

# 第九章 常用控制項(一)



9.1 Timer 計時控制項

9.2 PictureBox 圖片方塊控制項

9.3 GroupBox/Panel 容器控制項

9.4 RadioButton/CheckBox 選  
擇控制項

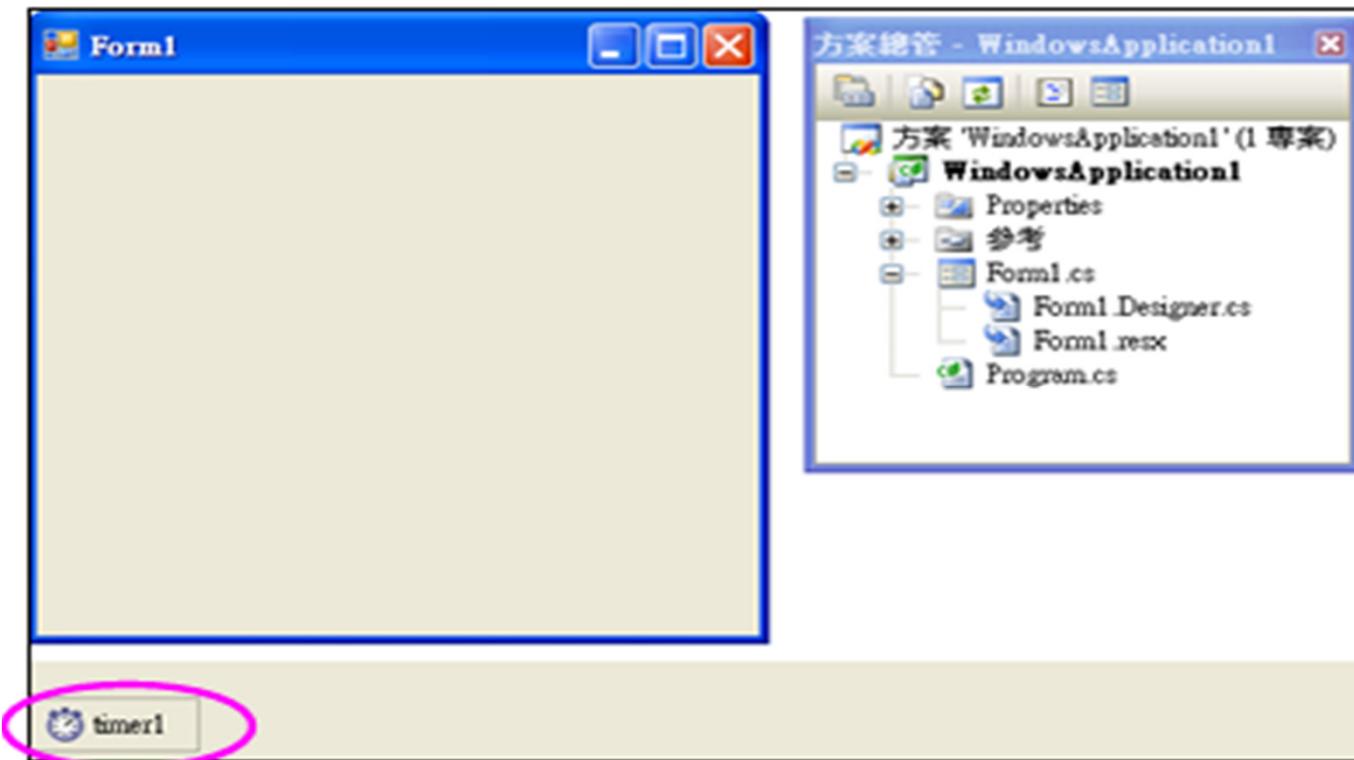
備註：可依進度點選小節



## 9.1 Timer計時控制項



- 設計程式時，若想製作動畫、延遲時間或每隔多少時間就執行某項工作等都可用計時控制項來完成。
- 如使用 **for**、**while{...}**、**do{...} while** 迴圈來控制時間延遲程式，相同程式碼會因不同電腦不同速度的**CPU**，得到不同時間延遲。
- 改用 **Timer** 計時控制項，因用 **CPU** 本身的計時器來計時，不會發生上述問題。
- 在表單建立計時控制項，該控制項不置於表單上，在表單正下方，表該控制項屬於非視覺化控制項。
- 非視覺化控制項在程式執行時是幕後執行，在表單上看不到計時控制項。





## 一. Timer常用屬性

屬性	說明
Interval	用來設定計時器的週期時間，預設值為1000，單位為毫秒(10-3秒)。所以Interval屬性值若設為1000表示每隔一秒就會去執行計時控制項的Tick事件內的程式碼一次，其先決條件是Enabled屬性必須先設為True，將電腦的計時器啟動才有效。
Enabled	用來設定是否啟動計時器。若設為False，表示計時器不啟動。若設為True時，表示啟動計時器開始計時。預設值為False。



## 二. Timer常用事件

- **Tick** 事件是計時控制項特有的事件
- 如 **Enabled =True**，每當設定的 **Interval** 屬性值週期一到，會觸動 **Tick** 事件。將該事件要執行的程式碼，撰寫在**Tick**事件處理函式中。



