

# CHAPTER 8

## 方法 (Method)

- ◇ 認識方法
- ◇ 學習如何使用亂數物件
- ◇ 學習控制項陣列的應用
- ◇ 學習如何定義自定方法
- ◇ 學習如何呼叫自定方法
- ◇ 學習傳值呼叫與參考呼叫方式
- ◇ 學習撰寫自定共用事件
- ◇ 學習方法間陣列的傳遞
- ◇ 學習方法多載

## 8.1 方法

方法(Method)是由多行敘述(Statements)所組成，是具有特定功能的程式碼區塊。程式呼叫(Calling)方法時，呼叫方法的敘述可以將引數傳給方法，提供給該方法內的程式區塊使用。Visual C# 程式主要是由方法所組成，而方法都置於類別中。Windows Forms 模式下所編寫的程式，方法預設置於 Form1 類別裡面。

方法是類別(Class)的成員之一，Visual C# 中的「方法」在 Visual Basic 程式語言稱為「函數」或「程序」；在 C 和 C++程式語言則稱為「函式」。所以，方法就是函數、函式或程序，只是在不同的程式語言而有不同稱呼。由於 Visual C# 是屬於物件導向程式語言，為符合物件導向封裝的特性，所以本書以「方法」來代替函式、函數或程序。方法具有下面特點：

1. 方法是類別的成員之一，具有特定功能的程式碼區塊，可重複呼叫使用。
2. 方法擁有自己的名稱，須使用合法的 Visual C# 識別字來命名。但不能和變數、常數或定義在類別內的屬性名稱相同。
3. 方法內所宣告的變數是屬於方法範圍的區域變數，其有效範圍侷限在該方法內，所以在不同的方法內允許使用相同的區域變數名稱。
4. 方法的程式碼功能明確，可以提高程式的可讀性，而且容易除錯和維護。

在 Visual C# 程式語言中，方法依其來源可分成三大類：

1. **系統提供的方法**：系統提供的方法可直接使用，譬如 ToInt32()是 Convert 類別的方法。本書附錄 B 有介紹亂數、數學...等類別的方法供參考。
2. **事件(Event)**：Visual C# 中每個控制項物件都有其事件，預設事件中沒有敘述，程式碼可自行視需要而寫入。應用程式常會使用多個物件，當在某個物件上做動作，就會觸發該物件針對這個動作所指定的事件，而執行事件處理函式內的程式碼。「事件」並不會自動執行，必須在程式執行時透過使用者或由其他程式來觸動。
3. **使用者自定的方法**：使用者依程式需求自己定義的方法稱為「使用者自定方法」，簡稱「自定方法」(或稱自訂方法)。

## 8.2 亂數物件

亂數在生活中常被應用，例如隨機抽題、數學統計、實驗數據和樂透彩的電腦選號...等。Visual C# 可以使用 Random 類別來建立亂數物件，然後就可以使用亂數物件所提供的方法來產生指定範圍的亂數。

### 8.2.1 如何使用 Random 亂數物件

由於 Random 類別是屬於非靜態類別，所以必須使用 new 敘述來建立一個屬於 Random 類別的亂數物件實體，其語法如下：

語法

```
Random 物件名稱 = new Random();
```

**例** 建立名稱為 rnd 屬於 Random 類別的亂數物件實體。

```
Random rnd = new Random();
```

亂數物件建立後，可用 Next()與 NextDouble()方法來產生指定範圍的亂數值。

#### 一、Next()方法

Next 方法可傳回介於指定範圍的整數亂數值。有下列三種語法：

語法

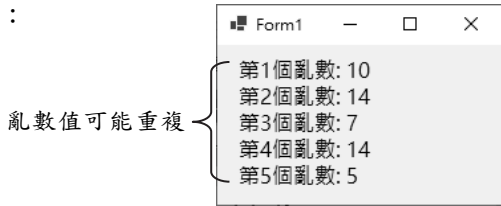
1. 亂數物件.Next(); // 傳回 0 ~ 2,147,483,647 的整數亂數。
2. 亂數物件.Next(max); // 傳回 0 ~ (max-1)的整數亂數值。
3. 亂數物件.Next(min, max); // 傳回 min(含)到(max-1)的整數亂數值。

**例** 產生 3~14 間的五個亂數，並在 LblMsg 標籤控制項顯示。(Rnd1 專案)

```
Random rnd = new Random();  
LblMsg.Text = "";  
for (int i = 1; i <= 5; i++)  
{
```

```
LblMsg.Text += $"第{i}個亂數: {rnd.Next(3, 15)}\n";
}
```

結果：

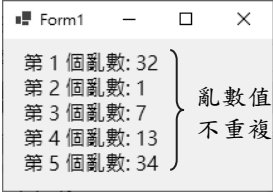


產生亂數的步驟：

- ① 先使用 `new Random()` 建立一個名為 `rnd` 的亂數物件。
- ② 在 `for` 迴圈內，使用亂數物件的 `Next()` 方法產生介於兩數間的多個整數亂數，產生的亂數可以逐一置入陣列中或直接顯示出來。

**例** 產生 1~49 間的五個不重複亂數，並在 `LblMsg` 標籤控制項顯示。(Rnd2 專案)

```
int[] num = new int[5];           // 陣列 num 記錄 5 個整數亂數
Random rnd = new Random();       // 建立 rnd 亂數物件
LblMsg.Text = "";               // 清空 LblMsg
for (int i = 0; i <= 4; i++)
{
    int r = rnd.Next(49) + 1;     // 產生 1~49 亂數
    bool same = false;           // 檢查亂數是否重複，預設為不重複
    foreach (int n in num)        // 逐一檢查 num 陣列元素
    {
        if (r == n) { same = true; break; } //若相同就設 same=true 並離開迴圈
        if (n == 0) { same = false; break; } //若是 0 表沒亂數設 same=false 離開迴圈
    }
    if (same == false)           // 若 same=false
        num[i] = r;              // 將亂數存入陣列中
    else
        i--;                      // i 減 1 重新產生亂數
}
for (int i = 0; i <= 4; i++)    // 逐一顯示亂數
{
    LblMsg.Text += $"第 {i + 1} 個亂數: {num[i]}\n";
}
```



