

Python

If and while (邏輯判斷)

根據你設定的“條件”，讓電腦判斷應該
要執行或跳過某一段程式

“條件” → 某個 “邏輯判斷” 成立(true)

(例如: 當a變數 > 5的時候，才print(a))

IF Block的基本語法

你設定的條件(邏輯判斷)，
這行用if開頭，用:結尾

- **if logical_expression :**

```
python command1  
python command2  
...
```

這一段程式只有邏輯判斷成立
才會被執行
如果不成立就會被跳過

在if...:之後的幾行，只要開頭
對齊(縮排情況一樣)，都會被
當成同一個if block的內容

IF Block的基本語法:例子

你設定的條件(邏輯判斷)

- **if a>5 :**

```
    c=a+10
```

```
    print(c)
```

如果a>5成立，才會令c=a+10並顯示c的值在螢幕上

如果a>5不成立，這兩行不會被執行

```
print(a)
```

這行沒有縮排，不受前面這個if block的影響，無論a變數的值是多少，這行都會執行

基本邏輯判斷符號

- == (equal)
- != (not equal)
- > (greater than)
- >= (greater than or equal to)
- < (less than)
- <= (less than or equal to)

提醒:

- == 邏輯關係判斷左右是否相等，
- = 把右邊的資料存到左邊的變數

邏輯判斷符號: is, is not

- is

比 "==" 更為嚴格，
必須連data type 都相同才是
“相等”

- is not

比 "!=" 更為寬鬆，
連data type 不同都當作“不
相等”

```
>>> a = 4
>>> print (a)
4
>>> a is 4.
False
>>> a is 4.0
False
>>> a is 4
True
>>> a == 4.
True
>>> a == 4
True
```

IF Block: 稍微複雜一點的結構 if...elif...(分岔)

- **if logical_expression :**

```
python command  
...
```

} 邏輯判斷成立才會被執行
並且執行後直接跳到if block外

else:

```
python command2  
...
```

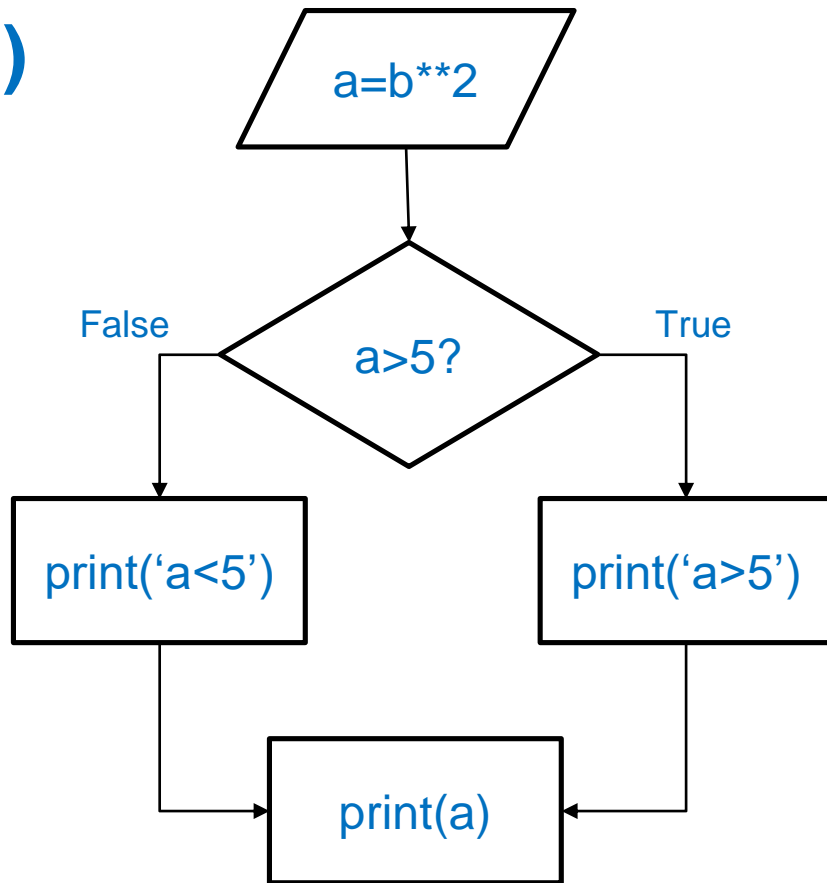
} 邏輯判斷**不**成立才會被執行
並且執行後直接跳到if block外

```
python command3
```

else後面也要有 : 符號

IF Block: if...else (分岔)

```
a=b**2
if a>5 :
    print('a>5')
else:
    print('a<5')
print(a)
```