## 範例演練

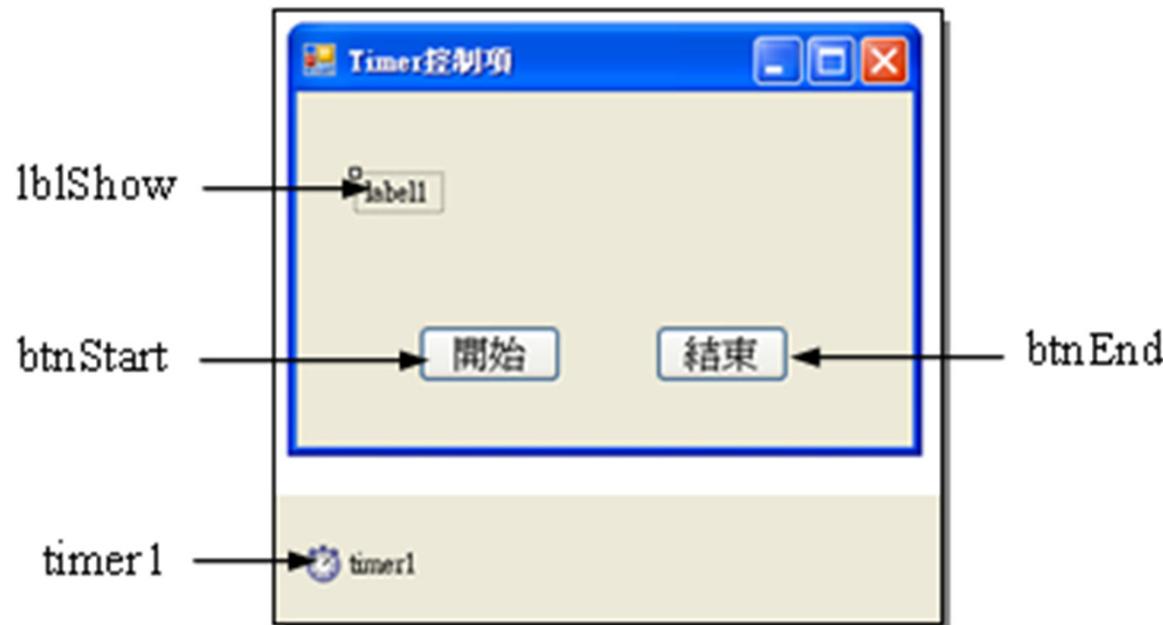
試設計一個看板，當按 **開始** 鈕開始計時，每隔一秒鐘，交互顯示淺藍底的”Show Time...”及黃底的”好戲上場!”訊息。按 **結束** 鈕結束程式執行。





上機

## ■ Step1 設計輸出入介面





## ■ Step2 撰寫程式碼

```
// FileName : TVWall.sln

01

02 private void Form1_Load(object sender, EventArgs e)

03 {

04     lblShow.Text = "";

05     lblShow.Font = new Font("標楷體", 24, FontStyle.Bold);

06     //設定timer1控制項每1秒執行timer1_Tick事件處理函式一次

07     timer1.Interval = 1000;

08 }
```



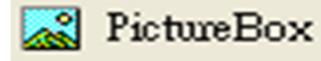
```
09 //按下btnStart鈕時執行btnStart_Click()事件處理函式
10 private void btnStart_Click(object sender, EventArgs e)
11 {
12     timer1.Enabled = true; //啟動timer1計時器
13 }
14 //按下btnEnd鈕時執行btnEnd_Click()事件處理函式
15 private void btnEnd_Click(object sender, EventArgs e)
16 {
17     Application.Exit(); //結束程式
18 }
19 //每1秒執行timer1_Tick事件處理函式一次
20 private void timer1_Tick(object sender, EventArgs e)
21 {
22     if (lblShow.Text == "好戲上場!")
23     {
24         lblShow.Text = "Show Time...";
25         lblShow.BackColor = Color.Aqua;
26     }
```



```
27     else  
28     {  
29         lblShow.Text = "好戲上場!";  
30         lblShow.BackColor = Color.Yellow;  
31     }  
32 }
```



## 9.2 PictureBox圖片方塊控制項



- 圖片方塊控制項主要是用來顯示圖檔，它也允許您在該控制項上面繪製圖形。
- 若要將圖片或影像顯示在圖片控制項上，可以在設計階段預先載入，也可以在執行階段透過 `Image.FromFile()`方法載入。
- **PictureBox**圖片控制項允許載入的圖檔格式主要有 **bmp**、**jpg**、**gif**、**wmf** 等。
- 若一幅動畫是由 2-4 張連續 **bmp**、**wmf** 或 **jpg** 圖檔組成，只要將這些連續圖檔交互置於變動位置的 **PictureBox** 控制項上，即可展現出生動的動畫。



