

CH1

Visual C# 2017 的基本概念

目錄

- 1-1 認識程式語言
- 1-2 Visual Studio Community 2017(內含Visual C#)開發環境介紹
- 1-3 物件導向程式設計的基礎概念
- 1-4 簡單加法計算機

1-1 認識程式語言

- 1-1-1 何謂程式語言
- 程式語言是人與電腦溝通的媒介，要了解程式語言才能命令電腦完成想要的功能。
- 需要先了解程式語言的儲存資料的機制、計算所需要的運算子、選擇結構與重複結構的語法。

1-1-2 程式語言的分類

- 機器語言(Machine Language) 是最早的程式語言，程式設計師必須查詢指令的二進位碼，並將資料轉成0 與1
- 組合語言(Assembly Language)改進機器語言以二進位方式撰寫程式，提供以英文字串表達指令與資料，透過組譯器(Assembler) 將組合語言轉成機器語言

- 高階語言(High-Level programming language) , 透過編譯器(Compiler) 轉成機器語言，編譯器支援各類型的處理器，就可將程式轉成機器語言
- 高階語言也可以透過直譯器(Interpreter) 執行，直譯器會將程式從頭開始一行接著一行執行，透過不同處理器版本的直譯器達成跨平台。

- 高階語言又可以分成程序導向與物件導向兩類。
程序導向是以功能為出發點，可以將一個大程式分割成幾個小程式，完成這些小程式，再利用呼叫小程式組合成大程式
- C 屬於程序導向的程式語言，Visual C# 與Java 屬於物件導向的程式語言，而C++、PHP 與Python 是程序導向的程式語言，也是物件導向的程式語言。

1-1-3 Visual C# 程式語言

- Visual C# 提供整合開發環境(integrated development environment，簡稱IDE)，整合開發環境(IDE) 為程式撰寫、編譯、執行與除錯等環境整合在一起
- Visual C# 可以用於開發Web 應用程式與視窗化的軟體等，並提供元件可連結資料庫。

1-2 Visual Studio Community 2017

開發環境介紹

- 1-2-1 下載、安裝與執行Visual Studio Community 2017
- **STEP 01** 到以下網址下載Visual Studio Community 2017 :
<https://www.visualstudio.com/zh-hant/downloads/>

- **STEP 02** 選擇Visual Studio Community 的「免費下載」，如圖1-1 所示。
- 圖1-1 選擇「免費下載」



- **STEP 03** 瀏覽器會自動下載安裝程式「vs_community.exe」，該程式會再從網路下載完整的安裝程式。



安裝Visual Studio Community 2017

- **STEP 01** 點選安裝程式「**vs_community.exe**」，出現如下安裝畫面，點選「繼續」。



- **STEP 02** 勾選想要安裝的功能，可以勾選「通用 Windows 平台開發」、「.NET 桌面開發」、「使用C++ 的桌面開發」，順便安裝網頁開發程式「**ASP.NET 與網頁程式開發**」，將來可能會使用到資料庫，所以安裝「**資料儲存與處理**」，接著卷軸往下移動。

工作負載 個別元件 語言套件

Windows (2)



通用 Windows 平台開發

使用 C#、VB、JavaScript 或選用 C++，建立適用於通用 Windows 平台的應用程式。



使用 C++ 的桌面開發

使用 Visual C++ 工具組、ATL 以及 MFC 和 C++/CLI 這類選用功能的能力，來建置傳統 Windows 應用程式。



.NET 桌面開發

使用 .NET Framework 建置 WPF、Windows Forms 和主控台應用程式。



Web 與雲端 (7)



ASP.NET 與網頁程式開發

使用 ASP.NET、ASP.NET Core、HTML、JavaScript 及 CSS 建置 Web 應用程式。



Azure 開發

用於開發雲端應用程式及建立資源的 Azure SDK、工具及專案。



Python 開發

對 Python 進行編輯、偵錯、互動式開發及原始檔控制。



Node.js 開發

使用非同步的事件驅動 JavaScript 執行階段 Node.js 建置可調整的網路應用程式。



資料儲存和處理

使用 SQL Server、Azure Data Lake、Hadoop 或 Azure ML 連線、開發及測試資料解決方案。



資料科學與分析應用程式

用於建立資料科學應用程式的語言與工具，包括 Python、R 及 F#。



Office/SharePoint 開發

使用 C#、VB 及 JavaScript 建立 Office 與 SharePoint 增益集、SharePoint 解決方案，以及 VSTO 增益集。



摘要

- > 使用 C++ 的桌面開發
- > .NET 桌面開發
- > 資料儲存和處理
- > Visual Studio 擴充功能開發

✓ ASP.NET 與網頁程式開發

包含

- ✓ .NET Framework 4.6.1 開發工具
- ✓ .NET Core 1.0 - 1.1 開發工具
- ✓ ASP.NET 與網頁程式開發工具
- ✓ Developer Analytics Tools
- ✓ WebSocket4Net

選擇性

- ☒ .NET Framework 4 - 4.6 開發工具
- ☒ 容器開發工具
- ☒ Cloud Explorer
- ☒ 分析工具
- ☒ Entity Framework 6 工具
- ☒ Windows Communication Foundation
- ☐ ASP.NET MVC 4
- ☐ 類別設計工具
- ☐ F# 語言支援

i 繼續進行即表示您同意所選 Visual Studio 版本的授權。我們同時讓您使用 Visual Studio 下載其他軟體。此軟體為分開授權，如同協力廠商聲明或其隨附的授權中所述。繼續進行即表示您也同意該授權。

位置

C:\Program Files (x86)\Microsoft Visual Studio\2017\Community



安裝大小: 14.5 GB

安裝

- **STEP 03** 可以考慮安裝「**Visual Studio 擴充功能開發**」，最後點選「**安裝**」。

Visual Studio

正在安裝 - Visual Studio Community 2017 - 15.2 (26430.15)

工作負載 個別元件 語言套件



Office/SharePoint 開發

使用 C#、VB 及 JavaScript 建立 Office 與 SharePoint 增益集、SharePoint 解決方案，以及 VSTO 增益集。

行動與遊戲 (5)



使用 .NET 進行行動開發

使用 Xamarin 建置適用於 iOS、Android 或 Windows 的跨平台應用程式。



使用 JavaScript 進行行動開發

使用 Apache Cordova 工具建置 Android、iOS 及 UWP 應用程式。



使用 C++ 進行遊戲開發

使用 C++ 的完整功能建置由 DirectX、Unreal 或 Cocos2d 所提供的專業遊戲。



使用 Unity 進行遊戲開發

使用強大的跨平台開發環境 Unity 建立 2D 與 3D 遊戲。



使用 C++ 進行行動開發

使用 C++ 建置適用於 iOS、Android 或 Windows 的跨平台應用程式。



Visual Studio 擴充功能開發

建立適用於 Visual Studio 的增益集與擴充功能，包括新的命令、程式碼分析器與工具視窗。



使用 C++ 進行 Linux 開發

建立及偵錯在 Linux 環境中執行的應用程式。



.NET Core 跨平台開發

使用 .NET Core、ASP.NET Core、HTML、JavaScript 及 CSS 建置跨平台應用程式。

位置

C:\Program Files (x86)\Microsoft Visual Studio\2017\Community

摘要

- > 使用 C++ 的桌面開發
- > .NET 桌面開發
- > 資料儲存和處理
- > Visual Studio 擴充功能開發

✓ ASP.NET 與網頁程式開發

包含

- ✓ .NET Framework 4.6.1 開發工具
- ✓ .NET Core 1.0 - 1.1 開發工具
- ✓ ASP.NET 與網頁程式開發工具
- ✓ Developer Analytics Tools
- ✓ WebSocket4Net

選擇性

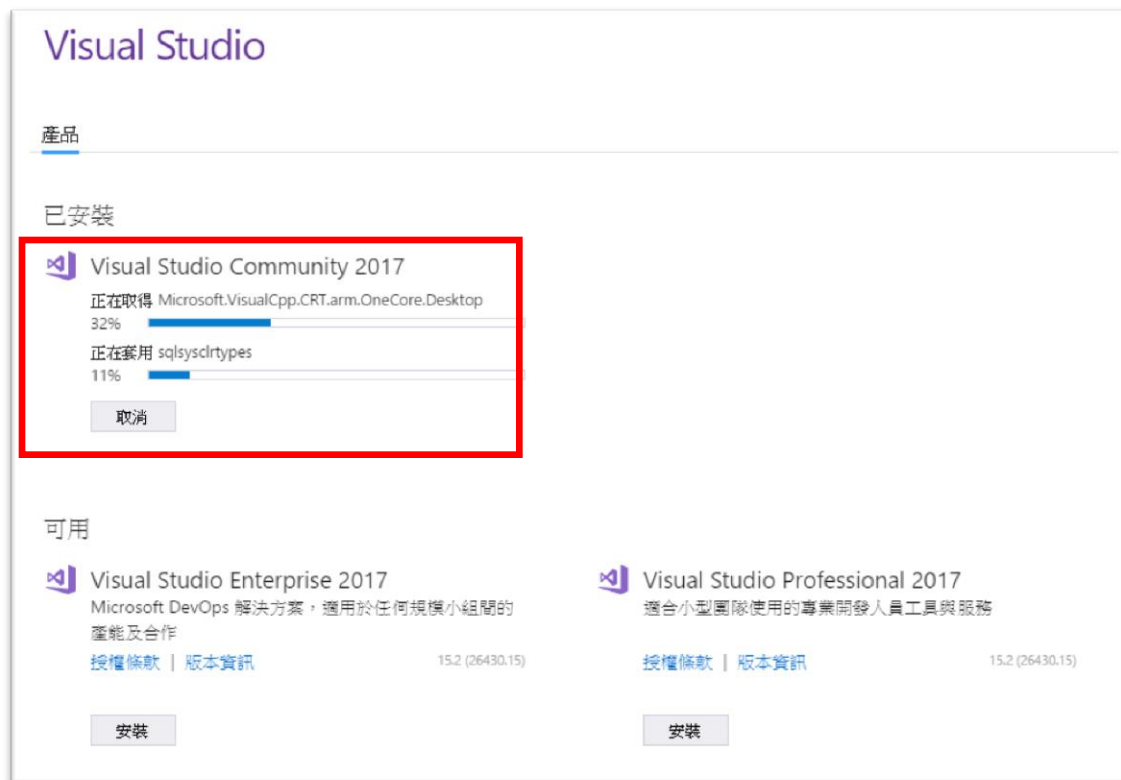
- ☒ .NET Framework 4 – 4.6 開發工具
- ☒ 容器開發工具
- ☒ Cloud Explorer
- ☒ 分析工具
- ☒ Entity Framework 6 工具
- ☒ Windows Communication Foundation
- ☐ ASP.NET MVC 4
- ☐ 類別設計工具
- ☐ F# 語言支援

i 繼續進行即表示您同意所選 Visual Studio 版本的授權。我們同時請您使用 Visual Studio 下載其他軟體。此款軟體為分開授權，如同[協力廠商聲明](#)或其隨附的授權中所述。繼續進行即表示您也同意該授權。

