

# 判斷式、布林運算與條件分支

滄海《Python 程式設計初學指引》二版·CH03

## § 1 名詞速查表

中文	English	一句話定義	例	易混
判斷式	condition	結果為布林的運算式，決定走哪個程式區塊	<code>score &lt; 60</code>	—
布林型別	bool	只有兩個值： <code>True</code> / <code>False</code>	—	<code>True/False</code> 首字大寫
比較運算	comparison	<code>&lt;</code> <code>&lt;=</code> <code>&gt;</code> <code>&gt;=</code> <code>==</code> <code>!=</code> 值比較； <code>in/is</code> 等	<code>7 &gt; 5</code>	<code>==</code> 比較 vs <code>=</code> 指定
布林運算	boolean	<code>and</code> / <code>or</code> / <code>not</code> 邏輯運算	<code>a&gt;0 and b&gt;0</code>	—
短路運算	short-circuit	<code>and</code> 遇 <code>False</code> 就停、 <code>or</code> 遇 <code>True</code> 就停	—	—
<code>if</code> / <code>elif</code> / <code>else</code>	—	條件分支：成立執行對應區塊	—	多分支用 <code>elif</code> 不是多個 <code>if</code>
縮排	indentation	Python 用縮排界定程式區塊（同區塊靠左對齊）	4 空格	縮排錯 → 區塊錯 / <code>IndentationError</code>
巢狀	nested	判斷式區塊裡再放判斷式	—	—
三元運算子	ternary	值1 if 條件 else 值2，一行做選擇	<code>'及格' if s&gt;=60 else '不及格'</code>	—
連鎖比較	chained	<code>5 &lt; x &lt; 10</code> 可直接連寫	—	—

## § 2 核心概念

### 核心概念

**判斷式 (condition)** 是結果為布林 (True / False) 的運算式；程式在決策點依判斷式選擇要執行哪個**程式區塊**，這就是**選擇性流程**。判斷式由兩類運算組成：**比較運算** (< <= > >= == != 、in 、is) 與**布林運算** (and / or / not)。優先序：not > and > or；整體上「數值 > 比較 > 布林 > =」。

Python 最關鍵的語法特性是**用縮排界定區塊**：if 條件後的程式碼**往右縮排**才屬於該條件；縮排錯了區塊就錯，執行結果跟著變（甚至 IndentationError）。三種分支結構：if（單分支）、if-else（雙分支）、if-elif-else（多分支）；分支裡還能再放判斷式，就是**巢狀**。