## 9.2.1 PictureBox常用屬性

下表為PictureBox圖片方塊控制項常用屬性。

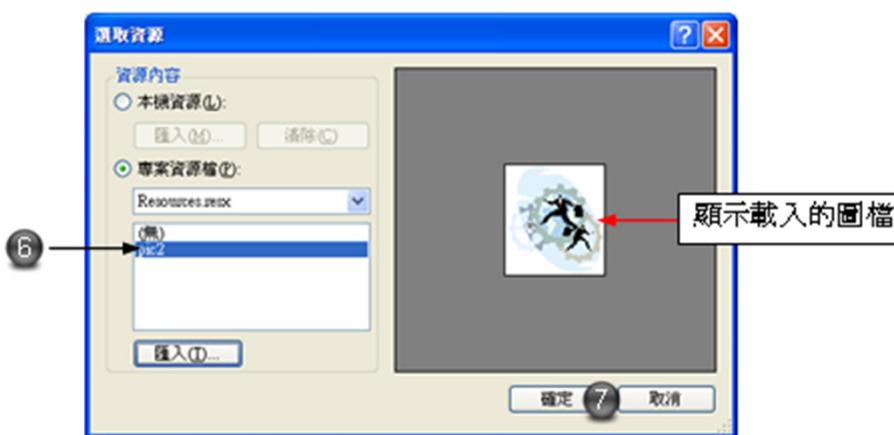
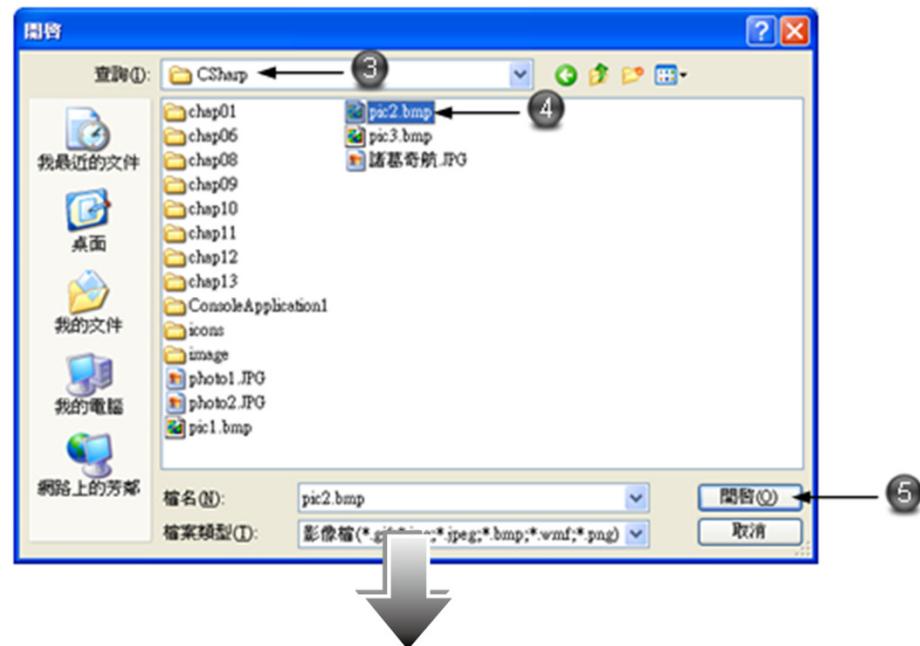
屬性	說明
Image	設定顯示的圖形檔，預設無。
SizeMode	設定圖形在PictureBox控制項上顯示的位置： ① <b>Normal</b> (預設值)：圖形由控制項的左上角開始放起，圖形檔長寬不變。 ② <b>CenterImage</b> ：顯示在控制項的正中央。 ③ <b>AutoSize</b> ：控制項依圖形大小縮放。 ④ <b>StretchImage</b> ：圖形依控制項大小縮放。
Visible	設定控制項是否顯現。 ① <b>True</b> ：表示可顯示(預設值)。 ② <b>False</b> ：表示隱藏該控制項。



## 9.2.2 圖檔的載入與移除

- 圖形檔可在設計階段或者在執行階段才載入。
- 使用時機當然視當時需求而定。
- 在設計階段載入圖檔







## 二. 如何在執行階段載入圖檔

■ 程式中使用 `Image.FromFile()` 靜態方法和 `new Bitmap()` 物件來設定 `Image` 屬性值，達到載入圖檔的目的。

■ 語法

1. `pictureBox1.Image=Image.FromFile(path); // path為檔案路徑`
2. `pictureBox1.Image=new Bitmap(path);`



## ■ 簡例：

1. 載入固定路徑圖檔，若將程式安裝在不同的硬碟或資料夾時會產生錯誤。圖檔必須安裝在固定路徑，如下兩種寫法是載入 C:\image\ok.bmp 至 **pictureBox1** 的 **Image** 屬性：

寫法1：

```
pictureBox1.Image=Image.FromFile("C:\\image\\ok.bmp");
```

寫法2：

```
pictureBox1.Image=new Bitmap("C:\\image\\ok.bmp");
```



2. 載入相對路徑的圖檔，路徑以 C# 執行檔(\bin\debug 資料夾)和圖檔的相對位置來表示，可以避免使用者安裝在不同資料夾產生的錯誤，如下四種簡例示範：