## 二、NextDouble() 方法

亂數物件的 NextDouble 方法，可以傳回 0.0 到 1.0 之間的倍精確度亂數。

語法

```
亂數物件.NextDouble();
```

**例** 建立 rnd 亂數物件後，使用 NextDouble 方法傳回 0.0 ~ 1.0 間倍精確度亂數，最後再將亂數指定給浮點數變數 d。

```
Random rnd = new Random();  
double d = rnd.NextDouble(); // 產生 0.0~1.0 之間的亂數值
```



**實作** FileName : Poker1.sln

按  鈕啟動計時器，計時器會以每 0.05 秒用亂數切換 poker1.jpg ~ poker13.jpg 撲克牌圖。當按  鈕後，會顯示點數及對應撲克牌圖。

### ► 輸出要求

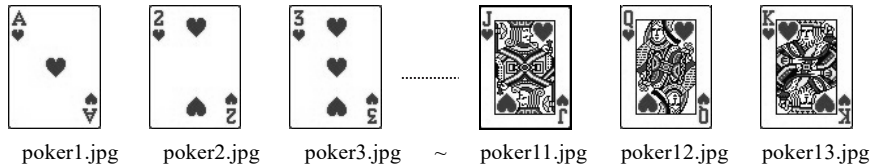


### ► 解題技巧

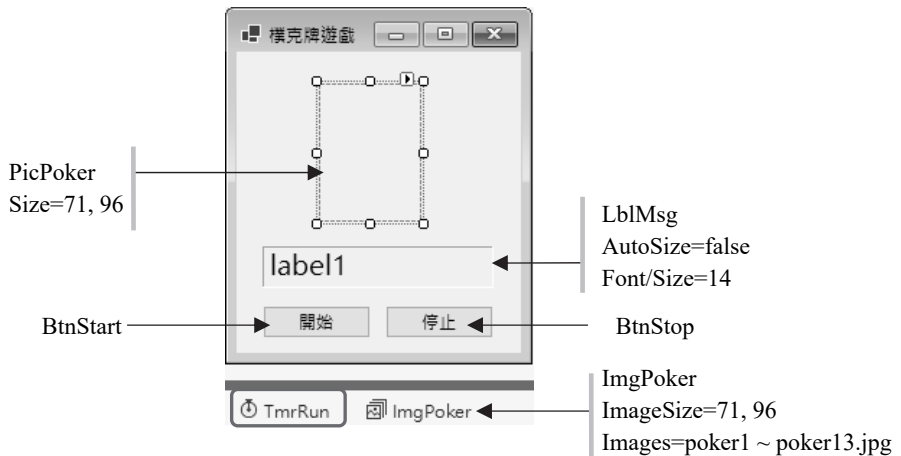
#### Step 1 建立輸出入介面

1. 新增專案並以「Poker1」為新專案名稱。
2. 由輸出要求必須在表單建立下列控制項：
  - ① 圖片方塊控制項 PicPoker 用來顯示撲克牌圖，並將 Size 屬性值設為 71, 96，因為撲克牌圖檔大小為 71 x 96。

② 影像列示控制項 `ImgPoker` 用來存放 `poker1 ~ poker13.jpg` 撲克牌圖，將 `ImageSize` 屬性值設為 71, 96，撲克牌圖檔在範例資料夾中請依序加入。



- ③ 一個標籤控制項 `LblMsg` 用來顯示亂數產生的點數。
- ④ 兩個按鈕控制項 `BtnStart`、`BtnStop`，分別為 、 鈕。
- ⑤ 計時器控制項 `TmrRun` 用來每 0.05 秒產生一個亂數。



### Step 2 分析問題

1. 宣告 `point` 整數欄位變數用來存放亂數產生的撲克牌點數，此變數在計時器的 `TmrRun_Tick` 及  鈕的 `BtnStop_Click` 事件處理函式都會使用。
2. 在 `Form1_Load` 事件處理函式定 `TmrRun` 的 `Interval` 屬性值為 50，當 `TmrRun` 啟動後每 0.05 秒會執行 `TmrRun_Tick` 事件處理函式切換撲克牌圖。
3. 按  鈕執行的 `BtnStart_Click` 事件處理函式，使用 `TmrRun` 的 `Start` 方法啟動計時器。
4. 按  鈕執行的 `BtnStop_Click` 事件處理函式時，將 `TmrRun` 的 `Stop` 方法關閉計時器，並將所得到的撲克牌點數(`point+1`)顯示在 `LblMsg` 標籤上。

5. 在計時器的 Tick 事件處理函式內，使用 Next(13)方法產生 0 ~ 12 的整數亂數 point，point 值對應到 ImgPoker 控制項的 Image 屬性的註標值，就可以取得 poker1 ~ poker13.jpg 的撲克牌圖。

### Step 3 撰寫程式碼

#### FileName: Poker1.sln

```
01 namespace Poker1
02 {
03     public partial class Form1 : Form
04     {
05         public Form1()
06         {
07             InitializeComponent();
08         }
09         int point; // 宣告point整數欄位變數來存放得到的點數
10         // 表單載入時執行
11         private void Form1_Load(object sender, EventArgs e)
12         {
13             TmrRun.Interval = 50; //設每50毫秒(即0.05秒)執行一次TmrRun_Tick事件
14             LblMsg.Text = "";
15         }
16         // 按 <開始> 鈕執行
17         private void BtnStart_Click(object sender, EventArgs e)
18         {
19             TmrRun.Start(); // 啟動TmrRun計時器
20         }
21         // 按 <停止> 鈕執行
22         private void BtnStop_Click(object sender, EventArgs e)
23         {
24             TmrRun.Stop(); // 停止TmrRun計時器
25             LblMsg.Text = $"得到 {point+1} 點 !!";
26         }
27         // 每50毫秒(即0.05秒)執行一次TmrRun_Tick事件
28         private void TmrRun_Tick(object sender, EventArgs e)
29         {
```

```

30         Random rnd = new Random();
31         point = rnd.Next(13); // 產生0~12變數
32         PicPoker.Image = ImgPoker.Images[point];
33     }
34 }
35 }

