

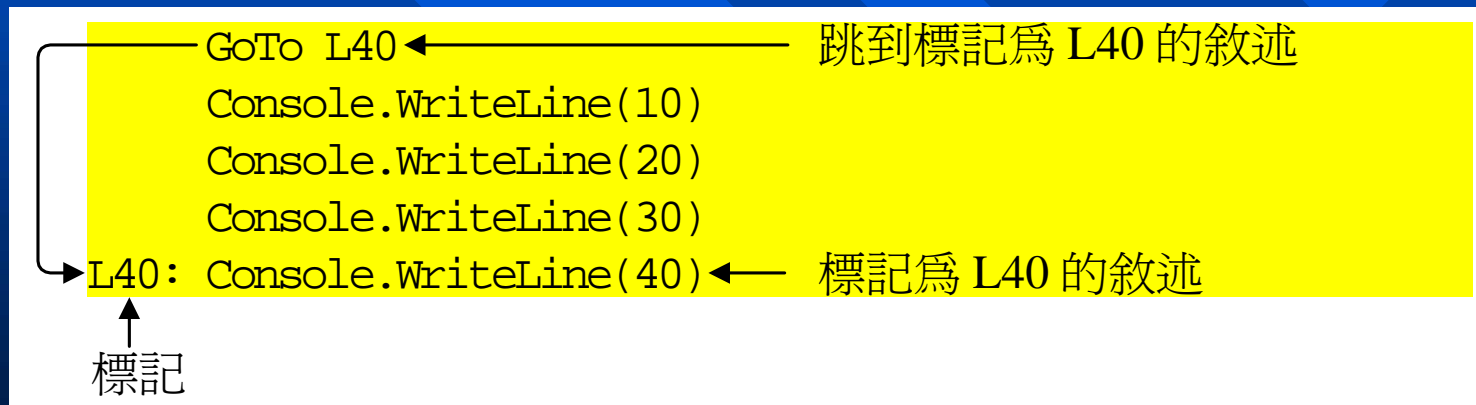
# 新觀念的 Visual Basic.NET 教本

## 第 6 章 我的程式會轉彎

# 6-1 強制跳躍的 GoTo

# GoTo 敘述

- “跳到...” 的意思，執行之後可以跳到前面的敘述，也可以跳到後面的敘述。
- 使用GoTo以前必須先有「標記」存在，例如：



# 標記的設定

標記名稱 = Label\_1

```
Label_1: Console.WriteLine("Hello, VB.NET")
```

用 ':' 號將標記與敘述分開

- 標記必須放在敘述的前面，並且以冒號 ':' 分隔。
- 標記之後可以不必接續任何敘述，唯 ':' 仍不可省略。
- 標記的命名規則與識別字的命名規則相同，有關識別字的命名規則請參考2-1節「敘述」。
- 除了標記名稱可以作為標記之外，數字0~9223372036854775807也可以作為標記。
- 已使用的標記不能重複使用。

# 標記的設定

正確用法：

使用例

```
123: Console.WriteLine()
```

```
標記_X: Dim X = 10
```

```
Label_A:
```

說明

數字標記

標記名稱也可以使用中文

標記後面可以不必接續任何敘述

錯誤用法：

錯誤的使用例

```
-123: Dim X = 10
```

```
Label-X: Console.WriteLine()
```

```
100: Console.WriteLine("100")
```

```
100: Console.WriteLine("200")
```

說明

超過數字標記的範圍


使用減號 '-', 不符合識別字命名規則

標記 100 重複使用

# 上機：GoTo 敘述與無窮迴圈

1. 建立一主控台應用程式專案，專案名稱定為 ch0601。
2. 在程式模組視窗中輸入以下程式：

```
Sub Main()  
    Dim I  
L10: I = I + 1  
    Console.WriteLine( I )  
    GoTo L10      ' GoTo 跳到前面的敘述  
End Sub
```



3. 執行程式，結果數字會一直累加，並且不停地顯示，如圖-1。
4. 程式無法正常停止，按下Ctrl+C或選取功能表的「偵錯>>停止偵錯」結束程式。

# 避免無窮迴圈

- 將 GoTo 標記改成：  
If 條件式 Then GoTo 標記
- 實例：

```
Sub Main()  
    Dim I  
L10: I = I + 1  
    Console.WriteLine( I )  
    If I < 10 Then GoTo L10  
End Sub
```

— I 小於 10 才跳回 L10，不然就向下執行

# 上機：只能輸入正數的程式

1. 建立一主控台應用程式專案，專案名稱定為 ch0602。
2. 在程式模組視窗中輸入以下程式：

```
Sub Main()  
    Dim X  
    L10: Console.Write("請輸入正數：")  
        X = Console.ReadLine()  
        If Val(X) <= 0 Then GoTo L10  
        Console.WriteLine("你輸入了 " & X)  
End Sub
```

3. 執行程式，第一次輸入 -99，第二次輸入0，第三次輸入123，結果直到輸入123之後，程式才會結束執行。



## 6-2 條件式

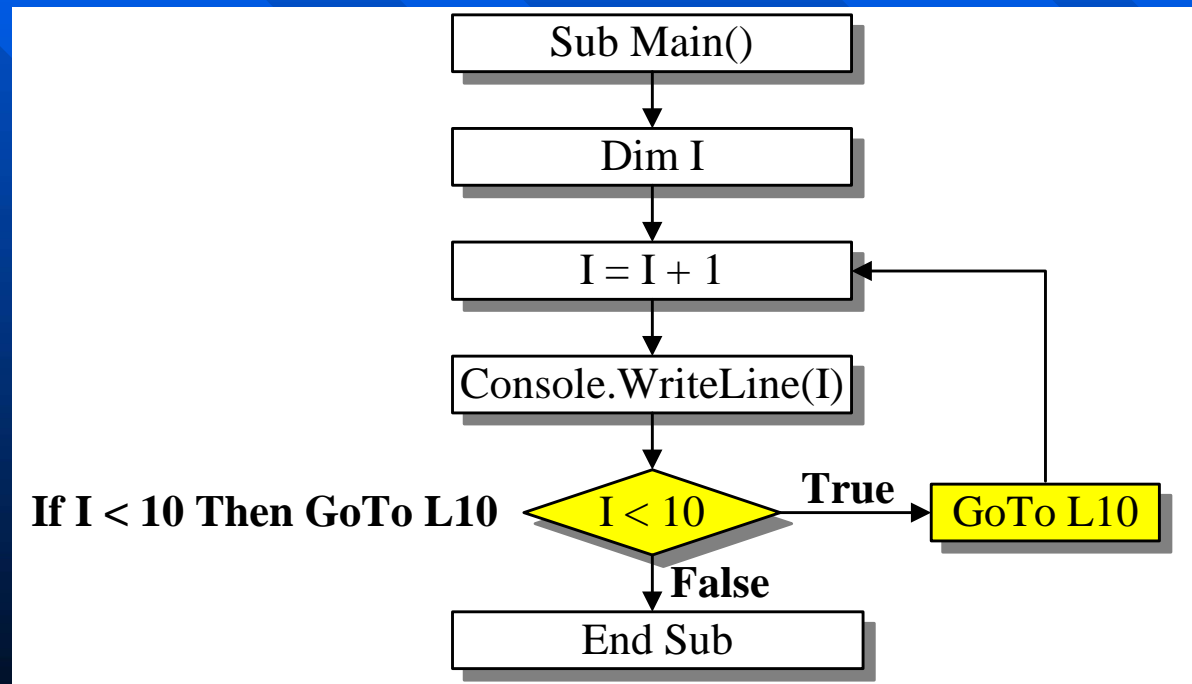
# If 敘述的流程圖



If 敘述的下一行(條件式不成立時)