① 欲載入的 ok.bmp 圖檔和執行檔在同一資料夾下，可採用下面兩種寫法：

寫法1：pictureBox1.Image=Image.FromFile("ok.bmp");

寫法2：pictureBox1.Image=new Bitmap("ok.bmp");



- ② 欲載入的ok.bmp圖檔位在程式執行檔的上一層資料夾內，可採用下面兩種寫法：

寫法1：**pictureBox1.Image=Image.FromFile("..\\ok.bmp");**

寫法2：**pictureBox1.Image=new Bitmap(" ..\\ok.bmp");**

- ③ 欲載入的 ok.bmp 圖檔位在執行檔的上一層的 image 資料夾內，可採用下面兩種寫法：

寫法1：

**pictureBox1.Image=Image.FromFile("..\\image\\ok.bmp");**

寫法2：**pictureBox1.Image=new Bitmap(" ..\\image\\ok.bmp");**



④ 欲載入的 **ok.bmp** 圖檔位在執行檔的上兩層的 **image** 資料夾內，可採用下面兩種寫法：

寫法1：

```
pictureBox1.Image=Image.FromFile("..\\..\\image\\ok.bmp");
```

寫法2：

```
pictureBox1.Image=new Bitmap(" ..\\..\\image\\ok.bmp");
```



### 三. 如何在設計階段移除圖檔

1. 點選 Image 屬性欄，然後按 **Del** 鍵，可移除原先載入的圖檔。
2. 在 Image 屬性上面或是右邊的 **...** 鈕上按右鍵，點選「重設」功能即可。

### 四. 如何在執行階段移除圖檔

將 PictureBox 的 Image 屬性值設為 null 就可將圖檔清除，寫法將 pictureBox1 的圖檔清除。

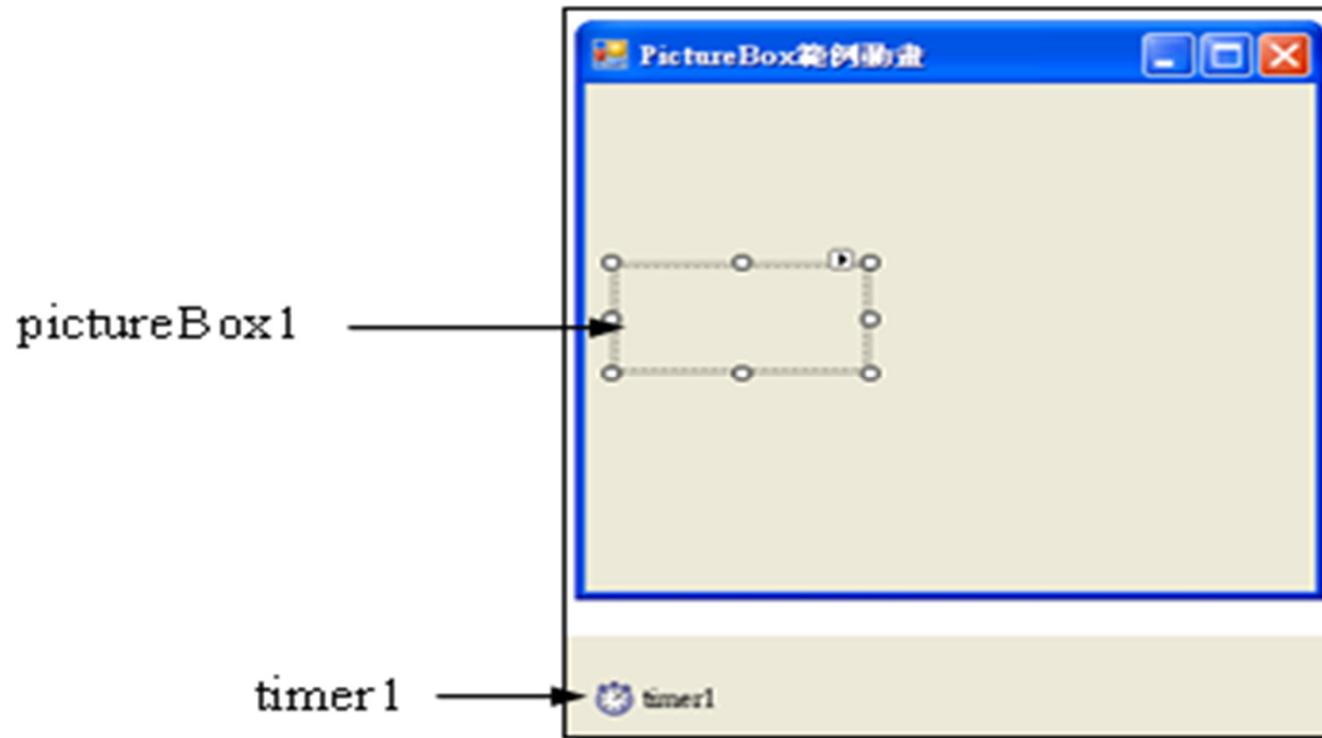
**pictureBox1.Image = null;**



## 範例演練

以「山景.jpg」當表單背景圖，以 bird0.png~bird5.png 六個連續圖片製作一個小鳥由左往右移動並依序輸流播放 bird0.png~bird5.png的動畫，當小鳥超出表單的最右邊時會再由表單最左邊出現並繼續由左往右移動。







