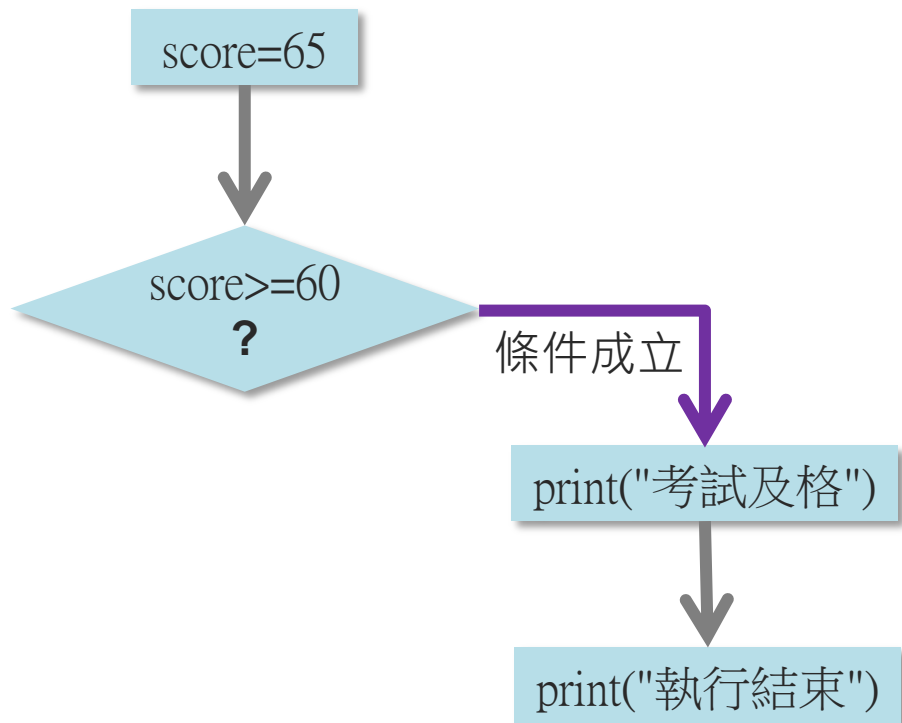
指令敘述



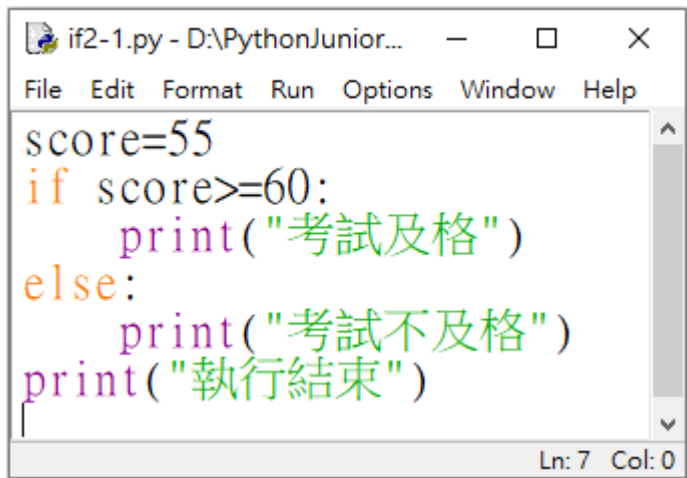
# 程式範例



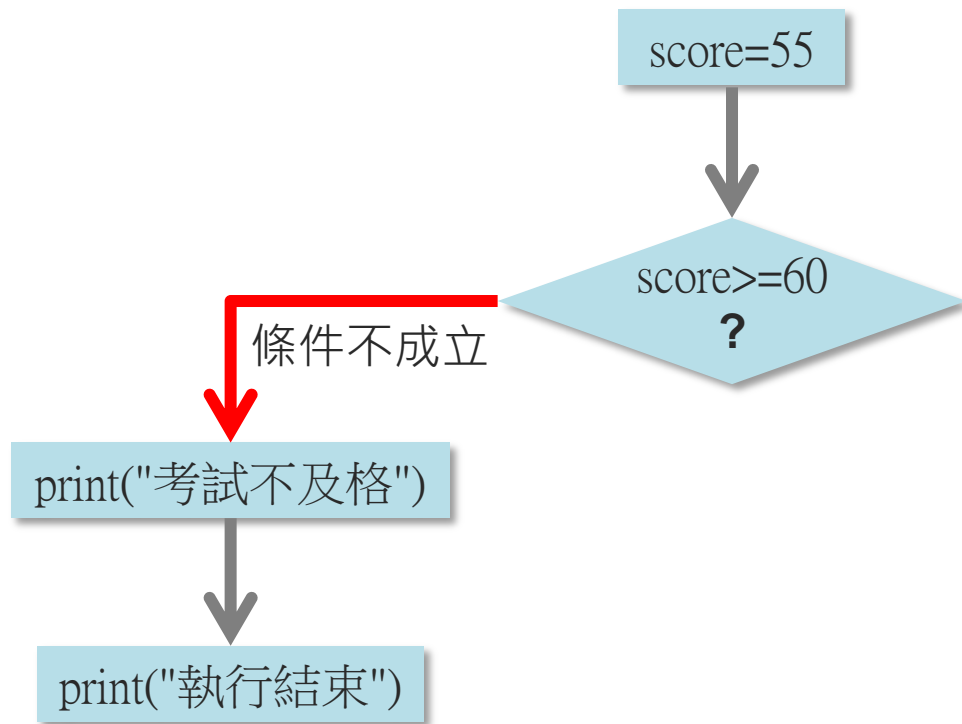
```
if2.py - D:\PythonJunior\E...  -  □  ×  
File Edit Format Run Options Window Help  
score=65  
if score>=60:  
    print("考試及格")  
else:  
    print("考試不及格")  
print("執行結束")  
Ln: 7 Col: 0
```



# 程式範例



```
if2-1.py - D:\PythonJunior...  
File Edit Format Run Options Window Help  
score=55  
if score>=60:  
    print("考試及格")  
else:  
    print("考試不及格")  
print("執行結束")  
Ln: 7 Col: 0
```