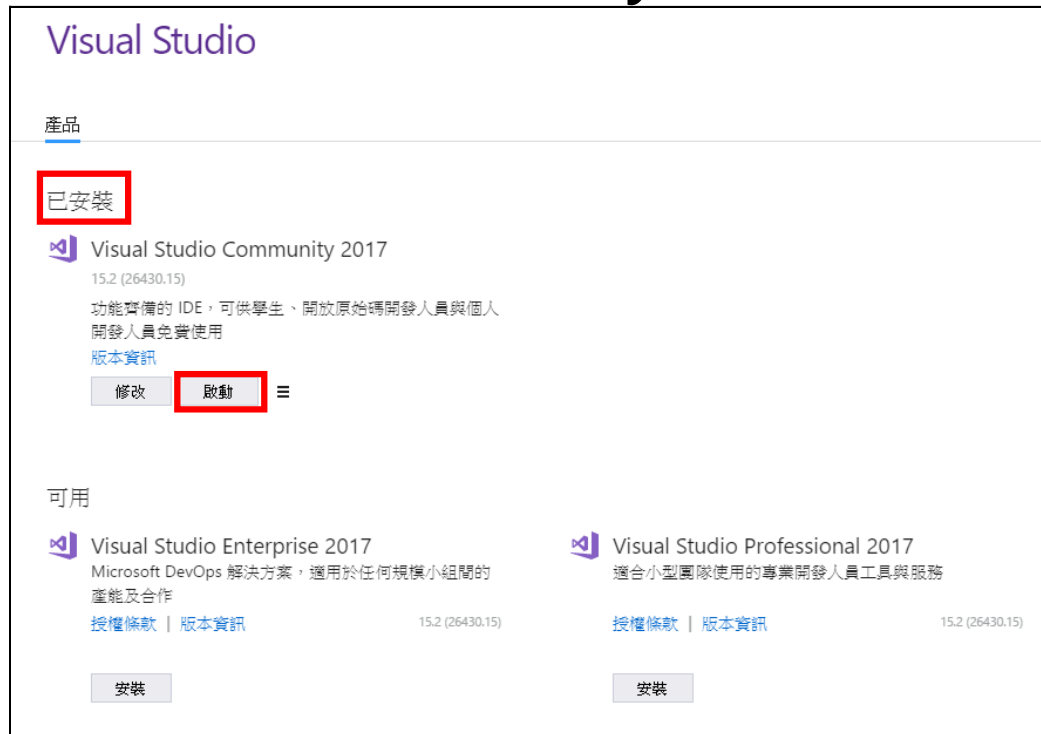
安裝大小: 14.5 GB

安裝

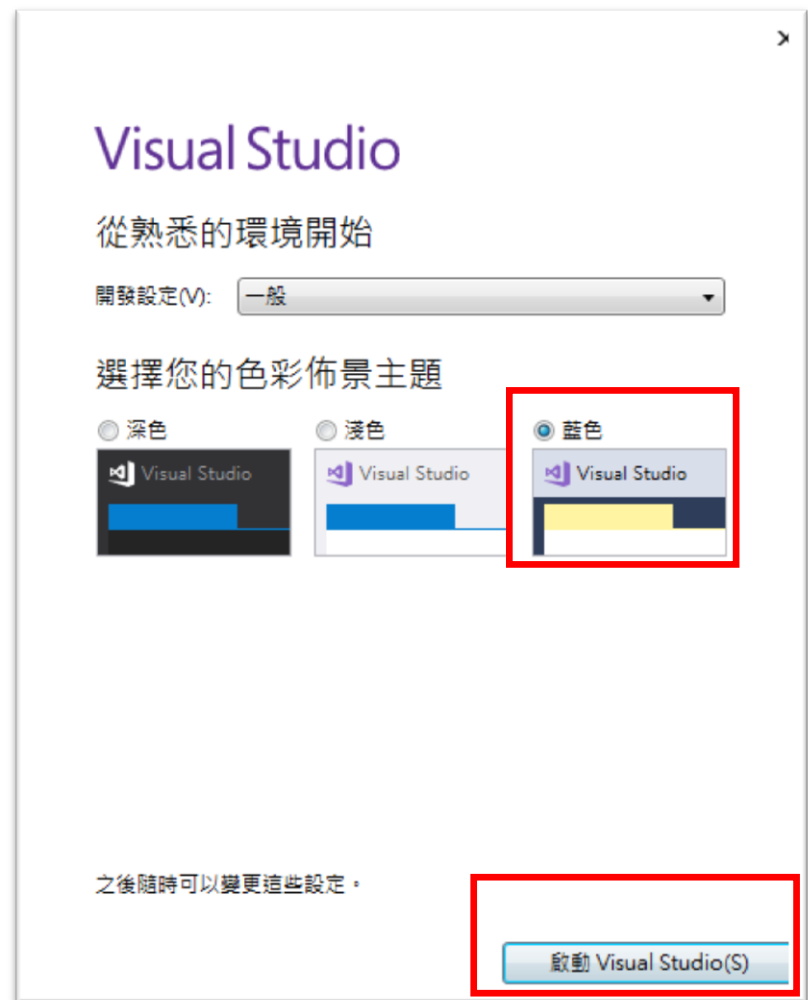
- **STEP 04** 出現下圖，表示開始下載與安裝。



- **STEP 05** 安裝完成後，會出現下圖，顯示「已安裝」表示安裝完成，點選「**啟動**」就可以啟動 Visual Studio Community 2017。



- **STEP 06**選擇佈景主題，
例如：藍色，接著點選
「**啟動Visual
Studio(S)**」。



- **STEP 07** 啟用 Visual Studio 後，如下圖。



1-2-2 專案管理與操作環境介紹

- 新增專案
- 點選「檔案→新增→專案」
- ❶ Visual C#
- ❷ Windows 傳統桌面
- ❸ Windows Forms App」
- ❹ 選擇程式要儲存的資料夾
- ❺ 「名稱」部分可以輸入想要的名稱
- ❻ 勾選「為方案建立目錄」
- ❼ 點選「確定」。

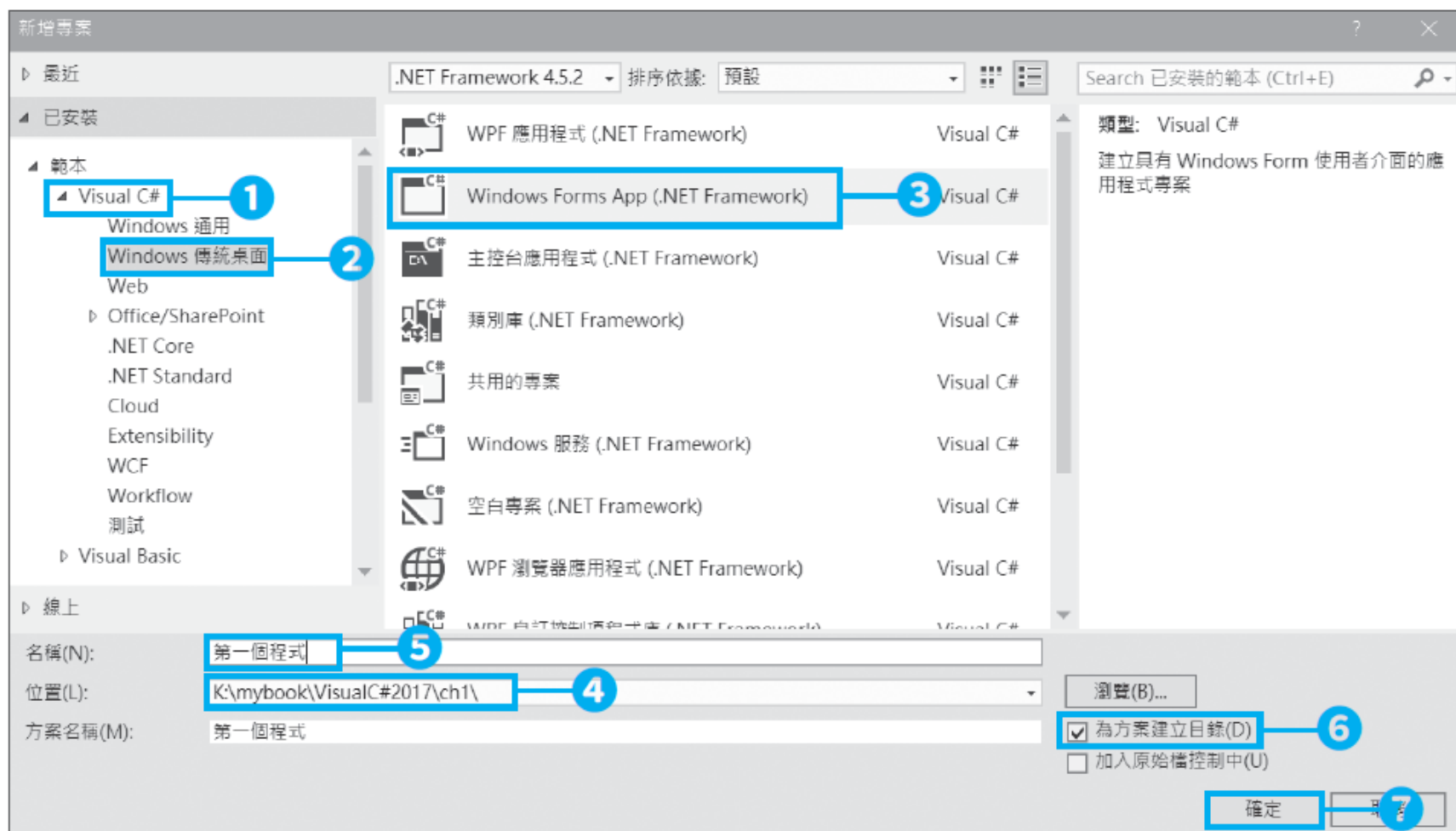


圖 1-11 新增專案

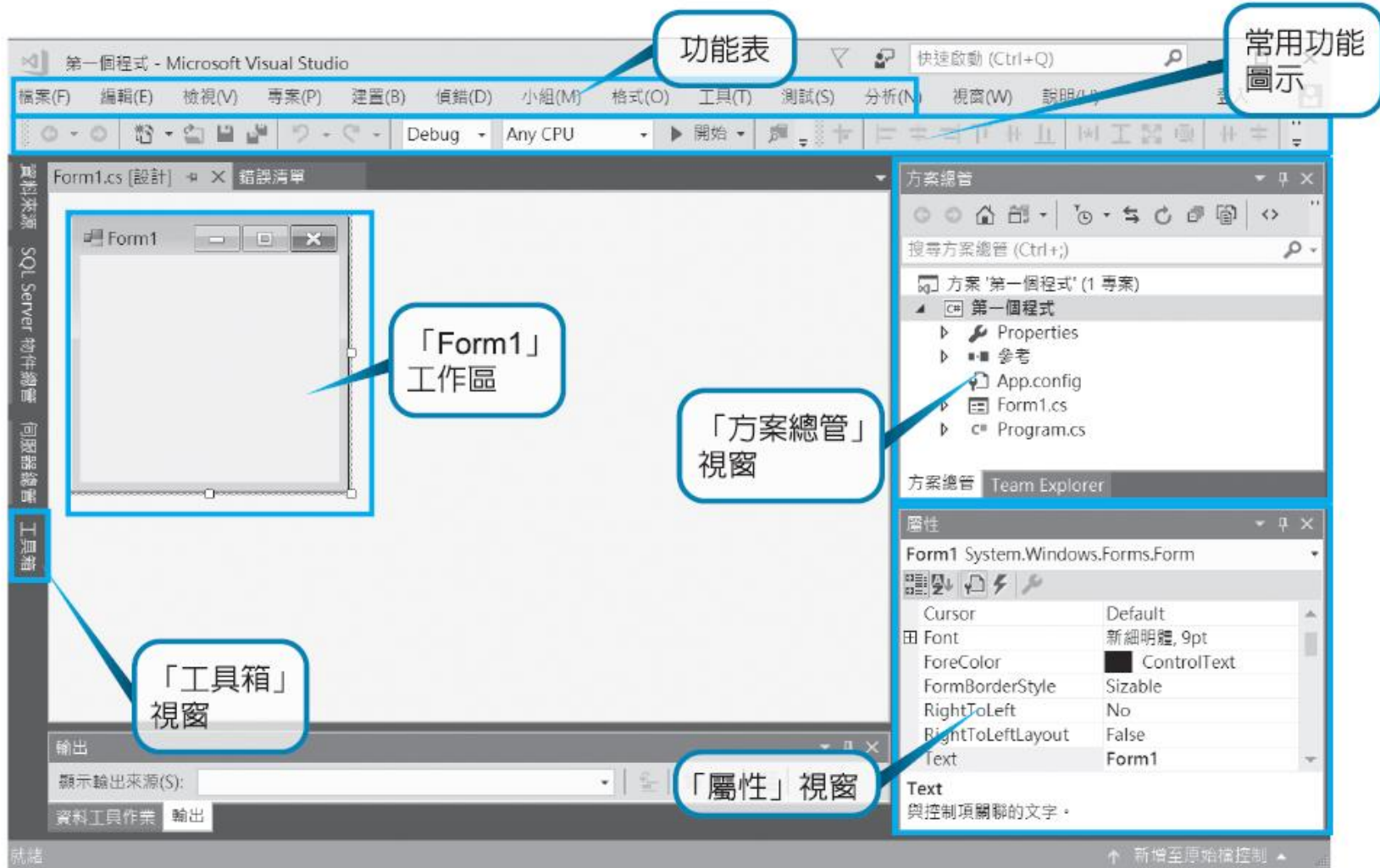


圖 1-12 操作畫面

操作環境介紹

- 功能表
 - 有「檔案」、「編輯」、「檢視」等為Visual C# 的所有功能選單，每個選單下拉之後又有多種選項可以選擇，讀者若有興趣可以點選每個功能選單看看，來熟悉Visual C# 的操作環境。