若 pictureBox1(小鳥)超出表單的右邊寬度，則馬上由表單左邊出現



## ■ 完整程式碼

01 **FileName : Animation.sln**

02 **int n = 0; // 成員變數**

03 **Bitmap[] img = new Bitmap[6];**

04 **private void Form1\_Load(object sender, EventArgs e)**

05 {

06

07 **for (int i = 0; i <= img.GetUpperBound(0); i++)**

08 {

09 **img[i] = new Bitmap("bird" + i.ToString() + ".png");**

10 }



```
11  this.BackgroundImage = new Bitmap("山景.jpg");
12  this.Width = 430;
13  this.Height = 300;
14  //設定表單無法調整大小
15  this.AutoScaleMode = AutoSizeMode.GrowAndShrink;
16  this.MaximizeBox = false;
17  timer1.Enabled = true;
18  timer1.Interval = 50;
19  pictureBox1.Image = img[0];
20  pictureBox1.Width = 70;
21  pictureBox1.Height = 55;
22  pictureBox1.SizeMode = PictureBoxSizeMode.Zoom;
23  pictureBox1.BackColor = Color.Transparent;
24 }
25
```

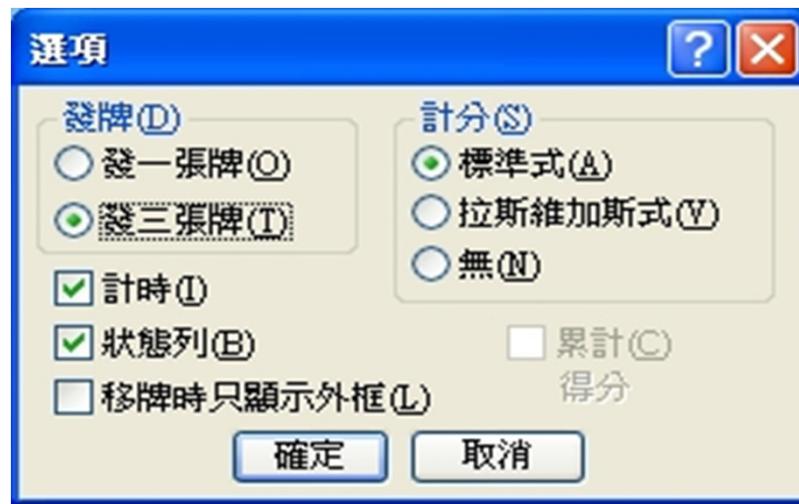


```
26 private void timer1_Tick(object sender, EventArgs e)
27 {
28     n++;
29     // 若n大於等於img陣列的最大維度，則n設為0，由brid0.png開始播放
30     if (n > img.GetUpperBound(0))
31     {
32         n = 0;
33     }
34     // 若pictureBox1超出表單的寬度時，使pictureBox1由表單左邊出現
35     if (pictureBox1.Left >= this.Width)
36     {
37         pictureBox1.Left = 0 - pictureBox1.Width;
38     }
39     pictureBox1.Left += 5;
40     pictureBox1.Image = img[n];
41 }
```



## 9.3 GroupBox / Panel 容器控制項

- **GroupBox 群組控制項和 Panel 面板控制項和表單**  
一樣都具備有容器（Container）的功能，上面都可放置其他控制項，以便對上面控制項分門別類，使得畫面排列整齊容易操作。
- 如下圖「接龍」遊戲的「選項」對話方塊中，使用「發牌」和「計分」兩個群組控制項。





### 9.3.1 GroupBox 群組控制項



- 群組控制項或稱為框架控制項可以和表單一樣，在該控制項內放置其他的控制項。
- 使用群組控制項好處是可將控制項分門別類，調整輸出入介面時，搬移群組控制項時裡面的控制項亦跟隨移動。
- 同性質選項按鈕或核取方塊可用 **GroupBox** 或 **Panel** 來存放。
- **GroupBox**內的控制項位置是以容器的左上角為基準，而不是以表單為基準。



## ■ GroupBox 常用屬性

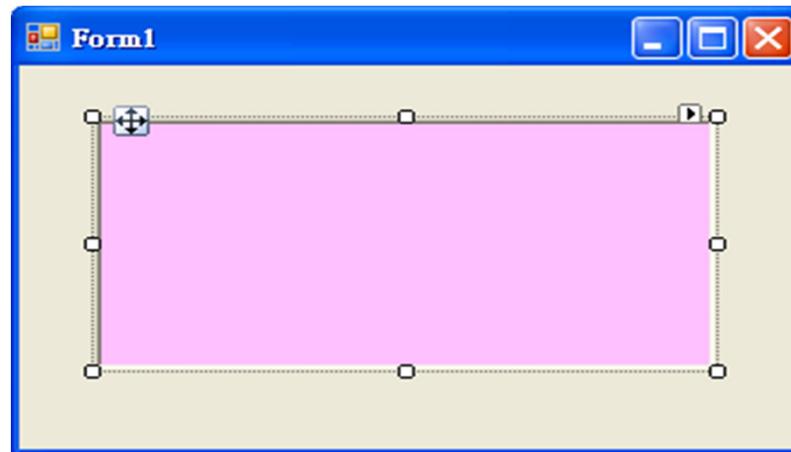
屬性	說明
<b>Text</b>	設定群組控制項左上角顯示的標題文字，該文字可以提示使用者。預設名稱為 <b>groupBox1</b> 。



