

副程式 Wisely



1040810更新

主程式

```
main()
```

```
{
```

```
    int x,y;
```

```
    x=5;
```

```
    y=x+1;
```

```
}
```

主程式以main為開頭，以大括號{}
做為開始及結束

副程式(又稱「函數」)

副程式為一個功能獨立的個體，其架構與主程式類似，副程式可以接受資料傳入自己，也可以將處理完的結果傳回呼叫它的程式，格式如下：

副程式型態 副程式名稱 (變數1, 變數2, ……)

副程式(又稱「函數」)

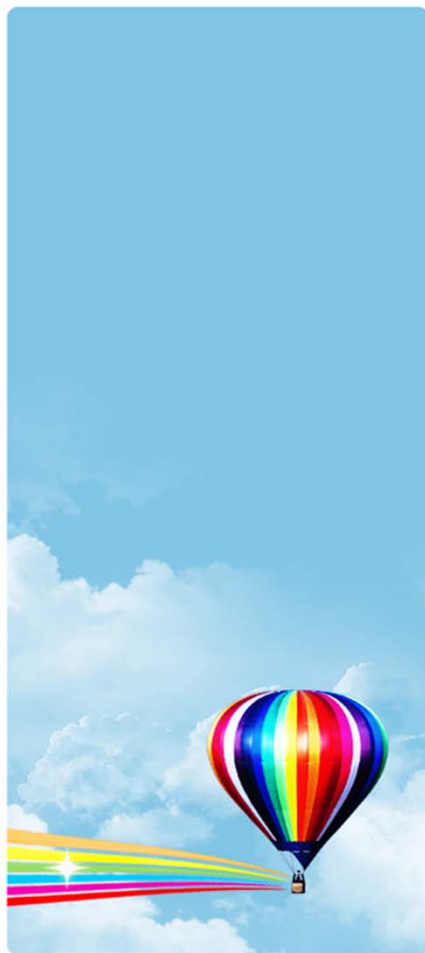
副程式型態 副程式名稱 (變數1, 變數2, ……)

副程式型態是指函數的傳回值型態，該值可以是C 語言中的任何一個資料型態，如未設定，則C 語言會自動設為整數型態。

傳入/傳回區分

依使用者須求，副程式可接受變數傳入計算後傳回，可分成以下四種方式：

1. 有傳入值，有傳回值
2. 有傳入值，無傳回值
3. 無傳入值
4. 無傳入值，有傳回值



有傳入值，有傳回值

```
#include <regx51.h>
```

```
int ABC(int c)
```

```
{
```

```
    c=2*c+1;
```

```
    return(c);
```

```
}
```

```
main()
```

```
{
```

```
    int i;
```

```
    i=11;
```

```
    P1=ABC(i);
```

```
}
```

```
//傳進來的值放在c 中(c=i=11)
```

```
//c=2*11+1=23
```

```
//返回並將c=23結果值傳回給呼叫
```

```
//程式
```

```
//設定 i=11
```

```
//呼叫ABC 副程式並傳入i 值
```

```
//執行後P1=23
```



有傳入值，無傳回值

```
#include <regx51.h>
```

```
void ABC(int c)
```

```
{
```

```
    c=2*c+1;
```

```
}
```

```
main()
```

```
{
```

```
    int i;
```

```
    i=11;
```

```
    ABC(i);
```

```
}
```

//傳進來的值放在c 中(c=i=11) ,

//c=2*11+1=23

//結果值不回傳

//設定 i=11

//呼叫ABC 副程式並傳入i 值。



有傳入值，無傳回值

```
#include <regx51.h>
```

```
void ABC(char c)
```

```
{
```

```
    c=2*c+1;
```

```
}
```

```
main()
```

```
{
```

```
    char i;
```

```
    i=11;
```

```
    ABC(i);
```

```
}
```

//傳進來的值放在c 中(c=i=11) ,

//結果值不回傳

//c=2*11+1=23

注意：ABC(i)不可以寫成接收回傳值的型式P1=ABC(i)，如此會造成錯誤。



無傳入值

僅呼叫程式不傳值給函數，則函數名稱後可空括號或為(void)。

```
#include <regx51.h>
```

```
void ABC(void)
```

