

### 學習要點:

1. If...then...else...end if
2. Select case...select end
3. for...next
4. do while...while wend

## 3.1 認識流程控制結構

我們上一章所用的範例都是一些相當簡單的程序。所謂的“簡單”就是指程序執行只有一個由上而下的方向，而不會轉彎或跳行，但實際上平常所應用的程序並不會這麼簡單，一般的程序都會依據不同的情況而換行，因為這樣的程序才能應付各種不同的情況，這時就需要流程控制結構來幫組設計師控制程序執行的方向。流程控制結構通常要借助邏輯數據的幫助才能運行，流程控制結構共有 6 種，我們一一來介紹。

## 3.2 if 的第一種格式: 如果...就.. (單一選擇)

If 條件 then 程序代碼

這是 if 指令最簡單的格式“單一選擇”，其中的“條件”是一個表達式，他所計算的結構必須是一個邏輯數據，如果:“條件”的計算結果是“True”，那麼就會執行 Then 後面的“程序代碼”: 如果“條件”的計算結果是“False”，那麼就會跳過整個 if 語句，而不會執行 Then 後面的“程序代碼”。 Then 後面的“程序代碼”如果跟 Then 不是同一行或是程序代碼有很多行，那就要在最後加上 end if 以結束 if 條件語句。

```
<script language=vbs">  
    Score=70  
    If score>=60then msgbox "及格!"  
</script>
```

## 3.3 if 的第二種格式: 如果...就...否則...(雙向選擇)

```
If 條件 then  
    程序代碼一  
else  
    程序代碼二  
end if
```

```

<script language="vbs">
    score=InputBox("請輸入計算機成績:","輸入成績")
    if score>=60 then
        msg="及格!"
    else
        msg="不及格!"
    end if
    msgbox"您的計算機成績"&msg
</script>

```

### 3.4 輸入函數 InputBox 用法

變量=InputBox("顯示信息", "窗口標題")

InputBox 函數可以提供對話框讓用戶輸入信息，並將用戶輸入的信息存入等號左邊的變量中;

### 3.5 if 的第三種格式：多項選擇

```

If 條件 then
    程序代碼一
Elseif
    程序代碼二
Elseif
    程序代碼三
.....
.....
Else
    程序代碼 N
End if

```

```

<script language="vbs">
    score=InputBox("請輸入計算機成績:","輸入成績")
    if score>=90 then
        msg="優等"
    elseif score<90 and score>=80 then
        msg="甲等"
    elseif score<80 and score>=70 then
        msg="乙等"

```

```
elseif score<70 and score>=60 then
    msg="丙等"
else
    msg="丁等"
end if
msgbox"您的計算機成績為"&msg
</script>
```

### 3.6 select...case

這種流程控制結構可以根據第一個變量的值而有不同的執行方向。

```
select case 變量
case 值一
    程序代碼一
case 值二
    程序代碼二
case 值三
    程序代碼三
...
case else
    程序代碼 N
end select
```

```
<script language="vbs">
no=InputBox("請輸入一個 1~5 之間的數字: ","漢譯英")
select case no
    case 1
        msg="ONE"
    case 2
        msg="TWO"
    case 3
        msg="THREE"
    case 4
        msg="FOUR"
    case 5
        msg="FIVE"
    case else
        msg="本字典查尋不到你要查詢的單詞"
```

```

        end select
        msgbox no&"的英文為 "&msg
    </script>

```

### 3.7 for-next (計數循環)

計算機最擅長的功能就是重複執行某項功能，如果每執行一次就要寫一次程序代碼的話，大部分的程序必然非常的龐大，"循環"指令就是用來解決重複執行的問題。

for 變量=起始值 to 終止值 [step 步長]

```

<script language="vbs">
    total=0
    for i=1 to 10
        total=total+i
    next
    msgbox "1 加到 10 的總合為"&total
</script>

```

```

<script language="vbs">
    total=0
    for i=2 to 100 step 2
        total=total+1
    next
    msgbox "2+4+6+...+100="&total
</script>

```

```

<script language="vbs">
    '用循環顯示九九表
    Document.write"<table border='4'>"
    for i=1 to 9
        total="<tr>"
        for j=1 to 9
            total=total&"<td>"&i&"*"&i&"="&i*j&"</td>"
        Next
        total=total&"</tr>"
    document.write total

```

Next

Document.write"</table>"

</script>

注: **document.write** 是瀏覽器中的一個方法，在瀏覽器中有許多對象的屬性和方法可供設計人員使用，瀏覽器將自己本身的各部分，包括窗口、屬性、標記等分成各種對象，而每個對象又提供不同的屬性和方法。

```
<script language="vbs">
    dim a(99)
    for i=0 to 99
        a(i)=i+1
        document.write a(i)&"<br />"
    next
</script>
```

```
<script language="vbs">
    dim score(3)
    score(0)=90
    score(1)=86
    score(2)=99
    score(3)=54
    '用循環打印每一位學生的成績
    For i=0 to 3
        msg=msg&"第"&i+1&"個學生分數為"&score(i)&vbcrif
    next
    msgbox msg
</script>
```