

# Python

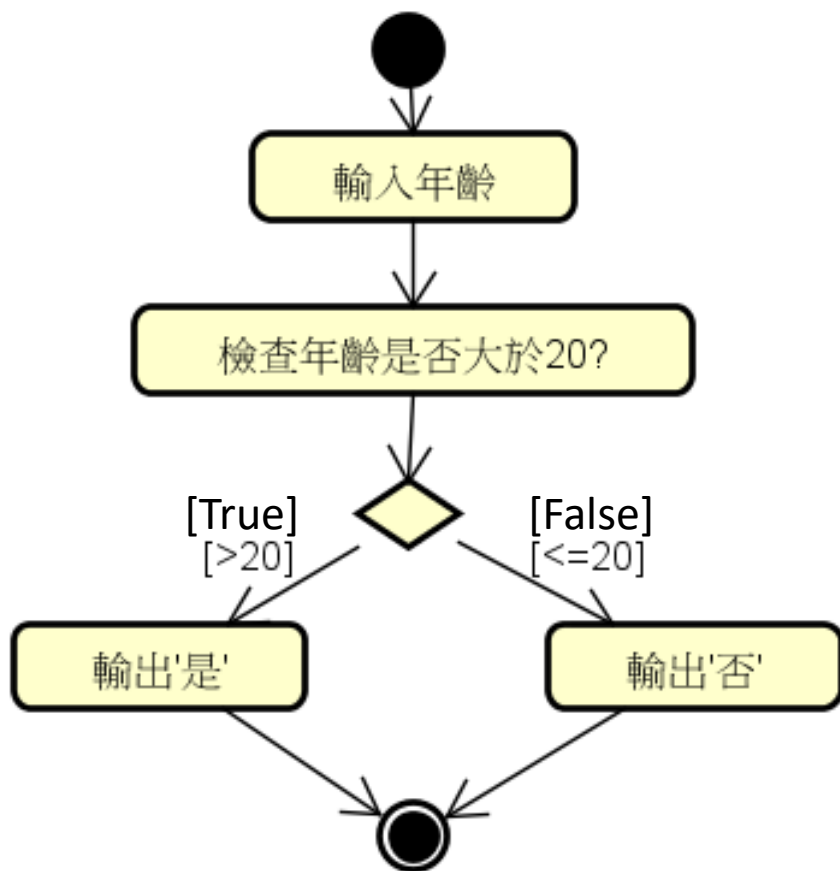
比較-真假

請說出你的年齡，  
並回答是否有大於20歲

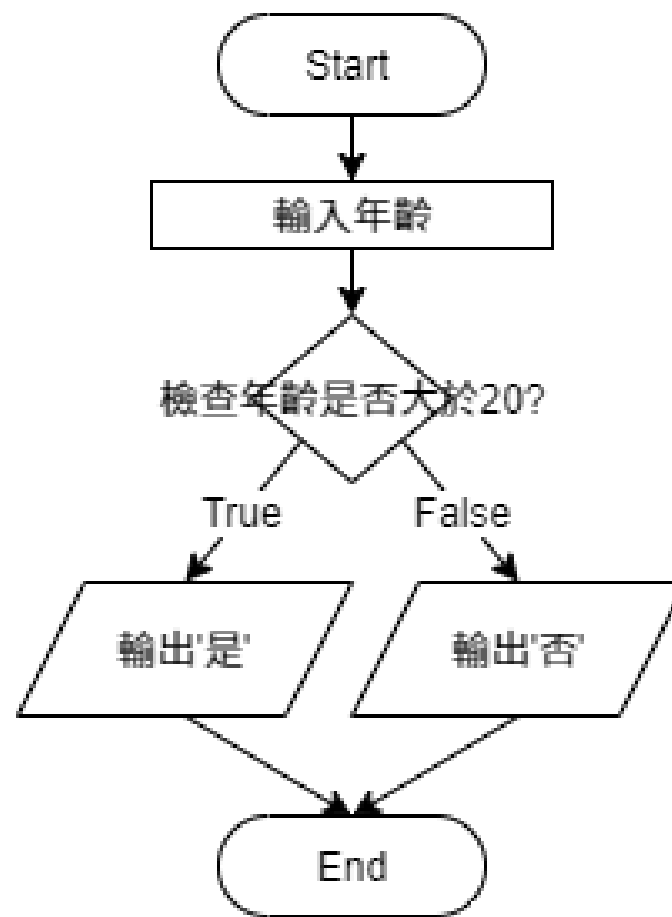
問題



# 流程



UML流程圖



傳統流程圖

# Python 嘛ㄟ通的數學比較符號

符號	數學意義	Python 比較運算子 (Comparison Operators)
=	等於	==
≠	不等於	!=
<	小於	<
≤	小於等於	<=
>	大於	>
≥	大於等於	>=