//沒有傳入值

```
{
```

```
    int c=12;
```

```
    c=2*c+1;
```

//C=2*12+1=25

```
}
```

```
main()
```

```
{
```

```
    ABC();
```

//呼叫ABC 副程式

```
}
```



無傳入值，有傳回值

```
#include <regx51.h>
```

```
ABC(void){
```

```
    int c=12;
```

```
    int b=9;
```

```
    c=2*c+1;
```

```
    b=2*b+1;
```

```
    return(b);
```

```
}
```

```
main() {
```

```
    int P1;
```

```
    P1=ABC();
```

```
}
```

//不接收傳入值,但可以回傳結果值

//不指定回傳型態，預設為 int

//C=2*12+1=25

//b=2*9+1=19

//指定回傳 b=19 的值

//呼叫 ABC 副程式,P1=b=19 或 0x13



主程式與函數的位置

先寫主程式,後寫函數,函數必須先宣告

先寫函數,後寫主程式,函數不須宣告

先主後函,函數必須先宣告

```
#include <regx51.h>
```

```
int ABC(void);
```

//宣告 ABC 是一個整數函數,其型態
//必須與下面函數型態符合

```
main() {
```

```
    P1=ABC();
```

//呼叫 ABC 副程式,P1=b=19 或 0x13

```
}
```

```
int ABC(void) {
```

```
    int c=12;
```

```
    int b=9;
```

```
    c=2*c+1;
```

```
    b=2*b+1;
```

```
    return(b);
```

//不接收傳入值,但可以回傳結果值

```
}
```



先函後主,函數不須宣告

```
#include <regx51.h>
```

```
int ABC(void){ //不接收傳入值,但可以回傳結果值
```

```
    int c=12; int b=9; c=2*c+1; b=2*b+1;  
    return(b);
```

```
}
```

```
main() {
```

```
    P1=ABC();
```

```
}
```





課堂練習1

請設計二運算元算式，讓使用者可以在步驟一時選擇運算法，步驟二時輸入運算子，如

步驟一：輸入"+"

步驟二：輸入"3" 及 "4"

步驟三：程式執行 $3+4=7$ 並印出

...

再按一次"空白鍵"立即重複開始。

限用函數方式書寫程式。

(步驟一中，加減乘除及取餘數皆可輸入)

宣告技巧

- 全域宣告
 - 所宣告的變數是提供所有函數使用包括主程式。
- 區域宣告
 - 所宣告的變數則僅提供該函數使用。

全域宣告例：

```
#include <regx51.h>
```

```
int x=5;
```

//全域變數

```
test(){
```

```
    P2=x;
```

//因為 x 是全域

//宣告因此 P2 等於 5

```
}
```

```
main() {
```

```
    P1=x;
```

//P1 等於 x=5

```
}
```

區域宣告例：

```
#include <regx51.h>
```

```
test(){
```

```
    int x=5;
```

```
    P2=x;
```

```
}
```

```
main() {
```

```
    P1=x;
```

```
}
```

//宣告為區域變數

// x 已在此副程式宣告因

//此 P1 等於 5

//x 是區域宣告,因此 x 在此

//會出現會造成錯誤 }

1040810更新

課堂練習2 1/2：

家庭用電100度以下，每度2.5元

家庭用電101度以上，300度以下，每度3.3元

家庭用電301度以上，每度4.2元

工業用電契約馬力，每馬力150元

工業用電每度2.9元

營業用電300度以下，每度6元

營業用電301度以上，每度6.8元

課堂練習2 2/2：

請讓使用者可藉由輸入以下資訊，得到應繳電費

- 1)用電類型
- 2)用電度數
- 3)若是工業用電，需額外輸入契約馬力
- 4)將電費計算完成後，show在畫面上

課堂練習3

- 方程式方程式 $ax^2+bx+c=0$ ，使用者在輸入a、b、c值後。程式可計算出解答(有一解、無解、無限多解、有兩解...等)