```

### ► 馬上練習

將上面實作修改為比大小遊戲，按  鈕會顯示電腦的點數撲克牌(亂數產生)，玩家撲克牌圖會用亂數快速切換。當按  鈕後會顯示比賽結果。



## 8.2.2 控制項陣列的應用

在上例撲克牌程式只顯示一張撲克牌，若同時出現多張撲克牌時，則都需要寫一段除名稱不同之外，其他都相同的程式碼。如果能像陣列一樣，將圖片控制項物件指定給陣列元素，此時就可以使用迴圈來處理，不但能縮短程式碼，而且可讀性和維護性都提高。這種將控制項存放在陣列中，該陣列就稱為「控制項陣列」。

例如有 Pic1 ~ Pic3 三個圖片方塊控制項要分別載入 poker1 ~ poker3.jpg 圖檔。此時可使用控制項陣列，透過 for 迴圈逐一載入圖檔到陣列中。步驟如下：

**Step ①** 先建立名稱為 Pic1 ~ Pic3 共三個圖片方塊控制項。

**Step ②** 建立名稱為 arrPic 的圖片方塊控制項陣列，有 arrPic[0] ~ arrPic[2] 三個陣列元素，用來存放圖片方塊控制項 PictureBox 型別的物件。如果要存放其他控制項例如標籤，宣告時只要指定控制項陣列型別為 Label 即可。

```
PictureBox[] arrPic = new PictureBox[3];
```



Step 3 將 Pic1 ~ Pic3 圖片方塊控制項，指定給 arrPic[0] ~ arrPic[2] 陣列元素。

```
arrPic[0] = Pic1 ; arrPic[1] = Pic2 ; arrPic[2] = Pic3;
```

Step 4 使用 for 迴圈改變陣列註標值，將註標值加 1 當作檔名，即可將 poker1 ~ poker3.jpg 圖檔分別指定給 arrPic[0] ~ arrPic[2] 陣列元素的 Image 屬性。

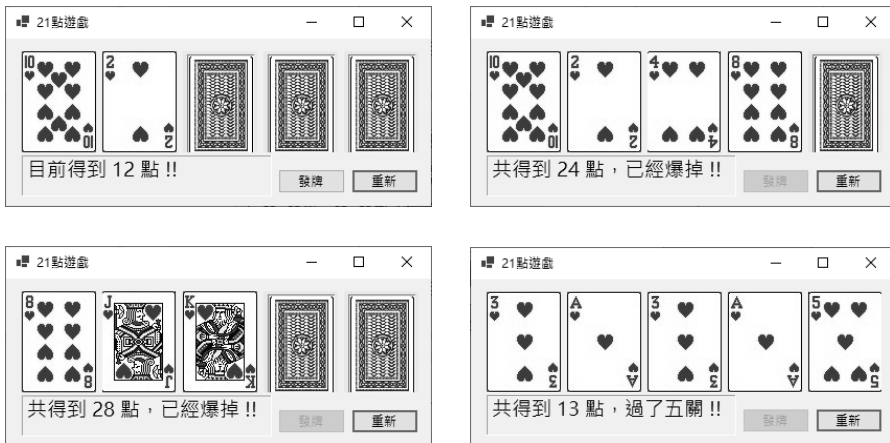
```
for(int i=0 ; i<arrPic.Length; i++)
    arrPic[i].Image = Image.FromFile($"poker{i+1}.jpg");
```



### 實作 FileName : Poker2.sln

設計 21 點遊戲，共有五張撲克牌，1~10 點的牌以牌面點數計算，J、Q、K 以 10 點計算。開始時隨機出兩張牌，並顯示總點數。每按一次  鈕，會由第三張牌起隨機出牌並顯示總點數，若超過 21 點就不能按  鈕。到第五張若沒有超過 21 點，就顯示「過五關」，按  鈕可重玩。

### ► 輸出要求

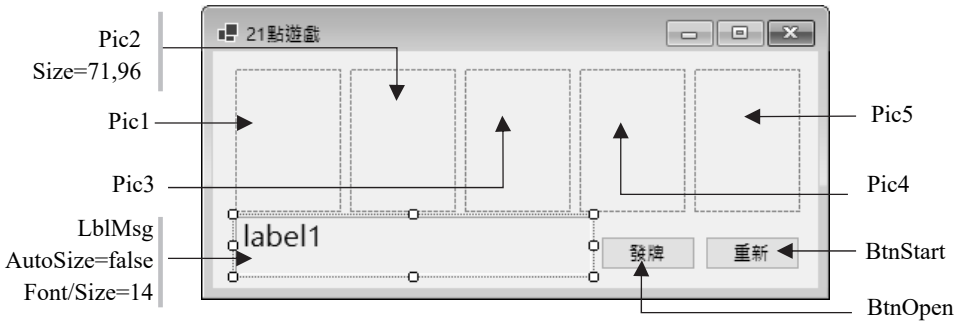


### ► 解題技巧

Step 1 建立輸出入介面

1. 新增專案並以「Poker2」為新專案名稱。

2. 由輸出要求可知，本實作必須在表單上建立下列各控制項：
  - ① 在表單的標題欄顯示「21 點遊戲」標題。
  - ② 表單上建立五個圖片方塊控制項 Pic1 ~ Pic5 用來顯示撲克牌，並將 Size 屬性設為 71, 96，讓圖片方塊控制項和圖片的大小相等。
  - ③ 建立標籤控制項 LblMsg 用來顯示總點數和相關訊息。
  - ④ 建立兩個名稱為 BtnStart、BtnOpen 的按鈕控制項。