- 常用功能圖示

- 將常用功能做成圖示，直接點選就可以使用，不需要再由功能表一步一步選，若想要新增其他功能圖示可以點選「檢視→工具列」，選擇想要新增的功能圖示。

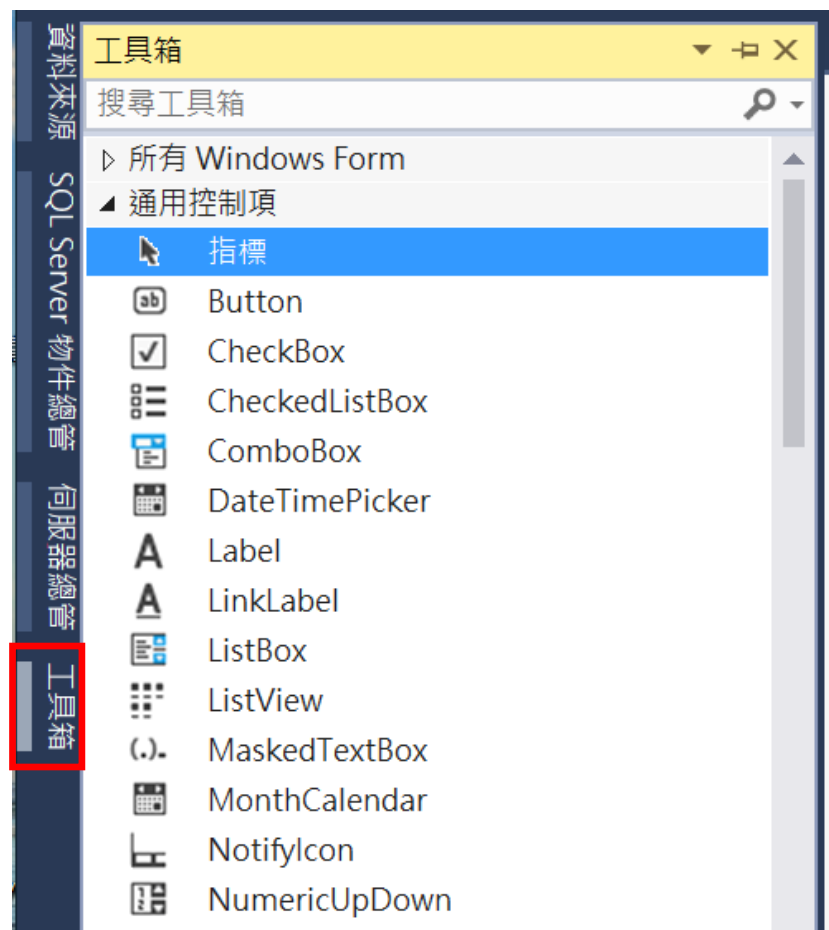
- 「方案總管」視窗
 - 列出方案中的所有檔案與資源，方便管理與使用方案。
- 「Form1」工作區
 - 「Form1」為Visual C# 預設的第一個表單，使用者可以新增、刪除與設定元件，「Form1」表單所呈現的外觀，就是程式執行時的畫面。



- 「工具箱」視窗

- 表列Visual C# 所提供的視窗化元件，直接拖曳到「Form1」工作區，就可以於「Form1」工作區新增該元件。
- 「屬性」視窗當點選「Form1」內的元件時，屬性視窗就會顯示該元件的屬性狀態，可以進行屬性的設定與更改。

