

函數

函數

1

- 函數(function)
- 可以簡化主程式的結構，也可以節省撰寫相同程式碼的時間，達到程式模組化的目的
- 利用前置處理器不但可以完成簡單函數的定義，同時還可以含括所需的檔案到程式中
- C語言使用方式
 - 函數原型(prototype)宣告
 - 函數定義
 - 呼叫函數

範例-函數範例

```
#include <stdio.h>
#include <stdlib.h>
void star(void);           /* star() 函數的原型 */
int main(void)
{
    star();                /* 呼叫star函數 */
    printf("歡迎使用C語言\n");
    star();                /* 呼叫star函數 */
    system("pause");
    return 0;
}

void star(void)
{
    printf("*****\n");    /* 印出13個星號 */
    return;
}
```

範例詳解

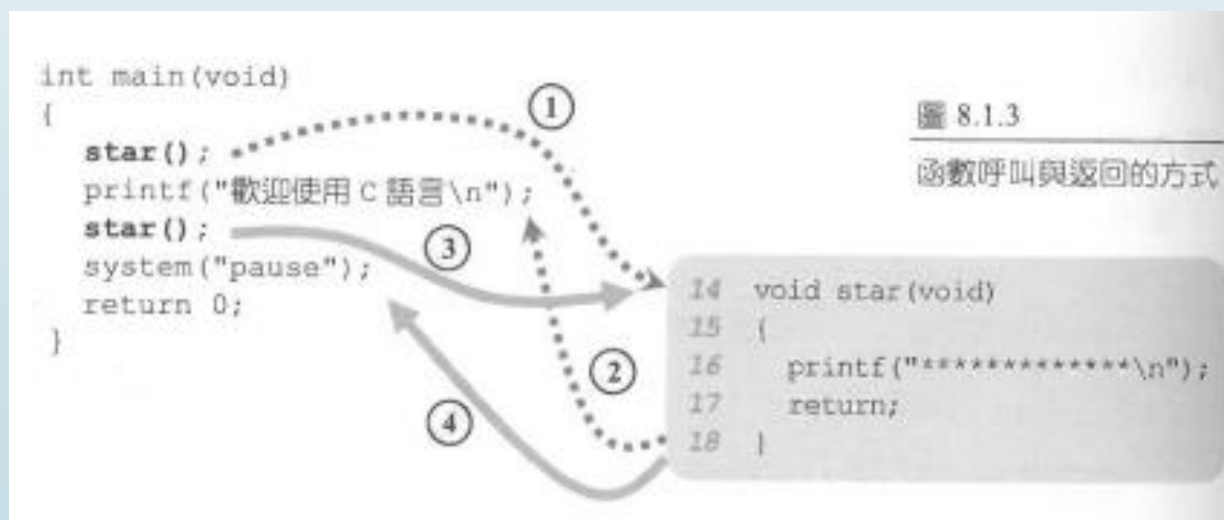
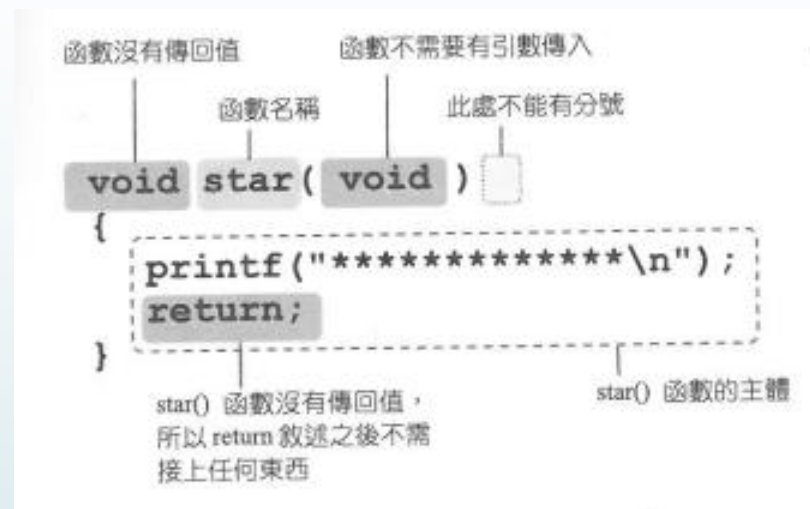
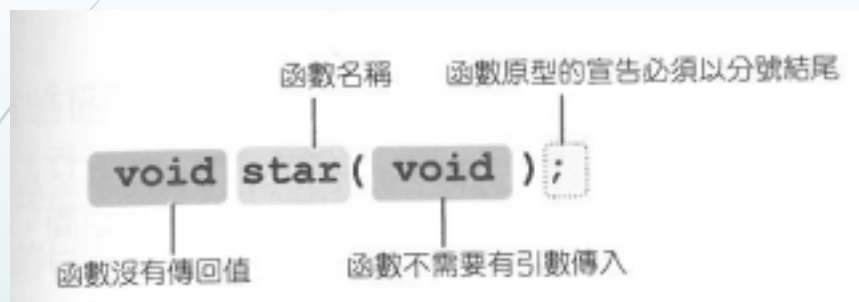