課堂練習: if...else

- 給予變數x一個值 (e.g., $x=25$)
- 寫出if判斷式：
 - 如果 $x>0$ ，則在螢幕顯示 $\log_{10}(x)$ ；
 - 其餘情況，則在螢幕顯示 'x out of range'
- 測試是否正確
- 第一行改為 $x= -10$ ，測試是否正確

IF Block: 更複雜一點的結構 `if...elif...else` (多重條件)

- `if logical_expression1 :`

```
python command1
```

```
...
```

} 條件1成立才會執行
執行後直接跳到if block外

- `elif logical_expression2 :`

```
python command2
```

```
...
```

} 條件2成立才會執行
執行後直接跳到if block外

- `elif logical_expression3 :`

```
python command3
```

```
...
```

} 條件3成立才會執行
執行後直接跳到if block外

`else:`

```
python command4
```

} 條件1~3都不成立才會執行
執行後直接跳到if block外

IF Block: if...elif...else

多重條件

a=b/2

if a<0 :

 print('a<0')

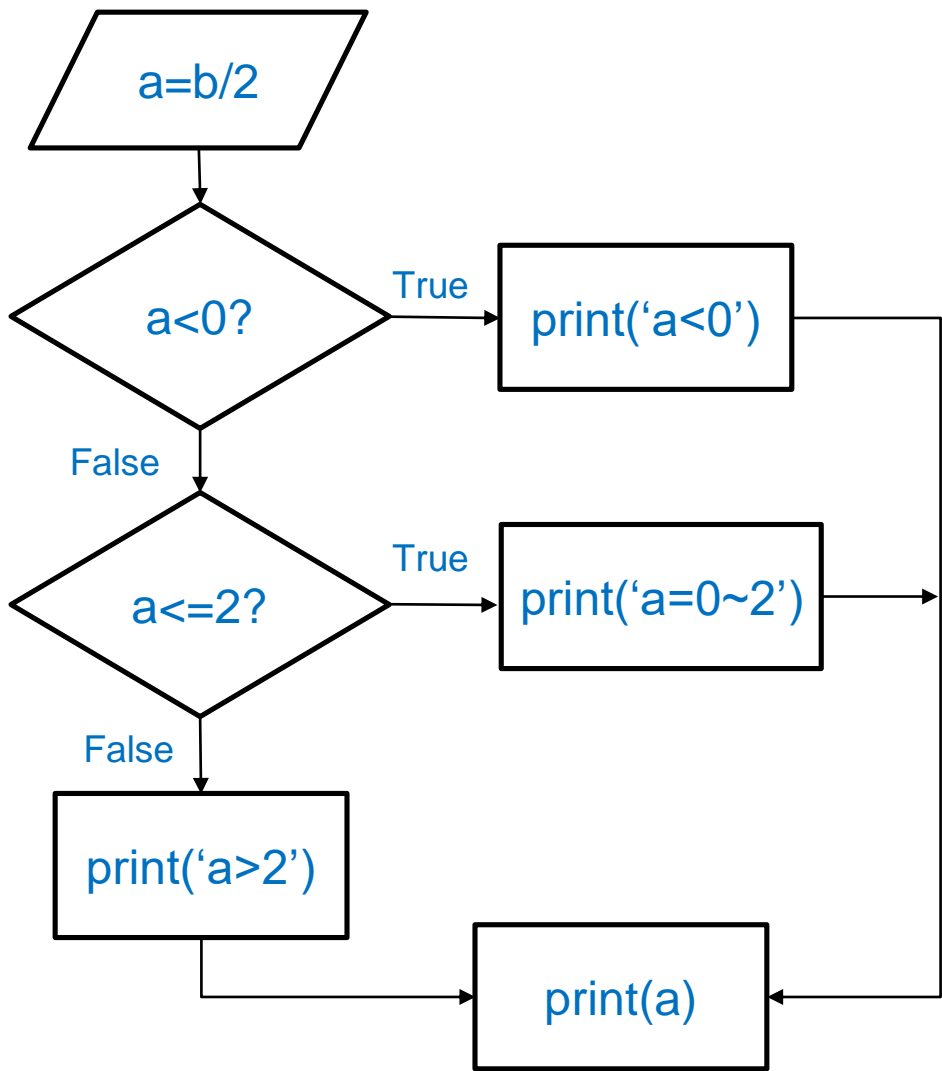
elif a<=2 :

 print('a=0~2')

else:

 print('a>2')

print(a)



多層IF Block結構

- **if logical_expression1 :**

```
python command1
```

```
...
```

```
if logical_expression2 :
```

```
python command2
```

```
...
```

```
...
```

```
else:
```

```
python command3
```

條件1成立才會執行

條件1成立且條件2成立才會執行，執行後直接跳到if block外(注意縮排情況)

條件1**不成立**才會執行
執行後直接跳到if block外

多層 IF Block

a=b/2

if a<0 :

 print('a<0')