3. 將範例所附 poker1.jpg ~ poker13.jpg 和 pokerbk.jpg(撲克牌背面)圖檔，複製到目前製作專案的 [bin\Debug\net6.0-windows] 子資料夾下。

Step 2 分析問題

1. 本實作需在五個圖片方塊控制項顯示對應的撲克牌圖檔，建立控制項陣列 arrPic(arrPic[0] ~ arrPic[4])存放圖片方塊控制項，程式碼較簡潔且易維護。
2. 建立 rnd 為亂數物件來產生亂數；total 整數變數用來存放撲克牌的總點數；point 整數變數存放亂數產生的點數；times 整數變數存放發牌的次數。上述變數在多個函式皆有參用，因此將變數宣告於事件處理函式之外。
3. 在 Form1\_Load 事件處理函式中，將 Pic1 ~ Pic5 圖片方塊控制項指定成為 arrPic[0] ~ arrPic[4]圖片方塊控制項的陣列元素。接著執行 BtnStart\_Click 事件，來設定遊戲的初始狀態。
4. 在 BtnStart\_Click 事件處理函式中，完成下列動作：
  - ① 使用for迴圈設 Pic3 ~ Pic5的 Image 屬性值為 pokerbk.jpg 來顯示背景圖。

- ② 使用 for 迴圈產生 1~13 亂數，依 point 值設定 Pic1 ~ Pic2 的 Image 屬性值為對應的圖檔，並設 times 為 2 表發兩張牌。使用三元運算子，當 point 值大於 10 就重設為 10(因 J、Q、K 為十點)，然後加入 total 總點數。

```
total += (point < 10 ? point : 10);
```

- ③ 在標籤控制項 LblMsg 上顯示總點數。
- ④ 設定 BtnOpen 的 Enabled 屬性值為 true，讓使用者可以點按。
5. 在 BtnOpen 的 Click 事件處理函式中，完成下列動作：
- ① times 加 1 表新發一張牌，times-1 就是要處理控制項陣列的註標值。
- ② 產生 1 ~ 13 亂數，在指定控制項陣列用 FromFile 方法載入對應的圖檔。因為本例不用快速顯示圖片，所以使用 FromFile 方法讀檔即可。
- ③ 計算目前發牌點數，並加到 total 總點數中，
- ④ 檢查 total 是否超過 21，若超過就顯示「爆掉」訊息。如果 total 未超過 21 且發牌次數為 5，就顯示「過五關」訊息。如果 total 未超過 21 且發牌次數不為 5，就顯示總點數。
- ⑤ 當爆掉或過五關時，就設 BtnOpen 的 Enabled 屬性為 false 使不能點按。
6. 30~32 和 42~44 行程式碼相同可以抽出成為自定方法，作法在下一節介紹。

### Step 3 撰寫程式碼

#### FileName: Poker2.sln

```
01 namespace Poker2
02 {
03     public partial class Form1 : Form
04     {
05         public Form1()
06         {
07             InitializeComponent();
08         }
09         PictureBox[] arrPic = new PictureBox[5]; // arrPic為圖片方塊陣列大小為5
10         Random rnd = new Random();//rnd為亂數物件
11         int total = 0; // total整數變數用來存放撲克牌的總點數
```