<=> 有提過，等號是將右邊的內容指派給左邊變數的指派運算子，因此為了避免搞混，使用了兩個等號(==)代表等於的比較運算

限於打字輸入的方便性，比較符號的寫法會有部分差異，但代表的意義都是相同的喔！

# True & False

- 進行比較後會得到兩種特別的值:
  - True
  - False
- True & False 兩種值是屬於布爾(Bool)型態
- 布爾(Bool)型態也只有 True & False 兩種值

```
age = int(input())  
print(age > 20)
```

10

False

30

True

# True & False

True 可以當做是:

- 真的
- 對的
- 正確
- Yes
- 整數的 1



False 可以當做是:

- 假的
- 錯的
- 錯誤
- No
- 整數的 0

由於 True 跟 False 可以視為整數的 1 與 0，因此也可將布爾型態視為整數型態的子集合

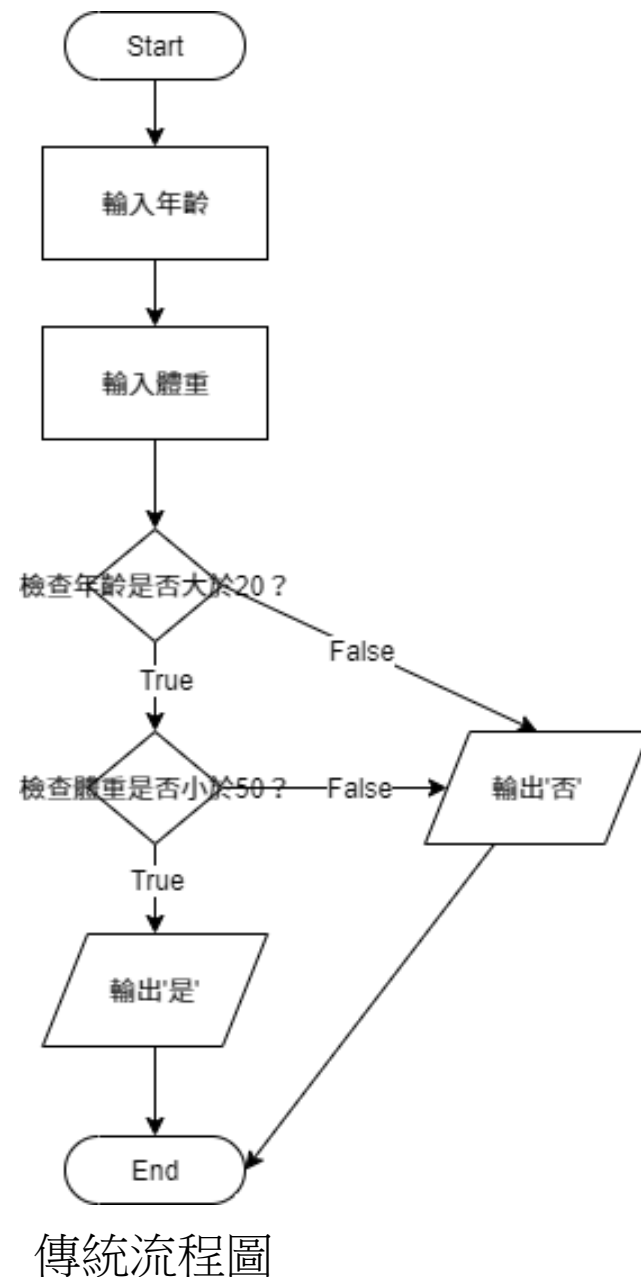
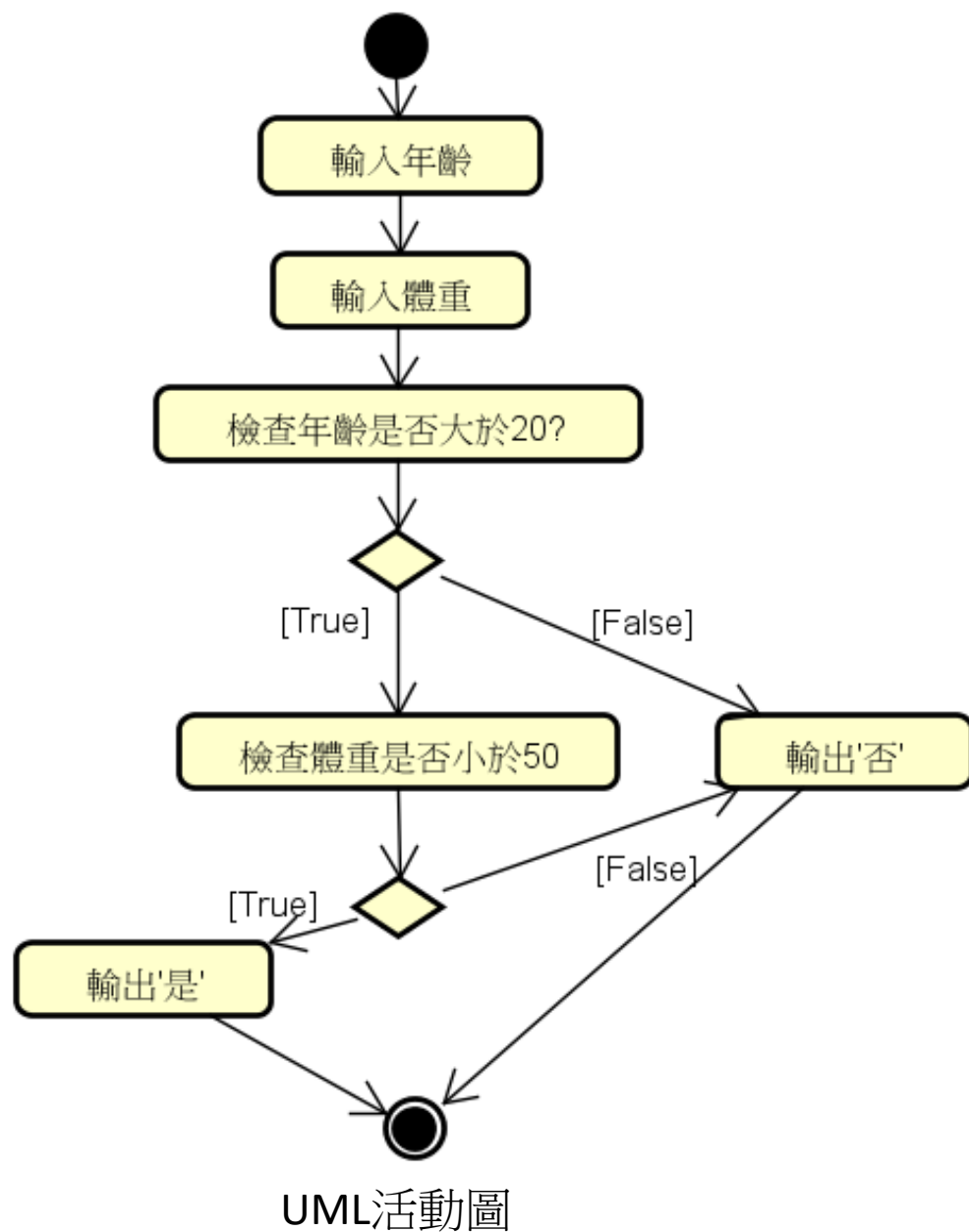
布爾	整數	代表的意義
True	1	真(對)
False	0	假(錯)

