第3單元

資料的運算

前面我們已經介紹資料的數位化，本單元則要進行資料的運算，要進行資料的運算必須先介紹一些電腦的基本運算思維。例如，運算元、資料型態、運算子、算術運算、變數宣告、識別字、保留字等等，分別說明如下：

運算元與資料型態

一般的計算機進行算術運算，就直接運算，例如進行 3 * 2，就直接按『3』、『*』、『2』、『＝』就可得到答案，但是以上步驟當要重複計算時，都要重複按以上運算元與運算子（3 與 2 稱爲運算元，*與 = 稱爲運算子），這樣非常耗時與不便，所以就有電腦程式語言的發展。所謂電腦程式設計，就是利用代號儲存以上運算元，這些代號在程式設計領域稱爲變數，意思是說，它所代表的值，運算過程隨時可以改變，再使用變數寫出計算結果的敘述，結果也是先以變數儲存，那此一敘述就可重複利用，例如，計算長方形面積，我們會先將資料數位化：

```
int a=3,b=4;
int c;
```

計算結果與輸出結果如下：

```
c=a*b;
Serial.print(c);
```

以上代號 a, b, c，在程式設計領域裡，我們稱爲『變數, variable』、『int b=3;』稱爲變數宣告與指派初值；『a*b』稱爲『運算式, Expression』；『c=a*b;』稱爲『敘述, Statement』。其次，在上一單元的資料數位化單元，我們已經介紹不同的數字（整數、負整數與實數）與不同的數字的大小（數字的位數），其數位化的方式也不同，所佔用的記憶體大小也不同。電腦爲了有效率的處理這些資料，就有資料型態（Data Types）的規劃，也就是大的資料用大盒子裝，小的資料用小盒子裝，如此才可節省記憶體，並加快處理效率。反過來說，若不分資料大小，通通用大盒子裝資料，那將會非常浪費記憶體，也拖垮執行效率。例如，所有的東西都用冰箱的盒子裝當然也可以，但這樣非常浪費空間，還有，搬運時也很耗時。Arduino 所提供的資料型態、所佔用記憶體、所能代表的數值範圍如下表：

資料型態	中文名稱	佔用記憶體的大小（位元）	所能代表的數值的範圍	備註
byte	位元組	8	0 ～ 255	
int	整數	16	-32768 ～ 32767	
long	長整數	32	-2147483648 ～ 2147483647	
float	浮點數	32	+/-3.4E+-38	
double	倍精度浮點數	32	+/-3.4E+-38	Arduino 的 double 同 float
unsigned char	正字元	8	0 ～ 255	
char	字元	8	-128 ～ 127	
unsigned long	正長整數	32	0 ～ 42949667295	

資料型態	中文名稱	佔用記憶體的大小（位元）	所能代表的數值的範圍	備註
unsigned int	正整數	16	0 ～ 65535	
bool	布林	8	true or false	boolean 也可，但不鼓勵
String	字串			

變數宣告

變數的功能是用來輸入、處理及儲存外界的資料，而變數在使用以前則要事先宣告才可使用。在一些舊式的 Basic 語言中，變數使用前並不需要事先宣告，卻也帶來極大的困擾。例如，以下敘述即爲變數未宣告的後果：編譯器無法回應使用者在拼字上的錯誤，而造成除錯上的困難。

```
student=studend+1;
```

上式若事先宣告 student 如下：

```
int student;
```

則編譯器遇到 studend 時，便會出現 studend 未宣告的錯誤訊息，提醒使用者補宣告或注意拼字錯誤。其次，變數宣告的優點是可配置恰當的記憶體而提高資料的處理效率。例如，有些變數的值域僅爲整數，則不用宣告爲 float。此即爲小東西用小箱子裝，大東西用大箱子裝，才能有效運用空間與提升搬運效率。Arduino 語言的變數宣告語法如下：

```
資料型態 變數名稱[=初值];
```