1-2-3 詳細操作環境介紹

- 工具箱的展開與收合
 - 點選左側「工具箱」圖示可以展開工具箱。



- 點選「」按鈕一次固定視窗，如圖
- 再點選「」按鈕一次收合
- 以上操作可用於 Visual C# 其他視窗操作的共同準則。

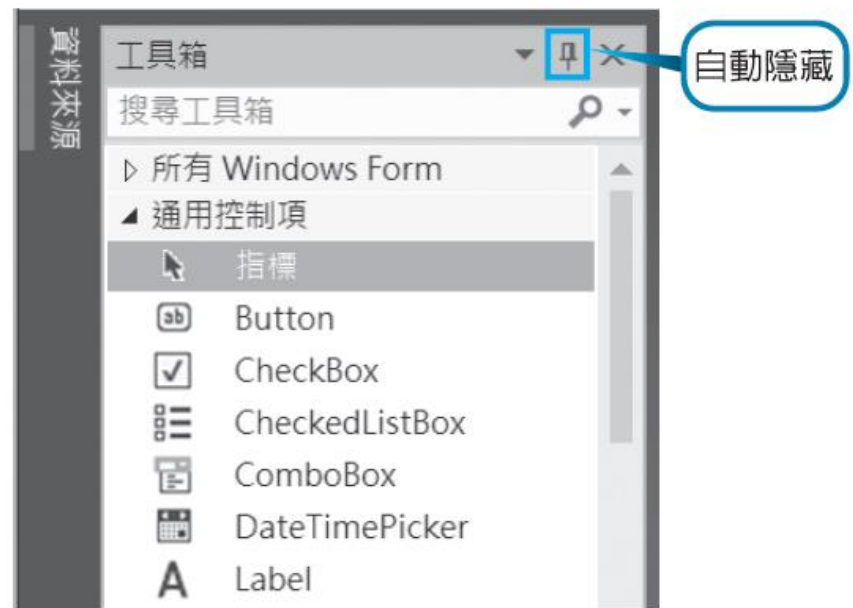


圖 1-14 點選按鈕一次固定視窗，再點選一次收合

「設計工具檢視」與「檢視程式碼」 模式切換

- 點選「Form1」表單滑鼠右鍵後，點選「檢視程式碼」按鈕。

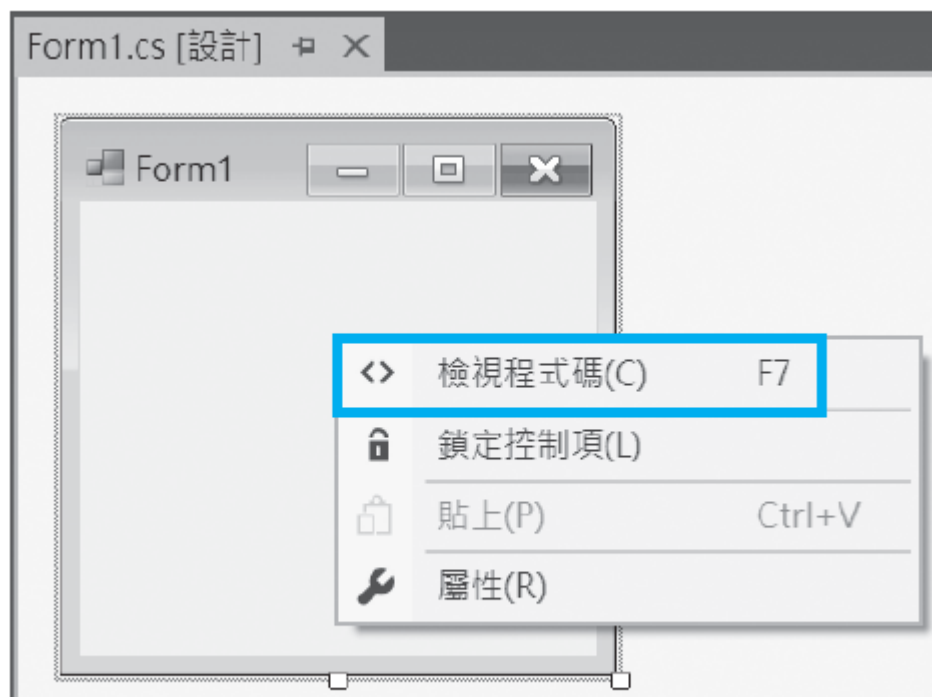


圖 1-15 點選「檢視程式碼」按鈕

- 「設計工具檢視」轉換成「檢視程式碼」。

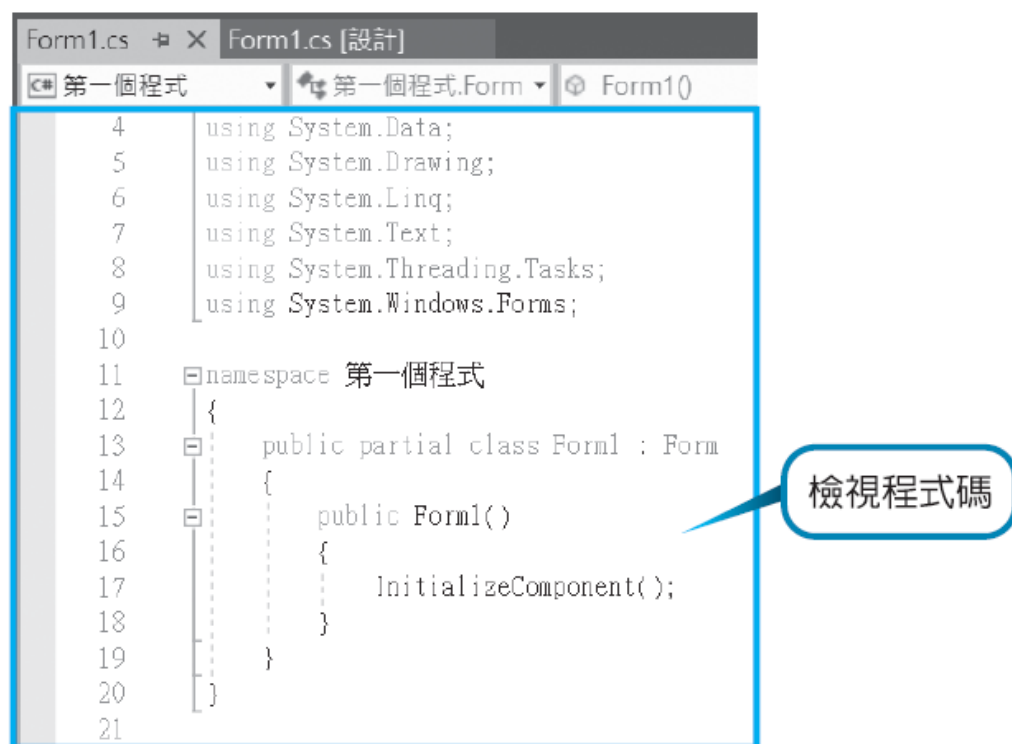


圖 1-16 檢視程式碼

- 點選「Form1.cs[設計]」，由「檢視程式碼」轉換成「設計工具檢視」。

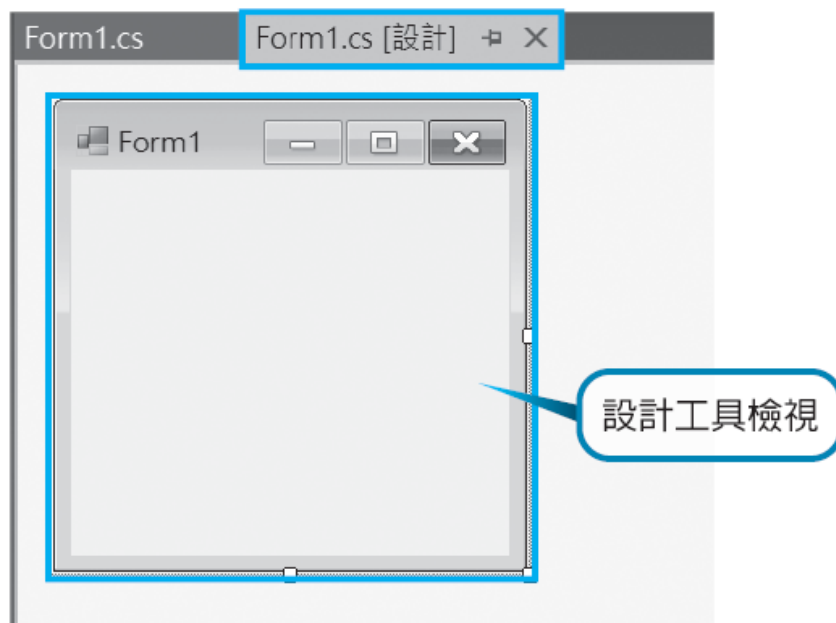


圖 1-17 設計工具檢視

視窗的顯示與隱藏

- 點選「檢視」或「檢視→其他視窗」，經由點選可以顯示或隱藏視窗，如方案總管、伺服器總管與屬性管理員等。

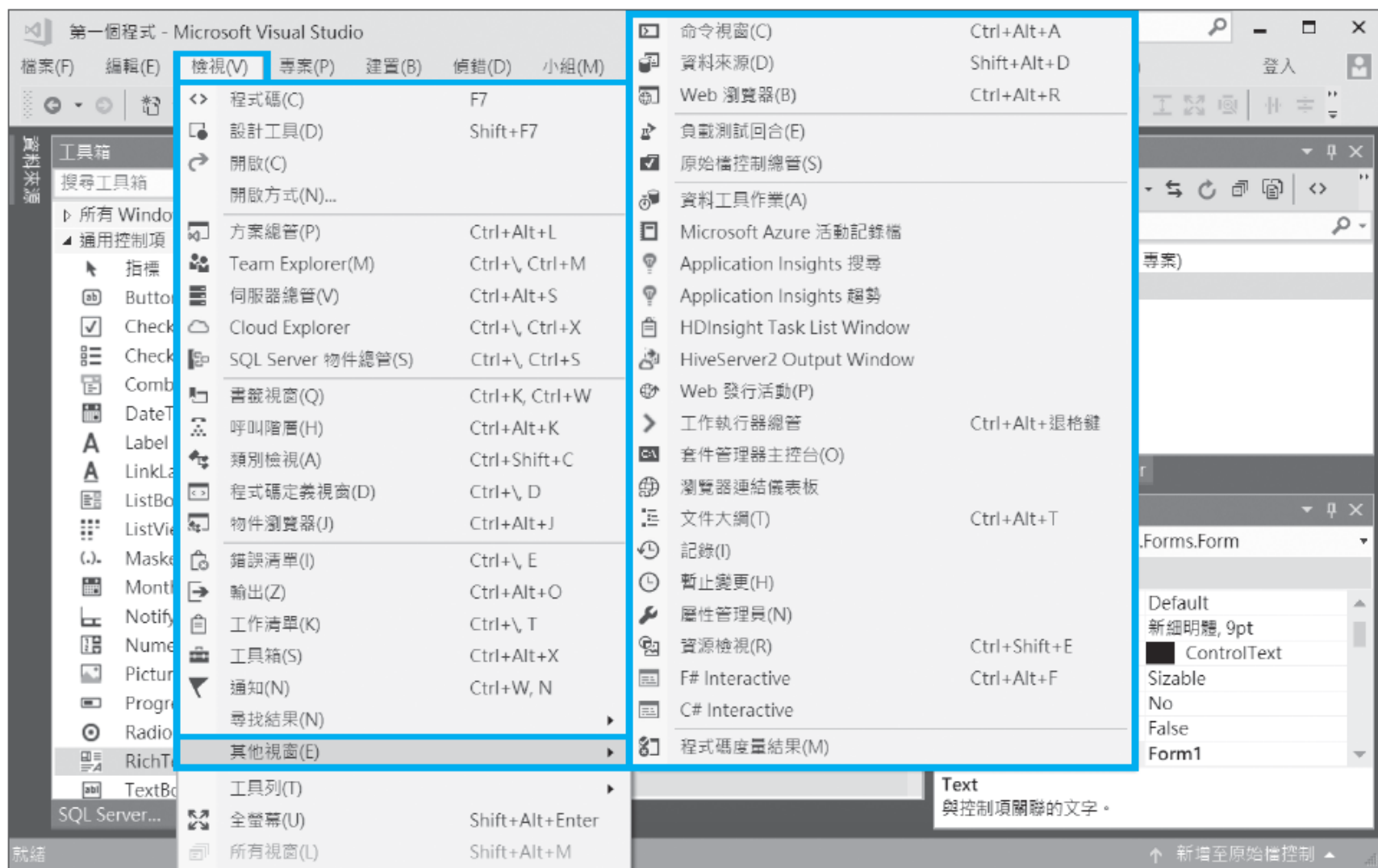


圖 1-18 經由點選可以顯示或隱藏視窗

視窗的大小與位置

- Form1 工作區、工具箱、屬性視窗、方案總管等視窗可以調整大小與位置，在視窗的邊界處出現雙向箭頭就可以調整大小，拖曳視窗標題就可以移動視窗到另一個視窗的上下左右或疊合在一起。

調整視窗大小

- 以滑鼠點選視窗的邊界，出現雙向箭頭就可以利用滑鼠拖曳方式，調整視窗到想要的大小。視窗調整過程如圖

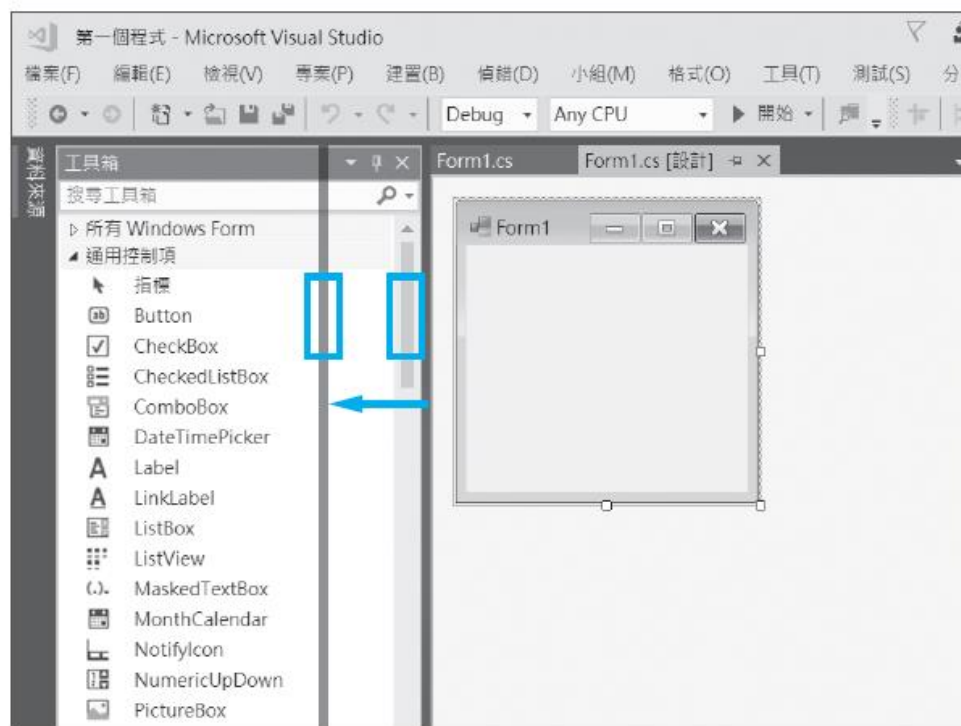
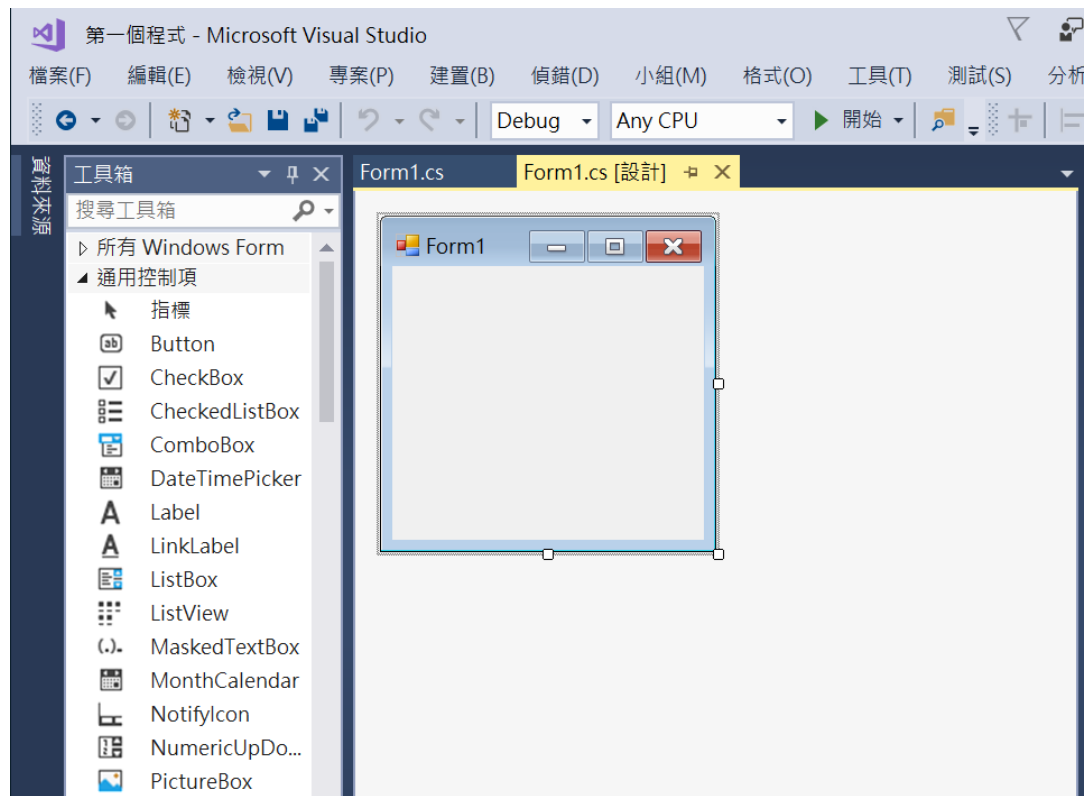