請說出你的年齡與體重，  
並回答是否有大於20歲，  
且體重小於50公斤

問題



# 流程





# 所有敘述都為true

```
age = int(input())  
weight = int(input())  
print(age>20 and weight<50)
```

執行結果

```
30  
40  
True
```

```
30  
60  
False
```

# 只要一個敘述為true

```
age = int(input())  
weight = int(input())  
print(age > 20 or weight < 50 )
```

執行結果

30  
40

True

30  
60

True

# 一起比較吧

- 當需要多個一起比較時，就用 `and`, `or`, `not` 把所有比較連起來
- `and`, `or`, `not` 就稱為**邏輯運算子(Logical Operators)**

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

PYTHON

延伸的概念

指定布爾值的變數

```
Y = True  
N = False
```

```
print(type(Y))  
print(type(N))
```

```
print(10 > 5)  
print(5 != 5)
```

```
print(type(10 > 5))  
print(type(5 != 5))
```

像 True 跟 False 這種有特別意義的詞不能當變數名稱，這種詞稱為『**關鍵詞**』(keyword)

執行結果

```
<class 'bool'>  
<class 'bool'>  
True  
False  
<class 'bool'>  
<class 'bool'>
```

# 概念 1: is , is not

- 還有一種運算符號稱為**身分運算子(Identity Operators)**
- 顧名思義，就是用來判斷兩種資料身分是否是相同的喔

運算符號	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

# 概念 2: in , not in

- 最後一種運算符號：**隸屬成員子(Membership Operators)**
- 用來判斷前者是否有被包含在後者資料中

運算符號	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'

# 整理一下吧

類型	包含的符號
算數運算子 (Arithmetic Operators)	+ , - , * , / , % , ** , //
指派運算子 (Assignment Operators)	+= , -= , *= , /= , %= , **= , //=
比較運算子 (Comparison Operators)	== , != , < , <= , > , >=
邏輯運算子 (Logical Operators)	and , or , not
身分運算子 (Identity Operators)	is , is not
隸屬成員子 (Membership Operators)	in , not in
位元運算子 (Bitwise Operators)	&   ^ ~ << >>

部分沒說明到的符號，可參考：

<http://www.runoob.com/python3/python3-basic-operators.html>