# If 敘述的流程圖

```
Sub Main()  
    Dim I  
L10: I = I + 1  
    Console.WriteLine( I )  
    If I < 10 Then GoTo L10  
End Sub
```



# 布林資料與條件式

- 爲了方便我們在程式中表示條件式是否成立，VB.NET用**True**來表示條件式成立，用**False**來表示條件式不成立。

- 實例：

```
Dim I, X
```

```
I = 0
```

```
X = I < 10      ' X 等於「I < 10」條件式
```

```
Console.WriteLine( X )      ' 輸出: True
```

```
I = 20
```

```
X = I < 10      ' X 等於「I < 10」條件式
```

```
Console.WriteLine( X )      ' 輸出: False
```

# 比較運算式

- 比較運算在比較 '運算符號' 兩邊的資料，若結果成立，得到True，否則得到False。
- 在VB.NET的程式設計中，常用的比較運算符號有6種，如下表：

比較運算符號	相當於數學上的	使用例
=	=	1 = 2 得到False
>	>	1 > 2 得到False
<	<	1 < 2 得到True
>=	≥	1 >= 2 得到False
<=	≤	1 <= 2 得到True
<>	≠	1 <> 2 得到True

# 日期時間的比較

1. 在時間的比較中，後面的時間大於前面的時間。
  - 實例：  
`#9:30# > #8:30#` 得到 `True`
  - 注意事項：  
`#12:00:00 AM#` 被視為 '0 時'，而  
`#12:00:00 PM#` 被視為一天的中午12時

# 日期時間的比較

2. 在日期的比較中，後面的日期大於前面的日期。

#7/1/2003# > #1/1/2003# 得到 True

3. 同時涵蓋日期及時間的比較中，先比較日期再比較時間。

#1/1/2003 0:00:00# > #12/31/2002  
23:59:59#  
得到 True

#1/1/2003 3:00:00# > #1/1/2003  
0:00:00#  
得到 True

# 日期時間的比較

4. 如果比較的一方有日期，一方沒有日期，則將沒有日期那一方的日期視爲 `#01/01/0001#`，例如 `#20:00:00#` 相當於 `#01/01/0001 20:00:00#`，所以：  
`#1/1/2003 9:00:00# > #20:00:00#` 得到 True
5. 如果比較的一方有時間，一方沒有時間，則將沒有時間那一方的時間視爲 `#0:00:00#`，所以：  
`#1/1/2003# < #1/1/2003 0:0:1#` 得到 True



# 字串的比較

- 比較的依據是字元的字元碼。
  - 比較方式是從字串的第一個字元開始依次比較其字元碼，直到比較出大小為止。
  - 如果未比較出大小，而某一個字串已先達到最後一個字元，則長度較小的字串小於長度較大的字串。

V	B	.	N	E
---	---	---	---	---

1. 先依序比較字元碼

V	B	.	N	E	T
---	---	---	---	---	---

2. 若比較不出，再比較長度

# 取得字元的字元碼

- 英文範圍(0~127)內的字元:利用Asc(“字元”)得知該字元的字元碼。
- 中文字:利用Asc(“中文字”) + 65536得知該中文字的字元碼。
- 實例:
  - `Console.WriteLine( Asc("A") )`  
' 輸出: 65, 表示字元A的字元碼等於65
  
  - `Console.WriteLine( Asc("B") )`  
' 輸出: 66, 字元B的字元碼大於A
  
  - `Console.WriteLine( Asc("程") + 65536 )`  
' 輸出: 46459, 中文字 '程' 的字元碼等於46459

# 判斷字元碼大小的規則

- 英文字母順序在前面者，其字元碼較小，例："a" < "b"。
- 小寫英文字母 "大於" 大寫英文字母，例："a" > "A"。
- 數字 "小於" 英文字母，例："2" < "A"。
- 中文字 "大於" 英文字母，例："程" > "a"。

數字字元 < 英文大寫字母 < 英文小寫字母 < 中文字

# 判斷字元碼的大小

## ■ 實例：

```
"kg" > "KG"           得到 True
"COPY" < "COPYRIGHT"  得到 True
"ANGEL" < "ANGLE"     得到 True
"3-ANGEL" > "ANGLE"   得到 False
"中文" > "English"   得到 True
```

## ■ 練習一：

```
(1) "中國" > "USA"
(2) "cm" < "CM"
(3) "ABC" > "123"
```

## ■ 練習二：

```
(1) Dim A = 100
Dim B = 100 - 50
Console.WriteLine( A > B )
```

## ■ (2) Dim X = "100"

```
Dim Y = "100-50"
Console.WriteLine( X > Y )
```

```
(3) Console.WriteLine( #10/01/1999# > #10/1/1998# )
```

```
(4) Console.WriteLine( "10/01/1999" > "10/1/1998" )
```

# 邏輯運算

運算符號	使用例	功 能	說 明
Not	Not A	反 A	若A為True, 則Not A為False 若A為False, 則Not A為True
And	A And B	A 且 B	只有當A及B同時為True時, 其結果方為True, 否則為False
Or	A Or B	A 或 B	只要A或B為True, 其結果便為True

A	B	Not A	A And B	A Or B
TRUE	TRUE	FALSE	TRUE	TRUE
TRUE	FALSE	FALSE	FALSE	TRUE
FALSE	TRUE	TRUE	FALSE	TRUE
FALSE	FALSE	TRUE	FALSE	FALSE

# 邏輯運算

- 實例：4 > 3 And 1 > 4

4 > 3 And 1 > 4    ‘ 先進行比較運算  
= True And False    ‘ 再進行邏輯運算  
= False

- 實例：(True Or False) And (Not True)

(True Or False) And (Not True)  
= True And False  
= False

- 實例：True Or False And Not True

True Or False And Not True    先運算 Not  
= True Or False And False    再運算 And  
= True Or False 最後才是 Or  
= True

# 實例研究：限制輸入範圍

- 目的：我們在第四章曾經撰寫程式讓使用者輸入「國文」、「數學」、及「英文」成績，然後計算其平均分數(ch0409專案)，現在讓我們來改良這個程式 -- 如果使用者輸入的成績小於0或大於100，則要求其重新輸入。
  1. 開啓ch04目錄的ch0409專案。

# 實例研究：限制輸入範圍

2. 開啓程式模組視窗，檢視其程式碼，如下：

```
Private Sub Button1_Click(...) Handles Button1.Click
```

```
    Dim A, B, C
```

```
    A = Val(InputBox("請輸入[數學]成績"))
```

```
    B = Val(InputBox("請輸入[英文]成績"))
```

```
    C = Val(InputBox("請輸入[國文]成績"))
```

```
    MsgBox((A + B + C) / 3, , "平均分數")
```

```
End Sub
```

以「數學」成績爲例，檢查使用者所輸入的資料是否位於0~100區間，如果不是，則跳回「A = Val(InputBox("請輸入[數學]成績"))」敘述的程式如下：

```
L10:
```

```
    A = Val(InputBox("請輸入[數學]成績"))
```

```
    If A < 0 Or A > 100 Then Goto L10
```