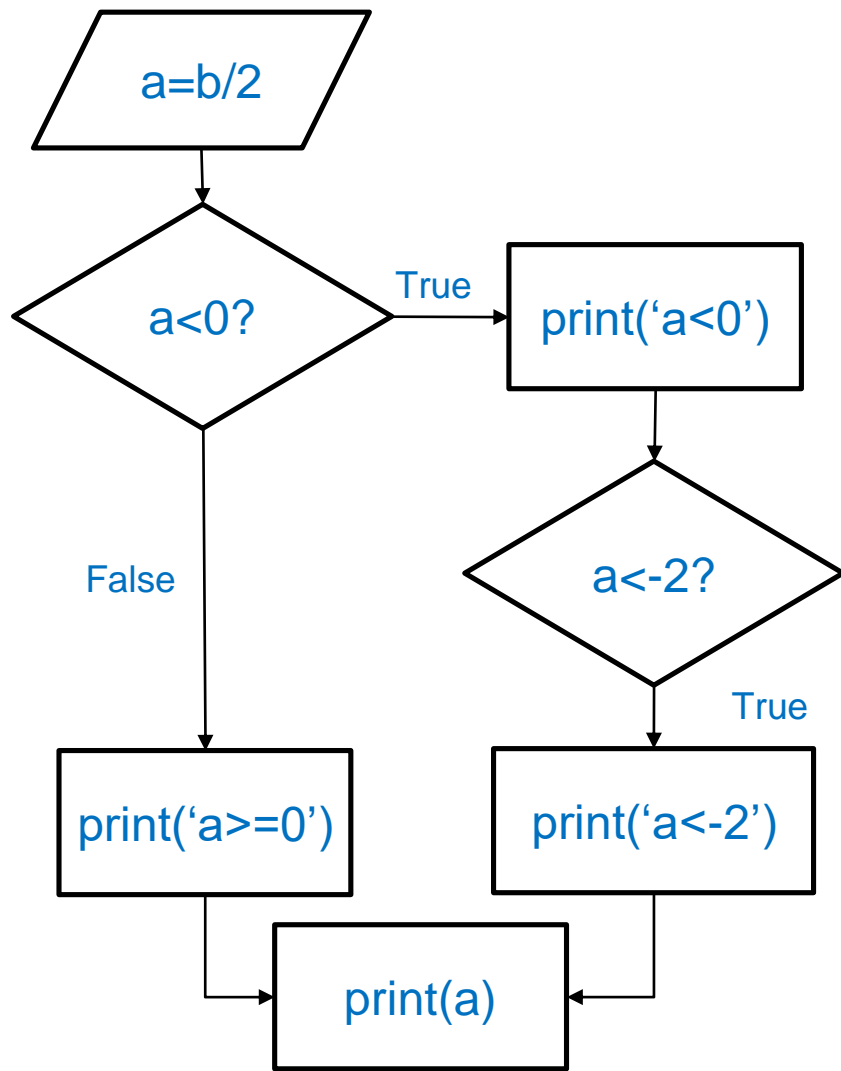
 if a<-2 :

 print('a<-2')

else:

 print('a>=0')

print(a)



邏輯判斷的組合(運算)

- **and** 交集 (且)

- 例如

```
if a>0 and a<10:  
    print('0<a<10')
```

- 兩個條件都要成立才執行

- 針對同一個變數的交集，可以連續寫，例如

```
if 0<a<10 :  
    print('0<a<10')
```

- **or** 聯集(或)

- 例如:

```
if a>2 or b<-5:  
    print(a, b)
```

- 任一條件成立就會執行

- **not** 非(...的相反)

- 例如:

```
if not a>10 :  
    print("a<=10")
```

邏輯判斷變數(Boolean variables)

- 除了數值變數、字串變數，還有另一種變數類型：邏輯判斷變數
- 有複雜邏輯運算的時候，利用這種變數可以讓程式變得簡潔易懂
- 邏輯判斷變數只會有兩個「值」：False (不成立)，True (成立)
- 如何給定邏輯判斷變數的值？
 - 方法一（邏輯關係） `a_positive = (a > 0.)`
 - 程式執行過程中，`a`變數的值會決定`a_positive`的值是True or False
 - 方法二（直接設定是/非） `ask = True`

使用邏輯判斷變數 讓判斷式更簡潔

```
a,b,c=input('give me three numbers').split(',')  
a,b,c=float(a),float(b),float(c)
```

```
if (a>0 and b<10) or (c>100):  
    print('good choice')
```

```
good_ab = (a>0 and b<10)  
good_c = (c>100)
```

```
a,b,c=input('give me three numbers').split(',')  
a,b,c=float(a),float(b),float(c)
```

```
if good_ab or good_c:  
    print('good choice')
```

用邏輯判斷變數改寫

While (條件成立則不斷重複，直到條件不成立為止)