例如，

```
byte a;
```

即是宣告變數 a 爲 byte 型態，佔用 1 個 Byte，此種型態僅能儲存 0 到 255。又例如，

```
int d;
```

那 d 就佔用 2 個 Byte，但可儲存 -32768 到 32767。又例如，

```
char b,c;
b='A';c='*';
```

則是宣告變數 b, c 爲 char 型態，此種型態可儲存單一字元。變數的宣告亦可連同初值一起設定，如以下敘述：

```
float d = 30.2;
```

宣告 d 是 float 型態，並設其初值是 30.2。以下敘述，宣告 e 是 char 型態，且其初值是字元 'a'。

```
char e='a';
```

以下敘述，同時宣告兩個變數，且設定其初值。

```
int f1=3,f2=3;
```

以下敘述可宣告布林型態:（布林型態用來表示，運算結果的『眞』與『僞』）

```
bool g=true;
```

C/C++/Arduino 都可用字元陣列表示字串，以下程式可宣告 a 爲字串型態，請留意字元是單引號，字串是雙引號。

```
char a[]="ABC";//字串用雙引號
```

C++/Arduino 才有字串型態 String，以下程式可宣告變數 h 爲字串型態：

```
String h="123";
```

變數經過宣告之後，編譯器即會根據該變數的資料型態配置適當的記憶體儲存此變數，所以若要提高程式的執行效率，則應儘量依照資料性質，選擇佔用記憶體較小的資料型態。

變數的有效範圍

任一變數的宣告，若無特殊聲明，均屬於區域變數，其有效範圍僅止於該變數所在的程式區塊。所以，以下變數 i 的有效範圍僅在 setup()，無法在 loop(){} 函式內存取。

```
void setup() {
  Serial.begin(9600);
  byte i=1;
}
void loop() {
  Serial.println(i);
}
```

其次，請比較下圖左與下圖右的差別，下圖左 byte i=0; 放在 void loop() {} 裡面，則每次執行 void loop() {} 時，byte i=0 都被執行，所以其值永遠都相同，沒有累加效果。此時就要把此敘述放在外面，成爲全域變數，所以此 i，稱爲計數器，也可以想像成 loop() 被執行的次數，如下圖右。

```
void setup() {
  Serial.begin(9600);
}
void loop() {
  byte i=0;
  i=i+1;
  Serial.println(i);
}
```

```
void setup() {
  Serial.begin(9600);
}
byte i=0;
void loop() {
  i=i+1;
  Serial.println(i);
}
```

變數有效範圍的優點

因爲一個大型程式通常是很多人合力完成，每個人各自開發自己的函式，各自宣告自己的變數。例如，以下變數 a，彼此獨立，互不干擾。

```
void setup() {
  Serial.begin(9600);
  byte a=1;
  Serial.println(a);
}

void loop() {
  byte a=2;
  Serial.println(a);
  delay(1000);
}
```

若沒有區域變數的觀念，彼此的變數就會互相干擾，例如，以下程式，您用 a，我也用 a，那彼此就互相干擾，執行結果就會錯亂。

```
void setup() {
  a=a+1;
}
void loop() {
  a=a+3;
}
```

若需要共用變數時，大家要先講好，將這一變數宣告爲全域，就可共用。例如，以下變數 a 稱爲全域變數，大家一起共用。

```
byte a=0;
void setup() {
  Serial.begin(9600);
  Serial.println(a);
  a=a+1;
  Serial.println(a);
}
```

```
void loop() {
  Serial.println(a);
  a=a+3;
  Serial.println(a);
  delay(1000);
}
```

識別字（Identifiers）命名規則

眞實的世界裏，每個人、事及物都有一個名稱，程式設計亦不例外，於程式設計時我們必須爲每一個變數、常數、函式命名，以上所有變數、常數、函式等名稱，統稱爲程式語言的識別字（Identifiers）。先以數學爲例，數學的變數命名規則是，已知數用 a, b, c，未知數用 x, y, z；物理則習慣用 v 代表速度，t 代表時間；而 C/C++/Arduino 語言的識別字命名規則如下：

