

C 程式語言 3

遞增與遞減運算子

- 遞增運算子(`++`)與遞減運算子(`--`)
 - 來自變數內容遞增 1 與遞減 1 的精簡版寫法
- 簡單的說
 - `a = a + 1` 可寫成 `a++`
 - 讓變數內容加 1
 - `a = a - 1` 可寫成 `a--`
 - 讓變數內容減 1

遞增與遞減運算子

- 遞增與遞減運算子的成員：

遞增與遞減運算子	意義	範例	說明
++	遞增，變數值加 1	a++	a 加 1 後再設定給 a 存放
--	遞減，變數值減 1	a--	a 減 1 後再設定給 a 存放

- a++ 會先執行整個敘述後，再將 a 的值加 1
- ++a 則是先把 a 的值加 1 後，再執行整個敘述

遞增與遞減運算子

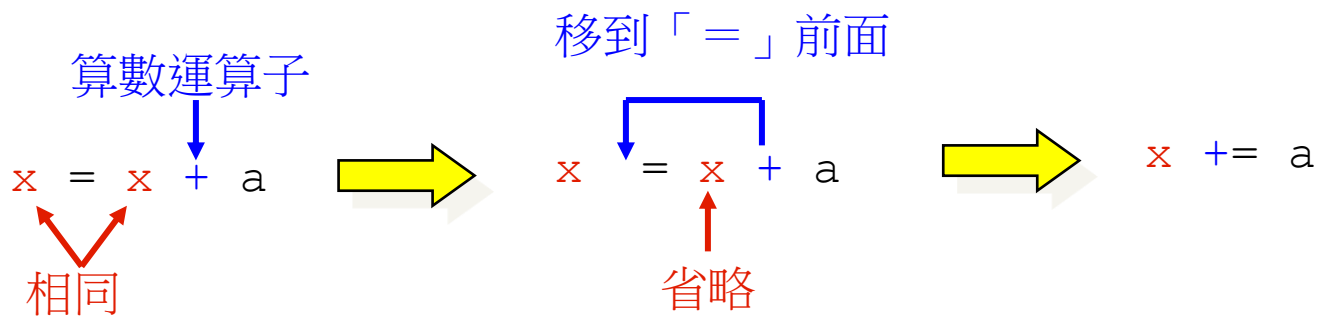
- 下面的程式是使用遞增運算子的範例：

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int a=3, b=3;
    printf("a=%d",a);
    printf(", a++ return value%d",a++);    /* 計算a++，並印出其傳回值 */
    printf("\n", a);
    printf("b=%d",b);
    printf(", ++b return value%d",++b);    /* 計算++b，並印出其傳回值 */
    printf("\n", b);