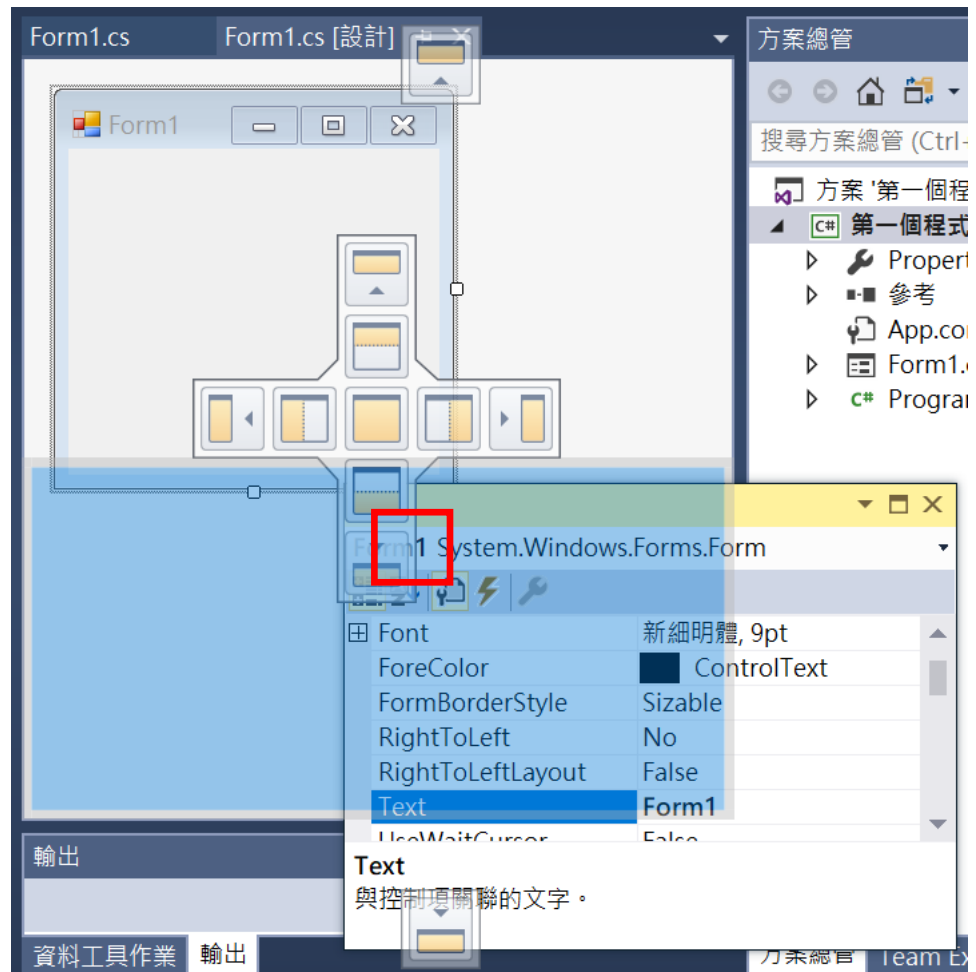
圖 1-19 利用滑鼠拖曳方式，調整視窗

- 調整後的視窗如圖1-20所示。

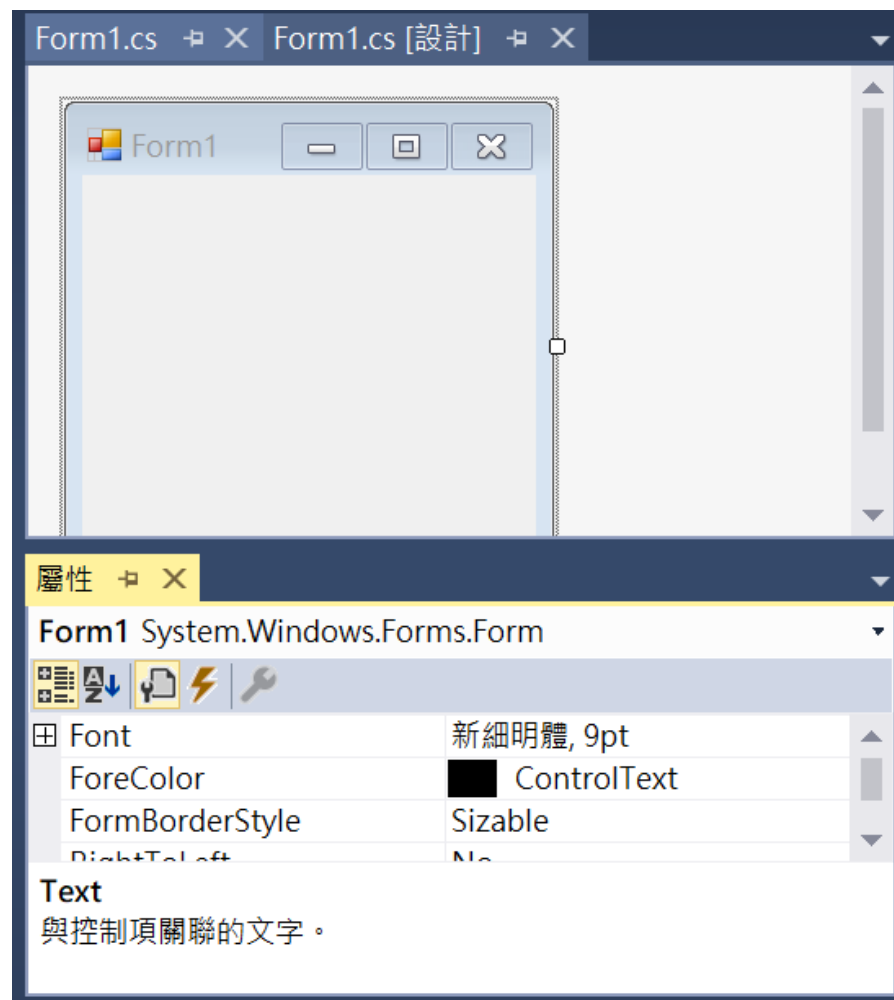


移動視窗

- 將「屬性」視窗移動到「Form1」視窗之下，「屬性」視窗將疊合「Form1」視窗之下。



- 「屬性」視窗與「Form1」視窗疊合後畫面，「Form1」視窗在上面，而「屬性」視窗在下面。



1-2-4 基礎元件介紹

- Form 用於放置各類元件
- Button 用於接受使用者按下Button 啟動事件程式
- Label用於顯示文字或計算的結果
- TextBox 用於使用者輸入資料

Form

- Form 為操作畫面的底層，為操作畫面的範圍，上面可以新增各類元件，如Button、Label 與 TextBox 等，也可以修改Form 的屬性

- 表1-1 Form1常用的屬性

| 物件名稱 | 屬性 | 說明 |
|-------|------|---|
| Form1 | size | 調整Form1的大小，程式設計人員可以利用滑鼠拖曳表單的右下角調整大小，對應的屬性size也會跟著修改，如圖。 |
| Form1 | Text | 設定Form1的標題，直接於屬性視窗設定，如「表單標題」，Form1標題會跟著修改，如圖。 |

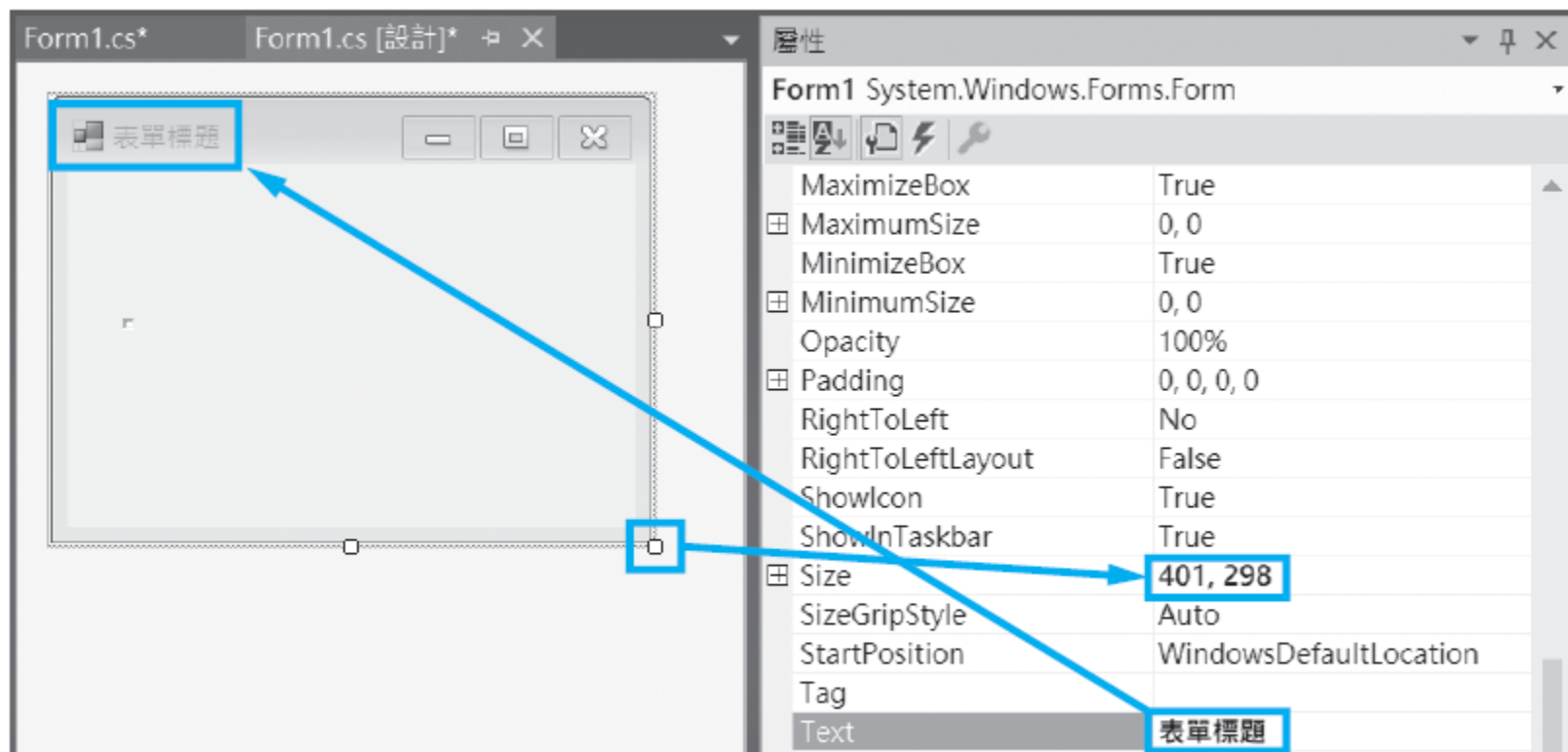


圖 1-23 更改 Form1 表單的大小與標題

Button

- Button 為按鈕元件，允許使用者按下時驅動事件，該事件內可以撰寫程式處理資料與產生輸出，Button 物件上有許多屬性可以設定，表1-2 所列為常用的屬性。

- 表1-2 Button常用的屬性

| 物件名稱 | 屬性 | 說明 |
|---------|------|---|
| button1 | Size | 調整button1的大小，程式設計人員可以利用滑鼠拖曳button1的右下角調整大小，對應的屬性Size也會跟著修改。 |
| button1 | Text | 設定button1的文字，直接於屬性視窗設定，如「我的按鈕」，button1的文字會跟著修改 |
| button1 | Font | 設定button1的字型，直接於屬性視窗設定。 |