圖 8.1.3

函數呼叫與返回的方式

函數原型宣告

- 宣告要有分號結尾
- 可放在main函數內或外皆可，內表示為只能作用在main函數中，外則表示再本檔案中皆可使用

- 語法

傳回值型態 函數名稱 (引數型態 1, 引數型態 2, ...);

- void表示無須傳入參數

傳回值型態是整數 有兩個引數傳入 add() 函數，型態
均為 int，各型態間以逗號分開

```
int add ( int , int );
```

函數名稱為 add

函數定義

- 放在main函數外
 - 若放在main前面則不需要宣告函數原型
- return可省略
 - 回傳值可以是一個運算式或值，須注意只能有一個回傳值
- 注意參數數量及資料型態是否與呼叫函數時的參數數量及資料型態一致
- 語法

```
傳回值型態 函數名稱 (型態 1 引數 1, ..., 型態 n 引數 n)
{
    變數宣告;
    敘述主體;
    return 運算式;    /* 傳回運算式的值 */
}
```

呼叫函數

- 若不需要傳參數的話引數內不需要傳任何內容
- 語法

```
函數名稱 (引數);
```

```
變數 = 函數名稱 (引數);
```

範例-函數加總

```
#include <stdio.h>
#include <stdlib.h>
int add(int,int);          /* add() 函數的原型 */
int main(void)
{
    int sum, a=5, b=3;
    sum=add(a,b);          /* 呼叫add() 函數，並把傳回值設給sum */
    printf("%d+%d=%d\n",a,b,sum);

    system("pause");
    return 0;
}

int add(int num1, int num2) /* add() 函數的定義 */
{
    int a;                  /* 於add() 函數裡宣告變數a */
    a=num1+num2;
    return a;               /* 傳回num1+num2 的值 */
}
```


範例-字元列印函數

```
#include <stdio.h>
#include <stdlib.h>
void display(char,int); /* display() 函數的原型 */
int main(void)
{
    int n;
    char ch;
    printf("請輸入欲列印的字元:");
    scanf("%c",&ch);
    printf("請問要印出幾個字元:");
    scanf("%d",&n);
    display(ch,n); /* 呼叫自訂的函數，印出n個ch字元 */

    system("pause");
    return 0;
}

void display(char ch,int n) /* 自訂的函數display() */
{
    int i;
    for(i=1;i<=n;i++) /* for迴圈，可印出n個ch字元 */
        printf("%c",ch); /* 印出ch字元 */
    printf("\n");
    return;
}
```

範例-絕對值函數

```
#include <stdio.h>
#include <stdlib.h>
int abs(int);          /* 宣告函數abs()的原型 */
int main(void)
{
    int i;
    printf("Input an integer:");    /* 輸入整數 */
    scanf("%d",&i);
    printf("abs(%d)=%d\n",i,abs(i)); /* 印出絕對值 */

    system("pause");
    return 0;
}

int abs(int n) /* 自訂的函數abs(),傳回絕對值 */
{
    if (n<0)
        return -n;
    else
        return n;
}
```