# 實例研究：限制輸入範圍

## 3. 輸入完整的程式。

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim A, B, C
L10:
    A = Val(TextBox1.Text)
    If A < 0 Or A > 100 Then Goto L10
L20:
    B = Val(TextBox2.Text)
    If B < 0 Or B > 100 Then Goto L20
L30:
    C = Val(TextBox3.Text)
    If C < 0 Or C > 100 Then Goto L30
    MsgBox((A + B + C) / 3, , "平均分數")
End Sub
```

## 6-3 有條件執行的 If 敘述

# If-Then 格式

- 格式：  
If 條件式 Then 敘述
- 意義：如果 '條件式' 成立，就執行 Then 之後的 '敘述'，否則繼續向下執行。

# If-Then 實例練習一

## ■ 求 X 的絕對值

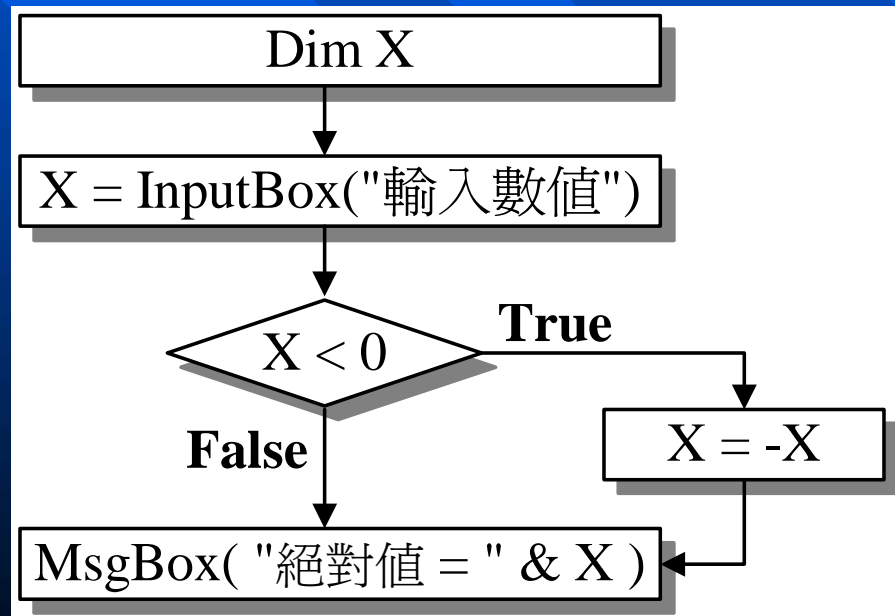
– Dim X

X = InputBox("輸入數值")

If X < 0 Then X = -X

MsgBox("絕對值 = " & X)

– 流程圖：



# If-Then 實例練習二

- 功能：購物促銷專案，凡購物滿500元，超出500元的部分打9折。

- 程式：

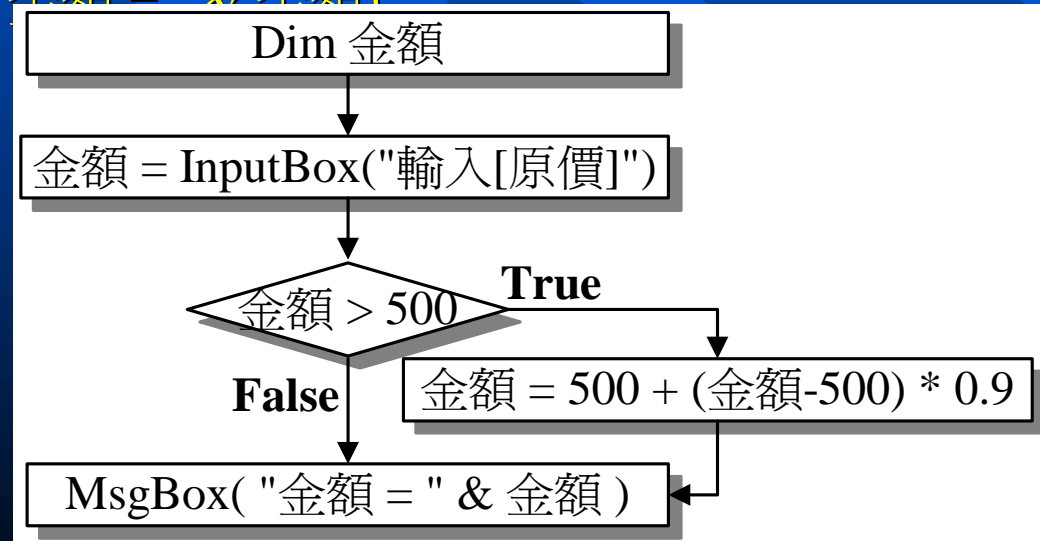
- Dim 金額

- 金額 = InputBox("輸入[原價]")

- If 金額 > 500 Then 金額 = 500 + (金額 - 500) \* 0.9

- MsgBox("金額 = " & 金額)

- 流程圖：



# 結構化的 If-Then 格式

- 當 Then 後面的敘述很多時，整個 If-Then 敘述將會拖得很長，對閱讀來說十分不方便，所以 VB.NET 又提供了另一種結構化的 If-Then 格式。
- 格式：

```
If 條件式 Then
    敘述 1
    敘述 2
    :
End If
```

} 敘述區塊

# 上機：資料互換

- 功能：寫一個含有兩個輸入欄位(TextBox)的表單，輸入資料後比較大小，然後將較大的資料放在第一個欄位，較小的資料放在第二個欄位。
  1. 建立一Windows應用程式專案，專案名稱定為ch0604。
  2. 在表單上佈置兩個TextBox及一個Button，如圖-7。

# 上機：資料互換

3. 在Button1\_Click之中輸入程式.
4. 執行程式，然後分別在TextBox1及TextBox2之中輸入10, 99，接著按下「重排」鈕，結果所輸入之資料會重新排成99, 10，如圖-8。

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim Temp
    If Val(TextBox2.Text) > Val(TextBox1.Text) Then
        Temp = TextBox1.Text
        TextBox1.Text = TextBox2.Text
        TextBox2.Text = Temp
    End If
End Sub
```