# Python 分支結構

## ◆ Python 分支結構

### ◆ 多重分支 if-elif-else 結構

**if** 條件式 :

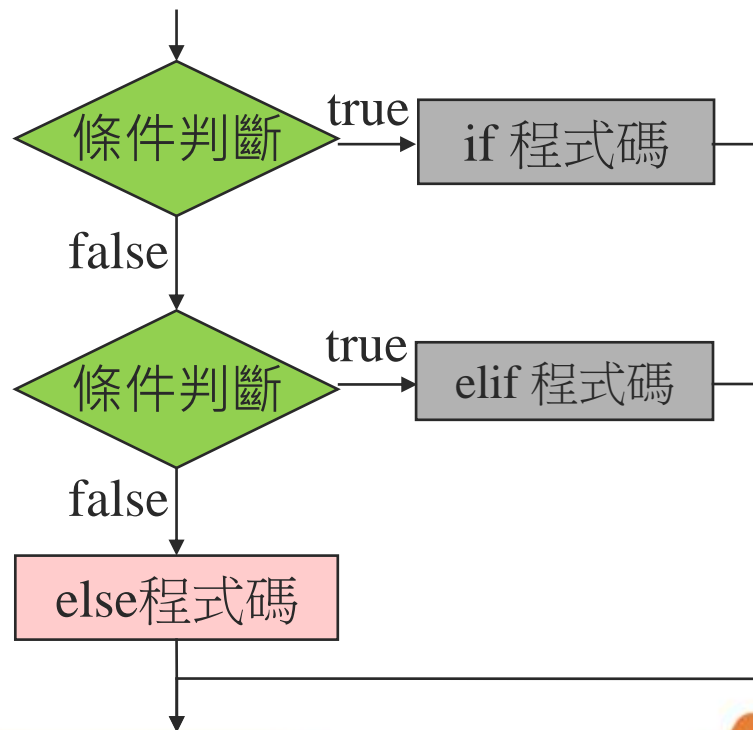
指令敘述

**elif** 條件式 :

指令敘述

**else** :

指令敘述



# Python 分支結構

## ◆ 多重分支 if-elif-else 結構

### ◆ if 或 elif 判斷成立

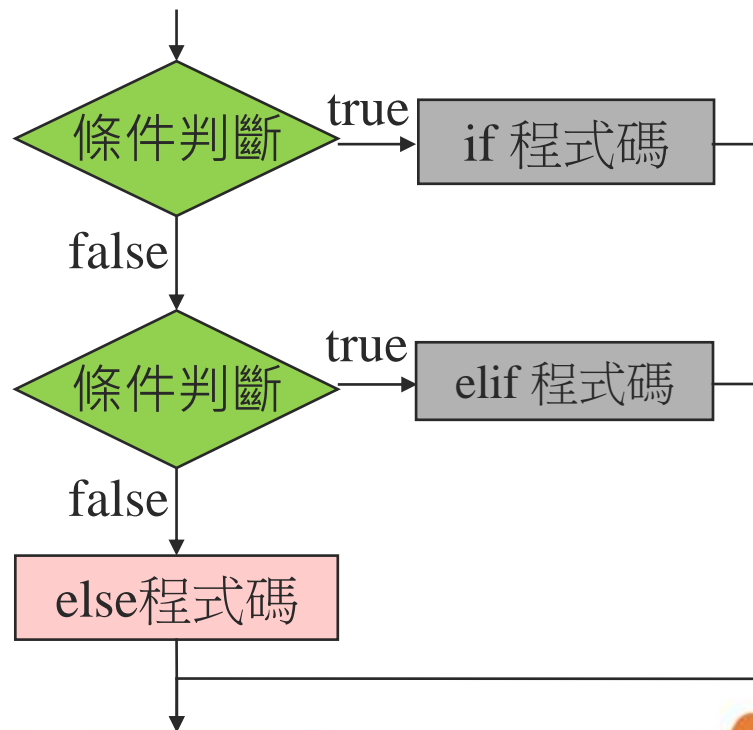
- 不執行下一個條件分析

### ◆ if 或 elif 不成立

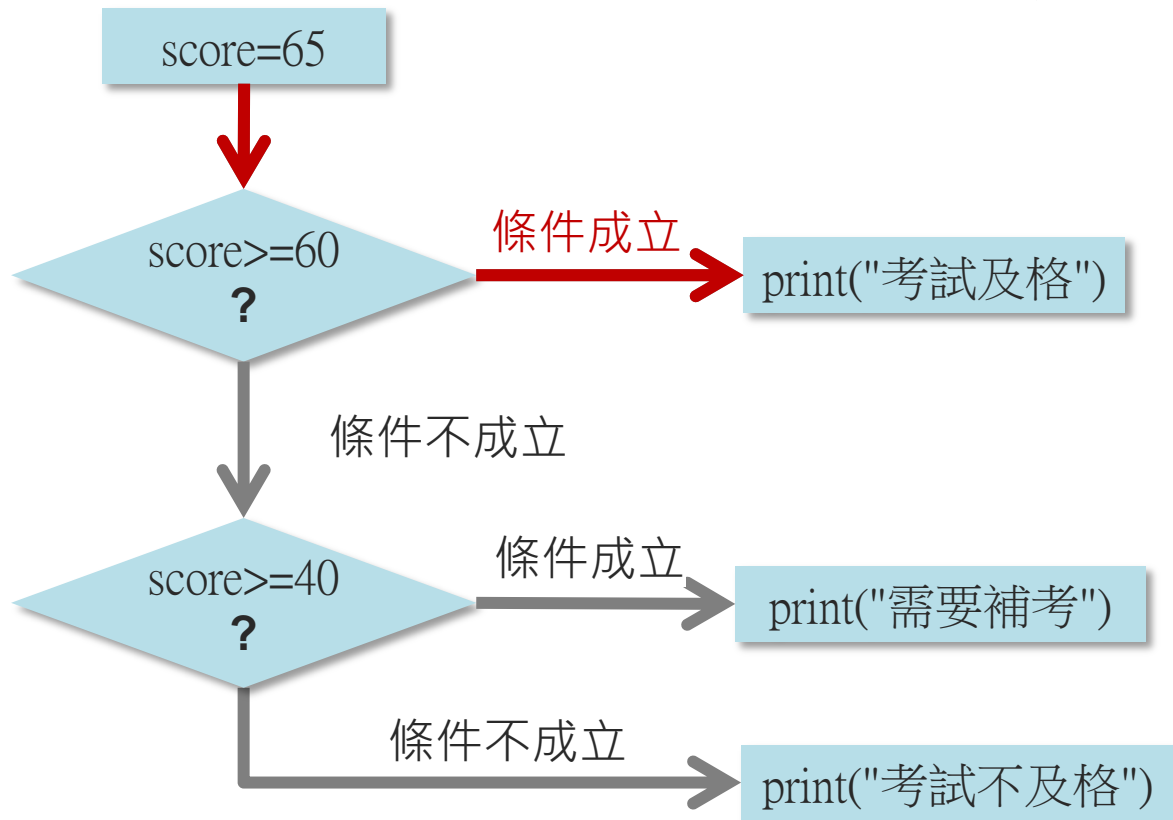
- 才會執行下一個 elif 分析

## ◆ Python沒有switch結構

### ◆ 使用if-elif-else達到相同效果



# 程式範例

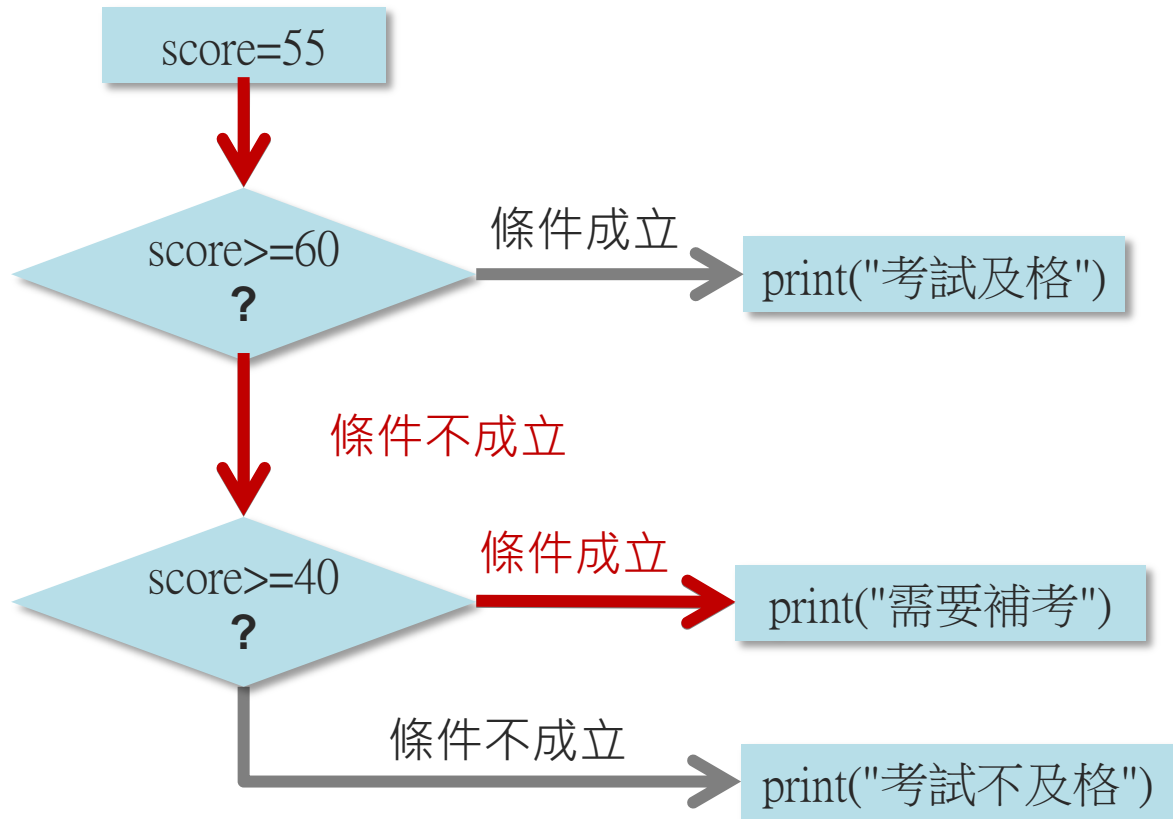


```
if3.py - D:\PythonJunior\Exempl...
File Edit Format Run Options Window Help

score=65
if score>=60:
    print("考試及格")
elif score>=40:
    print("需要補考")
else:
    print("考試不及格")
print("執行結束")

Ln: 9 Col: 0
```

# 程式範例



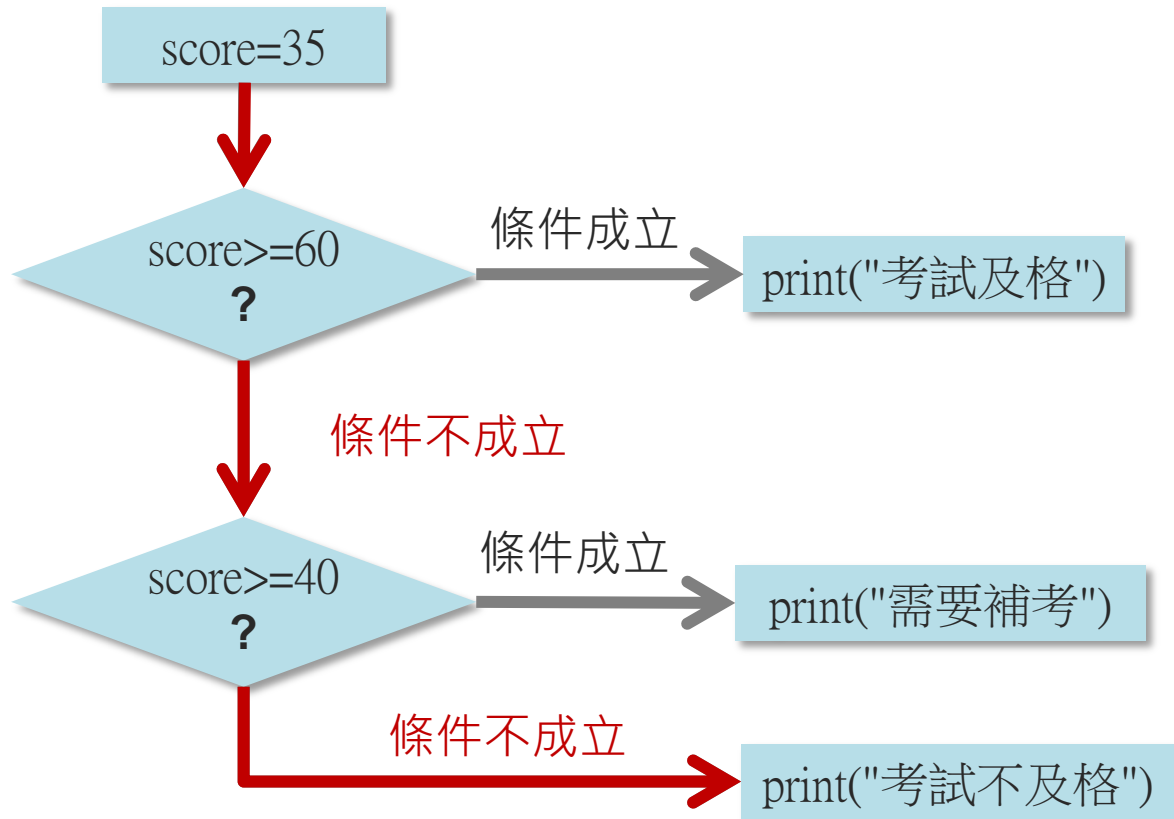
```
if3-1.py - D:/PythonJunior/E...
File Edit Format Run Options Window Help

score=55
if score>=60:
    print("考試及格")
elif score>=40:
    print("需要補考")
else:
    print("考試不及格")
print("執行結束")

Ln: 9 Col: 0
```



# 程式範例



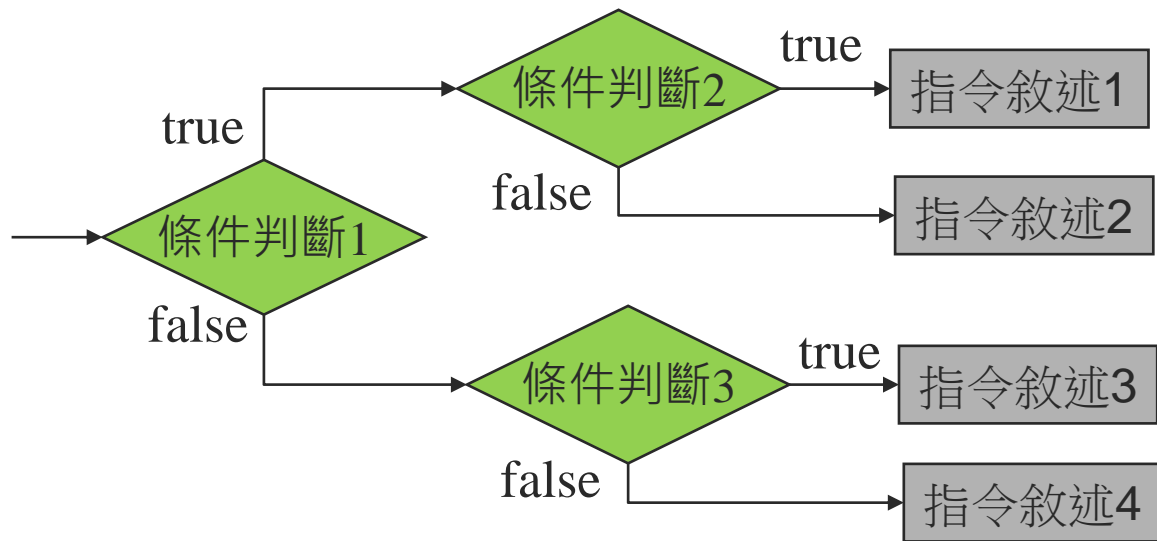
```
if3-2.py - D:/PythonJunior/E...
File Edit Format Run Options Window Help

score=35
if score>=60:
    print("考試及格")
elif score>=40:
    print("需要補考")
else:
    print("考試不及格")
print("執行結束")

Ln: 9 Col: 0
```