    system("pause");
    return 0;
}
```

運算式與簡潔運算子

- 算數運算子與指定運算子「=」的簡便寫法
－ 例如： $x = x + a$ 可以寫成 $x += a$



運算式與簡潔運算子

運算式 為運算子與運算元組成

- 簡潔運算子可簡化運算式

運算子	範例用法	說明	意義
<code>+=</code>	<code>a+=b</code>	a+b 的值存放到 a 中	<code>a=a+b</code>
<code>-=</code>	<code>a-=b</code>	a-b 的值存放到 a 中	<code>a=a-b</code>
<code>*=</code>	<code>a*=b</code>	a*b 的值存放到 a 中	<code>a=a*b</code>
<code>/=</code>	<code>a/=b</code>	a/b 的值存放到 a 中	<code>a=a/b</code>
<code>%=</code>	<code>a%=b</code>	a%b 的值存放到 a 中	<code>a=a%b</code>

條件運算子

- ?: 是一個非常有趣的運算子
- 語法如下：

條件判斷 ? 運算式1 : 運算式2

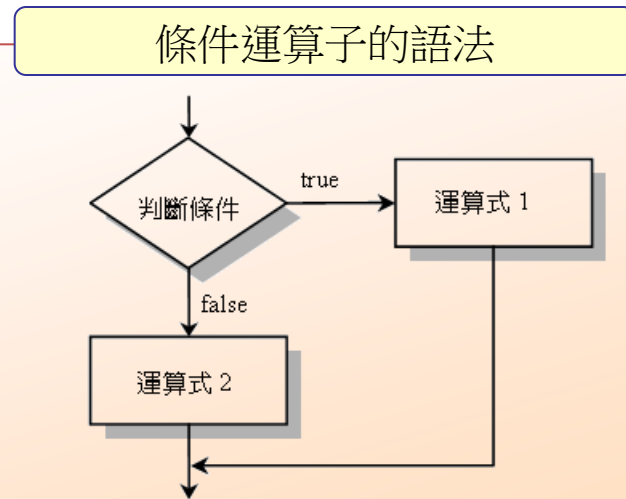
- 規則
 - 當「條件判斷」的計算結果為 **true** 時
 - 傳回「運算式1」的計算結果
 - 當「條件判斷」的計算結果為 **false** 時
 - 傳回「運算式2」的計算結果。

條件運算子

- 條件運算子可代替簡單的 if-else 敘述

條件運算子	意義
?:	根據條件的成立與否，來決定結果為?或:後的運算式

條件判斷 ? 運算式1 : 運算式2



條件運算子的範例

- 利用條件運算子判斷兩個數中較大的數：

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int num1,num2,larger;
    printf("Please input two integers: :");
    scanf("%d %d",&num1,&num2);

    num1>num2 ? (larger=num1) : (larger=num2); /* 條件運算子 */
    printf("%d greater value \n",larger);

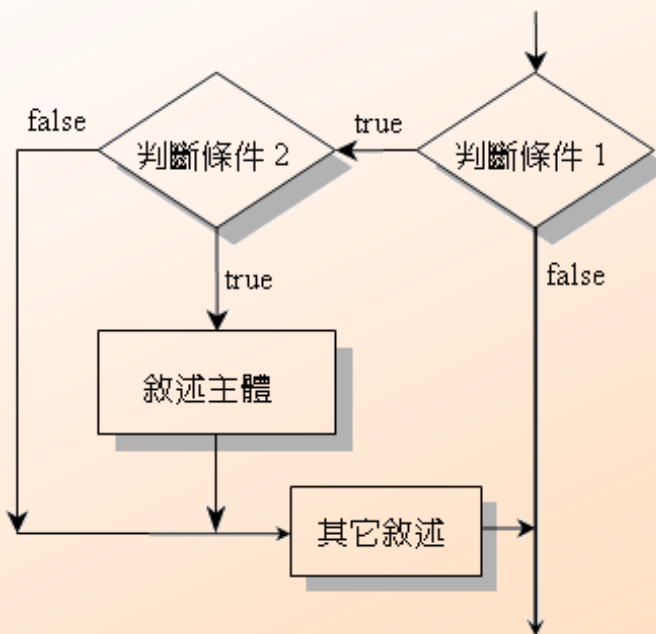
    system("pause");
    return 0;
}
```

巢狀 if 敘述

- if 裡面還有其它的if 敘述，則稱為巢狀 if 敘述

巢狀 if 敘述的語法

```
if (判斷條件1)
{
    if (判斷條件2)
    {
        敘述主體;
    }
    ...
    其它敘述;
}
```



巢狀 if 敘述的練習

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int score;
    printf(" Please input a score: ");
    scanf("%d",&score);

