

VB.NET的流程控制

資科系
林偉川

流程控制的基礎

- 程式語言撰寫的程式碼大部分是一列指令接著一系列指令循序的執行，但是對於複雜的工作，為了達成預期的執行結果，需要使用「流程控制結構」
- 流程控制可以配合條件判斷以便執行不同區塊的程式碼，或是重複執行區塊的程式碼，流程控制指令可以分為兩類，如下所示：
 - **條件控制**：條件控制可能為**單一選擇(if)**或**多選一(select case)**，依照條件運算子的結果，決定執行那一個區塊的程式碼。
 - **迴圈控制**：**迴圈控制**是重複執行區塊的程式碼，其中擁有**結束條件**，以便結束迴圈。

2

If ... Then是否選條件敘述

- If...Then條件敘述是一種是否執行的條件，如果If條件為True，就執行Then...End If間的程式碼，如下所示：

```
If score >= 60 Then
    Response.Write(name & "成績及格!")
    Response.Write("學生成績: " & score & "<br>")
End If
```

3

If ... Then ... Else二選一條件敘述

- 如果是2個執行區塊，只能二選一，我們可以加上Else指令，如果If條件為True，就執行Then...Else間的程式碼，False就執行Else...End If間的程式碼，如下所示：

```
If sex = "男" Then
    Response.Write(name & "男士您好!")
Else
    Response.Write(name & "女士您好!<br>")
End If
```

4

If ... Then ... ElseIf多選一條件敘述

- If ... Then ... ElseIf條件敘述屬於If ... Then條件敘述的延伸，使用ElseIf指令建立多選一條件指令，如下所示：

```
dtWeekday = Weekday(Today)  
If dtWeekday = 1 Then  
    wdstring="星期日"  
Elseif dtWeekday = 2 Then  
    wdstring="星期一"  
Elseif dtWeekday = 3 Then  
    .....  
Else  
    wdstring="無法分辨是星期幾"  
End If
```

5

Select Case多選一條件敘述

- Select Case直接依照符合的條件執行不同區塊的程式碼，如下所示：

```
dtWeekday = Weekday(Today)  
Select Case dtWeekday  
    Case 1: wdstring="星期日"  
    Case 2: wdstring="星期一"  
    Case 3: wdstring="星期二"  
    Case 4: wdstring="星期三"  
    Case 5: wdstring="星期四"  
    Case 6: wdstring="星期五"  
    Case 7: wdstring="星期六"  
    Case Else  
        Wdstring="無法分辨是星期幾"  
End Select
```

6

For ... Step ... Next迴圈

- For Step Next迴圈指令可以執行固定次數的迴圈，其結束條件是Step量每次增加或減少一個值，當到達結束條件值就結束迴圈，例如：

```
Dim i, total As Integer
For i = 1 To 10 Step 1
    total = total + i
Next i
```

7

巢狀的For ... Next迴圈

- 巢狀迴圈指的是在For Next迴圈中擁有其它For Next迴圈，如下所示：

```
For i = 1 To 9
    For j = 1 To 9
        .....
    Next j
Next i
```

8

For Each ... In ... Next迴圈

- For Each迴圈是的使用在集合物件或陣列，可以顯示集合物件或陣列的所有元素，特別適合那些不知道到底有多少元素的集合物件或陣列，如下所示：

```
For Each userName in names
    Response.Write("<td>" & userName & "</td>")
Next
```

9

Exit For中斷For迴圈

- 我們可以使用Exit For指令強迫跳出For Next迴圈，如下所示：

```
For i = 1 To 100 Step 1
    ...
    Exit For
    ...
Next
```

10

Do ...Exit Do ... Loop迴圈

- 搭配條件測試可以在3種位置測試迴圈的結束條件，如下：
 - 迴圈開始：在迴圈開始使用While或Until指令測試迴圈條件。
 - 迴圈結尾：在迴圈結尾使用While或Until指令測試條件，結尾測試迴圈至少執行一次。
 - 中斷迴圈：在迴圈中使用Exit Do指令中斷迴圈的執行。

11

Do ...Exit Do ... Loop迴圈

- 我們可以使用Exit Do指令結束迴圈，如下所示：

```
Do
....
Exit Do
i = i + 1
....
Loop
```
- Do Loop迴圈插入Exit Do指令，當迴圈執行到此指令就會中斷迴圈執行，Do Loop迴圈需要自行使用計數器變數來增減其值。

12

Do While ... Loop迴圈

- Do Loop迴圈使用While條件在迴圈開頭檢查，判斷是否允許進入迴圈，While指令的測試條件如果成立時才能進入迴圈，如下：

```
i = 1
total = 0
Do While i <= 10
    total = total + i
    i = i + 1
Loop
```

13

Do Until ... Loop迴圈

- Do Loop迴圈使用Until條件在迴圈開始檢查，Until直到條件成立，如果條件不成立就進入迴圈，如下所示：

```
i = 1
total = 0
Do Until i > 10
    total = total + i
    i = i + 1
Loop
```

14

Do ... Loop While迴圈

- 這個Do Loop迴圈至少會執行一次，如下所示：

```
i = 1
total = 0
Do
  total = total + i
  i = i + 1
Loop While i <= 10
```

15

Do ... Loop Until迴圈

- Do Loop迴圈使用Until條件在迴圈結尾檢查，這個迴圈至少會執行一次，如下所示：

```
i = 1
total = 0
Do
  total = total + i
  i = i + 1
Loop Until i > 10
```

16

Do/While ... Loop/Until的巢狀迴圈

- Do Loop迴圈如同For Next迴圈一樣，可以使用各種組合建立巢狀迴圈，如下所示：

```
Do While i <= 9
  j = 1
  Do
    Response.Write(i & "*" & j & "=" & i*j)
    j = j + 1
  Loop Until j > 9
  i = i + 1
Loop
```

17

While ... End While迴圈

- While End While迴圈是在迴圈開始測試條件，在功能上和Do While Loop迴圈相同，其架構如下所示：

```
While i <= 9
  j = 1
  While i <= 9
    Response.Write(i & "*" & j & "=" & i*j)
    j = j + 1
  End While
  i = i + 1
End While
```

18

VB.NET的錯誤處理

- 在VB.NET提供更結構化的錯誤處理敘述，即 Try End Try，如下所示：

```
Try
    ' 測試的錯誤程式碼
    .....
Catch e As Exception
    ' 錯誤處理的程式碼
    .....
Finally
    .....
End Try
```

19

VB.NET的錯誤處理

- 錯誤處理敘述可以分爲3個部分，如下：
 - **Try**程式區塊：在**Try**和**Catch**指令間的程式區塊是**VB.NET**需要錯誤處理的程式碼。
 - **Catch**程式區塊：如果Try程式區塊的程式碼發生錯誤，在**Catch**到**Finally**指令間的程式區塊將會傳入參數**e**的**Exception**例外物件，可以使用**e.ToString()**方法顯示錯誤資訊，或建立錯誤處理的補救程式碼。
 - **Finally**程式區塊：這是選擇性的程式區塊，**不論**錯誤是否產生，**都會**執行此區塊的程式碼，通常是用來善後的程式碼。

20