## 15.3 電腦視覺應用程式開發

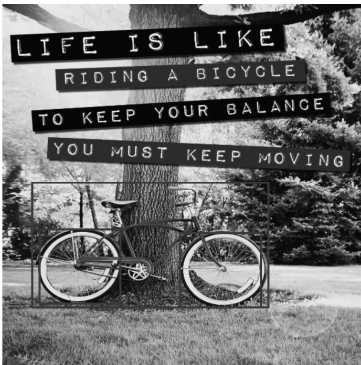
### 15.3.1 電腦視覺簡介

Azure 認知服務中的辨識服務提供電腦視覺(Computer Vision)服務，此服務不需要機器學習即可在應用程式中使用具電腦視覺分析與辨識影像的功能。例如：可解理影像中的物件數、文字、主題、品牌、影像格式、影像大小、成人資訊、人物性別或年齡、人物名稱與地標…等內容。您可連結到「<https://azure.microsoft.com/zh-tw/services/cognitive-services/computer-vision/>」網頁，此網頁展示電腦視覺功能與說明，如下兩圖使用電腦視覺服務分析影像的內容以及將分析結果以 JSON 呈現：



```

功能名稱 值
物件數 [{"rectangle": {"x": 6, "y": 390, "w": 48, "h": 40}, "object": "footwear", "confidence": 0.513}, {"rectangle": {"x": 104, "y": 104, "w": 127, "h": 323}, "object": "person", "confidence": 0.763}, {"rectangle": {"x": 174, "y": 236, "w": 113, "h": 74}, "object": "Laptop", "parent": {"object": "computer", "confidence": 0.56}, "confidence": 0.553}, {"rectangle": {"x": 351, "y": 331, "w": 154, "h": 99}, "object": "seating", "confidence": 0.525}, {"rectangle": {"x": 0, "y": 101, "w": 174, "h": 329}, "object": "person", "confidence": 0.855}, {"rectangle": {"x": 223, "y": 99, "w": 199, "h": 322}, "object": "person", "confidence": 0.725}, {"rectangle": {"x": 154, "y": 191, "w": 387, "h": 218}, "object": "seating", "confidence": 0.679}, {"rectangle": {"x": 111, "y": 275, "w": 264, "h": 151}, "object": "table", "confidence": 0.601}]
標籤 [{"name": "person", "confidence": 0.9945455}, {"name": "laptop",
    
```



```

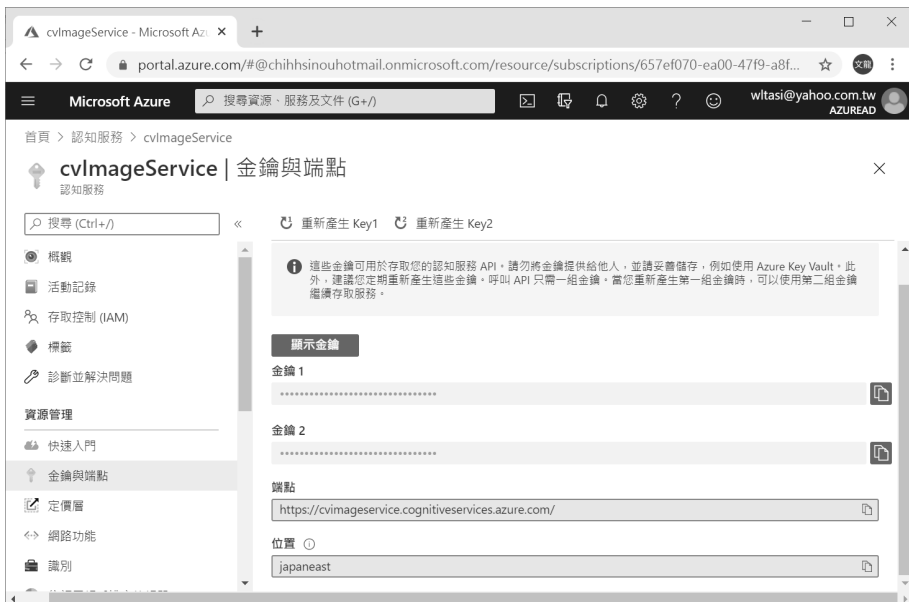
功能名稱 值
物件數 [{"rectangle": {"x": 112, "y": 711, "w": 269, "h": 243}, "object": "Bicycle wheel", "parent": {"object": "Wheel", "confidence": 0.775}, "confidence": 0.574}, {"rectangle": {"x": 566, "y": 723, "w": 268, "h": 243}, "object": "Wheel", "confidence": 0.585}, {"rectangle": {"x": 96, "y": 568, "w": 759, "h": 409}, "object": "bicycle", "parent": {"object": "cycle", "parent": {"object": "Land vehicle", "parent": {"object": "Vehicle", "confidence": 0.928}, "confidence": 0.927}, "confidence": 0.923}, "confidence": 0.91}]
標籤 [{"name": "text", "confidence": 0.9999137}, {"name": "grass", "confidence": 0.999893069}, {"name": "outdoor", "confidence": 0.9880197}, {"name": "bicycle", "confidence": 0.9697467}, {"name": "bicycle wheel", "confidence": 0.8976265}, {"name": "sign", "confidence": 0.842304945}, {"name": "bike", "confidence": 0.7690854}, {"name": "wheel", "confidence": 0.7583429}, {"name": "land vehicle",
    
```

## 15.3.2 電腦視覺服務的使用

下面步驟介紹如何在應用程式中內嵌雲端視覺功能。

### Step 1 申請電腦視覺服務

前往 Azure 雲端申請電腦視覺服務 API 的金鑰 (Key)與端點(URL)，端點即是要分析影像的雲端服務。



### Step 2 安裝電腦視覺 SDK

在專案安裝 Microsoft.Azure.CognitiveServices.Vision.ComputerVision 套件。

### Step 3 建立 ComputerVisionClient 電腦視覺物件，將分析的影像傳送到電腦視覺服務端點進行分析，程式寫法如下：

```
//引用如下命名空間才能使用 ComputerVisionClient
using Microsoft.Azure.CognitiveServices.Vision.ComputerVision;
using Microsoft.Azure.CognitiveServices.Vision.ComputerVision.Models;
.....
```

```
//建立 ComputerVisionClient 電腦視覺物件，同時指定電腦視覺服務的金鑰 Key
ComputerVisionClient 電腦視覺物件 = new ComputerVisionClient(
    new ApiKeyServiceClientCredentials("電腦視覺服務的金鑰(key)"),
    new System.Net.Http.DelegatingHandler[] { });

//指定電腦視覺服務端點 Url
電腦視覺物件.Endpoint = "電腦視覺服務的端點(Url)";

//執行 DescribeImageInStreamAsync()方法
//將影像分析結果傳給 ImageDescription 影像描述物件 res
ImageDescription res =
    await 電腦視覺物件.DescribeImageInStreamAsync(影像檔案串流);
```

#### Step 4 取得影像標籤

影像標籤即是影像中可能包含的項目，例如：在影像有車站月台、人、室內、室外、男人、女人 ... 等資訊。寫法如下：

```
// ImageDescription 影像描述物件的 Tags 集合內含分析後的影像標籤
string tagStr = "";
for(int i=0; i<res.Tags.Count(); i++) //使用 for 迴圈將所有的影像標籤放入 tagStr
{
    tagStr += $"{ res.Tags[i]}, ";
}
```

#### Step 5 取得影像資訊

使用 ImageDescription 影像描述物件 res 取得影像描述資訊，例如使用 Captions 集合 Text 屬性取得影像描述說明，使用 Captions 集合 Confidence 屬性取得影像描述信度。寫法如下：

```
String s1 = $"描述： {res.Captions[0].Text}";
String s2 = $"信度： {res.Captions[0].Confidence}";
```

進行電腦視覺分析的影像必須符合下列條件，否則會發生例外：

- 影像檔案大小須小於 4 MB
- 影像像素須大於 50 x 50
- 影像格式必須以 BMP、JPEG、PNG 或 GIF
- 呼叫電腦視覺 API，上傳的影像的大小必須介於 50 x 50 與 10000 x 10000 像素之間。