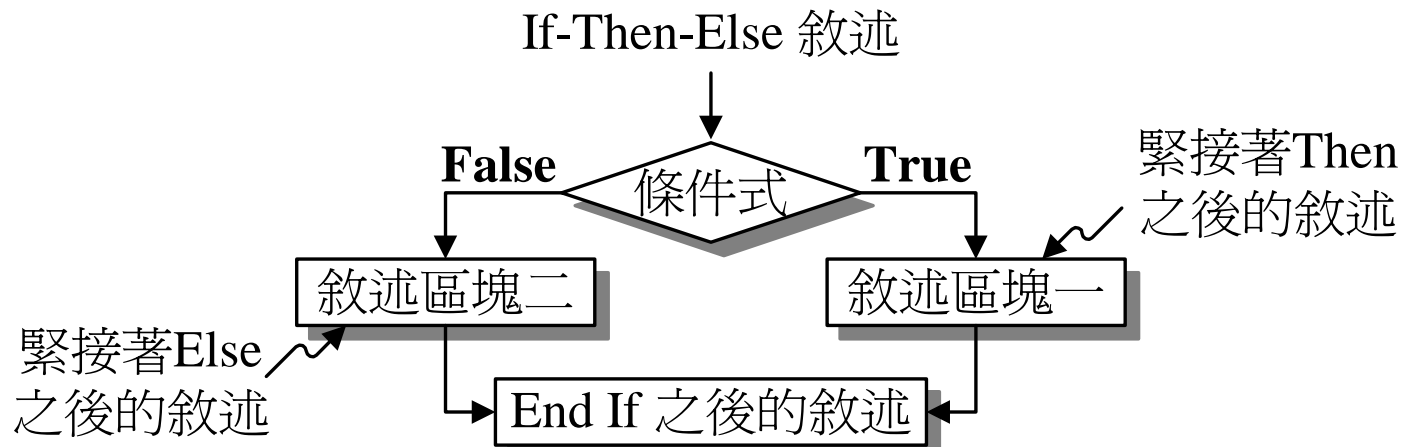


# If-Then-Else 格式

If 條件式 Then 敘述 1 Else 敘述 2

條件式成立      否則

If 條件式 Then 敘述區塊 1  
Else 敘述區塊 2  
End If



# 上機：顯示成績是否及格

1. 建立一主控台應用程式專案，專案名稱定為 ch0605。

2. 輸入程式：

```
Sub Main()  
    Dim 成績  
L10:Console.Write("輸入成績: ")  
    成績 = Console.ReadLine()  
    If 成績 = "" Then Exit Sub  
    If 成績 >= 60 Then  
        Console.WriteLine("及格")  
    Else  
        Console.WriteLine("不及格")  
    End If  
    GoTo L10  
End Sub
```

3. 執行程式，依序輸入90, 50，結果會顯示「及格」及「不及格」，[如圖-11](#)，最後按下 Enter 結束程式。

# 上機：根據時間顯示顏色

- 功能：寫一程式判斷系統時間，如果時間介於6時及18時之間，則將表單的背景顏色設定藍色、前景顏色設定白色、並顯示「日安, 您好!」；否則將表單的背景顏色設定黑色、前景顏色設定黃色、並顯示「晚安, 您好!」。
  1. 建立一Windows應用程式專案，專案名稱定為ch0606。
  2. 在表單上佈置一Label控制元件，並且將此一Label的字型設定成「18pt, 粗體」，如圖-13。

# 上機：根據時間顯示顏色

3. 在表單上Label1之外的位置雙按滑鼠，結果會顯示以下之程式結構：

```
Private Sub Form1_Load(...) Handles MyBase.Load  
End Sub
```

請在其中輸入以下程式：

```
Private Sub Form1_Load(...) Handles MyBase.Load  
    If Hour(Now()) >= 6 And Hour(Now()) < 18 Then  
        Me.BackColor = Color.Blue  
        Me.ForeColor = Color.White  
        Me.Text = "日安, 您好!"  
        Label1.Text = "日安, 您好!"  
    Else  
        Me.BackColor = Color.Black  
        Me.ForeColor = Color.Yellow  
        Me.Text = "晚安, 您好!"  
        Label1.Text = "晚安, 您好!"  
    End If  
End Sub
```

# 巢狀 If 敘述

```
If 條件式 1 Then ←—— If 裡面又有 If, 形成了巢狀 If 敘述
  If 條件式 2 Then
    敘述區塊 1-1
  Else
    敘述區塊 1-2
  End If
  其他敘述
  :
Else
  敘述區塊 2
End If
```

敘述區塊 1

# 上機：成績是否達錄取標準

- 功能：某校電子科招生，錄取標準為數學成績85分(含)以上、所有科目平均75分(含)以上。寫一程式輸入「數學、英文、國文」三科成績，然後計算是否達到錄取標準。
  1. 建立一Windows應用程式專案，專案名稱定為ch0607。
  2. 在表單上佈置三個Label、三個TextBox及一個Button，如圖-14。

# 上機：成績是否達錄取標準

3. 在 Button1\_Click 之中輸入以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim 數學, 英文, 國文 As Double

    數學 = Val(TextBox1.Text)
    英文 = Val(TextBox2.Text)
    國文 = Val(TextBox3.Text)
    If 數學 >= 85 Then
        If 數學 + 英文 + 國文 >= 75 * 3 Then
            MsgBox("達錄取標準!")
        Else
            MsgBox("平均分數未達錄取標準!")
        End If
    Else
        MsgBox("數學成績未達錄取標準!")
    End If
End Sub
```

# 上機：成績是否達錄取標準

4. 執行程式，然後輸入以下三組資料測試程式，結果第1組資料會顯示「達錄取標準!」、第2組資料顯示「平均分數未達錄取標準!」、第3組資料顯示「數學成績未達錄取標準!」。

組別	數學	英文	國文
1	90	70	80
2	90	70	50
3	80	80	90



# 逐一過濾條件的 ElseIf

```
If 條件式1 Then  
    敘述區塊1  
ElseIf 條件式2 Then  
    敘述區塊2  
:  
ElseIf 條件式N Then  
    敘述區塊N  
Else  
    敘述區塊N+1  
End If
```

條件1成立，執行敘述區塊1

條件1不成立，但條件2成立，  
執行敘述區塊2

條件1~N-1都不成立，但條件N  
成立，執行敘述區塊N

以上條件都不成立，執行敘述區塊N+1

# 上機：判斷數字是幾位數

1. 建立一Windows應用程式專案，專案名稱定為ch0608。
2. 在表單上佈置一個TextBox、兩個Label及一個Button，如圖-15。其中設定了以下屬性：

控制元件	屬性	屬性值
Label1	Text	輸入正整數：
Label2	ForeColor	紅色
TextBox1	Text	(清成空白)
Button1	Text	判斷

# 上機：判斷數字是幾位數

3. 在 Button1\_Click 之中輸入以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim 數值 = Val(TextBox1.Text)

    If Not IsNumeric(TextBox1.Text) Then
        Label2.Text = "不是數值!"
    ElseIf 數值 Mod 1 <> 0 Then
        Label2.Text = "不是整數!"
    ElseIf 數值 < 0 Then
        Label2.Text = "不是正整數!"
    ElseIf 數值 < 10 Then
        Label2.Text = "一位數!"
    ElseIf 數值 < 100 Then
        Label2.Text = "二位數!"
    ElseIf 數值 < 1000 Then
        Label2.Text = "三位數!"
    Else
        Label2.Text = "四位數(含)以上!"
    End If
End Sub
```

# 上機：判斷數字是幾位數

4. 執行程式，輸入以下資料進行測試：

輸入之資料	結果
ABC	不是數值!
10.2	不是整數!
-123	不是正整數!
1	一位數!
12	二位數!
123	三位數!
1234	四位數(含)以上!

# 上機：購物打折程式

- 功能：購物促銷方案，凡購物滿1000元，打折方式如下：

金額	折扣
超出3000元部分	7折
超出2000元但未超出3000元部分	8折
超出1000元但未超出2000元部分	9折
未超出1000元部分	不打折

1. 建立一Windows應用程式專案，專案名稱定為ch0609。
2. 在表單上佈置三個Label、一個TextBox及一個Button，如圖-16。

# 上機：購物打折程式

3. 在 Button1\_Click 之中輸入以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim 原價 = Val(TextBox1.Text)