# 巢狀分支結構

## ◆ 巢狀分支結構



if(條件式1):

if(條件式2):

指令敘述1

else:

指令敘述2

else:

if(條件式3):

指令敘述3

else:

指令敘述4

# 程式範例

```
if4.py - D:/PythonJunior/Examp...
File Edit Format Run Options Window Help

score=65
if score>=60:
    print("考試及格")
    if score>=90:
        print("上台領獎")
else:
    print("考試不及格")
    if score>=40:
        print("需要補考")
    else:
        print("明年重修")
print("分數:", score)
```

Ln: 13 Col: 0

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: D:/PythonJunior
/Examples/Ch3/if4.py =====
考試及格
上台領獎
分數: 95
>>>

===== RESTART: D:/PythonJunior
/Examples/Ch3/if4.py =====
考試及格
分數: 65
>>>

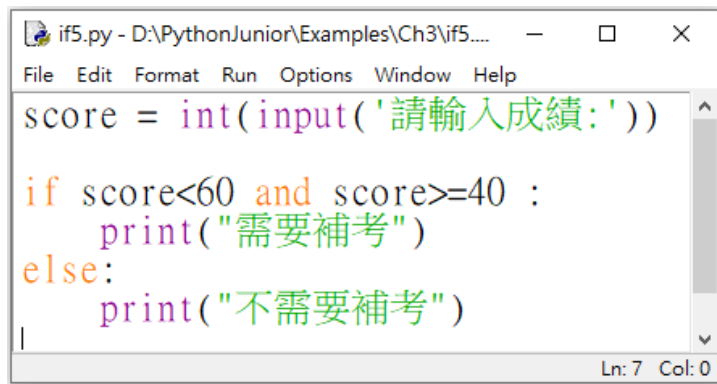
===== RESTART: D:/PythonJunior
/Examples/Ch3/if4.py =====
考試不及格
需要補考
分數: 55
>>>

===== RESTART: D:/PythonJunior
/Examples/Ch3/if4.py =====
考試不及格
明年重修
分數: 35
>>>
```

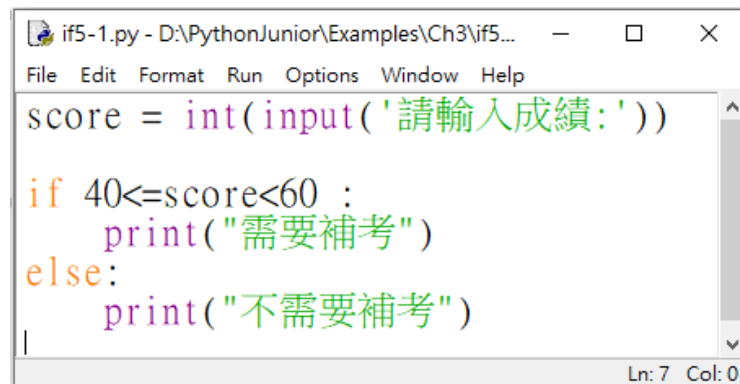
Ln: 22 Col: 4

# 直覺的邏輯運算寫法

- ◆ Python 允許使用直覺的方式進行邏輯分析。



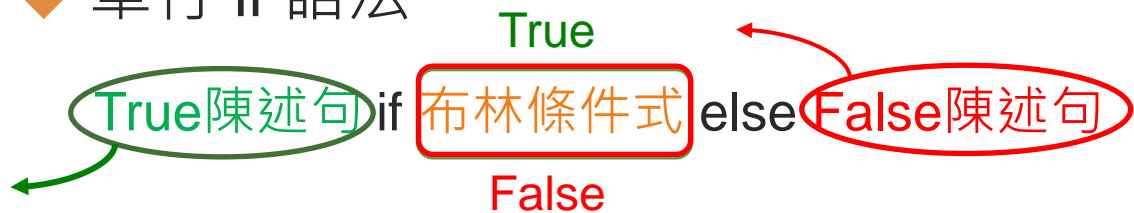
```
if5.py - D:\PythonJunior\Examples\Ch3\if5....
File Edit Format Run Options Window Help
score = int(input('請輸入成績:'))
if score<60 and score>=40 :
    print("需要補考")
else:
    print("不需要補考")
Ln: 7 Col: 0
```



```
if5-1.py - D:\PythonJunior\Examples\Ch3\if5...
File Edit Format Run Options Window Help
score = int(input('請輸入成績:'))
if 40<=score<60 :
    print("需要補考")
else:
    print("不需要補考")
Ln: 7 Col: 0
```

# 單行 if 語法

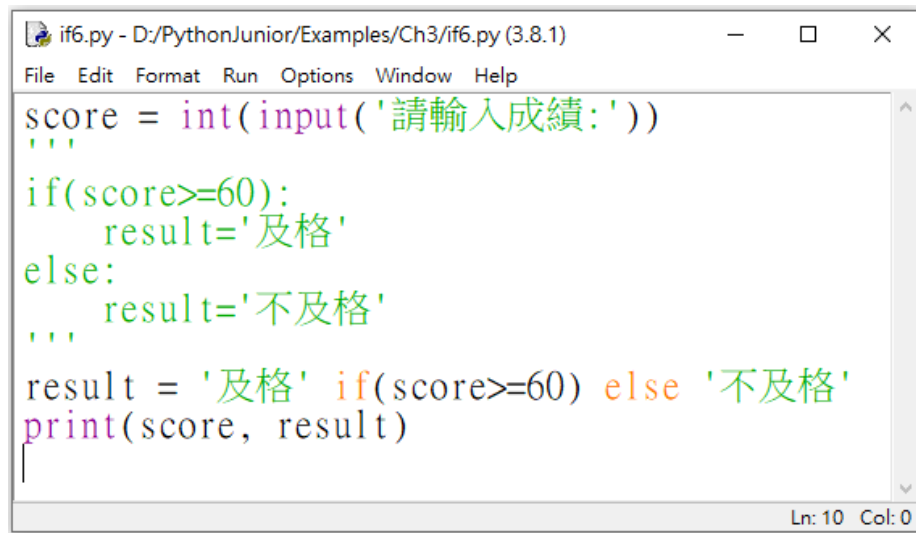
## ◆ 單行 if 語法



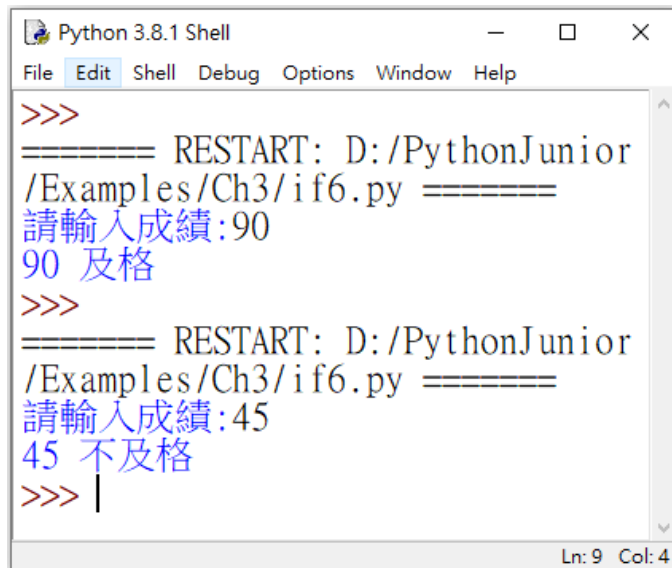
## ◆ 類似C及Java中的三元運算子

- (布林條件式)? True陳述句 : False陳述句

# 程式範例



