



學習 BASIC 語言

常有一些錯誤觀念

■ 尹玫君

一、前言

學習設計程式不是件容易的事，尤其對初學者來說，在學習電腦語言的初期，常常會產生一些困擾。一般而言，我們可以把一些初學者常有的困擾歸成五大類；

第一類是關於程式語言的定位問題，也就是說初學者常常不太明瞭學習設計程式是爲了什麼，可以用它來處理什麼樣的問題，以及花了那麼多時間、精力學習了設計程式的技能與知識，對我們有什麼益處。

第二類是初學者對於學習程式設計時，所以必須使用的有關電腦的一些基本設備及觀念，並不是很了解，尤其只要初學者將一些觀念上的電腦設備與實際的運作情形配合起來時，

更是困難。

第三類是與學習不同形式程式語言有關的語意及語法的一些問題。在學習設計程式時必須先了解所學的程式語言中的語意及語法及每一個敘述、指令的功能，如初學者對這些不甚了解，往往會產生困擾。

第四類是與獲得一些常用來解決問題的模式、結構有關的困難；例如如何知道可以用迴路（loop）的方式來計算數的總和。

第五類的困擾是與程式設計有關的一些實際的問題，例如學生需學會那些技巧，以便能設計程式，測試程式及找出程式中的錯誤。

當然，以上所提出的這五種困擾，它們彼此之間並不是完全沒有關連的，而常常是彼此很難斷然的區分，尤其當初學者在學習之初，

要同時面對所有這許多困難時，學習程式設計所造成的困難及產生錯誤的複雜性，更是可想而知。

因著上面所提及的初學者所產生的困擾，而引起初學者在學習 BASIC 語言時易犯的一些錯誤，大致可以歸納成下面幾種重要的類型，提供老師在教初學者 BASIC 程式語言時要特別的注意。

一、指定的敘述 (LET)

在 BASIC 程式語言中，不對稱的指定敘述常常會造成初學者的困擾。例如：

```
LET A = B
```

這個敘述的意思，實際上是將存放在 B 裡的值指定給 A，而初學者往往根據以前所學數學的觀念，認為是將 A 的值指定給 B，因此他們會認為 `LET 3 = A` 是正確的，而 `LET A = 3` 是不對的，尤其是當他們看到。

```
LET A = A + 1
```

這樣的敘述時，更認為是不可能成立的，因為他們對這個指定敘述執行的順序不了解，因而便根據他們的舊經驗來解釋，認為等號兩邊的 A 代表同樣的意識。其實在這個指定敘述中，等號兩邊的 A 並不相等，一個是代表位置，而另一個是代表一個值。初學者往往不了解變數所代表的真正意義，因而造成錯誤。

另外，有些初學者看到

```
LET B = A + 1
```

```
LET A = B
```

的敘述時，如不經過進一步的說明，也會造成一些錯誤的觀念，因此老師在教的時候應該特別的說明強調，每一個變數，一次只能儲存一個值，如不說明清楚，初學者不瞭解這個敘述中新的值會將已經存在的值覆蓋上去，而仍以爲這個被覆蓋了的值仍存在記憶體的某個地方，如需要時仍可隨時叫出來使用。

另有些初學者認為 `LET A=B` 的敘述中

，A 與 B 是相同的，所以所有發生在 A 的事件也同樣的會發生在 B。這些例子都是在說明教程式設計計算部份時，牽涉到同一 (identity) 和相等 (equal) 的觀念時，常常感到困難的地方。

此外，還有一些容易產生的錯誤，是關於指定敘述的暫時性的觀念，也就是說變數裡的值並不會自動清除，除非被更改、記憶體的內容被清除，或是把機器關掉才會改變。最常發生的錯誤是，使用了一個變數來儲存計算的累加值，而在執行的最初忘記將它設定爲零，因此當再次執行程式時，變數的內容會有不同：