### 15.3.3 電腦視覺應用程式實作



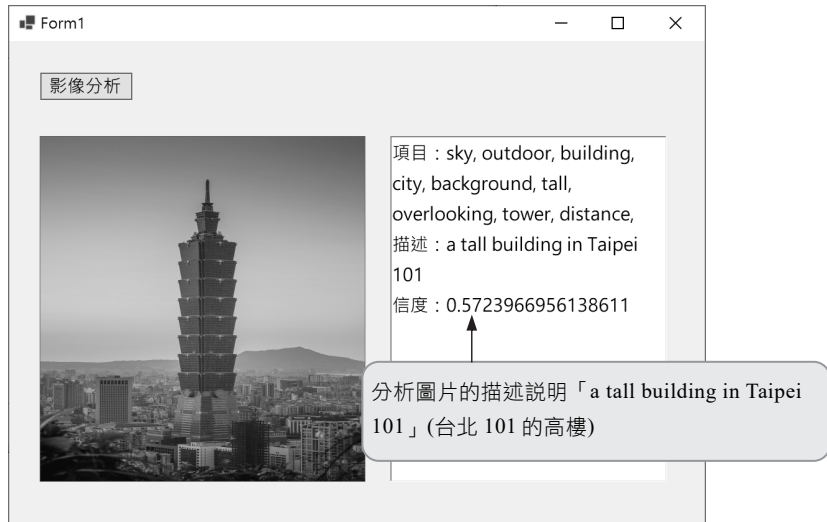
**實作** FileName : cv01.sln

練習製作可進行分析影像的程式。程式執行時按下 **影像分析** 鈕開啟開檔對話方塊並指定所要分析影像圖檔，接著會將影像中的項目、描述以及描述信度顯示於多行文字方塊。

#### ► 輸出要求




▲ 分析劉得華的影像



▲ 分析台北 101 影像

## ► 解題技巧

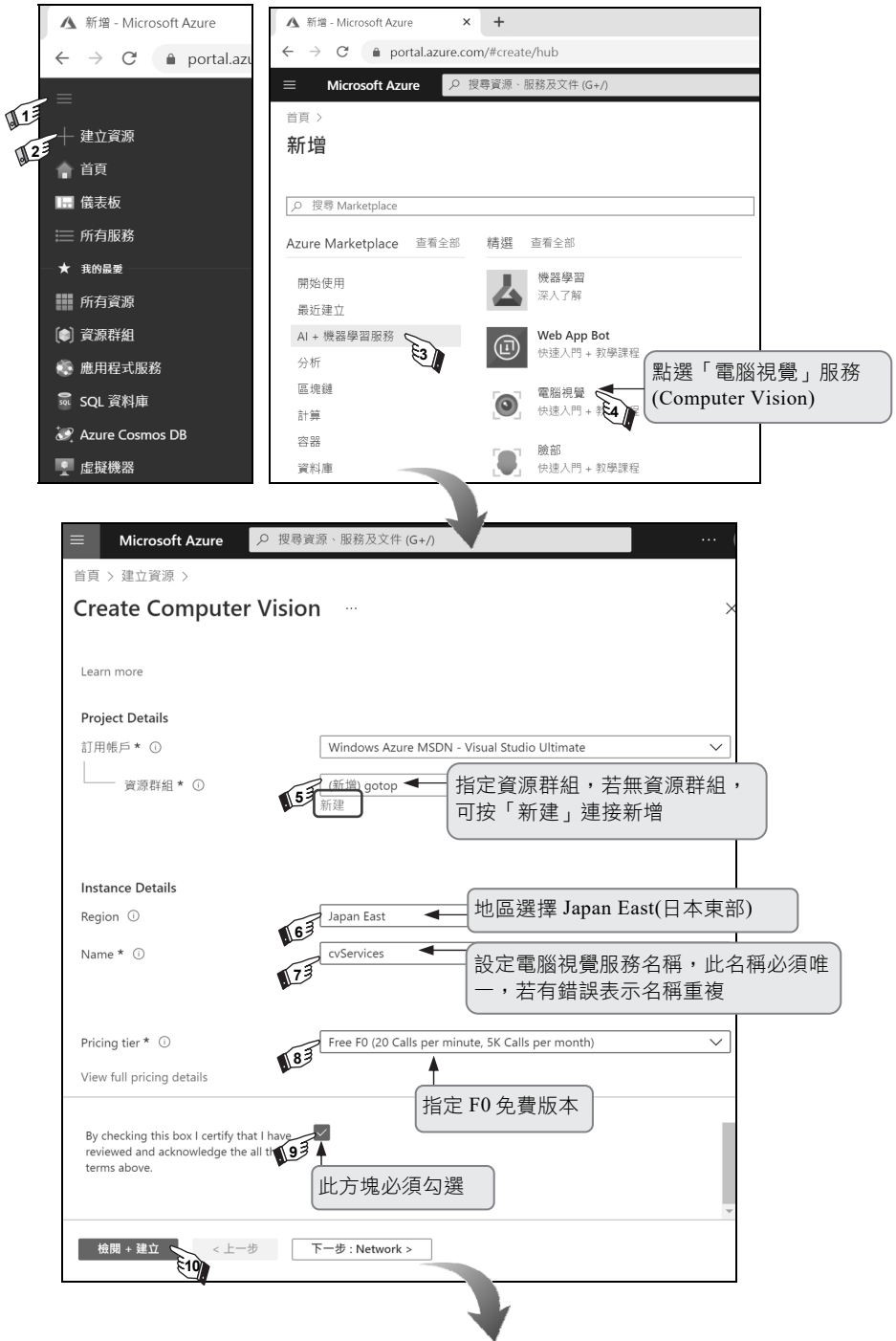
Step 1 申請電腦視覺服務的金鑰(Key)和端點(Url)：

1. 前往「<https://azure.microsoft.com/zh-tw/features/azure-portal/>」 Azure 入口網站，接著點選右上方「登入」連接進行登入網站。
2. 登入成功之後，請點選  鈕並由選單中點按「Azure 入口網站」連接即可進入 Azure 雲端平台網站。





3. 依下圖操作取得電腦視覺服務的金鑰(Key)和端點(Url)。



### 建立 電腦視覺

✓ 驗證成功

基本 標籤 檢閱 + 建立

#### 條款

按一下 [建立]，即表示我 (a) 同意上述 Marketplace 供應項目的相關法律條款及隱私權聲明; (b) 授權 Microsoft 向我目前的付款方式收取供應項目的相關費用，帳單週期與我的 Azure 訂用帳戶相同; 並 (c) 同意 Microsoft 將我的連絡資料、使用方式及交易資訊提供給供應項目的提供者，以用於支援、帳單及其他交易活動。Microsoft 不提供第三方供應項目的權利。如需其他詳細資料，請參閱 Azure Marketplace 條款。