```
if6.py - D:/PythonJunior/Examples/Ch3/if6.py (3.8.1)
File Edit Format Run Options Window Help
score = int(input('請輸入成績:'))
...
if(score>=60):
    result='及格'
else:
    result='不及格'
...
result = '及格' if(score>=60) else '不及格'
print(score, result)
Ln: 10 Col: 0
```



```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
>>>
===== RESTART: D:/PythonJunior/Examples/Ch3/if6.py =====
請輸入成績:90
90 及格
>>>
===== RESTART: D:/PythonJunior/Examples/Ch3/if6.py =====
請輸入成績:45
45 不及格
>>> |
Ln: 9 Col: 4
```

# 程式練習

- ◆ 請寫一個程式，判斷輸入的整數為否為2或3的倍數
  - ◆ 使用%

## 練習：BMI 計算後分析

- ◆ 請依據 BMI 值分析他人的身體狀況。

BMI值	<18.5	18.5~25	25~30	>30
身體狀態	太輕	正常	過重	肥胖



# 課程內容

## 1. 關係與邏輯運算

1-1. 關係運算

1-2. 邏輯運算

## 2. 分支結構

2-1. 分支結構

2-2. 單行分支語法

## 3. 迴圈結構

3-1. Range 範圍

3-2. for 迴圈

3-3. while 迴圈

# 重複執行概念

## ◆ 迴圈

- ◆ 讓程式中某一區段流程反覆執行

## ◆ for in 迴圈

- ◆ 執行固定次數
- ◆ for x in range(x)

## ◆ while迴圈

- ◆ 迴圈停止條件

# Range 範圍

## ◆ Range 範圍物件

- ◆ 內容為不可變更的數字序列
- ◆ 使用內建range函式建立範圍物件

`range(整數)`

- 建立一個由0至整數-1的範圍物件

- ◆ 使用內建list函式列出範圍物件內容

`list( range)`

# Range 範圍

## ◇ 自訂起始值及結束值的範圍

`range`(*起始值*, *結束值*)

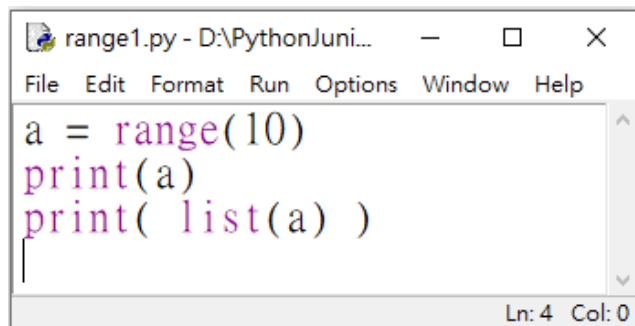
- 由起始值至結束值-1之間的範圍
  - ▶ 起始值與結束值均為整數

## ◇ 自訂遞增量的範圍

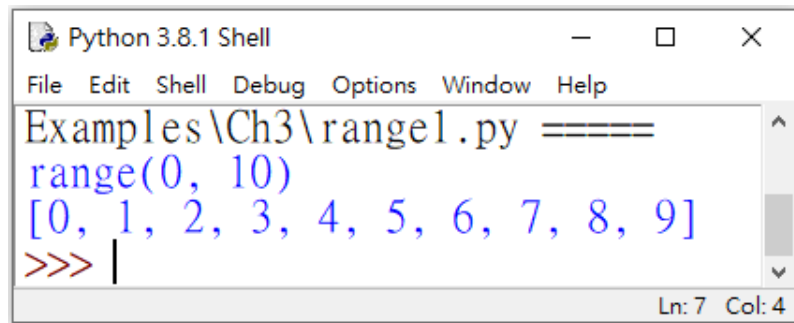
`range`(*起始值*, *結束值*, *遞增值*)

- 由起始值至結束值-1範圍的值，值之間相差指定遞增值
  - ▶ 起始值、結束值與遞增值均為整數

# 程式範例

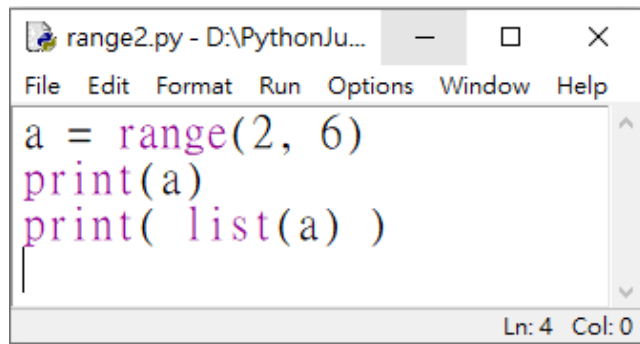


```
range1.py - D:\PythonJuni...  
File Edit Format Run Options Window Help  
a = range(10)  
print(a)  
print( list(a) )  
Ln: 4 Col: 0
```

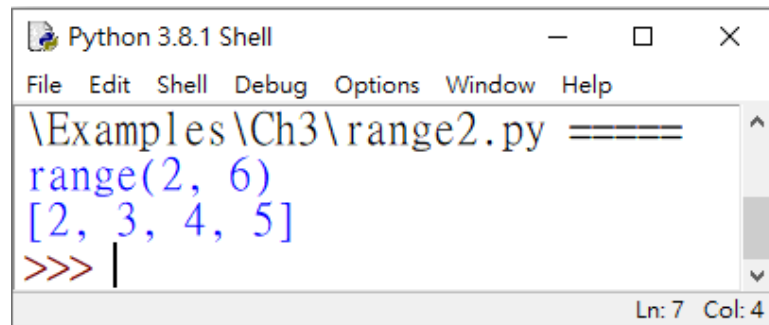


```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
Examples\Ch3\rangel.py =====  
range(0, 10)  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
>>> |  
Ln: 7 Col: 4
```

# 程式範例

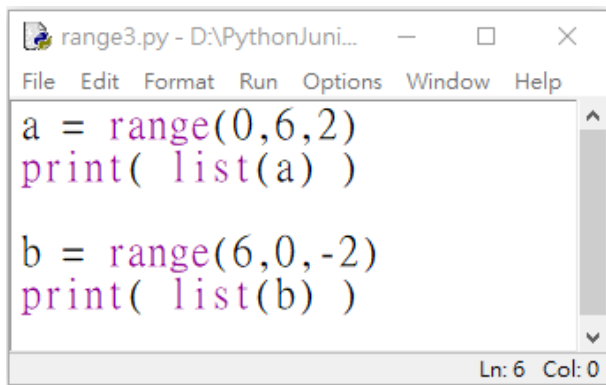


```
range2.py - D:\PythonJu...  
File Edit Format Run Options Window Help  
a = range(2, 6)  
print(a)  
print( list(a) )  
Ln: 4 Col: 0
```

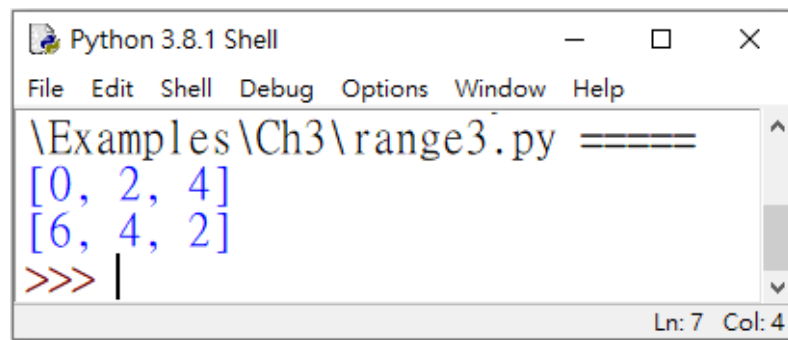


