

# 第五單元

## 程式的除錯

程式的除錯是所有程式設計者的惡夢，本單元就來分享一些程式除錯心得。程式設計常見的錯誤分別是指令與語法拚字錯誤、演算法錯誤等，分別說明如下：

### 指令拚字錯誤

拚字錯誤是初學者最容易發生的錯誤，但是很多整合開發環境已經幫您將程式解析，給予提示，下圖左是 Spyder 的解析畫面，下圖右是 Python 官版解譯畫面，兩者都能夠將指令、函式、常數解析，給予不同的顏色。其次，Spyder 更強，還會給指令提示，幫忙補冒號、幫忙將對稱的括號給顏色、幫忙將錯誤語法給提示。例如，下圖左 `as` 變數名稱就不行，因為誤用保留字了，且『`as`』前已經出現錯誤信息，由 Spyder 所標示顏色，您會發現『`as`』是保留字。

```
D:\jupyter\exam\p3.py
a1.py x p3.py x
1 as=2;b=-7;c=3#變數先指派
2 d=(b*b-4*a*c)**(1/2)
3 x1=(-b+d)/(2*a)
4 x2=(-b-d)/(2*a)
5 print(x1)#3.0
6 print(x2)#0.5
7
```

```
File Edit Format Run Options Wind
as=2;b=-7;c=3#變數先指派
d=(b*b-4*a*c)**(1/2)
x1=(-b+d)/(2*a)
x2=(-b-d)/(2*a)
print(x1)#3.0
print(x2)#0.5
```

## 演算法錯誤

以上程式都對了，但結果就是不對，此時就要一步一步執行與觀察變數的變化。觀察的方法有兩種，分別是自己用 `print()` 輸出變數內容，或使用編譯器所提供的 `Debug` 工具，本書僅介紹前者。

### `print()`

`print` 就是在程式中，認為可能發生錯誤的地方輸出變數的內容。例如，程式原本如下：

```
a,b,c=eval(input("input a,b,c:"))
d=(b*b-4*a*c)**(1/2)
x1=(-b+d)/(2*a)
x2=(-b-d)/(2*a)
print(x1)#3.0
print(x2)#0.5
```

那我們可以中途先輸出變數內容 `a,b,c,d`，這樣可以縮小範圍，找出程式哪裡錯。

```
a,b,c=eval(input("input a,b,c:"))
print(a,b,c)
d=(b*b-4*a*c)**(1/2)
print(d)
x1=(-b+d)/(2*a)
x2=(-b-d)/(2*a)
print(x1)#3.0
print(x2)#0.5
```

### 自我練習

請每兩個同學一組，每個同學將以上範例程式故意製造一個錯誤，再請另外一個同學除錯。（請大家複習第 2 單元，將變數故意弄錯或使用保留字等）