#### 基本

訂用帳戶	Windows Azure MSDN - Visual Studio Ultimate
資源群組	gotoprs
區域	日本東部
名稱	cvImageService
定價層	免費 F0 (每分鐘 20 個呼叫，每月 5K 個呼叫)

建立

< 上一步

下一頁

下載自動化的範本



Microsoft.CognitiveServicesComputerVision-2

您的部署已完成

部署名稱: Microsoft.CognitiveServices...  
訂用帳戶: Windows Azure MSDN - Vis...  
資源群組: gotoprs

前往資源

通知

✓ 已成功部署

目標為資源群組 'gotoprs' 的部署 'Microsoft.CognitiveServicesComputerVision-20200828172025' 成功。

前往資源群組

尚餘點數 \$4,555.00 點

訂用帳戶 'Windows Azure MSDN - Visual Studio Ultimate' 尚餘價值 \$4,555.00 元的點數。

服務建立完成會出現 **前往資源** 鈕，按下此鈕會直接跳到該服務設定畫面。

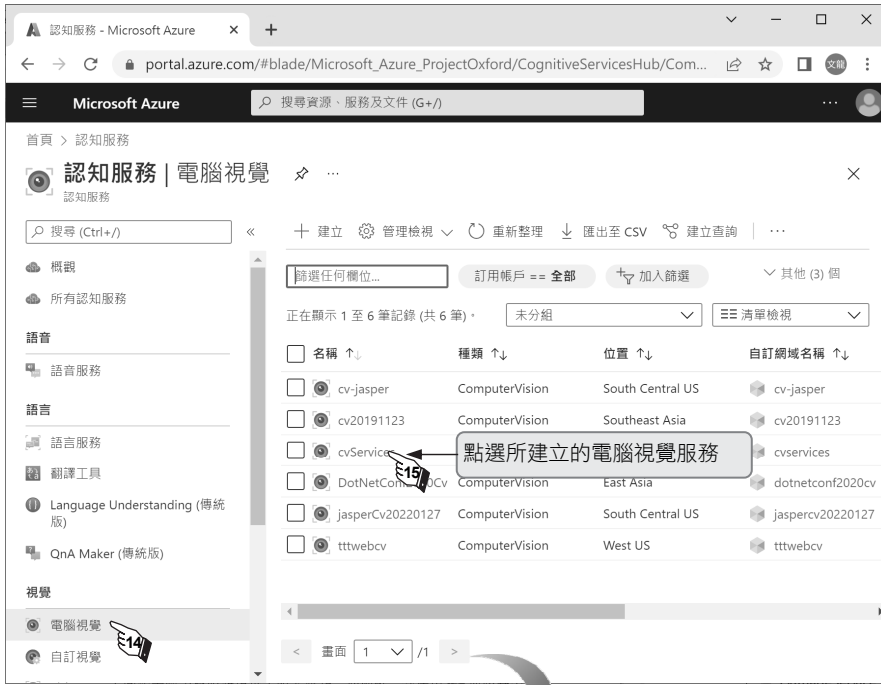
Microsoft Azure


Azure 服務

所有資源

也可以點選 [所有資源] 查詢目前帳號所建立的服務

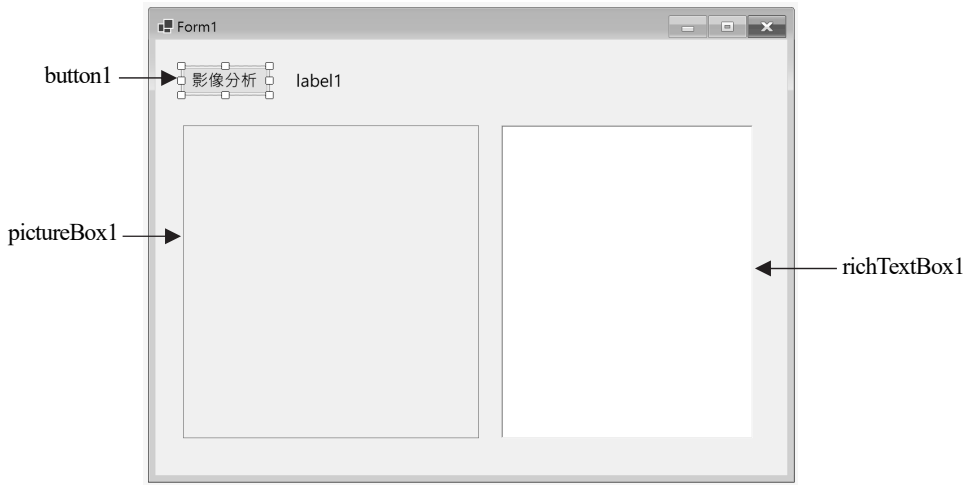




上圖電腦視覺服務提供兩組金鑰和一個端點。請使用  鈕將其中一組服務金鑰和端點複製到文字檔內，金鑰和端點撰寫程式需要使用。

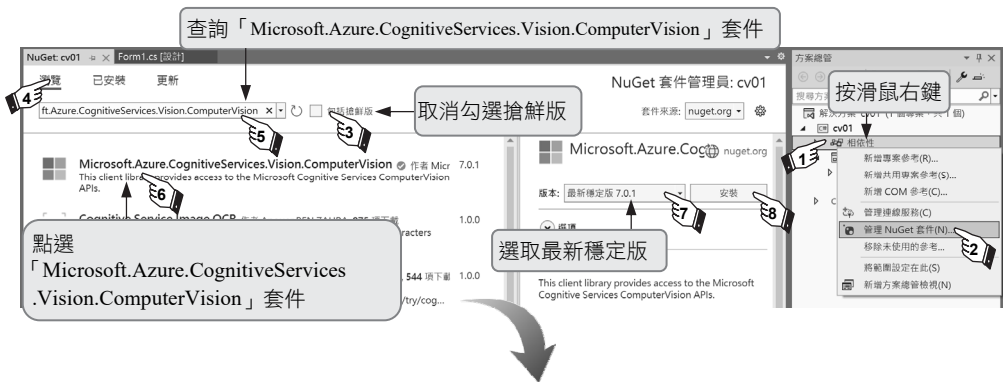
Step 2 新增專案並以「cv01」為新專案名稱。

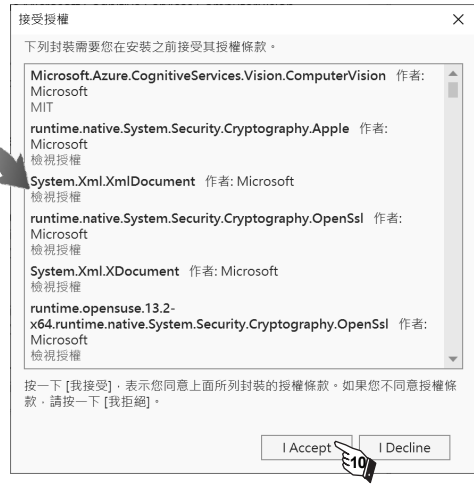
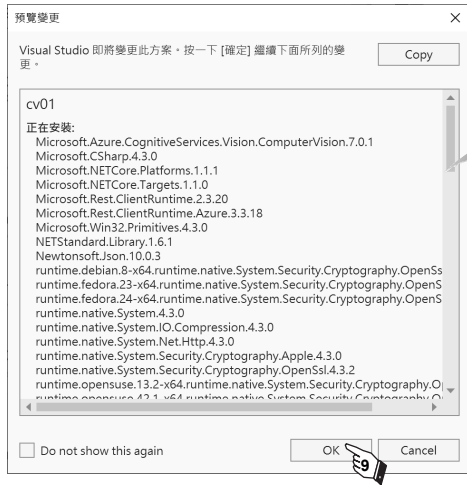
Step 3 建立表單輸出入介面，控制項名稱皆使用預設名稱。



Step 4 安裝電腦視覺 SDK

在方案總管視窗的「相依性」按滑鼠右鍵執行【管理 NuGet 套件 (N)】，接著依圖示操作安裝「Microsoft.Azure.CognitiveServices.Vision.ComputerVision」套件。





## Step 5 撰寫程式碼

**FileName:cv01.sln**

```

01 using Microsoft.Azure.CognitiveServices.Vision.ComputerVision;
02 using Microsoft.Azure.CognitiveServices.Vision.ComputerVision.Models;
03
04 namespace cv01
05 {
06     public partial class Form1 : Form
07     {
08         public Form1()
09         {
10             InitializeComponent();
11         }
12
13         //表單載入時執行
14         private void Form1_Load(object sender, EventArgs e)
15         {
16             label1.Text = "";
17         }
18
19         //按 [開檔] 鈕執行
20         private async void button1_Click(object sender, EventArgs e)
21         {

```

```
22 //顯示開檔對話方塊並判斷是否按下 [開啟] 鈕
23 if (openFileDialog1.ShowDialog() == DialogResult.OK)
24 {
25     try
26     {
27         //宣告cvApiUrl和cvApiKey用來存放服務與金鑰