### 9.3.2 Panel 面板控制項



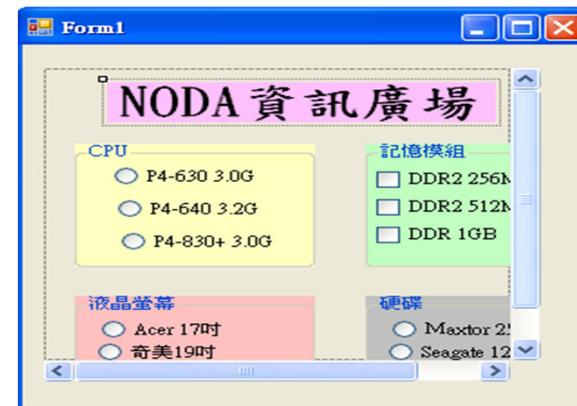
- 面板控制項也具備有容器的功能，裡面可放置其他的控制項。
- 和群組控制項外觀最大的不同是，左上角無法顯示文字。
- 但面板控制項允許有捲軸，輸出入介面佔用較小的表單空間。





## ■ Panel常用屬性

屬性	說明
BorderStyle	設定面板控制項的外框，其屬性值有 <b>None</b> （無邊框）-為預設值、 <b>FixedSingle</b> （單線框）、 <b>Fixed3D</b> （立體邊框）。面板控制項在執行階段，使用者是不能拖曳邊框來調整大小。
AutoScroll	設定面板控制項內的控制項大小若比面板大時是否顯示捲軸。 預設值為 <b>True</b> 表顯示捲軸；若設為 <b>False</b> 表不顯示捲軸。 右圖是Panel上面的控制項範圍超過面板大小且此屬性設為 <b>True</b> 時，會自動顯示捲軸的情形：





## 9.4 RadioButton / CheckBox 選擇控制項

- 當我們要填寫一張申請表時，表中有些資料是使用勾選，例如性別、年收入、購買項目...等。
- 如將這張申請表，改以電腦方式輸入，這些勾選的資料就可以使用**RadioButton**選項按鈕和**CheckBox**核取方塊控制項來設計。
- **RadioButton**選項按鈕控制項，具有排他性也就是只能選擇其中之一，所以單選性的選項可用它如性別、年收入。
- **CheckBox** 核取方塊控制項，每個選項都可獨立選擇互不影響，複選性問題可用它，
- 例如下圖購物選單。左邊在群組控制項上面置入選項按鈕控制項；在右邊群組控制項置入核取方塊控制項。





## 9.4.1 RadioButton選項按鈕控制項

- 選項按鈕控制項，具有排他的特性
- 一組多個 **Radio Button** 選項按鈕控制項中只能選擇其中之一。
- 若有兩組以上選項按鈕時，可用群組或面板控制項來加以區隔。
- **RadioButton** 選項按鈕控制項上面除可顯示文字外，也可顯示圖片。



## 一. RadioButton常用屬性

屬性	說明
Appearance	設定選項按鈕控制項的外觀，其值為： ① Normal :  (預設值) ② Button :
CheckAlign	設定圓形按鈕顯示的位置，當 Appearance 屬性值為 Button 時本屬性無效。其屬性值有九種可供設定，預設值為 MiddleLeft。本屬性再配合 TextAlign 屬性，就可以設計出所要的控制項外觀。
Checked	設定選項按鈕控制項是否被選取，若屬性值為 True 表示被選取 (按鈕外觀為 ); 屬性值為 False 表示未被選取 (按鈕外觀為  )。程式執行時，有一個選項按鈕被選取，同一群組其他選項按鈕的 Checked 屬性值都會改為 False。
AutoCheck	設定選項按鈕控制項被按下去時，是否會自動變更 Checked 屬性值。預設值為 True 表示會自動更改。
Text	設定選項按鈕控制項顯示的文字，如果希望使用者可以用快捷鍵來選取，可以在文字前加「&」。例如 Text 屬性值為「&1 開啓」，程式執行時選項按鈕控制項會顯示「1 開啓」，使用者只要按  鍵，就選取該選項按鈕。



## 二. RadioButton 常用事件

- 當在選項按鈕控制項按一下，會變更 **Checked** 屬性值，且同時依序觸動 **CheckedChanged** 和 **Click** 兩個事件。
- 若該按鈕已被選取，重複點選時因 **Checked** 屬性值不改變，只會觸發 **Click** 事件。
- 判斷勾選狀態的程式碼，都寫在 **CheckedChanged** 事件處理函式中。