還有，有些初學者看到：

```
LET A = 2
```

```
LET B = A
```

的敘述時，常常會認為當把一個變數的值，指定給另一個變數時，會使變數裡的內容也移走了。因此在這個敘述中，初學者會認為 A 變數裡已經空了，而 B 變數裡的內容爲 2。

二、PRINT 的敘述

初學者對於 PRINT 這個敘述也會產生一些錯誤的觀念，這種錯誤說明了初學者對語言的特性不了解，因而有許多不同的解釋方式。如在 PRINT 指令中用到的引號。

```
PRINT "A:" ; A
```

許多初學者會忽略在引號中的文字，而僅僅將 A 的值印出來，或是認為引號只是會將變數的值印出來而已，印出的結果是 4:4。另有人認為一旦要印出一個變數時，所有包含在變數中的值，都會被印出來。

三、Read 的敘述

Read 這個敘述也常引起初學者在學習 BASIC 語言時的困擾。許多與 Read 有關問題的產生是由於初學者認為 Read 這個指令可

以自行自 Data 這個敘述中，選擇適當的值。通常這是由於初學者過度的相信電腦的能力所造成的。初學者會認為 Read 的敘述中，如使用一個有意義的變數，會使程式自動根據變數的意義而自 Data 的敘述中，選擇適當的資料，而不是按照資料排列的順序。

例如在下列的程式中

```
10 Read   Smallest
20 Read   First
30 Read   Second
40 Read   Third
50 Read   First
100 Data  50, 19, 1, -10, 29
```

有些初學者會認為 Smallest 這個變數裡的值是 -10，而 50 是屬於變數 First 的，Second 的值是 19，而 Third 的值是 1，他們以為有意義的變數名稱可以讓電腦自行決定自 Data 中所讀取的值。

四、迴路 (loop)

另外有一些錯誤是因為迴路的結構所造成的。如在下列這個程式中，PRINT 敘述跟在一個迴路後面，而初學者往往把它當做是在迴路中，而重覆的執行。

```
10 LET S=0
20 Read A
30 IF A=0 Then Go To 70
40 IF A<0 Then Go To 60
50 LET S=S+A
60 Go To 20
70 PRINT "A=" ; A
80 Data 5, -8, 10, 25, -3, 43,
      72, -31, 0
90 END
```

有些初學者認為每執行迴路一次，就會將 A 的值印出一次來。

另外，像 FOR……NEXT 的迴路也容易讓初學者產生錯誤的觀念，常常初學者會不太明白在 FOR……NEXT 迴路中的控制值每次都會改變的現象。

例如

```
10 FOR I=1 To 8
20 PRINT I;
30 NEXT I
40 PRINT I
```

初學者往往認為最後的結果將會印出 1 2 3 4 5 6 7 8，而不知道 I 的值最後會變成 9，忽略了印出 9 的結果。

IF 的敘述也同樣的會造成一些困擾，初學者對程式的執行或是在 IF 敘述之後的控制流程，常有些不正確的觀念。尤其當碰到 IF 後面的條件不成立時，程式的流程不知該往何處繼續。

另像下面的程式中

```
10 A = 3 : B = 6
20 IF A < B Then A = A+B
30 IF A > B Then A = A-B
40 PRINT A, B
```

初學者通常認為結果是 9 和 6，而忽略去執行第 30 號。

程式設計工作或是程式語言的學習非常複雜，即使專門為了初學者而設計的程式語言如 BASIC，仍舊有非常多的地方，如稍不注意或觀念不甚清楚，就會造成初學者的許多錯誤。因此，在教 BASIC 語言的同時，應該讓初學者自程式語言如何被執行的角度來了解電腦內部到底是如何工作的，這樣可以幫助初學者對程式執行流程的認識。另一方面老師在教 BASIC 語言時，充份了解那些敘述會造成初學者的困擾或是產生不正確的觀念，在教授，講解這些敘述時，應該特別的注意，詳細的說明和解釋。

(作者：省立台南師院電算中心主任)