



條件與判斷式

Wisely

指令：if (判斷條件式內容...)

- 使用格式：

```
if (條件判斷)
```

```
{
```

```
.....
```

```
}
```

- 大括號的內容，會在條件判斷式為「真」(ture)時執行

指令：if (判斷條件式內容...)

- 範例1：

```
if (a > 200)
{
    printf( "It is a good day" );
}
```

- 範例2：

```
if (a < 20)
    printf( "good morning?\n");
```

指令：if (判斷條件式內容...)

- 範例3：(巢狀寫法)

```
if (a > 100)
{
    if (b > 70)
    {
        if (c > 90)
        {
            if (d > 130)
            {
                printf("This is a perfect maid!\n");
            }
        }
    }
}
```

指令：ifelse

- 若希望條件成立時做一件事，不成立時做另一件事，可加入“ else”
- 使用格式：

```
if(判斷式)
```

```
{
```

```
.....
```

```
}
```

```
else
```

```
{
```

```
.....
```

```
}
```

指令：ifelse

- 範例：

```
if (a < 1)
{
    printf( "可樂\n");
}
else
{
    printf( "果汁\n");
}
```

注意：else 不能單獨存在 (一定要與if搭檔)

指令：ifelse if else

- 範例：

```
if (a < 5)
{
    printf( "藍色!!\n");
}
else if (a < 10)
{
    printf( "紅色!!\n");
}
else
{
    printf( "黑色!!\n");
}
```

指令 for

- for 指令是要讓迴圈{}重覆執行N 次，直到設定條件不成立時，才跳出迴圈。
 - 格式：for(運算式1：運算式2：運算式3)
 - {敘述A;}
 - 敘述B;
- 運算式1：初值設定。
- 運算式2：條件式判斷，當條件成立時則一直執行迴圈{} 內的敘述A，當條件不成立時則離開for 迴圈，執行敘述B。
- 運算式3：運算式，當條件成立時則繼續執行此運算式，常見的運算式有i++遞增及i--遞減。

指令 for

- 範例1：

```
for (i=0;i<8;i++)  
{  
    j=i;  
}
```

//先設定 i=0，然後判斷有沒有小於8

//若有，j=0，i=i+1=1

//再判斷 i有沒有小於8

//因為 i=1仍小於8，所以 j=1，i=i+1=2

//(以上一直重複至i累增至8時

//因為i已不小於8，所以大括號內的內容不執

//行，直接跳出並離開for迴圈

指令 for

- 範例2 (無窮迴圈)：

```
for (;;) { //此處放你想執行的內容 }
```

- 範例3：(若迴圈未達到跳出的條件，因其他因素必須跳出for 迴圈時，則可在迴圈內加入break 指令)

```
for(i=0;i<8;i++)  
{  
    if(k==0){break;}  
    c=x+i;  
}
```

指令while...當什麼成立時

- 格式：

```
while(運算式)
{
    敘述A;
}
    敘述B;
```

- 範例1：

```
while(i<10)
{
    p1=i;
    i++;
}
```

指令while...當什麼成立時

- 範例2 (可寫成無窮迴圈) :

```
while(1)
{
    敘述;
}
```

- 範例3 (同樣可利用break跳出) :

```
while (1)
{
    if(p0==1)break;
    敘述;
}
```

指令do-while...先執行再判斷

- do
 {
 敘述A;
 }while(運算式);
 敘述B;

先執行敘述A 後，才測試while 的運算式是否為真？

是則繼續執行敘述A，若為**假**則跳出迴圈執行敘述B。

指令do-while...先執行再判斷

- 範例1：

```
main()
```

```
{
```

```
    int i; sum=0;
```

```
    do
```

```
    {
```

```
        P1=i;i++;
```

```
        //i=i+1
```

```
    } while(i<10);
```

```
    //若i 小於10，則繼續執行大括號內的程式，  
    //否則跳出。
```

```
}
```

指令do-while...先執行再判斷

- 範例2：

```
do  
{  
    敘述.....  
}while(1);
```

while 的運算式如果為1則形成無窮迴圈

課堂練習：

- 題目1：

請問下列程式之執行結果A=？、B=？

```
A=0; B=0;
```

```
for(i=1 ; i<10 ; i+2)
```

```
{
```

```
    A=A+B;
```

```
    B++;
```

```
}
```