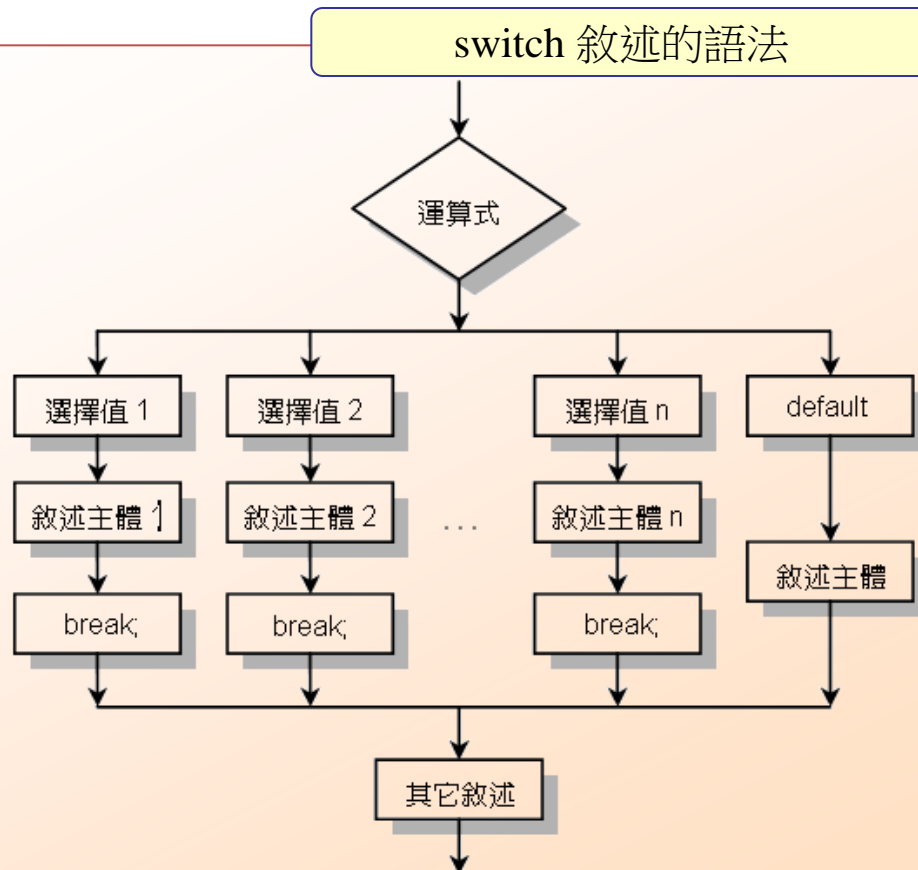
    if (score<60)                                /* 如果score<60 */
    {
        if(score>=50)                            /* 如果score>=50 */
            printf(" make up exams \n");
        else
            printf("Fail\n");
    }
    else
        printf("Pass\n");

    system("pause");
    return 0;
}
```

switch 敘述

- switch 敘述可用來進行多重選擇

```
switch (運算式)
{
    case 選擇值1:
        敘述主體1;
        break;
    case 選擇值2:
        敘述主體2;
        break;
    ...
    default:
        敘述主體;
}
```



switch 敘述的範例

- 依據選擇值進行四則運算：

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    float a,b;
    char oper;
    printf("Please input the expression:(ex:3+2):");    /* 輸入運算式 */
    scanf("%d %c %d",&a,&oper,&b);

    switch(oper)
    {
        case '+':
            printf("%d+%d=%d\n",a,b,a+b);    /* 印出a+b */
            break;
```

switch 敘述的範例

```
case '-':  
    printf("%d-%d=%d\n",a,b,a-b);          /* 印出a-b */  
    break;  
case '*':  
    printf("%d*%d=%d\n",a,b,a*b);          /* 印出a*b */  
    break;  
case '/':  
    printf("%d/%d=%.3f\n",a,b,a/b); /* 印出a%b */  
    break;  
default:  
    printf("input error!!\n");              /* 印出字串 */  
}  
system("pause");  
return 0;  
}
```

使用while 迴圈

- **while** 最適合用在迴圈執行次數為未知時

while 迴圈的語法

設定迴圈初值;

while (判斷條件)

{

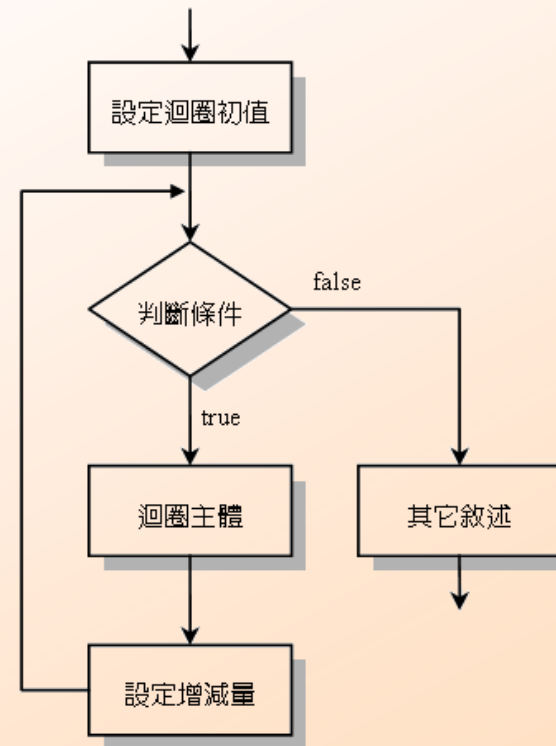
 迴圈主體;