```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
\Examples\Ch3\range2.py =====  
range(2, 6)  
[2, 3, 4, 5]  
>>> |  
Ln: 7 Col: 4
```

# 程式範例



```
range3.py - D:\PythonJuni...  
File Edit Format Run Options Window Help  
a = range(0,6,2)  
print( list(a) )  
  
b = range(6,0,-2)  
print( list(b) )  
Ln: 6 Col: 0
```



```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
\\Examples\\Ch3\\range3.py =====  
[0, 2, 4]  
[6, 4, 2]  
>>> |  
Ln: 7 Col: 4
```

# for 迴圈

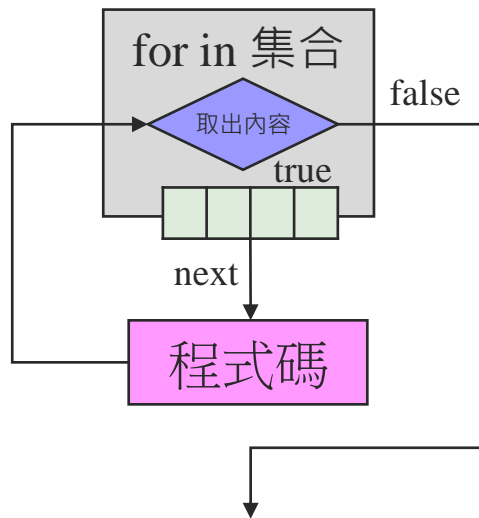
## ◆ for 迴圈

◆ 重複執行某段程式碼指定次數

◆ 語法

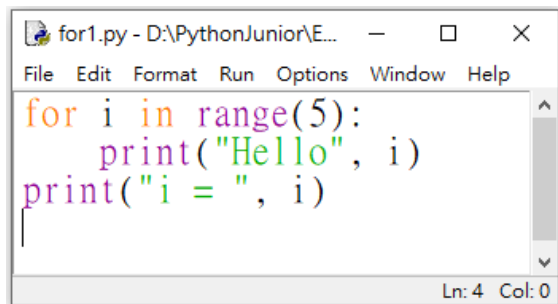
**for** 變數 **in** 集合 :

- 宣告變數，儲存當次由集合取出的值
- 集合可以是單一整數、兩個整數(起始值, 結束值)、三個整數(起始值, 結束值, 遞增減值) 的範圍物件、序列、字串





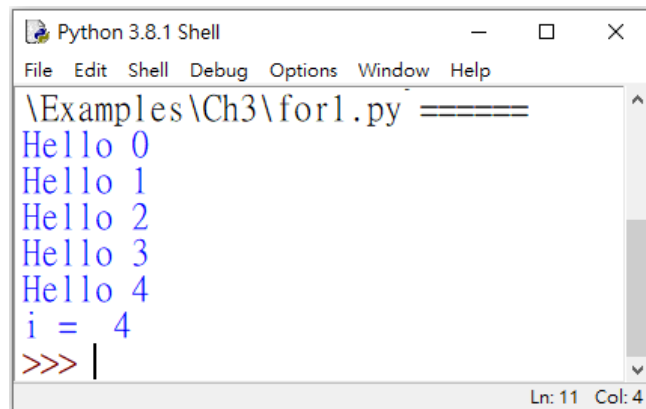
# 程式範例



A screenshot of a Python IDE window titled "for1.py - D:\PythonJunior\E...". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following Python code:

```
for i in range(5):  
    print("Hello", i)  
print("i = ", i)
```

The status bar at the bottom right indicates "Ln: 4 Col: 0".

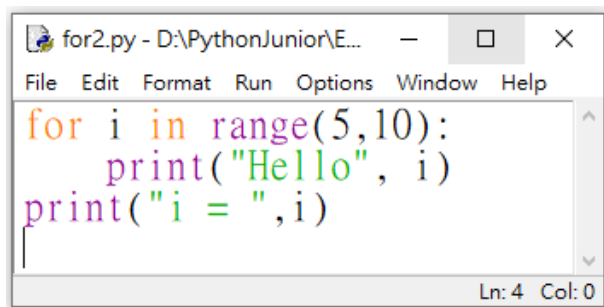


A screenshot of a "Python 3.8.1 Shell" window. The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The output of the script is displayed in the text area:

```
\Examples\Ch3\for1.py =====  
Hello 0  
Hello 1  
Hello 2  
Hello 3  
Hello 4  
i = 4  
>>> |
```

The status bar at the bottom right indicates "Ln: 11 Col: 4".

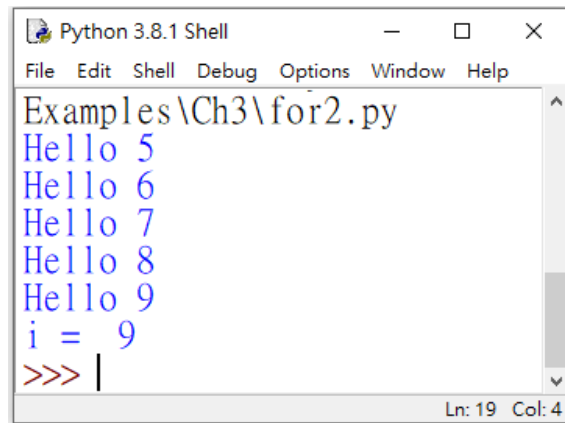
# 程式範例



A screenshot of a Python IDE window titled "for2.py - D:\PythonJunior\E...". The window contains a Python script with the following code:

```
for i in range(5,10):  
    print("Hello", i)  
print("i = ",i)
```

The status bar at the bottom right indicates "Ln: 4 Col: 0".

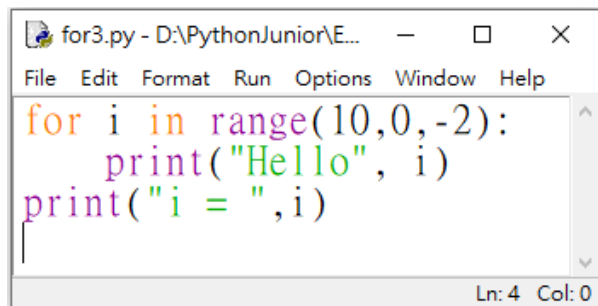


A screenshot of a Python 3.8.1 Shell window titled "Python 3.8.1 Shell". The window shows the output of the script from the previous image:

```
Examples\Ch3\for2.py  
Hello 5  
Hello 6  
Hello 7  
Hello 8  
Hello 9  
i = 9  
>>> |
```

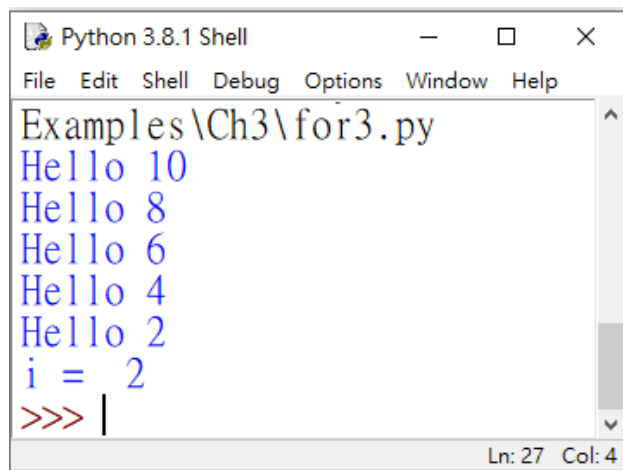
The status bar at the bottom right indicates "Ln: 19 Col: 4".

# 程式範例