### 1. **CheckedChanged** 事件

當選項按鈕控制項的 **Checked** 屬性值改變時，會觸發 **Checked Changed** 事件。

### 2. **Click** 事件

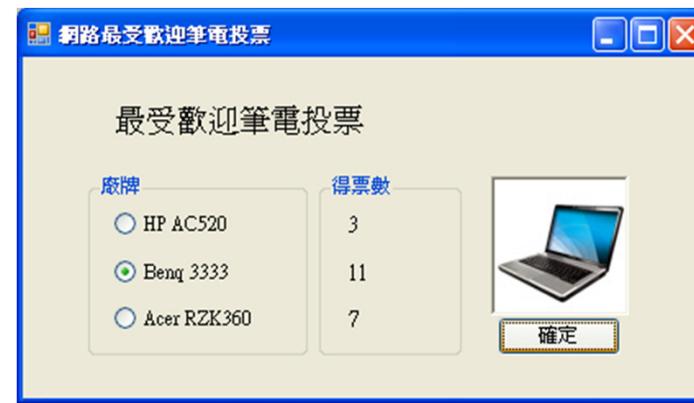
當選項按鈕控制項被滑鼠點選時，會觸發 **Click** 事件。

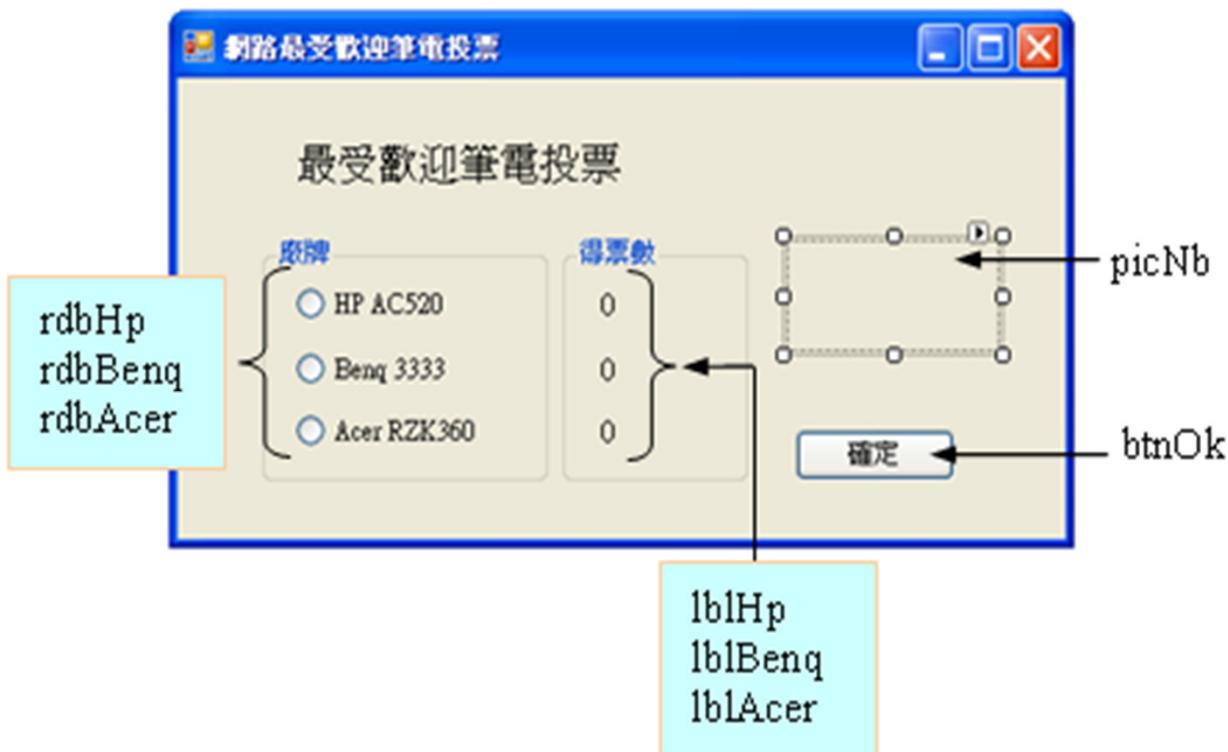


## 範例演練

使用 **GroupBox**、**RadioButton**、**Label**、**Button**、**PictureBox**  
... 等控制項設計最受歡迎筆電投票程式。

下圖可在廠牌 **GroupBox** 內選取筆電型號的選項按鈕  
後 會顯示該台筆電的圖示，接著按 **確定** 鈕則  
**GroupBox** 得票數內對應的票數會進行 加 1。