STEP 01 於表單上新增一個 Button 元件，拖曳到 Form1 後，新增一個名為 button1 的元件。

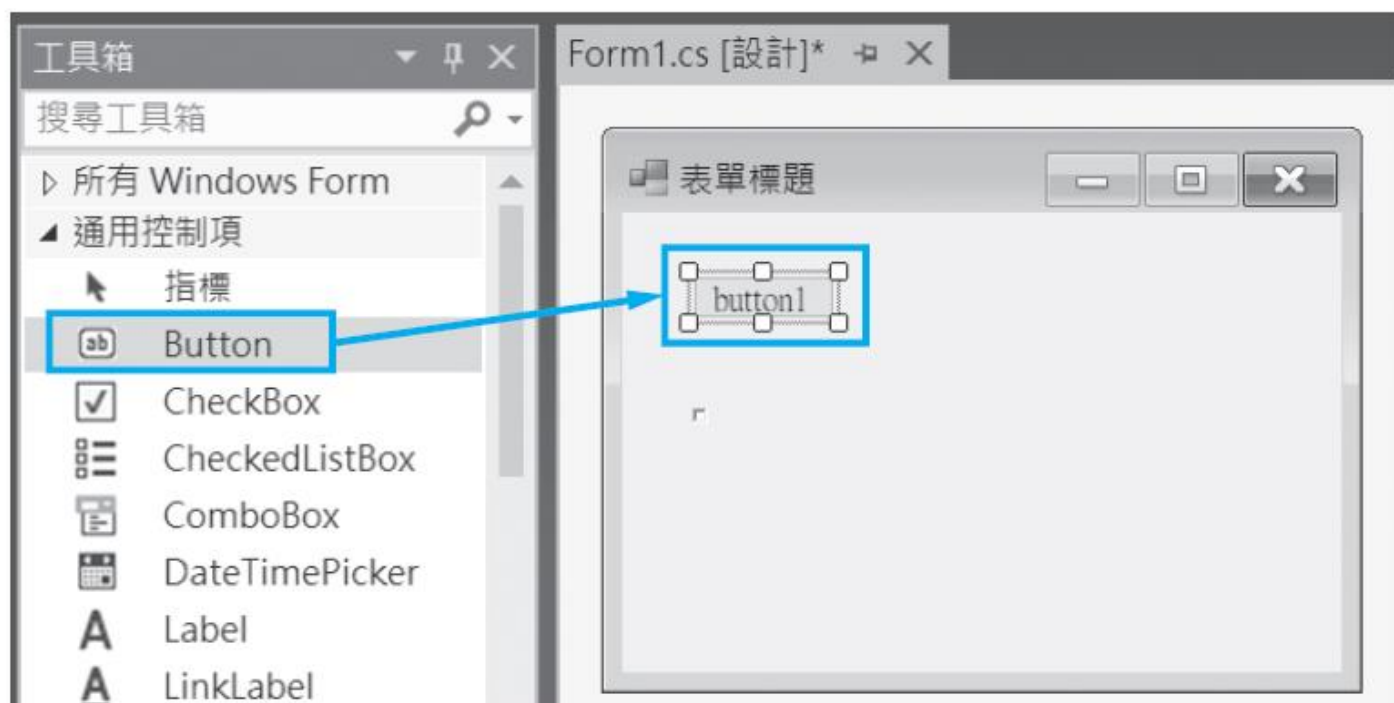


圖 1-24 新增一個 Button 元件

STEP 02 調整元件大小，於 button1 元件右下角縮放元件大小。

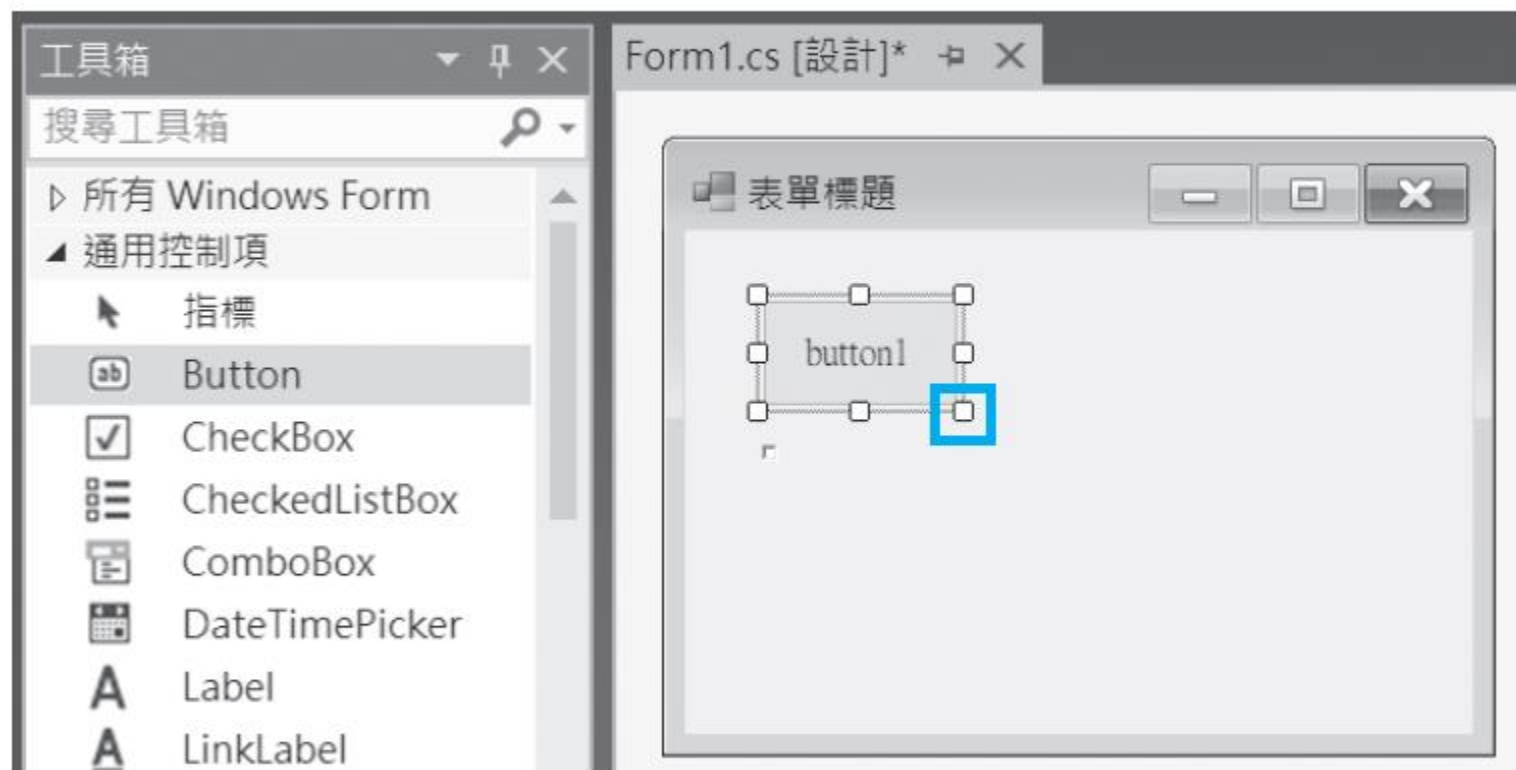


圖 1-25 元件右下角縮放大小

- 註解：
- 其實點選Button 元件後，出現八個設定點皆可調整大小，至於八個設定點的差異，讀者可透過拖曳每個設定點就可以了解。

STEP 03 設定 button1 的屬性 Text，改成「我的按鈕」。點選 button1 元件，於屬性視窗的 Text 屬性，改成「我的按鈕」。

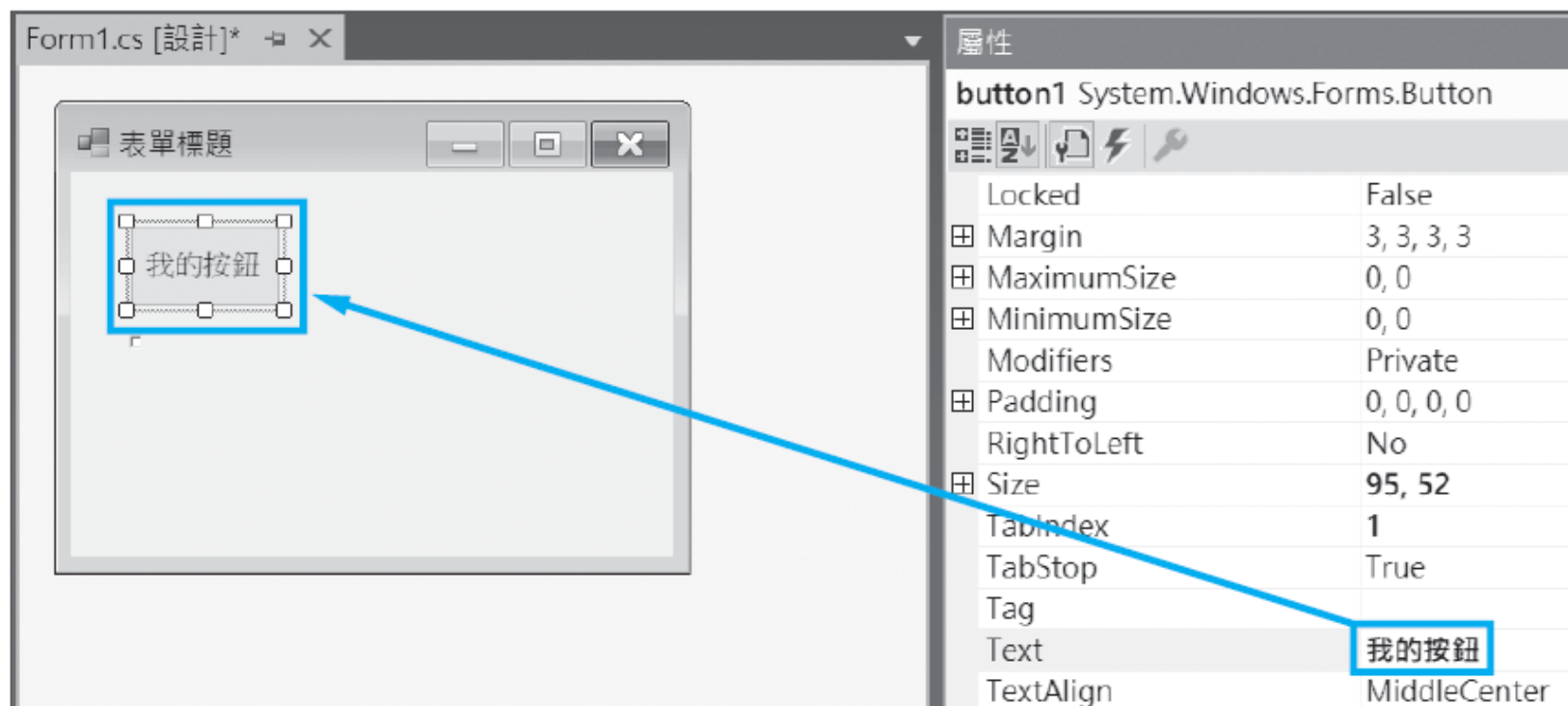


圖 1-26 修改 ext 屬性

STEP 04 修改 button1 按鈕文字的字形，修改 button1 的 Font 屬性，原本為「新細明體, 9pt」改成「新細明體, 14pt」，點選右邊「...」進行字型與大小的選擇。