1. 識別字必須是以字母（大小寫的 A 至 Z）或底線（_）開頭。例如，以下是一些合法的識別字。

```
a
i
sum
_sum
Income
```

以下是一些非法的識別字。

```
7eleven      //  不能由數字開頭
%as          //  不能由符號開頭
```

2. 識別字由字母開頭後，僅可由字母、底線及數字組合而成，且不得包含空白與符號。例如，以下是一些合法的識別字。

```
a123
a123b
_a_b
```

以下是一些非法的識別字。

```
A=          //  不能含有 = 號
sum!        //  不能含有 ! 號
Age#3       // 不能含有 # 號
a c         //  不能含空白
c+3         //  不得含有加號
```

3. 識別字的大小寫均視爲不同，例如 Score、score 及 SCORE 皆代表不同的識別字。
4. 識別字不得使用保留字，如 if、for 等。但是，if1、fora 等則可使用。
5. 識別字要用有意義的單字，例如，sum、avg、StudentNumber 或 AverageIncome。除非有效範圍很小（或稱生命週期極短）的變數才用 x、i 或 a 等當識別字，也千萬不要用 k23erp 等這種沒意義又難記的識別字。
6. 識別字有多個單字時，中間可以加上底線（_），例如上例的 StudentNumber 可寫成 student_Number，若擔心打字不靈光亦可寫成 Stu_Num、stu_num、stuNum 或 stunum，其中 stuNum 又稱駝峰表示法，因爲大寫字母看起來像駱駝駝峰一樣，這樣可以避免鍵入底線的困擾，且提昇閱讀效率，Arduino 的函數命名則採用此種命名習慣。

保留字（Keywords）

保留字（又稱關鍵字）是任一程式語言已事先賦予某一識別字（可識別的文字或字串，稱爲識別字）一個特別意義，所以程式設計者不得再重複賦予不同的用途，就像現實生活中，電視、冰箱已經有特別意義了，所以沒有人取名字爲電視或冰箱。Arduino 也是一樣，例如 if 已賦予決策功能，程式設計者當然不得再定義

if 爲另外的用途，請自行線上查詢點選工具列的『說明 / 參考文件』或『Language Reference』，例如表內的 loop、setup、break、coutinue、do 等等，大部分擷取 C/C++ 語言精華。

運算子

所謂運算子（Operator），指的是可以對運算元（Operand）執行特定功能的特殊符號。例如，3 ＋ 2 的『3』與『2』稱爲運算元，『+』稱爲運算子。Arduino 的運算子分爲五大類，分別是：算術（Arithmetic）運算子、比較（Comparison）運算子、布林（Boolean）運算子、位元操作（Bitwise）運算子及複合（Compound）運算子，分別說明如下：

■ 算術運算子

下表是 Arduino 的算術運算子。（請開啓 https://www.arduino.cc/reference/en/）

Arithmetic Operators

% (remainder)

* (multiplication)

+ (addition)

- (subtraction)

/ (division)

= (assignment operator)

以上算術運算子用來執行一般的算術運算，包括指派 (=)、取正負數 (+/-)、加 (+)、減 (-)、乘 (*)、除 (/)、取餘數 (%) 等，下表是以上算術運算子的功能說明：

運算子	定義	優先順序	結合律
=	指派	15	由右至左
+/-	正負號，一元運算子	2	由右至左
*	乘法運算	4	由左至右
/	除法運算	4	由左至右
%	求餘數（Modulus）	4	由左至右
+/-	加法 / 減法運算	5	由左至右

■『=』

『=』符號爲指派或稱賦值運算子，其作用爲將運算符號右邊運算式的值，指派給運算符號左邊的運算元。所以，以下敘述 sum=a+b 是先計算 a+b 的值，再將此值指派給 sum。

```
int sum = 0,a = 3,b = 5;
sum = a + b;
```

上式與數學的『相等』符號是不同的，在程式設計領域裡，此稱爲指派或賦值，右邊先運算，運算後再將結果指派給左邊的變數，所以不要一直糾結爲什麼 0 會等於 8。其次，你是不能將常數放在指派運算子的左邊，例如：

```
8 = x ;
```

爲一個不合法的敘述，但以下敘述將常數 8 指派給變數 x 是合法的。

```
x = 8 ;
```

■四則運算

以下是一些簡單四則運算：

```
int a=5,b=4;
Serial.println(a+b);//9
Serial.println(a-b);//1
Serial.println(a*b);//20
Serial.println(a%b);//1 取餘數
```