範例-次方函數

```
#include <stdio.h>
#include <stdlib.h>
double power(double, int); /* 宣告函數power()的原型 */
int main(void)
{
    double x;          /* x為底數 */
    int n;              /* n是次方 */

    printf("請輸入底數與次方:");
    scanf("%lf,%d",&x,&n); /* 輸入底數與次方 */
    printf("%lf的%d次方=%lf\n",x,n,power(x,n));

    system("pause");
    return 0;
}

double power(double base, int n) /* power()函數的定義 */
{
    int i;
    double pow=1.0;
    for(i=1;i<=n;i++)          /* for() 迴圈，用來將底數連乘n次 */
        pow=pow*base;
    return pow;
}
```

範例-質數測試函數

```
#include <stdio.h>
#include <stdlib.h>
int is_prime(int);          /* 宣告函數is_prime()的原型 */
int main(void)
{
    int i;
    for(i=2;i<=30;i++)      /* 找出小於30的所有質數 */
        if(is_prime(i))    /* 呼叫is_prime()函數 */
            printf("%3d",i); /* 如果是質數,便把此數印出來 */
    printf("\n");
    system("pause");
    return 0;
}

int is_prime(int num)      /* is_prime()函數,可測試num是否為質數 */
{
    int i;
    for(i=2;i<=num-1;i++)
        if(num%i==0)
            return 0;
    return 1;
}
```

範例-同時使用多個函數

```
#include <stdio.h>
#include <stdlib.h>
void sum(int), fac(int);          /* 定義函數的原型 */
int main(void)
{
    fac(5);                      /* 呼叫fac()函數 */
    sum(5);                      /* 呼叫sum()函數 */

    system("pause");
    return 0;
}

void fac(int a)                  /* 自訂函數fac(), 計算a! */
{
    int i, total=1;
    for(i=1; i<=a; i++)
        total*=i;
    printf("1*2*...%d=%d\n", a, total); /* 印出a!的結果 */
}

void sum(int a)                  /* 自訂函數sum(), 計算1+2+...+a的結果 */
{
    int i, total=0;
    for(i=1; i<=a; i++)
        total+=i;
    printf("1+2+...%d=%d\n", a, total); /* 印出加總的結果 */
}
```

範例-函數相互呼叫

```
#include <stdio.h>
#include <stdlib.h>
double Leibniz(int);          /* 宣告函數Leibniz()的原型 */
double power(double, int);    /* 宣告函數power()的原型 */
int main(void)
{
    int i;
    for(i=1;i<=10000;i++)      /* 找出前10000個π 的估算值 */
        printf("Leibniz(%d)=%12.10f\n",i,Leibniz(i));
    system("pause");
    return 0;
}

double Leibniz(int n) /* Leibniz()函數，可估算π 值到第n項 */
{
    int k;
    double sum=0.;
    for(k=1;k<=n;k++)
        sum=sum+power(-1.0,k-1)/(2*k-1); /* 萊布尼茲公式 */
    return 4*sum;
}

double power(double base, int n) /* power()函數，可計算base的n次方 */
{
    int i;
    double pow=1.0;
    for(i=1;i<=n;i++)
        pow=pow*base;
    return pow;
}
```

萊布尼茲公式

► 萊布尼茲公式

$$\begin{aligned}\pi &= 4 \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{2k-1} = 4 \left(\frac{(-1)^{1-1}}{2(1)-1} + \frac{(-1)^{2-1}}{2(2)-1} + \frac{(-1)^{3-1}}{2(3)-1} + \frac{(-1)^{4-1}}{2(4)-1} + \frac{(-1)^{5-1}}{2(5)-1} + \dots \right) \\ &= 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots \right)\end{aligned}$$

$$\text{Leibniz}(n) = 4 \sum_{k=1}^n \frac{(-1)^{k-1}}{2k-1}$$

```
20 sum=sum+power(-1.0,k-1)/(2*k-1); /* 萊布尼茲公式 */
```

習題

15

- 試寫 `int mod(int x,int y)` 函數，計算 x/y 的餘數，並利用此函數來計算 `mod(17,5)` 即計算 $17/5$ 的餘數。

習題

16

- 試寫一函數 `int prime(int n)`，可以用來找出第 `n` 個質數(第一個質數為2，第二個質數為3，以此類推)，以此函數找出第100個質數。