    If 原價 > 3000 Then
        Label3.Text = (原價 - 3000) * 0.7 + 2700
    ElseIf 原價 > 2000 Then
        Label3.Text = (原價 - 2000) * 0.8 + 1900
    ElseIf 原價 > 1000 Then
        Label3.Text = (原價 - 1000) * 0.9 + 1000
    Else
        Label3.Text = 原價
    End If
End Sub
```

# 上機：購物打折程式

4. 執行程式，然後輸入以下幾組資料進行測試：

資料	符合之條件式	結果	計算過程
3900	原價 > 3000	3330	$(3900-3000)*0.7 + 2700 = 3330$
2900	原價 > 2000	2620	$(2900-2000)*0.8 + 1900 = 2620$
1900	原價 > 1000	1810	$(1900-100)*0.9 + 1000 = 1810$
900	Else	900	900

6-4

# 多執行路徑的 Select Case 敘述



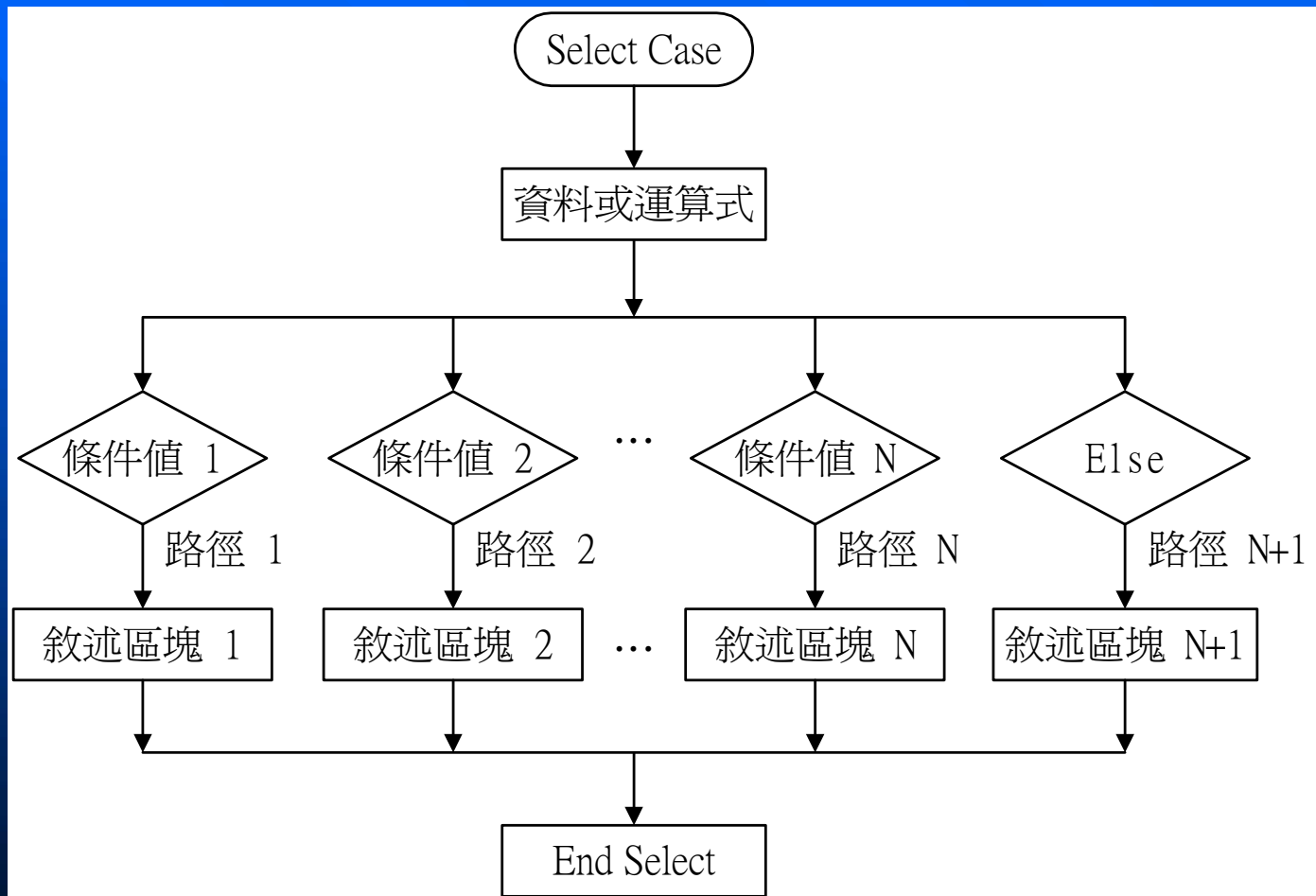
# Select Case 敘述的格式

## ■ 格式：

```
Select Case 資料或運算式
  Case 條件值 1
    敘述區塊 1
  Case 條件值 2
    敘述區塊 2
  :
  Case 條件值 N
    敘述區塊 N
  Case Else
    敘述區塊 N+1
End Select
```

- 資料(或運算式)的值符合條件值1時，執行敘述區塊1，符合條件值2時，執行敘述區塊2...，符合條件值N時，執行敘述區塊N，如果任何條件值都不符合，則執行敘述區塊N+1

# Select Case 敘述的流程圖



# 上機：顯示四季名稱

- 功能：輸入1, 2, 3或4，然後顯示對應的「春, 夏, 秋, 冬」季節名稱。
  1. 建立一Windows應用程式專案，專案名稱定為ch0610。
  2. 在表單上佈置三個Label、一個TextBox及一個Button，如圖-19。

# 上機：顯示四季名稱

3. 在 Button1\_Click 之中輸入以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim 季節順序 = Val(TextBox1.Text)

    Select Case 季節順序
        Case 1
            Label3.Text = "春"
        Case 2
            Label3.Text = "夏"
        Case 3
            Label3.Text = "秋"
        Case 4
            Label3.Text = "冬"
        Case Else
            Label3.Text = "輸入錯誤"
    End Select
End Sub
```

# 上機：顯示四季名稱

4. 執行程式，然後輸入以下幾組資料進行測試：

資料(TextBox1)	結果(Label3)
1	春
2	夏
3	秋
4	冬
5	輸入錯誤
1.2	輸入錯誤

# 上機：顯示英文季節名稱

- 功能：輸入中文季節名稱「春, 夏, 秋, 冬」、然後顯示對應的英文季節名稱「spring, summer, autumn, winter」。
- 1. 建立一Windows應用程式專案，專案名稱定為ch0611。
- 2. 在表單上佈置三個Label、一個TextBox及一個Button，如圖-20。

# 上機：顯示英文季節名稱

3. 在 Button1\_Click 之中輸入以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim 中文季節 = TextBox1.Text

    Select Case 中文季節
        Case "春"
            Label3.Text = "spring"
        Case "夏"
            Label3.Text = "summer"
        Case "秋"
            Label3.Text = "autumn"
        Case "冬"
            Label3.Text = "winter"
        Case Else
            Label3.Text = "輸入錯誤"
    End Select
End Sub
```

# 上機：顯示英文季節名稱

4. 執行程式，然後輸入以下幾組資料進行測試：

資料(TextBox1)	結果(Label3)
春	spring
夏	summer
秋	autumn
冬	winter
碰	輸入錯誤



# 上機：電影分級

- 功能：輸入年齡，然後顯示該年齡可以觀賞之電影等級。
  1. 建立一Windows應用程式專案，專案名稱定為ch0612。
  2. 在表單上佈置三個Label、一個TextBox及一個Button，如圖-21。

# 上機：電影分級

## 3. 在 Button1\_Click 之中輸入以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim 年齡 = Val(TextBox1.Text)

    If 年齡 Mod 1 <> 0 Then
        Label3.Text = "請輸入整數"
        Exit Sub
    End If

    Select Case 年齡
        Case 0 To 5 ' age 等於 0 到 5，都符合這個條件
            Label3.Text = "普通級"
        Case 6, 7, 8, 9, 10, 11 ' 逐一列出條件值
            Label3.Text = "保護級"
        Case 12 To 17
            Label3.Text = "輔導級"
        Case Is >= 18 ' age ≥ 18，都符合這個條件
            Label3.Text = "限制級"
    End Select
End Sub
```