你設定的條件(邏輯判斷)

- **while** logical_expression :

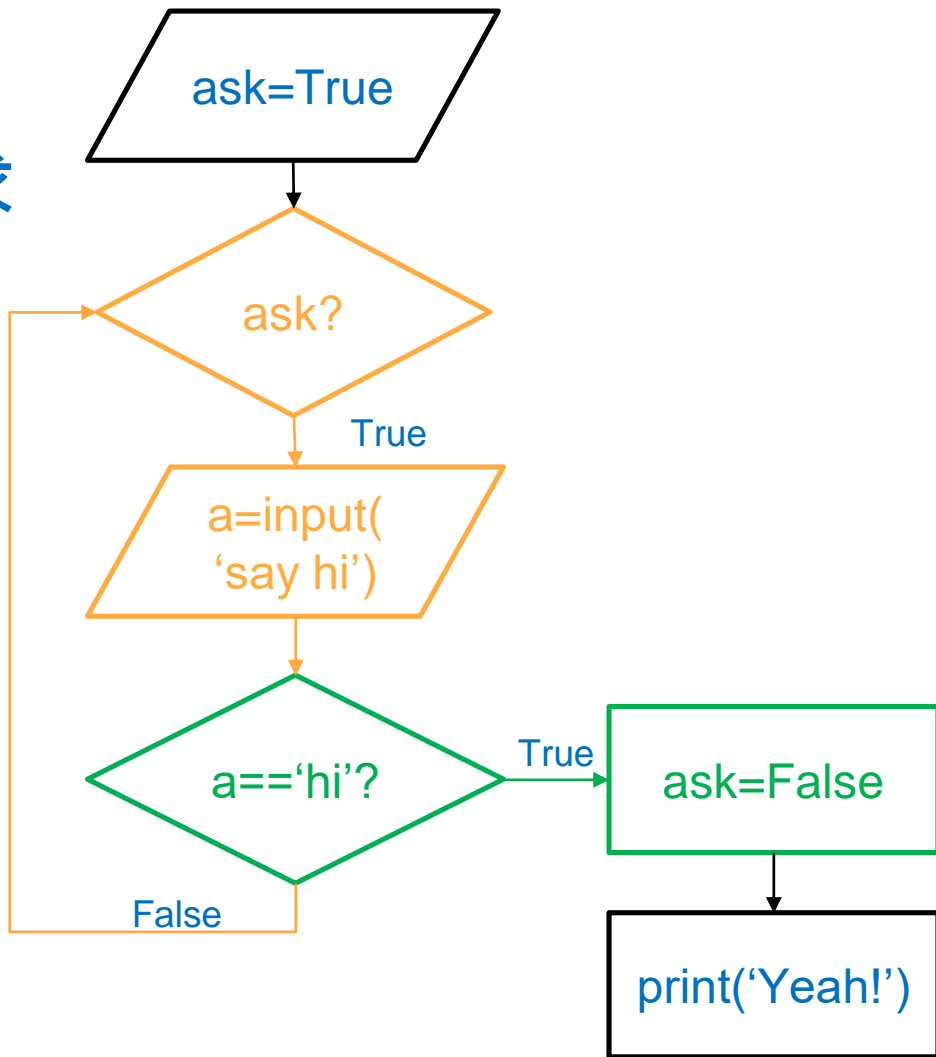
```
python command1  
python command2  
...
```

這一段程式在邏輯判斷成立時
會被無限次重複執行
如果不成立就會直接被跳過

注意!! 可能會製造出永不停止
的程式，如果發生了，請用
ctrl+c 強迫中止

While常見例子: 用來檢查輸入是否符合要求

```
ask=True
while ask :
    a=input('say hi:')
    if (a=='hi'):
        ask=False
print('Yeah!')
```



IF and while 語法規則提醒

- 縮排不同數量的空格（或tab），會被視為不同的if block
- 邏輯判斷式 如果有多個或很複雜，建議加上括號()以利閱讀，或利用邏輯判斷變數

if (a>5) or (b<3) and (c==0):

- if, elif, else, while開頭的那行，結尾都要加上 :
- 指令如果很簡短，可以直接在 : 後面寫出，例如
if (a%2 == 0): print (“a is even number”)
while (a>0): a=a+1

設計IF BLOCK

- 把問題寫成簡單的判斷式
 - 找出要進行邏輯判斷的變數
 - 列出各種條件，以及不同條件下要執行的動作
 - 可畫流程圖幫助理解
- 檢查 IF, ELIF等條件是否合理，要避免條件重複或自相矛盾的情況
- 是否考慮到所有可能發生的條件或情況？（ ELSE包含的條件狀況為何？ ）
- 寫成python語法
 - 用縮排讓程式整齊易讀
 - 注意語法規則、加上註解