兩個方便寫法：**連鎖比較** 5 < x < 10（等同 x>5 and x<10）；**三元運算子** 值1 if 條件 else 值2（一行做選擇）。

## § 3 主要內容

### 3.1 一、比較運算與布林運算

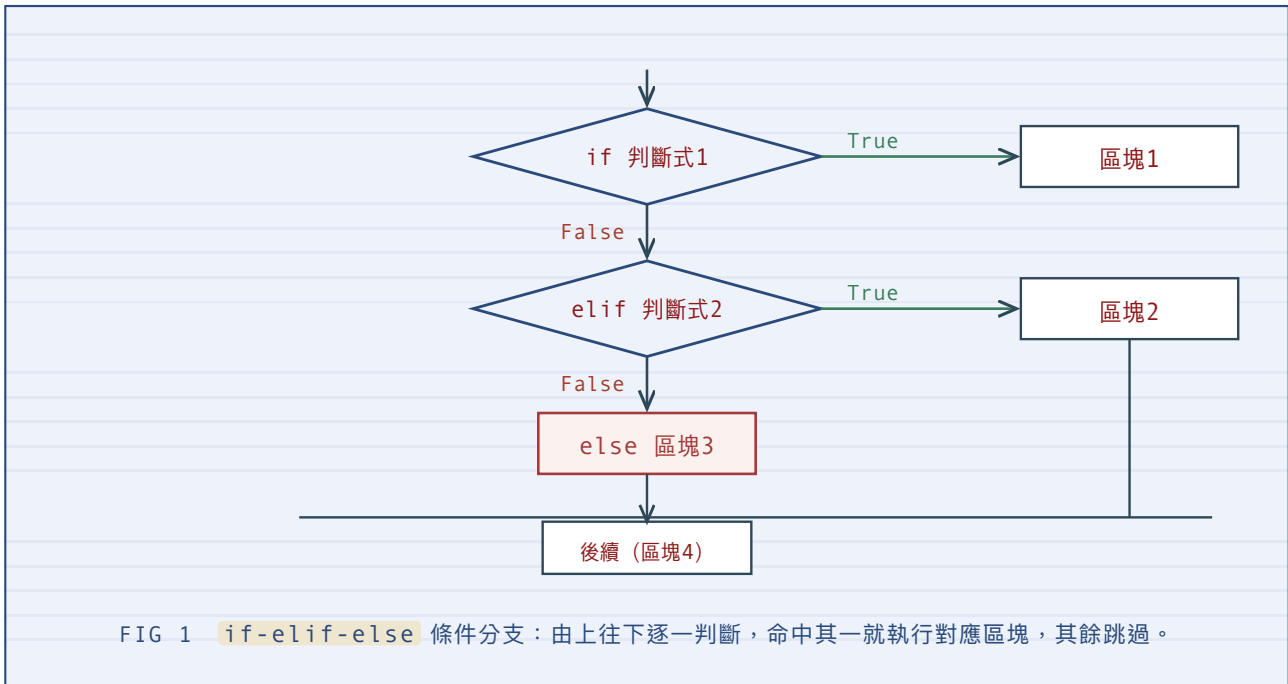
```
7 > 5           # True           值比較:< <= > >= == !=
'a' < 'A'      # False          字串比 Unicode ('a'=97 > 'A'=65)
'x' in 'cat'   # False          in / not in:從屬關係
a is b         # 兩變數是否同一物件 (is / is not)

True and False # False        and:兩者都 True 才 True
True or False  # True         or:有一個 True 就 True
not True       # False        not:反轉
5 < x < 10     # 連鎖比較, 等同 x>5 and x<10
```

### 短路運算 (short-circuit)

and 從左看，遇到第一個 False 就停、整體為 False（後面不算）；or 遇到第一個 True 就停、整體為 True。所以把「最可能失敗／成立」或「便宜」的條件放前面可省運算，也可避免後面出錯（例如 x != 0 and 10/x > 1）。

### 3.2 二、三種分支結構（招牌：條件分支流程）



```

# 1. if (單分支)
if score < 60:
    print('不及格')

# 2. if-else (雙分支)
if eng >= 60 and math >= 60:
    print('兩科都及格')
else:
    print('有科目不及格')

# 3. if-elif-else (多分支，成績等第)
if s >= 90: print('A')
elif s >= 80: print('B')
elif s >= 70: print('C')
elif s >= 60: print('D')
else: print('F')
  
```

### 3.3 三、縮排決定區塊（最重要）

同樣的程式碼，縮排不同、結果就不同：

```

# (A) print 縮排 → 屬於 if，只有不及格才印分隔線
if score < 60:
    print('!!! 不及格 !!!')
    print('=====') # 縮排：屬於 if 區塊
print('要繼續加油喔')

# (B) print 不縮排 → 屬於主程式，一定會印
if score < 60:
    print('!!! 不及格 !!!')
print('=====') # 沒縮排：不論及格與否都印
  
```

**縮排陷阱**

Python 用**縮排**界定區塊（慣例 4 個空格）。縮排多一層／少一層，程式碼就歸到不同區塊、執行結果不同；縮排不一致會 `IndentationError`。同一區塊的程式碼必須對齊。

**3.4 四、巢狀與三元運算子**

```
# 巢狀：分支裡再放判斷式（英文、數學各及格與否 → 四種情境）
if eng >= 60:
    if math >= 60: print('兩科都及格')
    else:          print('英文及格、數學要加油')
else:
    if math >= 60: print('數學及格、英文要加油')
    else:          print('兩科都不及格')

# 三元運算子：一行做選擇
result = '及格' if score >= 60 else '不及格'
```