# 上機：電影分級

4. 執行程式，然後輸入以下幾組資料進行測試：

資料(TextBox1)	結果(Label3)
1	普通級
6	保護級
12	輔導級
18	限制級
3.5	請輸入整數

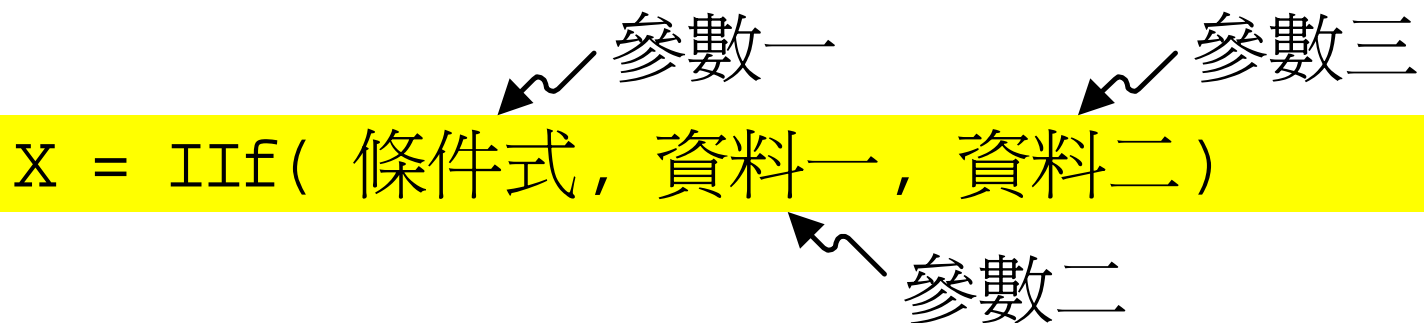
## 6-5 If、Choose、Switch函數

# Iif：兩選一的函數

X = Iif( 條件式, 資料一, 資料二 )

參數一      參數三

參數二

The diagram shows the function call 'X = Iif( 條件式, 資料一, 資料二 )' on a yellow background. Three arrows point from labels above and below to the parameters: '參數一' points to '條件式', '參數二' points to '資料一', and '參數三' points to '資料二'.

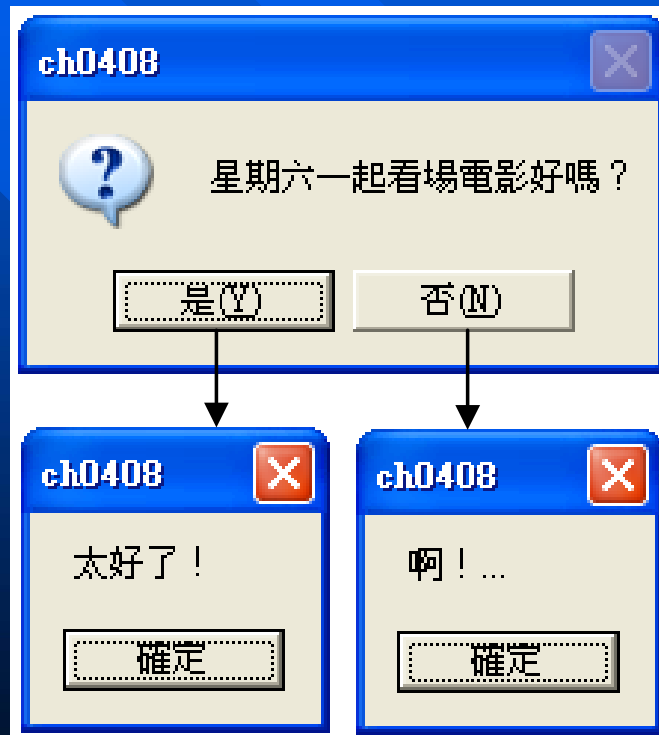
- 說明：當「條件式」(參數一)成立時，Iif將選擇「資料一」(參數二)指定給X、否則選擇「資料二」(參數三)指定給X。

# Iif 函數實例

- Dim 回答, 反應

```
回答 = MsgBox( "星期六一起看場電影好嗎?", _  
                MsgBoxStyle.YesNo + MsgBoxStyle.Question)
```

```
反應 = Iif( 回答 = MsgBoxResult.Yes, "太好了!", "啊! ..." )  
MsgBox( 反應 )
```



# 上機：改用 If 完成 p.217 程式

1. 開啓p.217 所建立的專案ch0605。
2. 開啓程式模組視窗，檢視以下程式：

```
Sub Main()  
    Dim 成績  
L10:Console.Write("輸入成績: ")  
    成績 = Console.ReadLine()  
    If 成績 = "" Then Exit Sub  
    If 成績 >= 60 Then  
        Console.WriteLine("及格")  
    Else  
        Console.WriteLine("不及格")  
    End If  
    GoTo L10  
End Sub
```

# 上機：改用 Iif 完成 p.217 程式

3. 將以上所列出之加底線程式刪除，然後改成如以下的粗體字部分：

```
Sub Main()  
    Dim 成績  
L10:Console.Write("輸入成績:")  
    成績 = Console.ReadLine()  
    If 成績 = "" Then Exit Sub  
    Dim 結果 = Iif(成績 >= 60, "及格", "不及格")  
    Console.WriteLine(結果)  
    GoTo L10  
  
End Sub
```



# Choose：對號選取的函數

- Choose是對號選取資料的函數，使用時第1個參數要傳入編號、第2, 3, 4...以後的參數則傳入被選取的資料，格式如下：

$X = \text{Choose}(\text{編號}, \text{資料一}, \text{資料二}, \dots)$

- 當「編號」等於1時，Choose將選擇「資料一」指定給X；等於2時，則選擇「資料二」指定給X...，依此類推。

# Choose：對號選取的函數

## ■ 實例：

```
Dim 編號 = InputBox(“請選擇遊戲等級：1.入門 2.進階 3.專家”)
```

```
Dim 結果 = Choose(編號, “入門”, “進階”, “專家”)
```

- 上程式經執行後會顯示一輸入窗讓使用者輸入「編號」，而所得到的「結果」一覽表如下：

編號	結果
1	入門
去除小數點後等於1的數值	入門
2	進階
去除小數點後等於2的數值	進階
3	專家
去除小數點後等於3的數值	專家
其他資料	(空字串)

# 上機:改用Choose完成 p.231 程式

1. 開啓 p.231 所建立的專案ch0610。
2. 開啓程式模組視窗，檢視以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Click  
    Dim 季節順序 = Val(TextBox1.Text)
```

```
    Select Case 季節順序
```

```
        Case 1
```

```
            Label3.Text = "春"
```

```
        Case 2
```

```
            Label3.Text = "夏"
```

```
        Case 3
```

```
            Label3.Text = "秋"
```

```
        Case 4
```

```
            Label3.Text = "冬"
```

```
        Case Else
```

```
            Label3.Text = "輸入錯誤"
```

```
    End Select
```

```
End Sub
```

# 上機:改用Choose完成 p.231 程式

3. 將以上所列出之加底線程式刪除，然後改成如以下的粗體字部分：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim 季節順序 = Val(TextBox1.Text)

    Label3.Text = Choose( 季節順序, “春”, “夏”, “秋”, “冬” )
End Sub
```