- 題目2：

請問下列程式之執行結果a=?、b=?

```
int a=1,b=0;
```

```
while(a<7)
```

```
{
```

```
    a++;
```

```
    b+=a;
```

```
}
```


指令switch

- 除if-else-if外，尚有switch-case指令，格式如下：

```
switch(變數){  
    case 常數值1 :                //變數值等於常數值1 時執行敘述1  
        {敘述1;}  
        break;  
    case 常數值2 :                //變數值等於常數值2 時執行敘述2  
        {敘述2;}  
        break;  
    case 常數值3 :                //變數值等於常數值3 時執行敘述3  
        {敘述3;}  
        break;  
    default :                      //變數值不等於以上常數值時執行敘述4  
        {敘述4;}  
        break;
```

指令switch

- 範例：

```
switch(x){  
  case 1 :           // x=1 時執行P0=1  
    {P0=1;}  
    break;  
  case 2 :           // x=2 時執行P0=2  
    {P0=2;}  
    break;  
  case 3 :           // x=3 時執行P0=3  
    {P0=3;}break;  
  default :          // x 不等於以上值時執行P0=FFH  
    {P0=0xff;}break;  
}
```

陣列

- 陣列是一種結構化的資料結構，主要是將相同型態的變數集合起來，以一個陣列名稱來代表，經由陣列的索引值來存取內部的某一筆資料。陣列同一般變數一樣，在使用前也要宣告，以便系統預留記憶體空間給宣告的陣列使用，宣告格式如下：

資料型態 陣列名稱[陣列長度]

EX..

```
int        abc[3];
```

```
char       new[5];
```

陣列

- 宣告範例1：

```
char seg[5]={0xc0,0xf9,0xa4,0xb0,0x99}; //宣告一組1x5的陣列
```

char seg[5]分配內容值為：

陣列位置	內容
seg[0]	c0
seg[1]	f9
seg[2]	a4
seg[3]	b0
seg[4]	99

陣列

- 宣告範例2：

```
char seg[]={a,b,c,d,e};
```

//如未指定陣列長度大小，

//則會由編譯器決定

陣列

- 宣告範例3：(也可宣告成二維陣列)

Ex..

```
int z[3][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12}};
```

- int z[3][4]分配內容值如右表：

陣列位置	內容
z[0][0]	1
z[0][1]	2
z[0][2]	3
z[0][3]	4
z[1][0]	5
z[1][1]	6
z[1][2]	7
z[1][3]	8
z[2][0]	9
z[2][1]	10
z[2][2]	11
z[2][3]	12

課堂練習 1：

- 請問下列程式之執行結果A[2][2]= ? 、 B= ?

```
B=0;
```

```
for (i=0 ; i<3 ; i++)
```

```
    for (j=0 ; j<3 ; j++)
```

```
    {        A[i] [j]=(i*i)+(j*j);
```

```
        B=B+A[i] [j];
```

```
    }
```

課堂練習 2：

- 請利用for 或while 指令完成「計算1 加到30」的迴圈控制程式
- 完成一個程式，以列印排列的數字。

例如輸入 6 時，輸出

654321
65432
6543
654
65
6

輸入 4 時，輸出

4321
432
43
4

回家作業：

- 停車費

某一停車場的停車費率規則如下：

前 30 分(含)鐘免費。停車 31 ~ 60 分鐘 30 元。停車 61 ~ 120 分鐘 60 元。接下來每一小時增加 30 元，

例如 121~ 180 分鐘 90 元、 181~ 240 分鐘 120 元，以此類推。

每日停車費最高為 210 元，例如停了 11 小時又 30 分鐘，原本應該是收費 $30 \times 12 = 360$ ，但只要收 210 元就好。

- 現在想請你寫一個程式，輸入四個整數 H1、M1、H2、M2 ($0 \leq H1$ 、 $H2 \leq 23$ ， $0 \leq M1$ 、 $M2 \leq 59$)，H1:M1 代表車子進場的時間，H2:M2 代表車子出場的時間，請你計算出他需要繳多少的停車費(不用考慮跨日的情形)。例如：

輸入1：7 10 7 30

輸出1：0

輸入2：8 10 9 20

輸出2：60

END