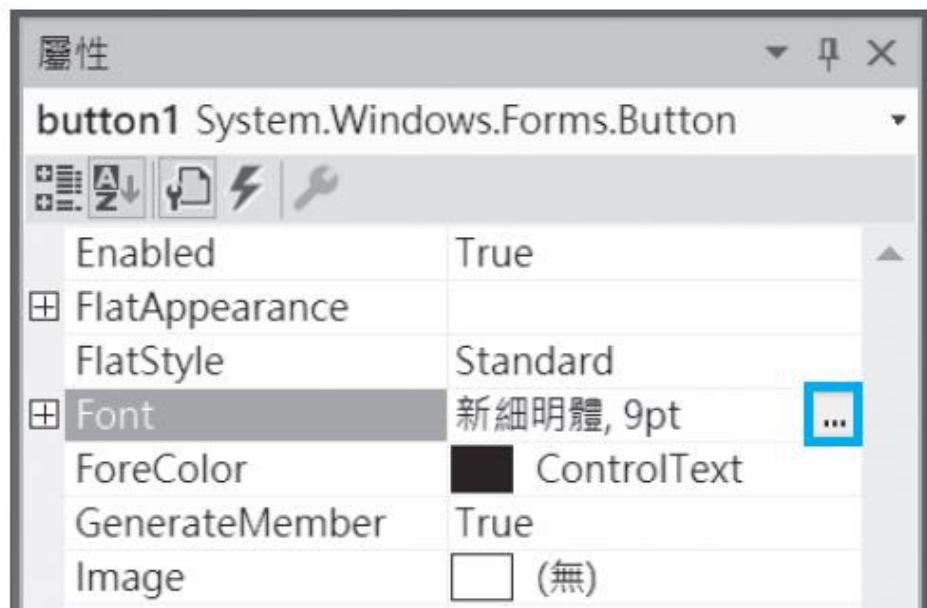


圖 1-27 修改 button1 屬性

STEP 05 於「字型」對話方塊中，大小選擇「14」，按下「確定」。

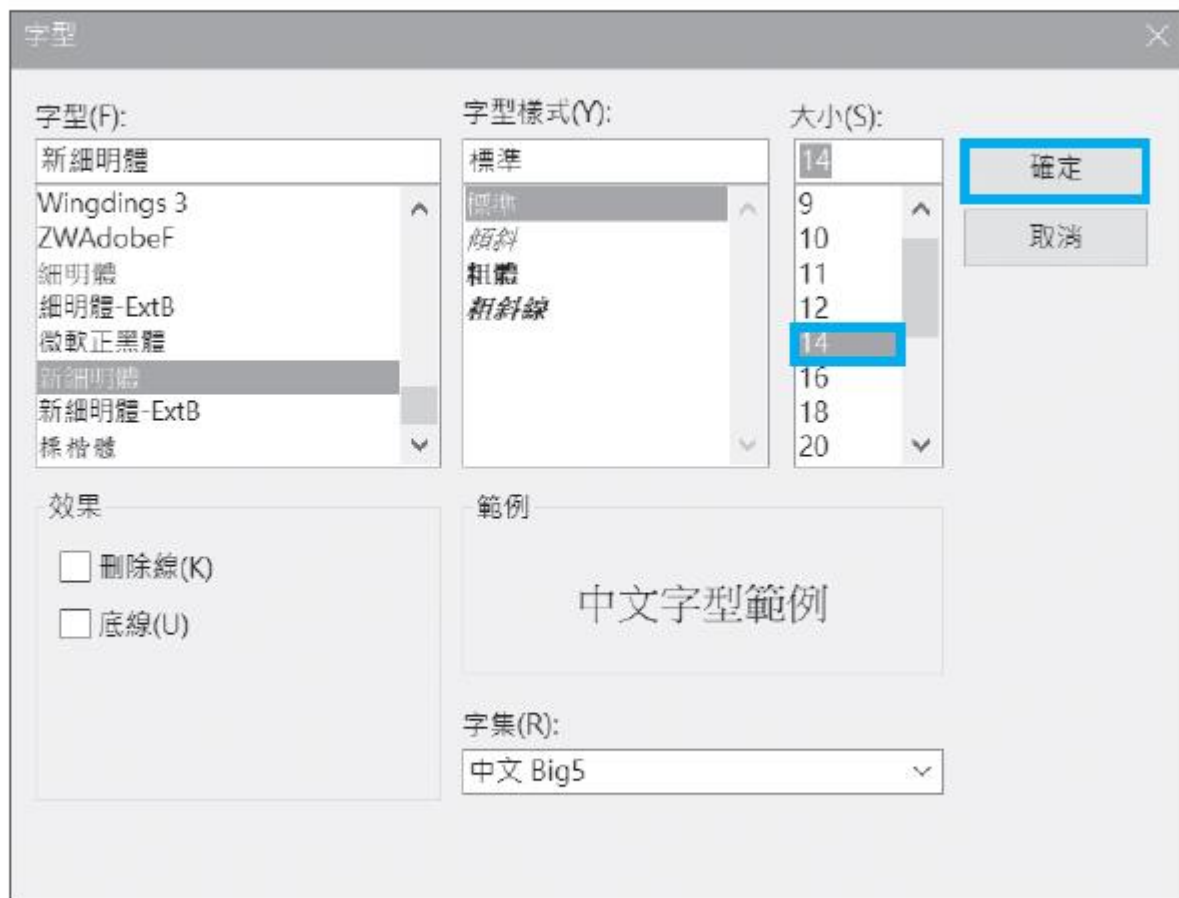


圖 1-28 大小選擇「14」並按下「確定」

STEP 06 設定完後，button1 元件上文字放大了。

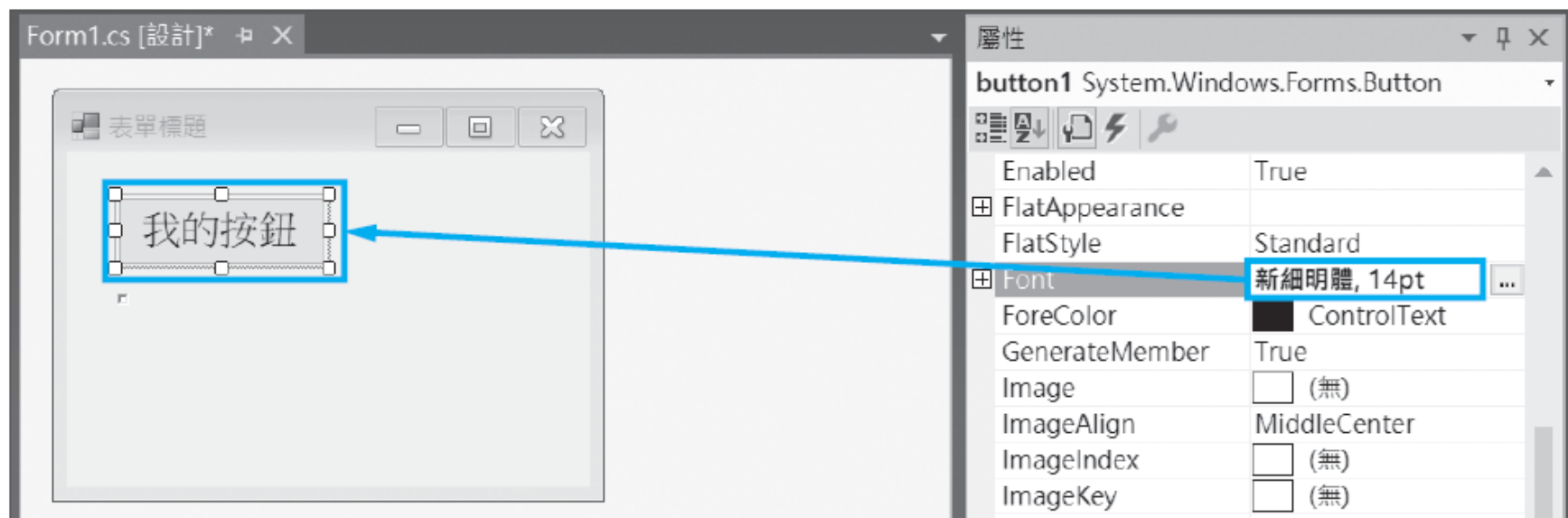
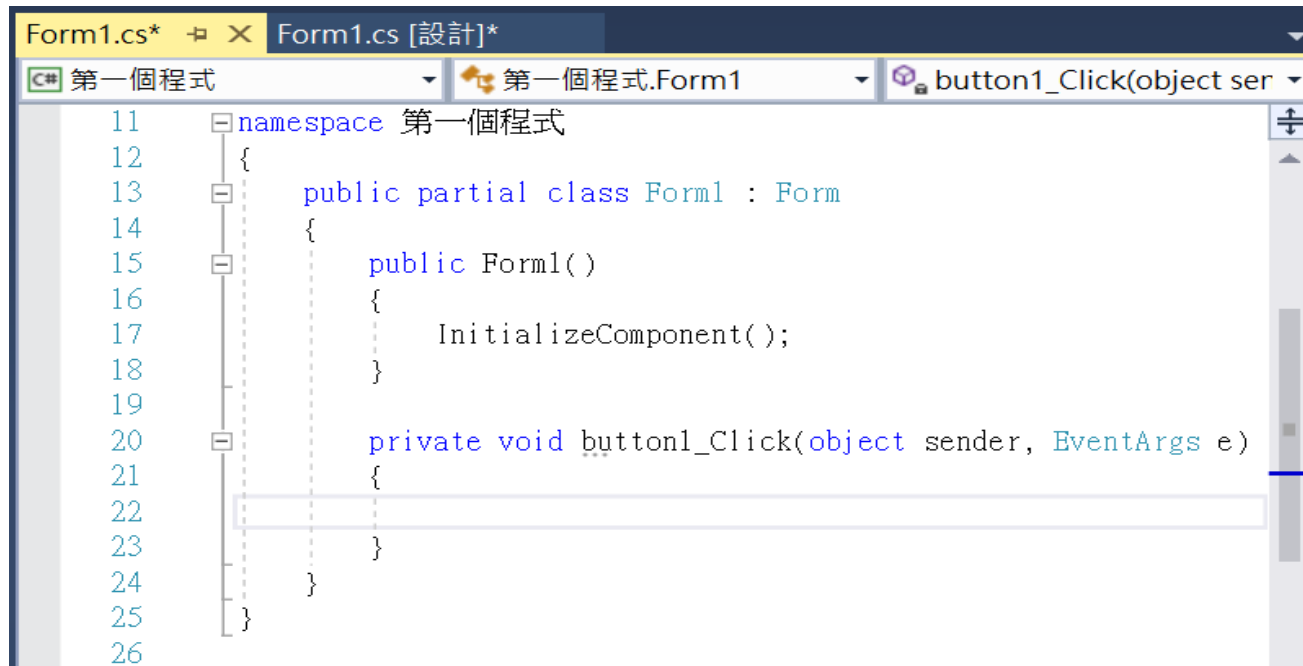


圖 1-29 設定完畢

- 表1-3 button1物件的Click事件

| 物件名稱 | 事件 | 說明 |
|---------|-------|---|
| button1 | Click | 點選button1物件後會驅動Click事件， 可以於button1_Click事件函式內，寫 入程式，程式處理資料，最後顯示結果 |

- 滑鼠左鍵點選button1 物件兩次進入button1 的 Click 事件程式編輯模式，可以於button1_Click 事件中新增程式，按下button1 就會驅動此事件。



```
Form1.cs*  Form1.cs [設計]*
C# 第一個程式 第一個程式.Form1 button1_Click(object sender, EventArgs e)
11 namespace 第一個程式
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19
20         private void button1_Click(object sender, EventArgs e)
21         {
22
23         }
24     }
25 }
26
```

Label

- Label 為顯示文字用，可以在屬性視窗預先輸入文字，或程式執行後再將結果輸出於此元件。
Label 物件上有許多屬性可以設定，表1-4 所列為常用的屬性。

| 物件名稱 | 屬性 | 說明 |
|--------|------|---|
| label1 | Text | 設定Label1的文字，直接於屬性視窗設定，如「我的標籤」，Label1的文字會跟著修改。 |

STEP 01 由工具箱拖曳 Label 元件到 Form1，在 Form1 上新增一個 label1 物件。

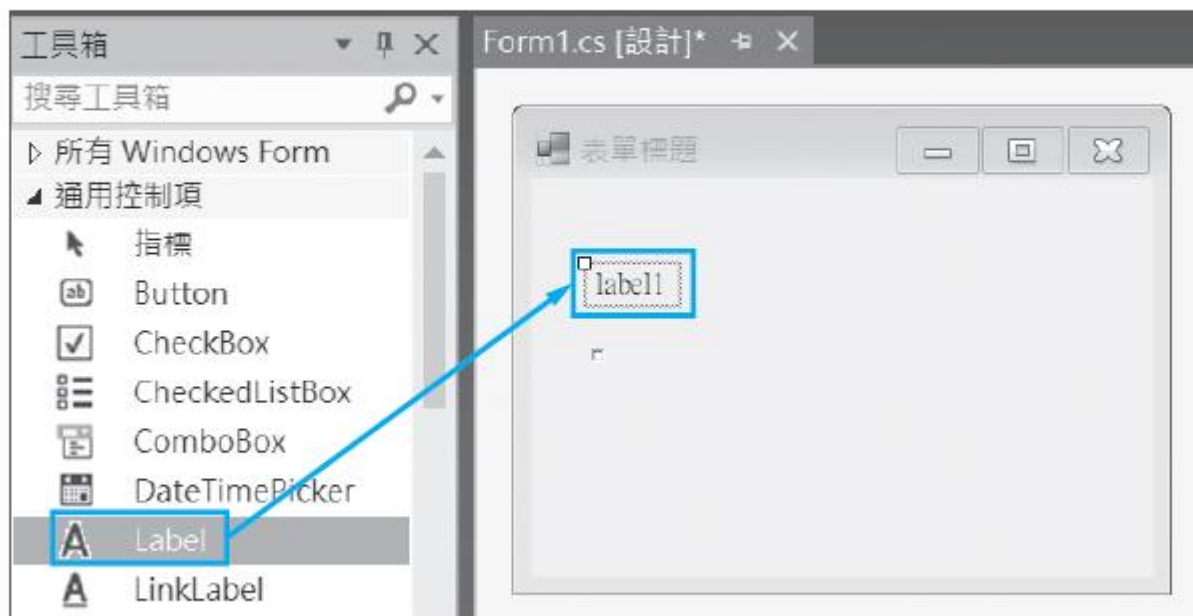


圖 1-31 在 Form1 上新增一個 label1 物件

STEP 02 設定 label1 物件的 Text 屬性為「我的標籤」，點選 label1 物件於屬性視窗的 Text 屬性改成「我的標籤」，如下圖，對應的 label1 物件也會顯示更改後的狀態。

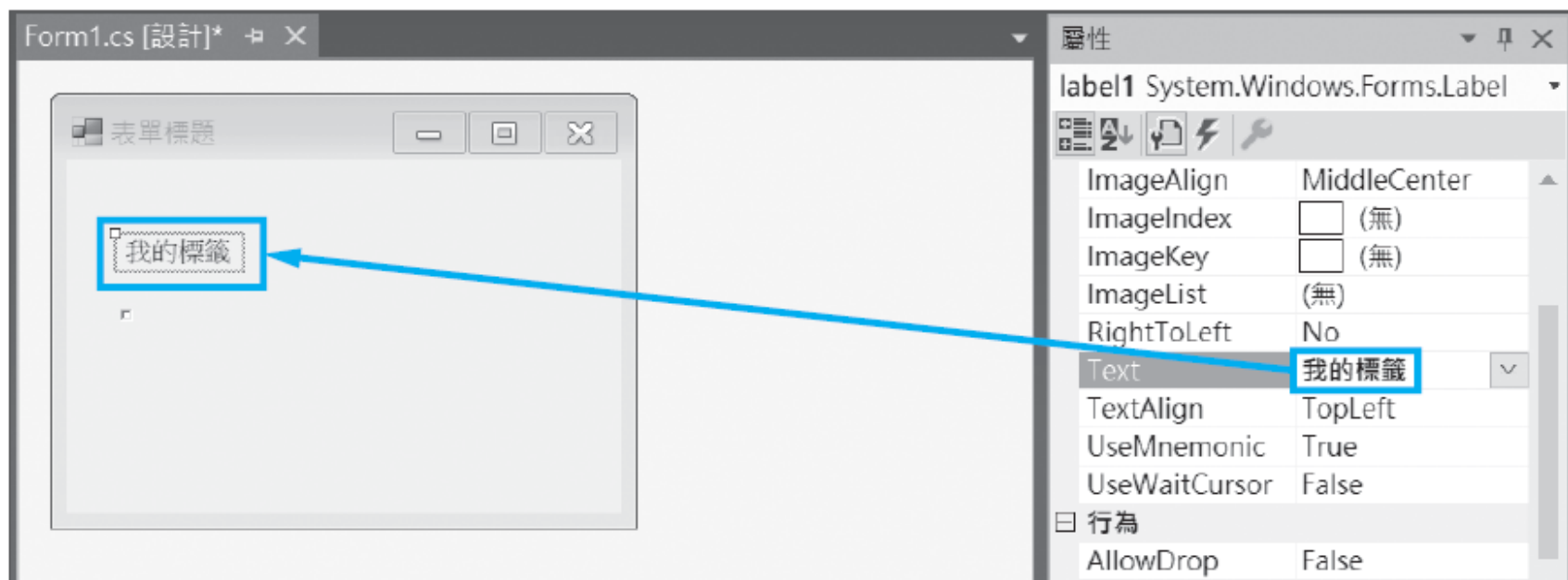


圖 1-32 設定 label1 物件的 Text 屬性為「我的標籤」

TextBox

- TextBox 用於輸入資料，使用者可以於TextBox 輸入資料，程式可以經由讀取TextBox 的屬性 Text 獲得輸入資料。
- TextBox 物件上有許多屬性可以設定，表1-5 所 列為常用的屬性。

- 表1-5 TextBox物件常用的屬性

| 物件名稱 | 屬性 | 說明 |
|----------|-----------|---|
| textBox1 | Multiline | 設定為「true」，textBox1物件才能夠多行輸入與輸出。 |
| textBox1 | Text | 設定textBox1的文字，直接於屬性視窗設定，如「我的文字方塊」，textBox1的文字會跟著修改。 |