```
for i in range(10,0,-2):  
    print("Hello", i)  
    print("i = ",i)
```

Ln: 4 Col: 0



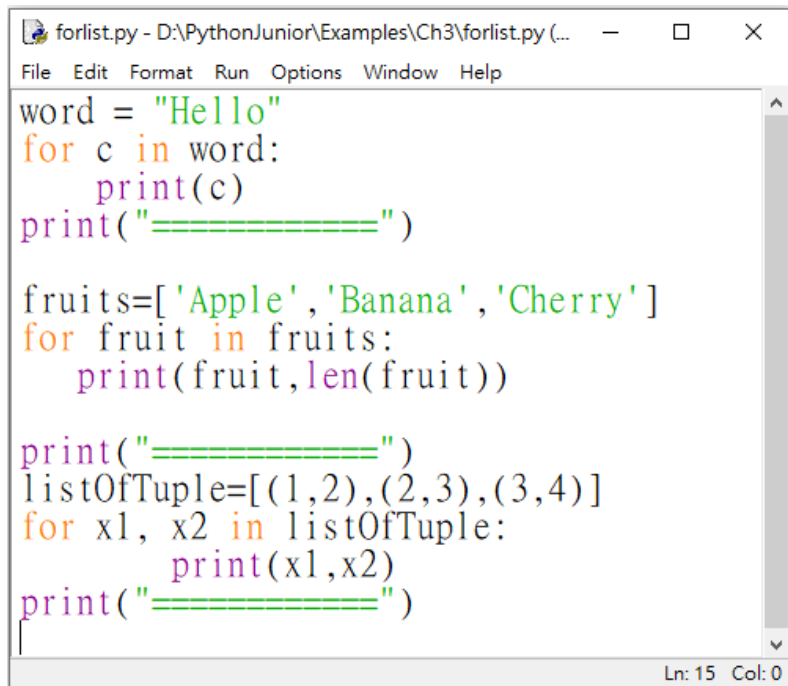
```
Examples\Ch3\for3.py  
Hello 10  
Hello 8  
Hello 6  
Hello 4  
Hello 2  
i = 2  
>>> |
```

Ln: 27 Col: 4

# for 迴圈與集合型資料

- ◆ in 可以搭配 string、set、tuple或list。
- ◆ 搭配 string 會逐一讀取字元。
- ◆ 搭配 set、tuple 與 list 會逐一取出集合元素。
  - ◆ (123, 456, 789)
  - ◆ ['Apple', 'Banana', 'Cherry']
  - ◆ len( ) 方法可計算字串的字數或集合內資料個數。

# 程式範例



```
forlist.py - D:\PythonJunior\Examples\Ch3\forlist.py (...)
```

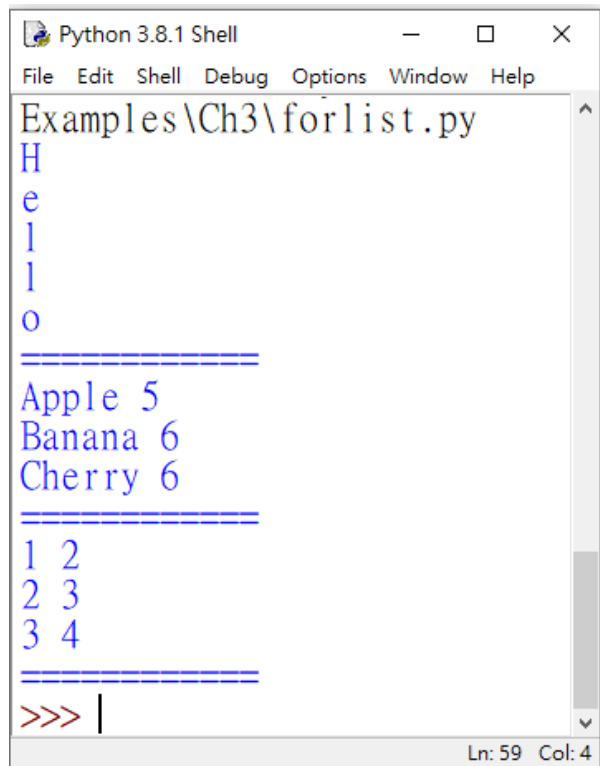
```
File Edit Format Run Options Window Help
```

```
word = "Hello"
for c in word:
    print(c)
print("=====")

fruits=['Apple','Banana','Cherry']
for fruit in fruits:
    print(fruit,len(fruit))

print("=====")
listOfTuple=[(1,2),(2,3),(3,4)]
for x1, x2 in listOfTuple:
    print(x1,x2)
print("=====")
```

Ln: 15 Col: 0



```
Python 3.8.1 Shell
```

```
File Edit Shell Debug Options Window Help
```

```
Examples\Ch3\forlist.py
H
e
l
l
o
=====
Apple 5
Banana 6
Cherry 6
=====
1 2
2 3
3 4
=====
>>> |
```

Ln: 59 Col: 4

# Zip 打包

◆ `for in` 結構只能針對單一集合逐一取出元素

◆ 取出多個集合相對索引位置的元素

◆ `zip()`

- 接受多個可迭代的對象作為參數，將對象中對應的元素打包成一個個 `tuple`

```
zipped = zip(subjs, scores)
```

subjs	=	['國語']	['數學']	['英文']
scores	=	[100,	80,	95]

◆ `for in` 取出打包完後的元素

```
for subj, score in zipped :
```

# 程式範例

```
ziptest.py - D:\PythonJunior\Examples\Ch3\ziptest.py (3.8.1)
File Edit Format Run Options Window Help
subjs = ['國語', '數學', '英文']
scores = [100, 80, 95]

for i in range(len(subjs)):
    print('%s分數:%d' %(subjs[i],scores[i]))

print(list(zip(subjs, scores)))