28         string cvApiUrl = "電腦視覺服務端點";
29         string cvApiKey = "電腦視覺服務金鑰";
30         string imagePath = openFileDialog1.FileName;
31         //建立FileStream物件fs開啟圖檔
32         FileStream fs = File.Open(imagePath, FileMode.Open);
33
34         //建立電腦視覺辨識物件，同時指定電腦視覺辨識的雲端服務Key
35         ComputerVisionClient visionClient = new ComputerVisionClient(
36             new ApiKeyServiceClientCredentials(cvApiKey),
37             new System.Net.Http.DelegatingHandler[] { });
38
39         //電腦視覺辨識物件指定雲端服務Api位址
40         visionClient.Endpoint = cvApiUrl;
41
42         //使用DescribeImageInStreamAsync()方法傳回辨識後的影像描述物件res
43         ImageDescription res = await visionClient.DescribeImageInStreamAsync(fs);
44
45         // 若辨識失敗則傳回null
46         if (res == null)
47         {
48             richTextBox1.Text = "辨識失敗，請重新指定圖檔";
49             return;
50         }
51
52         // 使用for迴圈將影像中所有標籤項目合併於tagStr字符串中
53         string tagStr = "";
54         for (int i = 0; i < res.Tags.Count(); i++)
55         {
56             tagStr += $"{ res.Tags[i]}, ";
57         }
58
59         // 進行圖片的描述的內容顯示於richTextBox1
```

```
57     richTextBox1.Text = $"項目：{tagStr}\n" +  
      $"描述：{res.Captions[0].Text}\n" +  
      $"信度：{res.Captions[0].Confidence}";  
58  
59     //pictureBox1顯示指定的圖片  
60     pictureBox1.Image = new Bitmap(imagePath);  
61  
62     //釋放影像串流資源  
63     fs.Close();  
64     fs.Dispose();  
65     }  
66     catch (Exception ex)  
67     {  
68         richTextBox1.Text = $"錯誤訊息：{ex.Message}";  
69     }  
70     }  
71 }  
72 }  
73 }
```

本例程式撰寫注意事項如下：

1. 電腦視覺 ComputerVisionClient 物件的 DescribeImageInStreamAsync() 為非同步方法，故呼叫時必須加上 await 關鍵字(第 41 行)，使用的事件處理函式也要定義為 async(第 20 行)。
2. 本例必須自行申請電腦視覺金鑰與端點才能正常執行，金鑰與端點設定於第 28~29 行處。

關於更多更詳細相關電腦視覺服務、認知服務與 Azure 雲端平台的功能可參閱峯出版的 Visual C# 2022 程式設計經典書籍。