以上程式，請放在 setup() 裡面，就可觀察執行結果。

```
void setup() {
  Serial.begin(9600);
  //執行一次的放這裡
}
void loop() {}//雖然沒用到，也不能省略
```

■整數除法或實數除法

Arduino/C/C++ 的除法運算，只有被除數與除數的型態均爲整數，才是整數除法，商的型態爲整數；否則即爲實數除法，得到實數商。例如：

```
int x=5, y=4,z;
float xf=5,yf=4;
Serial.println(x/y); // 1，被除數與除數的型態均爲整數
Serial.println(xf/y);//1.25
Serial.println(x/yf);//1.25
Serial.println(xf/yf);//1.25
```

其次，若運算結果爲實數，但若指派給整數型態的變數，結果也是整數。例如：

```
void setup() {
    Serial.begin(9600);
    int x=5, y=4,z;
    float xf=5,yf=4,zf;
    zf=xf/yf;//1.25
```

```
    Serial.println(zf);
    z=xf/yf;//1
    Serial.println(z); //原本運算結果是實數1.25，但指派給整數，所以會是1
}void loop() {}
```

■整數除法與取餘的應用

整數除法與取餘可以將一個整數分解爲數個數字。例如，若要使用七段顯示器顯示 152，那就要將此數字分解爲 1, 5, 2 三個數字，其方法如下：

```
int a=152;
int a1=152/100;//1百位數
int a2=(152-a1*100)/10;//5十位數
int a3=a %10;//個位數
```

■取餘

若要讓一個數字在累加的過程中，保持一定的循環，那也是用取餘。例如，

```
int i;
void loop() {
    i=(i+1) %4;
}
```

那 i 就永遠在 0, 1, 2, 3 循環。請特別留意，數學有

```
19/10=1…9
```

的書寫習慣，但電腦並沒有此運算方式，因此以上運算方式，一定要先取餘數，再求整數商如下：

```
int a=19,b=10;
int c=a%b;
int d=a/b;
```

順序錯也不行，以下程式先除再取餘，結果就不對。

```
int a=19,b=10;
int d=a/b;
int c=a%b;
```

運算子的優先順序（Precedence）

同一敘述，若同時含有多個運算子，此時即需定義運算子的優先順序。例如，

```
a=3+4*2;
```

『=』優先順序是 12，『+』優先順序是 5 ，『*』優先順序是 4，所以運算順序是

```
(a=(3+(4*2)));
```

運算子的結合律（Associativity）

當同一敘述，相鄰的運算子擁有相同優先順序的運算子，此時即需定義運算子是左結合或右結合。例如：

```
x=a-b-c;
```

連續兩個減號『-』，優先順序相同，此時就要靠定義結合律，減法結合律是由左至右，所以以上同義於

```
x=((a-b)-c);
```

而

```
x=y=z=2;
```

連續三個指派運算子『=』，指派運算子的結合律是由右至左，所以以上式子同義於

```
(x=(y=(z=2)));
```

所以，以上敘述，x、y、z 的結果都是 2。

範例 3a

請寫一個程式，可以指派長方形的長與寬，並計算周長與面積。

題目分析

1. 需要兩個變數儲存長與寬。請選擇此資料來源是整數還是浮點數，若是整數，還要分是 8 位元的 byte (0 ～ 255)、16 位元的 int (-32768 ～ 32767) 或 32 位元的 long (-2147483648 ～ 2147483647)。本書整數若沒特別註明，就一律折衷，通通取 int。
2. 需要兩個變數儲存面積與周長。請選擇整數或是浮點數，本例也是選擇 int。
3. 先使用變數指派方式，指派變數的值，程式如下。這樣可以省略冗長的變數輸入，先專注於演算法的實現。