for n, s in zip(subjs, scores):
    print('%s分數:%d分' %(n,s))
|
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
= RESTART: D:\PythonJunior\Examples\Ch3\ziptest.py
國語分數:100
數學分數:80
英文分數:95
[('國語', 100), ('數學', 80), ('英文', 95)]
國語分數:100分
數學分數:80分
英文分數:95分
>>> |
Ln: 76 Col: 4
```

# 程式練習

- ◆ 請寫一個程式，在螢幕上顯示九九乘法表
  - ◆ 使用巢狀 for 迴圈



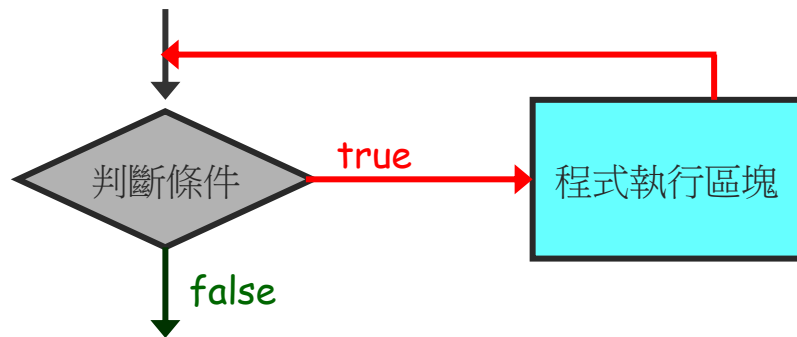
# while 迴圈

## ◆ while 迴圈

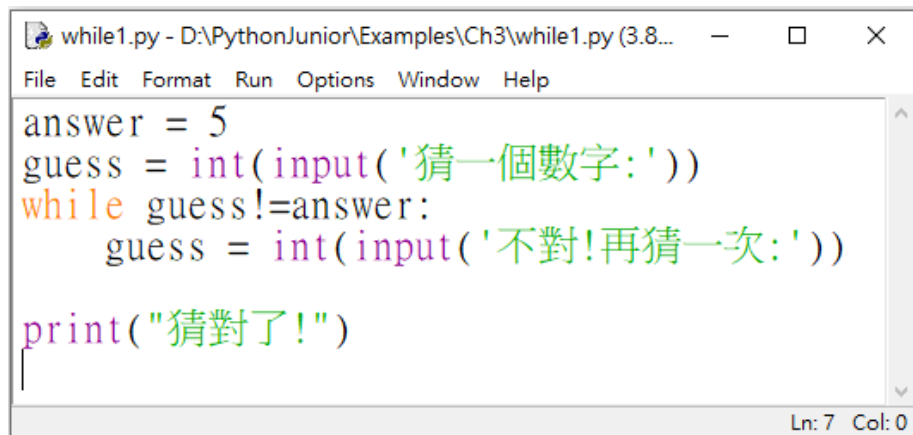
- ◆ 程式執行某些重複的動作
- ◆ 知道停止執行的條件, 不確定執行次數
- ◆ 語法

**while** 判斷條件：

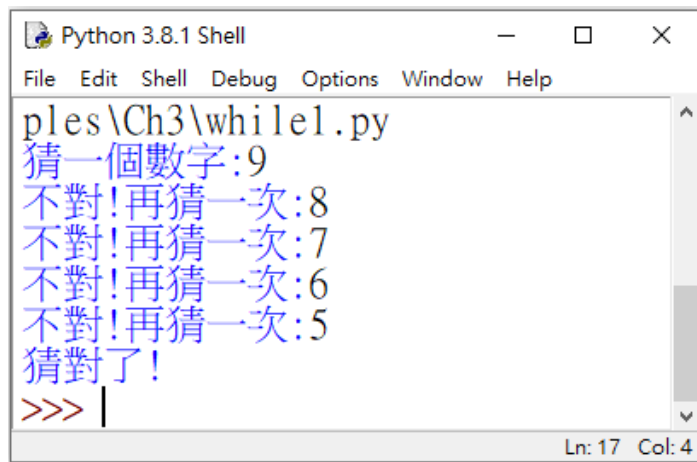
重複執行的動作  
重複執行的動作



# 程式範例



```
while1.py - D:\PythonJunior\Examples\Ch3\while1.py (3.8...  
File Edit Format Run Options Window Help  
answer = 5  
guess = int(input('猜一個數字:'))  
while guess!=answer:  
    guess = int(input('不對!再猜一次:'))  
  
print("猜對了!")  
Ln: 7 Col: 0
```



```
Python 3.8.1 Shell  
File Edit Shell Debug Options Window Help  
ples\Ch3\while1.py  
猜一個數字:9  
不對!再猜一次:8  
不對!再猜一次:7  
不對!再猜一次:6  
不對!再猜一次:5  
猜對了!  
>>> |  
Ln: 17 Col: 4
```

# 程式練習

- ◆ 請寫一個程式，顯示由1至1000之間的費伯那西數列
  - ◆ Fibonacci Sequence：由1、1開始，下一個數字為上兩個數字的和
  - ◆ 1、1、2、3、5、8、13、21...。

# 無窮迴圈

## ◆ 無窮迴圈

- ◆ 程式停止前持續不斷執行某個迴圈
- ◆ 經常用在遊戲的主迴圈
- ◆ 搭配break語法跳離迴圈，終止程式

while True :

重複執行的動作

重複執行的動作

# 迴圈中斷

## ◆ 迴圈中斷

- ◆ break：代表中斷迴圈執行後離開迴圈
- ◆ continue：代表迴圈中斷執行繼續執行下一次迴圈

# 程式範例

```
break1.py - D:\PythonJunior\Examples\Ch3\brea...
File Edit Format Run Options Window Help

count = 1
answer = 5
guess = int(input('猜一個數字:'))
while count < 5 :
    if guess==answer:
        print('猜對了')
        break
    count += 1
    guess = int(input('再猜一次:'))

print("遊戲結束!")

Ln: 12 Col: 0
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

\Examples\Ch3\break1.py =====
猜一個數字:10
再猜一次:9
再猜一次:8
再猜一次:7
再猜一次:6
遊戲結束!
>>>
===== RESTART: D:\PythonJunior
\Examples\Ch3\break1.py =====
猜一個數字:3
再猜一次:4
再猜一次:5
猜對了
遊戲結束!
>>>

Ln: 18 Col: 4
```

# 程式範例

```
continuel.py - D:\PythonJunior\Examples\Ch3\continuel.py
File Edit Format Run Options Window Help
results = {"David":0, "Amy":0, "Sean":0}
candidates = ("David", "Amy", "Sean")
for i in range(5):
    vote = input('投票給: ')
    if(vote not in candidates):
        print('選票無效')
        continue
    results[vote] = results[vote]+1
print("選舉結果:")
for name in candidates:
    print(name, results[name], "票")
|
```

Ln: 13 Col: 0

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Examples\Ch3\continuel.py
投票給: David
投票給: Amy
投票給: sean
選票無效
投票給: Sean
投票給: Amy
選舉結果:
David 1 票
Amy 2 票
Sean 1 票
>>> |
```

Ln: 58 Col: 4

# 程式練習

- ◆ 請寫一個程式，檢查所輸入的數是否為質數



# 程式練習

- ◆ 請寫一個程式，列出輸入整數的所有質因數

# for-else語法搭配break

## ◆ for-else語法

- ◆ 不用設置狀態變數來檢查 for 迴圈循環是否被break中斷
- ◆ for 迴圈中有循環中斷的break結構
  - break 將跳離整個 for-else 結構
- ◆ else部分定義循環中斷未發生時的處理邏輯

# 程式範例

```
forElse.py - D:/PythonJunior/Examples/Ch3/forEls...
File Edit Format Run Options Window Help

answer = 5
for i in range(5):
    guess = int(input('猜一個數字:'))
    if guess==answer:
        print('猜對了')
        break
else:
    print('答案是', answer)
print("遊戲結束!")

Ln: 10 Col: 0
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
= RESTART: D:/PythonJunior/
Examples/Ch3/forElse.py
猜一個數字:10
猜一個數字:9
猜一個數字:8
猜一個數字:7
猜一個數字:6
答案是 5
遊戲結束!
>>>
= RESTART: D:/PythonJunior/
Examples/Ch3/forElse.py
猜一個數字:3
猜一個數字:4
猜一個數字:5
猜對了
遊戲結束!
>>>

Ln: 85 Col: 4
```

# 程式練習

- ◆ 使用for-else結構改寫計算所輸入的數是否為質數程式



巨匠直播教學

[www.pcschoolonline.com.tw](http://www.pcschoolonline.com.tw)