## 完整程式碼

```
// FileName : RadioButtonDeom.sln

01 private void Form1_Load(object sender, EventArgs e)
02 {
03     rdbHp.Checked = true;
04     picNb.Image = new Bitmap("hp.jpg");
05     picNbSizeMode = PictureBoxSizeMode.AutoSize; //使picNb與圖的大小相同
06     picNb.BorderStyle = BorderStyle.Fixed3D;
07 }
09 private void rdbHp_CheckedChanged(object sender, EventArgs e)
10 {
11     picNb.Image = new Bitmap("hp.jpg");
12 }
14 private void rdbBenq_CheckedChanged(object sender, EventArgs e)
15 {
16     picNb.Image = new Bitmap("benq.jpg");
17 }
```



```
19 private void rdbAcer_CheckedChanged(object sender, EventArgs e)
20 {
21     picNb.Image = new Bitmap("acer.jpg");
22 }
24 private void btnOk_Click(object sender, EventArgs e)
25 {
26     if (rdbHp.Checked)
27     {
28         lblHp.Text = (Convert.ToInt32(lblHp.Text) + 1).ToString();
29     }
30     else if (rdbBenq.Checked)
31     {
32         lblBenq.Text = (Convert.ToInt32(lblBenq.Text) + 1).ToString();
33     }
34     else if (rdbAcer.Checked)
35     {
36         lblAcer.Text = (Convert.ToInt32(lblAcer.Text) + 1).ToString();
37     }
38 }
```



## 9.4.2 CheckBox 核取方塊控制項



- 核取方塊控制項，每個選項都可任意選取彼此間互不影響。
- 複選性選項可用它來設計。
- CheckBox 核取方塊控制項除可以顯示文字外，上面也可顯示圖片。



## 一. CheckBox常用屬性

屬性	說明													
ThreeState	設定核取方塊控制項是否允許三種核取狀態。預設值為 False 表示有勾選和不勾選兩種狀態。若設為 True 表示有勾選、不勾選和未確定三種狀態。													
CheckState	當 ThreeState 屬性值為 True 時，設定核取方塊控制項是有三種選取狀態。依 Appearance 屬性值而異：													
	<table border="1"><thead><tr><th>ThreeState</th><th>Appearance.Normal</th><th>Appearance.Button</th></tr></thead><tbody><tr><td>Checked</td><td>勾選 <input checked="" type="checkbox"/> CheckBox1</td><td>下凹 <input type="checkbox"/> CheckBox1</td></tr><tr><td>Unchecked</td><td>未勾選</td><td>突起</td></tr><tr><td>Indeterminate</td><td>不確定</td><td>平面</td></tr></tbody></table>		ThreeState	Appearance.Normal	Appearance.Button	Checked	勾選 <input checked="" type="checkbox"/> CheckBox1	下凹 <input type="checkbox"/> CheckBox1	Unchecked	未勾選	突起	Indeterminate	不確定	平面
ThreeState	Appearance.Normal	Appearance.Button												
Checked	勾選 <input checked="" type="checkbox"/> CheckBox1	下凹 <input type="checkbox"/> CheckBox1												
Unchecked	未勾選	突起												
Indeterminate	不確定	平面												
Checked	當 ThreeState 屬性值為 False 時，若核取方塊控制項被選取，其屬性值為 True；預設為 False 未被勾選。													



## 二. CheckBox常用事件

- 當 **ThreeState = False** 時，使用者按核取方塊控制項時會依序觸動 **CheckedChanged**、**CheckStateChanged** 和 **Click** 三個事件。
- 判斷核取方塊勾選狀態的程式碼，寫在其中一個事件中皆可。
- 當 **ThreeState =True** 時，若使用者點選勾選狀態為「未確定」時，不會觸動 **CheckedChanged** 事件。



## 1. CheckedChanged 事件

當核取方塊控制項的**Checked**屬性值改變時，  
會觸發此事件。

## 2. CheckStateChanged 事件

當核取方塊控制項的**CheckState**屬性值改變時  
會觸發此事件。

## 3. Click 事件

當核取方塊控制項被滑鼠點選時，就會觸發此事件。



## 範例演練

設計相機易購網程式。在「機型」GroupBox 選項只能單選，在「配件」GroupBox 選項可多選，只要有選取會隨時更新「購買總金額：」。  
各機型和配件單價如下圖表單所示。





上機

## ■ 設計輸出入介面





## 完整程式碼

```
// FileName : CheckBoxDemo.sln
01 int priceDsc, priceM1G, priceReader, priceChange;
02 private void rdbP10_CheckedChanged(object sender, EventArgs e)
03 {
04     if (rdbP10.Checked) //選取rdbP10
05     {
06         priceDsc = 15000;
07     }
08     if (rdbP20.Checked) //選取rdbP20
09     {
10         priceDsc = 10000;
11     }
```



```
12 if (rdbD70.Checked)
13 {
14     priceDsc = 25000;
15 }
16 if (chkM1G.Checked)
17 {
18     priceM1G = 2000;
19 }
20 else
21 {
22     priceM1G = 0;
23 }
24 if (chkReader.Checked)
25 {
26     priceReader = 500;
27 }
```



```
28     else
29     {
30         priceReader = 0;
31     }
32     if (chkCharge.Checked)
33     {
34         priceChange = 1500;
35     }
36     else
37     {
38         priceChange = 0;
39     }
41     lblTotal.Text = "購買總金額：" + (priceDsc + priceM1G +
42     priceReader + priceChange).ToString();
```



## 設定共用事件

當在三個選項按鈕控制項或三個核取方塊控制項上按一下都會觸動該控制項的 **CheckedChanged** 事件。由於這六個控制項都做同樣的事情，可共用同一個 **rdbP10\_CheckedChanged** 事件處理函式。

