

單元一：資料型別簡介

C 語言的資料型別：

	基本資料型別					複合資料型別			
型別	整數	浮點數	字元	字串	無值	結構	聯合	列舉	檔案
表示法	int	float	char	Char s[]	void	struct	union	enum	file

一、變數宣告：C 語言中所有的變數、函數、指標(pointer)，皆須宣告後才能使用。

【格式一】 資料型別 變數名稱； int a,b;

【格式二】 資料型別 變數名稱=初值； float p=3.14;

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a=1;
    float b=3.6;
    printf("a=%d\n",a);
    printf("b=%f\n",b);
    system("PAUSE");
    return 0;
}
```

執行結果
a=1
b=3.600000

二、資料型別所佔空間及詮釋法：

佔 byte	有號整數	數值範圍	無號整數	數值範圍	浮點數	有效位數	數值範圍
1	char	-128~127	unsigned char	0~255	無	無	無
2	(short)int	-32768~32767	unsigned int	0~65535	無	無	無
4	(long)int	-2147483648~2137483647	unsigned long	0~4294967295	float	7	3.4E-38~3.4E+38
8	無	無	無	無	double	15	1.7E-308~1.7E+308
10	無	無	無	無	long double	19	3.4E-4932~1.1E+4932

三、整數

1.宣告格式 int 變數名稱=初值； int a=5;

2.整數變數在記憶體中佔 2 個位元組(16 bits)。

3.整數常數可表示成十進位、八進位、十六進位。

(1)八進位整數常數之前需以 0 作前導字元，例如 0123 表示八進位的 123

(2)十六進位整數常數之前需以 0x 作前導字元，例如 0x123 表示十六進位的 123

四、字元

1.宣告格式 char 變數名稱=初值； char c='a';

2.字元常數必須以一對單引號括起來。例如 'a'

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char a='A';
    printf("a 字元為%c\n",a);
    system("PAUSE");
    return 0;
}
```

執行結果

a 字元為 A

五、字串

- 1.宣告格式 `char 字串變數[字串長度]=字串初值;` `char name[5]="Mary";`
- 2.字元常數必須以一對雙引號括起來。例如 `"happy day!"`
- 3.字串最長為 65535 個字。
- 4.字串以 `\0` 為字串結束符號，假若宣告字串長度為 `n` 時，最多只能存放 `n-1` 個字元。

M	a	r	y	\0
---	---	---	---	----

例：

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char a[]="welcome!";
    printf("a 字串為%s\n",a);
    system("PAUSE");
    return 0;
}
```

執行結果

a 字串為 welcome!

- 六、void 型別（無值型別）非使用於一般變數上，而是定義函數的型別。

```
#include <stdio.h>
#include <stdlib.h>
void star();
int main() //main()主程式
{
    printf("call procedure star()\n");
    star(); //呼叫 star()函數
    system("PAUSE");
    return 0;
}
void star() //star()函數
{
    printf("*****\n");
}
```

執行結果

call procedure star()

單元二：變數、運算符號與運算式

一、變數：變數名稱由英文字母、數字、或 _ (底線符號)構成，但第一個字元不可以是數字；變數名稱最長可設 32 個字元。例如 my_name。

二、算符

1. 算數算符：

(1) 四則算符、% 與 =

算術運算符號	範例	執行結果 (若 a=13,b=5)
+	a+b	18
-	a-b	8
*	a*b	65
/	a/b	2
%	a%-b	3 取 a 除以 b 的餘數 (a 及 b 必須為整數)
=	a=a+1	a=14 將 '=' 右邊算式的值設定給左邊的變數，且 '=' 左邊必定是一個單一變數

(2) 簡化的運算式

= 與運算符號的合併

運算符號	範例	相當於
+=	a+=b	a=a+b
-=	a-=b	a=a-b
=	a=b	a=a*b
/=	a/=b	a=a/b
%=	a%=b	a=a%-b

例

<pre>#include <stdio.h> #include <stdlib.h> int main() { int a=51,b=23; printf("原始的 a=%d,\t b=%d \n",a,b); a+=b; printf("a+=b 後 a=%d\t b=%d\n",a,b); system("PAUSE"); return 0; }</pre>
<p>執行結果</p> <p>原始的 a=_____ b=_____</p> <p>a+=b 後 a=_____ b=_____</p>

++與--算符

算符	功能	範例	執行結果
++	遞增：將變數值加 1 ++a 或 a++相當於 a=a+1 即 a 的值加 1	a=3; a++;(或++a;) b=a*2;	a=3 相當 a=a+1 即 a=3+1 故 a=4 b=4*2 故 b=8
--	遞減：將變數值減 1 --a 或 a--相當於 a=a-1 即 a 的值減 1	a=3; a--;(或--a;) b=a*2;	a=3 相當 a=a-1 即 a=3-1 故 a=2 b=2*2 故 b=4

1. 敘述句 i++ 與 ++i 的意義相同，都是將變數 i 值加 1，即 i=i+1

2. 若牽涉到變數值的設定，則遞增++、遞減--符號使用，則需留心，如下：

前置：++或--在變數前，表示++或--是先加 1 或減 1 後再設定給其他變數。

後置：++或--在變數後，表示++或--是先設定給其他變數後再加 1 或減 1。

	範例	執行結果
前置	a=2; b=++a;	a=3 b=3 即 a 值先加 1 之後再設定給 b
後置	a=2; b=a++;	a=3 b=2 即 a 值先設定給 b 之後 a 值加 1
前置	a=2; b=--a;	a=1 b=1 即 a 值先減 1 之後再設定給 b
後置	a=2; b=a--;	a=1 b=2 即 a 值先設定給 b 之後 a 值減 1

2. 關係算符：即比較用的算符，其運算結果只會傳回真(整數 1)或假(整數 0)。

執行優先順序	關係算符	說明
1	<	小於
	>	大於
	<=	小於等於
	>=	大於等於
2	==	等於
	!=	不等於

3. 邏輯算符：且(&&)、或(| |)、否(!)，運算優先順序為 ! > && > | | ；運算式或數值都只有二個值 0(假)與 1(真)。
4. 位元算符：

- (1) <<(左移)、>>(右移)、&(且)、^(互斥或)、|(或)、~(取補數)
- (2) 數字在進行位元運算時，在電腦內部會先將數字轉換為二進位數後，再進行運算。
- (3) <<(左移)運算：將二進位數的每一個位元向左遞移 n 位，最左邊的位元遞移後捨去，最右邊的空位補 0

【格式】 數值<<n

範 例	說 明
a=3;	整數 3 的二進位為 000000000000011
a=a<<1;	所有位元左移 1 位 0000000000000110
結果 a 值為 6	還原成十進位為 $2^2*1+2^1*1+2^0*0=6$

- (4) 就數值計算而言：左移運算是將數值乘上 2^n 即可(n 為遞移位數)，右移運算是將數值除以 2^n
- (5) &(且)、^(互斥或)、|(或)、~(取補數)等運算方式：將數值先轉換為二進位後，以位元為單位進行 AND、OR、XOR 等運算；補數運算是取二進位數值之 1's 補數。

5. 條件算符

【格式】 (判斷條件)? 條件成立處理情形：條件不成立處理情形；

例：成績大於等於 60 則輸出"pass"否則輸出"down"

if (score>=60) printf("pass\n"); else printf("down\n");	以條件算符表示如下： (score>=60)?printf("pass\n");printf("down\n");
--	--

例：將變數 a 的絕對值設定給變數 b

b=(a>0)?a:-a;

三、算符的優先順序：算數算符 > 關係算符 > 邏輯算符