**§ 4 語法與方法速查**

```
# 比較 / 布林
< <= > >= == !=      in not in is is not
and or not            # 優先序：not > and > or
5 < x < 10            # 連鎖比較 = x>5 and x<10

# 分支（注意冒號與縮排）
if 條件:
    區塊
elif 條件:
    區塊
else:
    區塊

# 巢狀：區塊內再縮排一層放 if
# 三元
值1 if 條件 else 值2
```

## § 5 常見錯誤

### 常見錯誤

- **縮排錯誤**：`if` 後的區塊要縮排；縮排不一致 → `IndentationError`，多縮／少縮 → 區塊歸屬錯、結果變。
- **忘了冒號**：`if` 條件後面要 `:`，漏掉 → `SyntaxError`。
- **用 `=` 當判斷**：條件要用 `==` 比較，寫成 `if x = 5:` 會 `SyntaxError`。
- **多分支用多個 `if` 而非 `elif`**：多個獨立 `if` 會逐一都判斷（可能多個成立都執行）；互斥分支該用 `elif`。
- **混淆 `and`／`or`**：`and` 全真才真、`or` 有真即真；複合條件記得加括號釐清。
- **連鎖比較寫錯**：想表達「`x` 在 5 與 10 之間」用 `5 < x < 10` 或 `x>5 and x<10`，不能寫 `5 < x and < 10`。

## § 6 練習題

### 練習 1 (一般題)：比較與布林

#### 引導步驟

1.  $x = 8$ ，印  $5 < x < 10$ 、 $x > 5$  and  $x < 10$ 、 $\text{not}(x \leq 5$  or  $x \geq 10)$ 。
2. 三者結果應一致。


#### 解答

```
x = 8
print(5 < x < 10)           # True
print(x > 5 and x < 10)    # True
print(not(x <= 5 or x >= 10)) # True
```



### 練習 3 (重要題)：成績等第 if-elif-else

#### 引導步驟

1. 給 score，用 if-elif-else 印等第：≥90 A、≥80 B、≥70 C、≥60 D、其餘 F。
2. 想想為什麼用 elif 而不是多個 if。


#### 解答

```
score = 85
if score >= 90: print('A')
elif score >= 80: print('B')
elif score >= 70: print('C')
elif score >= 60: print('D')
else: print('F') # 輸出 B
# 用 elif 因為等第互斥，命中一個就該停；多個 if 會每條都判斷
```

### 練習 4（一般題）：巢狀判斷式

#### 引導步驟

1. 給 eng、math，用巢狀 if 印出英數及格的四種情境之一。


#### 解答

```
eng, math = 88, 40
if eng >= 60:
    if math >= 60: print('兩科都及格')
    else:         print('英文及格、數學要加油')    # 此例輸出這行
else:
    if math >= 60: print('數學及格、英文要加油')
    else:         print('兩科都不及格')
```

### 練習 5（一般題）：三元運算子

#### 引導步驟

1. 給 `score`，用三元運算子一行決定 `result` 是「及格」或「不及格」並印出。


#### 解答

```
score = 55
result = '及格' if score >= 60 else '不及格'
print(result)      # 不及格
```

## § 7 自我檢核

- 能說出判斷式結果是布林，並用比較運算（`< == in is`）與布林運算（`and/or/not`）寫條件。
- 記得布林優先序 `not > and > or`，會用連鎖比較 `5<x<10`。
- 理解短路運算：`and` 遇 `False` 停、`or` 遇 `True` 停。
- 會寫 `if` / `if-else` / `if-elif-else` 三種分支，知道互斥分支用 `elif`。
- 能解釋縮排決定區塊，知道縮排錯會改變結果或 `IndentationError`。
- 會寫巢狀判斷式處理多條件組合。
- 會用三元運算子 `值1 if 條件 else 值2`。
- 認得錯誤：忘冒號 / 用 `=` 當判斷 → `SyntaxError`；縮排不一致 → `IndentationError`。