 設定增減量;

}



這兒不可以加分號



while迴圈的範例

- 將while用在迴圈執行次數為未知：

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int i=1,sum=0; /* 設定迴圈初值 */
    while(i<=10)
    {
        sum+=i;
        printf(" From 1 to accumulate %2d=%2d\n",i,sum);
        i++;
    }
    system("pause");
    return 0;
}
```


以巢狀while迴圈改寫九九乘法表

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int i=1, j=1;                /* 設定迴圈控制變數的初值 */
    while (i<=9)                 /* 外層迴圈 */
    {
        while (j<=9)             /* 內層迴圈 */
        {
            printf("%d*%d=%2d ",i,j,i*j);
            j++;
        }
        printf("\n");
        i++;
        j=1;
    }
    system("pause");
    return 0;
}
```

以巢狀迴圈印出幾何圖形

- 利用巢狀迴圈印出三角形

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int i,j,n=6;                /* 設定迴圈初值 */

    for (i=1;i<=n;i++)          /* 外層迴圈決定哪一系列要印星號 */
    {
        for (j=1;j<=i;j++)      /* 內層迴圈印出*星號 */
            printf("*");
        printf("\n");
    }

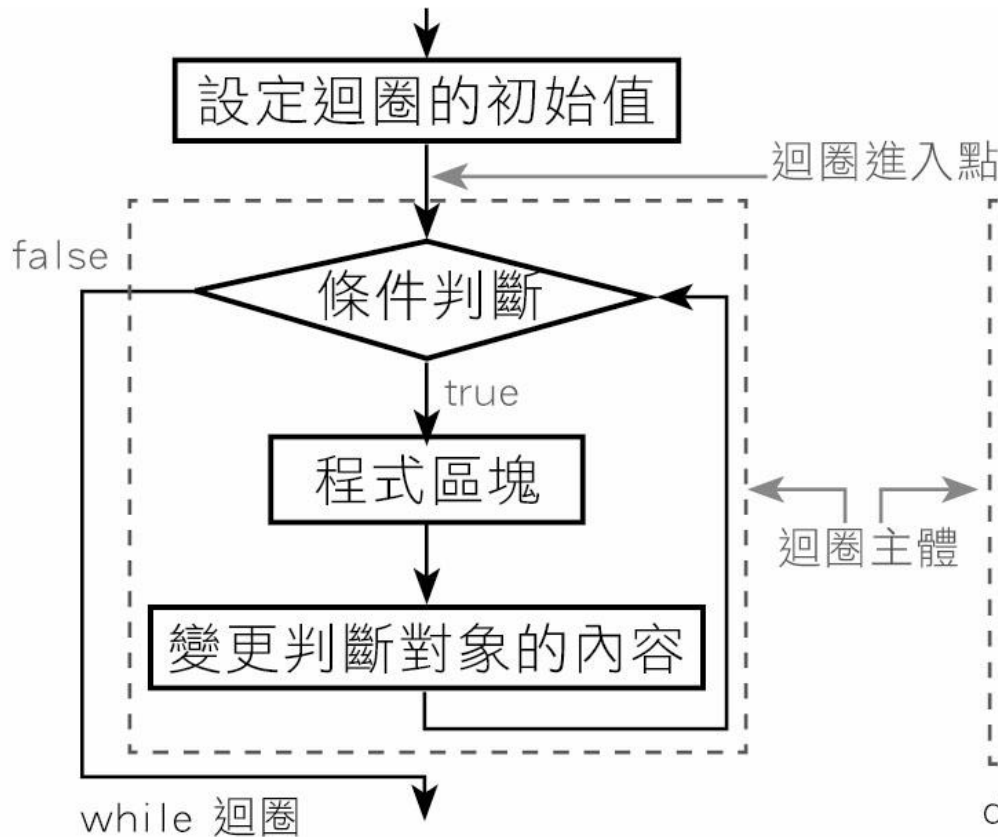
    system("pause");
    return 0;
}
```

while與do while

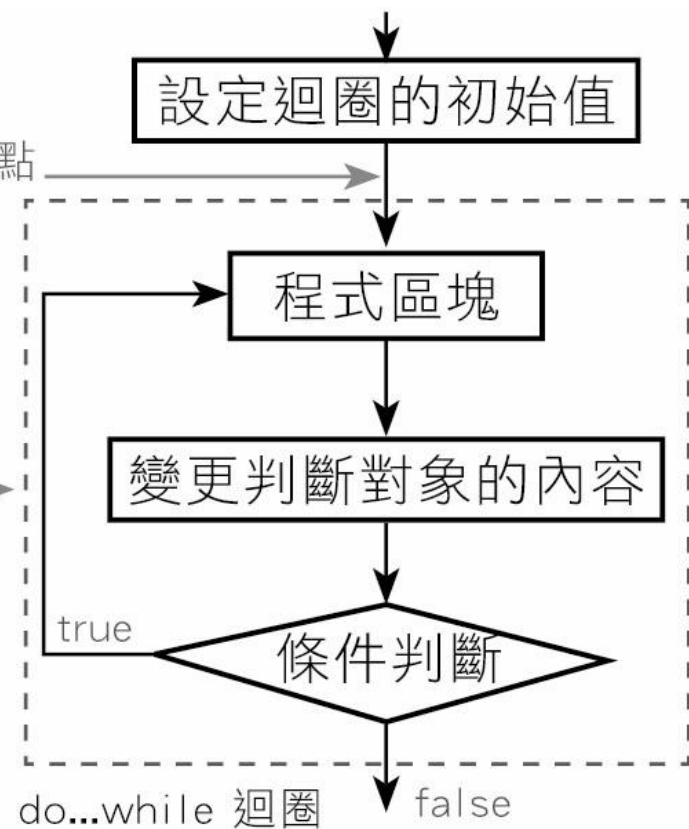
後測式do/while和前測式while迴圈的主要差異是在迴圈結尾檢查條件，因為先執行程式區塊的程式碼後才測試條件，所以do/while迴圈的程式區塊至少會執行「1」次

迴圈的概念與基本架構

- 先判斷再執行



- 先執行再判斷



使用do while迴圈

do-while 迴圈的語法

設定迴圈初值;

do

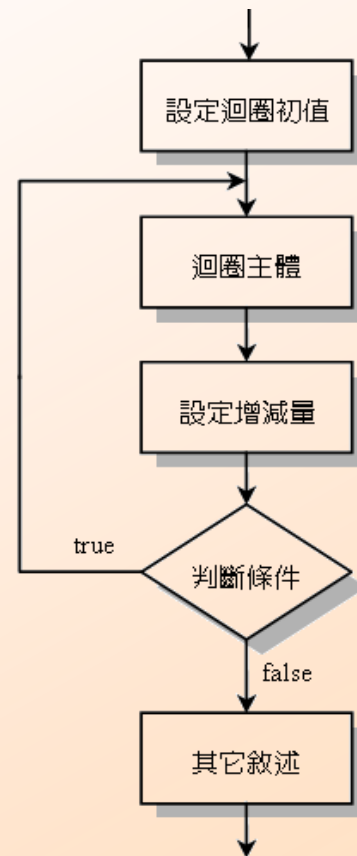
{

迴圈主體;

設定增減量;

} **while** (判斷條件);

這兒要加分號



以巢狀迴圈反印數字

- 將整數反印，例如 5123→3215：

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int a,r;

    while(1)
    {
        do
        {
            printf("Input an integer:");
            scanf("%d",&a);
        }
        while (a<=0);
```

/* 必須輸入大於0的正整數 */

以巢狀迴圈反印數字

```
printf("The reverse is ");
while (a!=0)          /* 將正整數倒過來輸出 */
{
    r=a%10;           /* 計算a/10的餘數 */
    a/=10;            /* 計算a/10，再把結果設回給a */
    printf("%d",r);
}
printf("\n\n");
}
system("pause");
return 0;
}
```

-The End-