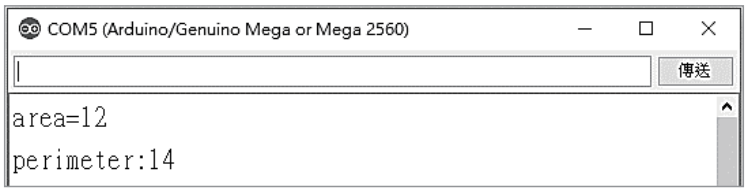
```
void setup() {
    Serial.begin(9600);
    int a,b,area,perimeter;
    a=3;
    b=4;
    area= a*b;
    perimeter=2*(a+b);
    Serial.print("area=");
    Serial.println(area);
    Serial.print("perimeter:");
    Serial.println(perimeter);
}void loop() {}
```

4. 請特別留意

```
area= ab;
perimeter=2(a+b);
```

爲數學語言，不是電腦語言，以上寫法電腦看不懂。

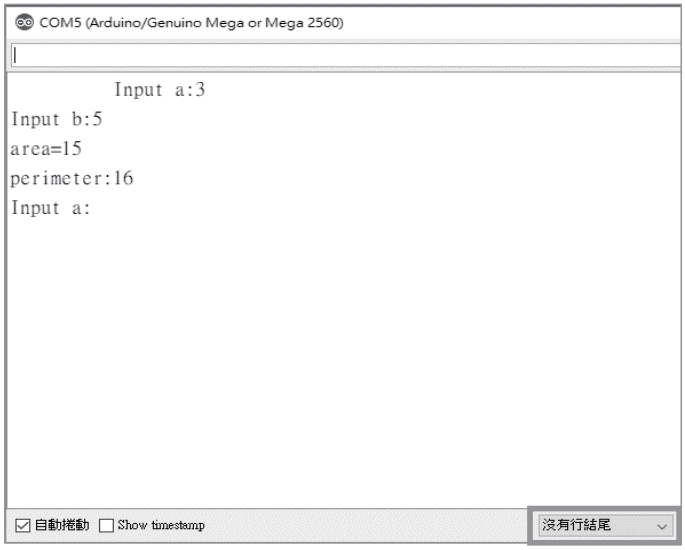
5. 以上程式執行結果如下：



6. 使用電腦的鍵盤輸入變數的數值，程式如下：

```
void setup() {
    Serial.begin(9600);
}
void loop() {
    int a,b,area,perimeter;
    Serial.print("Input a:");
    while(Serial.available() ==0) {}//等待使用者輸入資料
    a=Serial.parseInt();
    Serial.println(a);
    Serial.print("Input b:");
    while(Serial.available() ==0) {}//等待使用者輸入資料
    b=Serial.parseInt();
    Serial.println(b);
    area= a*b;
    perimeter=2*(a+b);
    Serial.print("area=");
    Serial.println(area);
    Serial.print("perimeter:");
    Serial.println(perimeter);
  }
```

6. 以上程式執行結果如下:(請將輸入方式點選『沒有行結尾』，如下圖，不然無法輸入第二個變數)



自我練習

1. 指派長方體的長、寬、高，計算其表面積與體積。
2. 請寫一個程式，可以指派一個台斤數，且轉爲公斤數輸出。
3. 請寫一個程式，可以指派一個 0 ～ 86399 的整數，且轉爲『時：分：秒』的格式輸出。
4. 請寫一個程式，可以指派一個 4 位數，並從千位數一一輸出、且求其數字和。例如，指派 1234，那就輸出 1,2,3,4 與 10。
5. 請寫一個程式，可以指派一個 4 位數，並從個位數一一輸出、且求其數字和。例如，指派 1234，那就輸出 4,3,2,1 與 10。

6. 請寫一程式，滿足以下條件。可以指派兩個座標。(x1=0;y1=3;x2=4;y2=0)，然後計算此兩點座標距離，輸出距離。

   提示：已知兩點座標分別是 (x1,y1),(x2,y2)，則其距離的公式的數學語言是：

$$d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

7. 指派三角形三邊長 a、b、c，求其面積。

   提示：先指派 a,b,c，然後計算 d=(a+b+c)/2，則三角形面積 s $=\sqrt{d(d-a)(d-b)(d-c)}$ ，輸出 s。本例假設所輸入的三角形三邊長可圍成三角形，例如指派 a=3;b=4;c=5 則得三角形面積 6，但並不是任意三條線都可圍成三角形，若要判斷是否可圍成三角形，請繼續研讀下一單元。