# Switch：多選一的函數

- Switch是多選一的函數，使用時每傳入一個條件式，就要相對地傳入一個選項，例如：

```
Dim 英文季節 = Microsoft.VisualBasic.Switch( _  
    中文季節 = "春", "spring", _  
    中文季節 = "夏", "summer", _  
    中文季節 = "秋", "autumn", _  
    中文季節 = "冬", "winter")
```

- 其中含有4個條件式及4個選項：

條件式	對應的選項
中文季節 = "春"	spring
中文季節 = "夏"	summer
中文季節 = "秋"	autumn
中文季節 = "冬"	winter

# 上機:改用Switch完成p.233 程式

1. 開啓 p.233 所建立的專案ch0612。
2. 開啓程式模組視窗，檢視以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Cli
    Dim 年齡 = Val(TextBox1.Text)

    If 年齡 Mod 1 <> 0 Then
        Label3.Text = "請輸入整數"
        Exit Sub
    End If

    Select Case 年齡
        Case 0 To 5 ' age 等於 0 到 5，都符合這個條件
            Label3.Text = "普通級"
        Case 6, 7, 8, 9, 10, 11 ' 逐一列出條件值
            Label3.Text = "保護級"
        Case 12 To 17
            Label3.Text = "輔導級"
        Case Is >= 18 ' age ≥ 18，都符合這個條件
            Label3.Text = "限制級"
    End Select
End Sub
```

# 上機:改用Switch完成p.233 程式

3. 將以上所列出之加底線程式刪除，然後改成如以下的粗體字部分：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim 年齡 = Val(TextBox1.Text)

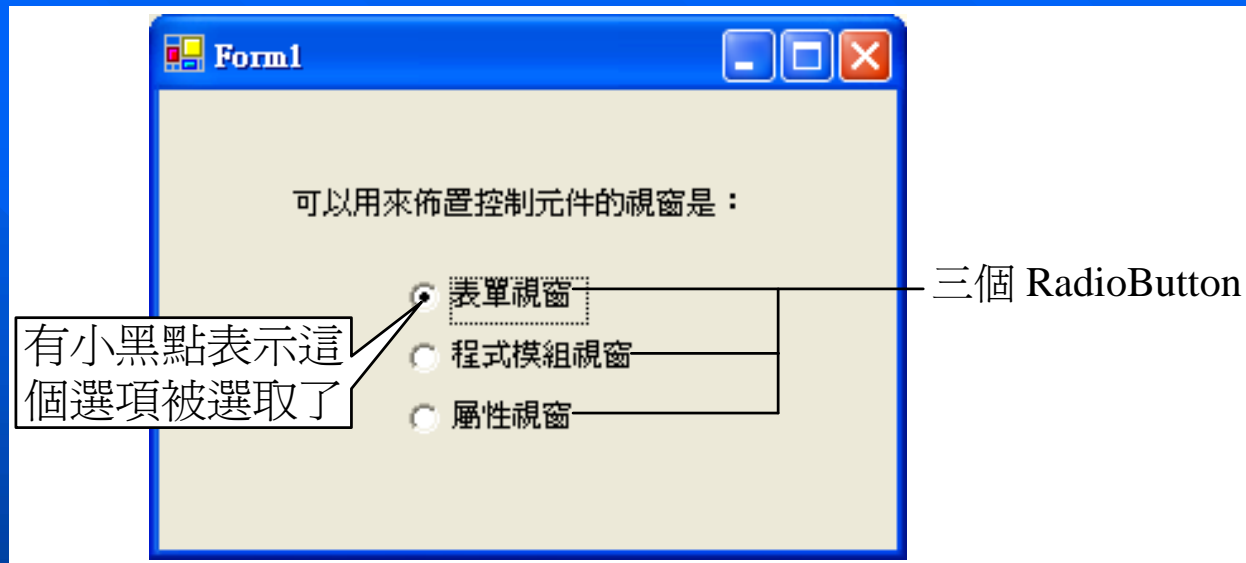
    Label3.Text = Microsoft.VisualBasic.Switch( _
        年齡 >= 0 And 年齡 < 6, "普通級", _
        年齡 >= 6 And 年齡 < 12, "保護級", _
        年齡 >= 12 And 年齡 < 18, "輔導級", _
        年齡 >= 18, "限制級")

End Sub
```

## 6-6 選擇類控制元件



# RadioButton(選擇鈕)



- RadioButton 的功用好像考試時的單選題，例如有一考題：「可以用來佈置控制元件的視窗是：(1) 表單視窗 (2) 程式視窗 (3) 專案視窗」，若使用RadioButton來設計其畫面，則如上圖，我們分別以三個RadioButton來代表三種選擇，而其中含有黑點的小圓圈表示被選擇的選項。

# 上機：體驗RadioButton的特性

1. 建立一Windows應用程式專案，專案名稱定為ch0613。
2. 在表單上佈置一個Label及三個RadioButton，如圖-24。其中設定的屬性如下：

控制元件	屬性	屬性值
Label1	Text	可以用來佈置控制元件的視窗是：
RadioButton1	Text	表單視窗
RadioButton2	Text	程式模組視窗
RadioButton3	Text	屬性視窗

# 上機：體驗RadioButton的特性

3. 執行程式，然後以滑鼠選取「表單視窗」，結果「表單視窗」這個RadioButton就含有小黑點，表示此一RadioButton被選取了。
4. 再以滑鼠選取「屬性視窗」，結果「表單視窗」這個RadioButton的小黑點消失，變成「屬性視窗」這個RadioButton含有小黑點，表示被選取的選項變成了「屬性視窗」，如圖-25。

# 讀取 RadioButton 的選項

- 當某一個RadioButton被選取時，該RadioButton的Checked屬性等於True，而其他RadioButton的Checked屬性則為False，因此我們可以利用以下程式碼來判斷哪一個RadioButton被選取了：
  - `If RadioButton1.Checked Then`
    - ‘ 第一個RadioButton被選取了
  - `ElseIf RadioButton2.Checked Then`
    - ‘ 第二個RadioButton被選取了
  - `ElseIf RadioButton3.Checked Then`
    - ‘ 第三個RadioButton被選取了
  - `End If`

# 上機：讀取 RadioButton 的選項

- 功能：修改剛才的ch0613專案，當使用者選取「表單視窗」時，顯示「正確」；選取「程式模組視窗」時，顯示「程式模組視窗用於撰寫程式」；選取「屬性視窗」時，顯示「屬性視窗用於設定物件的屬性」。
  1. 開啓ch0613專案。
  2. 在表單上增加一個Button，如圖-26。

# 上機：讀取 RadioButton 的選項

3. 在 Button1\_Click 之中輸入以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Click
    If RadioButton1.Checked Then
        MsgBox("正確", MsgBoxStyle.Information)
    ElseIf RadioButton2.Checked Then
        MsgBox("程式模組視窗用於撰寫程式", MsgBoxStyle.Critical)
    ElseIf RadioButton3.Checked Then
        MsgBox("屬性視窗用於設定物件的屬性",
        MsgBoxStyle.Critical)
    End If
End Sub
```

4. 執行程式，選擇任何一個RadioButton，然後按下「答題」鈕以測試程式。

# GroupBox 控制元件

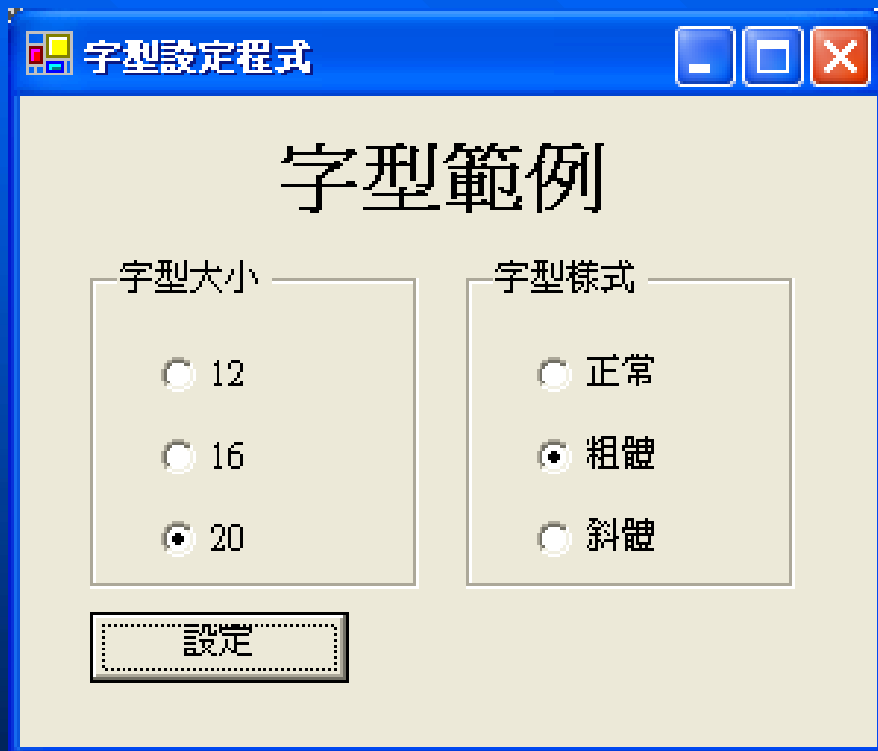
- 與 RadioButton 的分組
  - 剛才我們看到的表單只有1個題目，而這個題目含有3個選項，如果說我們想把表單改成含有3個題目，而每個題目各有3個選項(合計9個RadioButton)，那麼必須利用(群組方塊)控制元件把這9個RadioButton分成3組，使用者方才得以從中選取3個選項。
- GroupBox的功用是把數個控制元件框起來，構成控制元件群組，如圖-27。

# 上機：如何正確地將控制元件 佈置於 GroupBox 之中

1. 建立一Windows應用程式專案，專案名稱定為ch0614。
2. 先佈置一GroupBox，並將GroupBox拉曳至適當的大小。
3. 選取工具箱的RadioButton，然後將滑鼠移到GroupBox裡面，以拉曳的方式佈置RadioButton，如圖-28。
4. 仿照步驟3，佈置RadioButton2～RadioButton4。



# 上機：字型設定程式



- 功能：設計一個可以改變字型大小及樣式的程式。

# 上機：字型設定程式

1. 建立一Windows應用程式專案，專案名稱定為ch0615。
2. 在表單上佈置一個Label、兩個GroupBox、六個RadioButton及一個Button，如圖-30。其中所設定之屬性如右表：

控制元件	屬性	屬性值
Label1	Text	字型範例
	TextAlign	MiddleCenter
	Font	大小=12
GroupBox1	Text	字型大小
GroupBox2	Text	字型樣式
RadioButton1	Text	12
	Checked	TRUE
RadioButton2	Text	16
RadioButton3	Text	20
RadioButton4	Text	正常
	Checked	TRUE
RadioButton5	Text	粗體
RadioButton6	Text	斜體
Button1	Text	設定
Form1(表單)	Text	字型設定程式

# 上機：字型設定程式

3. 在Button1\_Click之中輸入程式。
4. 執行程式，選取「字型大小」12, 16, 20及選取「字型樣式」之中的 正常, 粗體, 斜體，然後按下「設定」鈕測試程式。

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim 字型名稱 As String = Label1.Font.Name
    Dim 字型大小 As Integer = Label1.Font.Size
    Dim 字型樣式 As FontStyle = Label1.Font.Style

    If RadioButton1.Checked Then 字型大小 = 12
    If RadioButton2.Checked Then 字型大小 = 16
    If RadioButton3.Checked Then 字型大小 = 20
    If RadioButton4.Checked Then 字型樣式 = FontStyle.Regular
    If RadioButton5.Checked Then 字型樣式 = FontStyle.Bold
    If RadioButton6.Checked Then 字型樣式 = FontStyle.Italic

    Label1.Font = New Font(字型名稱, 字型大小, 字型樣式)
End Sub
```

# CheckBox(核取方塊)

- CheckBox(核取方塊)也是用來選取資料的控制元件，它的用法與RadioButton幾乎完全一樣，不同的是CheckBox並不受單選的限制，也就是說，我們可以選擇0到多個CheckBox所代表的資料，[如圖-31](#)。在這個表單中，佈置了六個CheckBox，並且選取了其中三個。
- 程式中判斷使用者選取了哪幾個CheckBox，也是根據Checked屬性，當CheckBox的Checked屬性值為True時，表示該CheckBox被選取了，而由於可能複選的關係，所以每一個CheckBox都要獨立判斷。

# 上機：計算點餐金額

The screenshot shows a software application window titled "美餓美餐廳" (Mei E Mei Restaurant). The window contains a menu with four items, each with a checkbox and a quantity input field:

Item	Price	Quantity	Unit
<input checked="" type="checkbox"/> 滷肉飯	30 元	2	份
<input type="checkbox"/> 雞腿飯	70 元		份
<input checked="" type="checkbox"/> 排骨飯	65 元	2	份
<input type="checkbox"/> 美餓美特餐	99 元		份

At the bottom of the window is a button labeled "計算金額" (Calculate Amount). A dialog box titled "ch0616" is overlaid on the right side of the window, displaying the text "總金額=190" (Total Amount=190) and a button labeled "確定" (Confirm).

# 上機：計算點餐金額

1. 建立一Windows應用程式專案，專案名稱定為ch0616。
2. 在表單上佈置4個CheckBox、4個TextBox、4個Label及一個Button，如圖-33。其中所設定之屬性如下：

控制元件	屬性	屬性值	控制元件	屬性	屬性值
CheckBox1	Text	滷肉飯(30元)	CheckBox2	Text	雞腿飯(70元)
CheckBox3	Text	排骨飯(65元)	CheckBox4	Text	美餓美特餐(99元)
TextBox1	Text	(清成空白)	TextBox2	Text	(清成空白)
TextBox3	Text	(清成空白)	TextBox4	Text	(清成空白)
Label1	Text	份	Label2	Text	份
Label3	Text	份	Label4	Text	份
表單	Text	美餓美餐廳	Button1	Text	計算金額

# 上機：計算點餐金額

3. 輸入程式。
4. 執行程式，假設選取所有CheckBox，並且在4個TextBox之中分別輸入1, 2, 3, 1，則結果顯示的總金額應等於464。

```
Private Sub Button1_Click(...) Handles Button1.Clic
    Dim 總金額 As Long

    If CheckBox1.Checked Then      ' 滷肉飯
        總金額 = 總金額 + 30 * Val(TextBox1.Text)
    End If
    If CheckBox2.Checked Then      ' 雞腿飯
        總金額 = 總金額 + 70 * Val(TextBox2.Text)
    End If
    If CheckBox3.Checked Then      ' 排骨飯
        總金額 = 總金額 + 65 * Val(TextBox3.Text)
    End If
    If CheckBox4.Checked Then      ' 美餓美特餐
        總金額 = 總金額 + 99 * Val(TextBox4.Text)
    End If
    MsgBox("總金額=" & 總金額)
End Sub
```