STEP 01 由工具箱拖曳 TextBox 元件到 Form1，在 Form1 上新增一個 textBox1 物件。

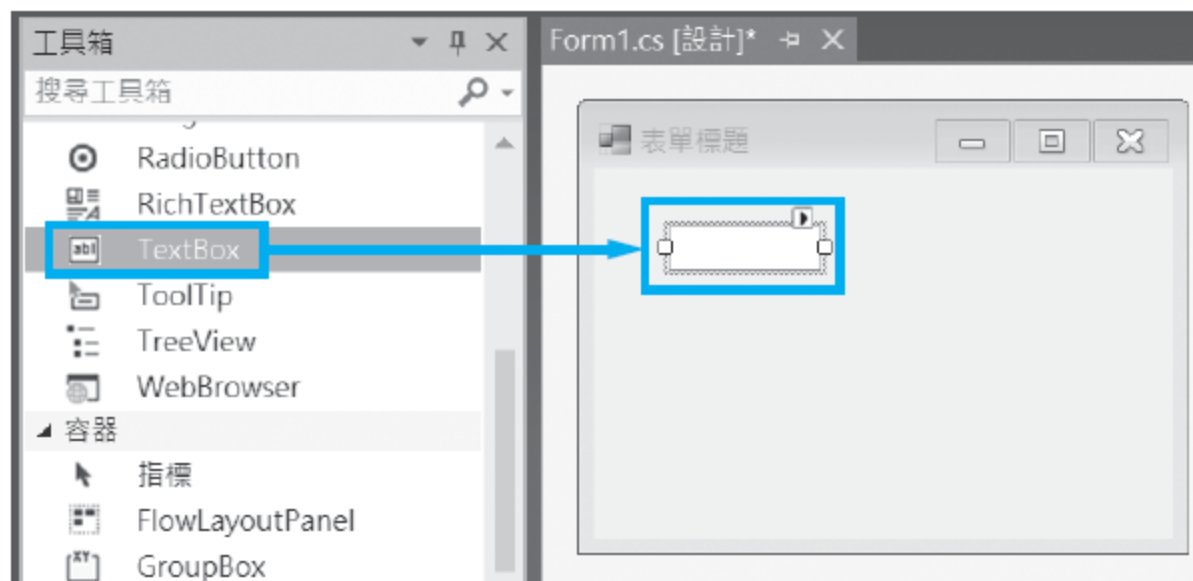


圖 1-33 在 Form1 上新增一個 textBox1 物件

STEP 02 將 textBox1 物件的屬性 Multiline 設定為「True」，拖曳 textBox1 的右下角控制點調整 textBox1 的大小，textBox1 可以輸入多行文字。

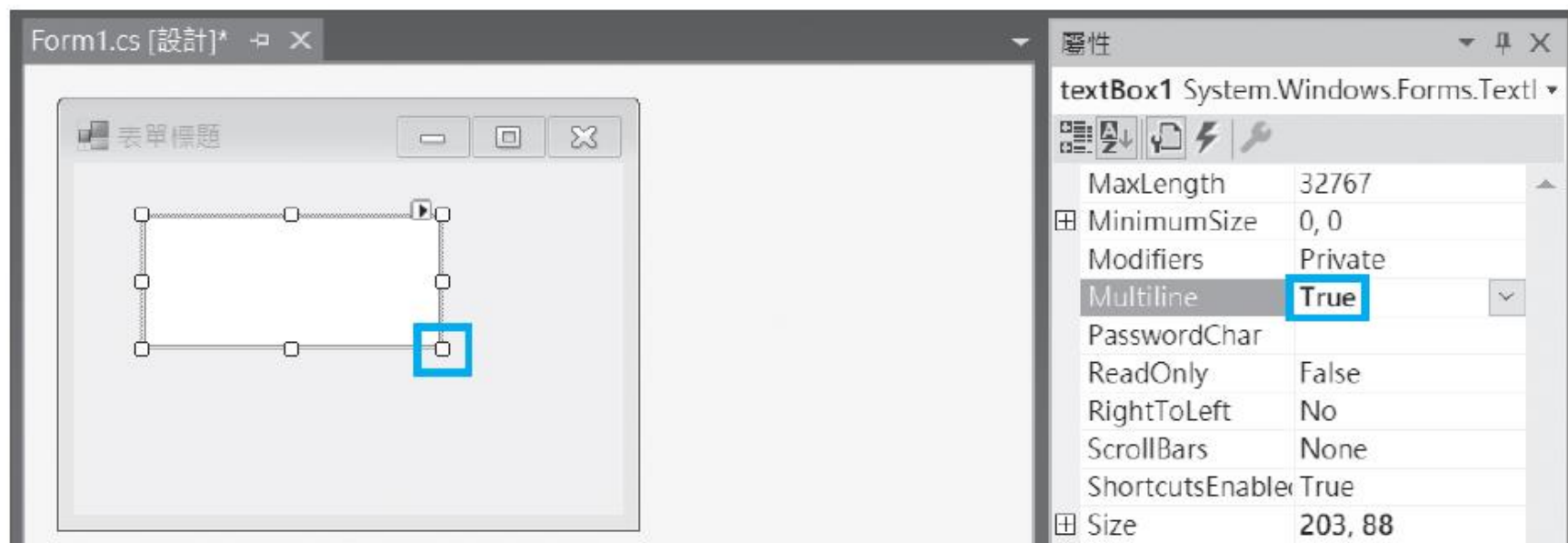


圖 1-34 調整 textBox1 物件的屬性及大小

STEP 03 設定 textBox1 的屬性 Text 設定為「我的文字方塊」。

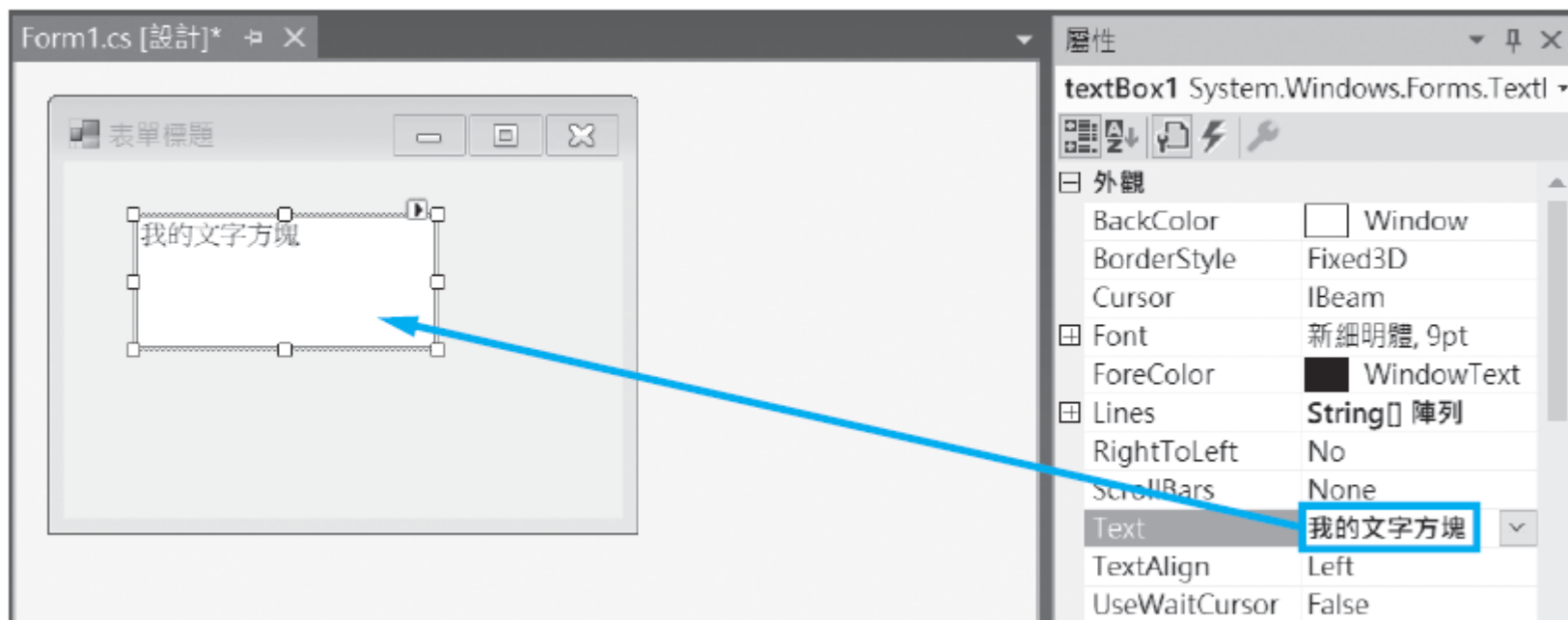


圖 1-35 設定 textBox1 的屬性 Text

1-2-5 Visual C# 視窗化程式設計的開發流程

- **STEP 01 新增專案**，Visual C# 是以「專案」為程式設計的最小單位，在專案中會自動新增第一個表單(自動命名為Form1)。
- **STEP 02** 在Form1 上使用者可以**新增控制元件**(如：Button、Label 與TextBox)，並設定元件的屬性以符合程式設計者的需求。
- **STEP 03 撰寫、測試與修正程式**，不斷重複將演算法轉換成程式碼，接著編譯執行程式，執行結果不正確就修正程式，直到程式功能符合需求。

- **STEP 04** 編譯成執行檔並執行，最後確認功能是否正確，這就是C# 程式設計的開發流程，本書所有程式範例皆遵守此開發流程。
- **STEP 05** 最後儲存專案。

範例1-1 第一個程式

- 寫一個程式，點選「請按我」按鈕，出現一個訊息視窗「這是我的第一個C# 程式」。

- STEP 01

- 點選「檔案→新增→專案」，選擇「Visual C# → Windows 傳統桌面→Windows Form App」，「名稱」部分可以輸入想要的名稱後，點選「確定」，新專案產生，同時會自動新增表單「Form1」。
- 將「Button」元件新增到「Form1」，點選工具箱中「Button」元件拖曳至「Form1」工作區，「Form1」工作區新增「button1」元件。

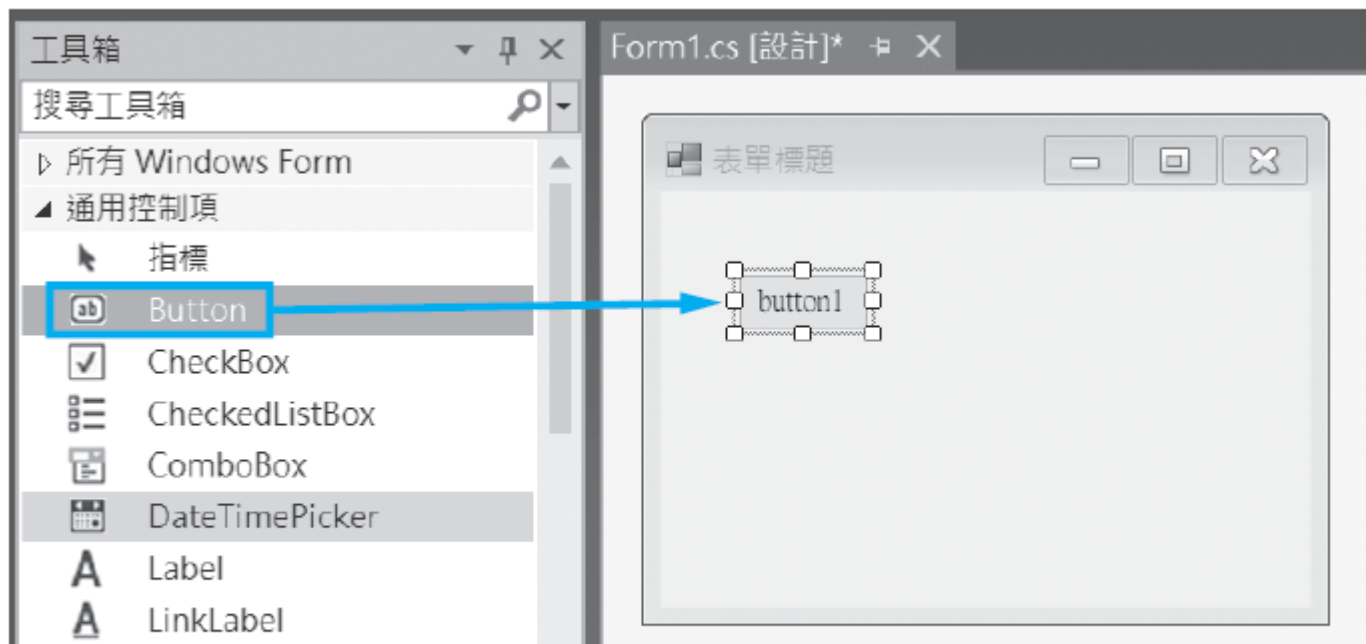


圖 1-36 「Form1」工作區新增「button1」元件

- STEP 02 屬性視窗的使用
 - 修改「button1」元件上的文字，點選「button1」元件，修改屬性視窗的Text為「請按我」，如圖1-37 所示，「button1」元件上所呈現文字也改成「請按我」。

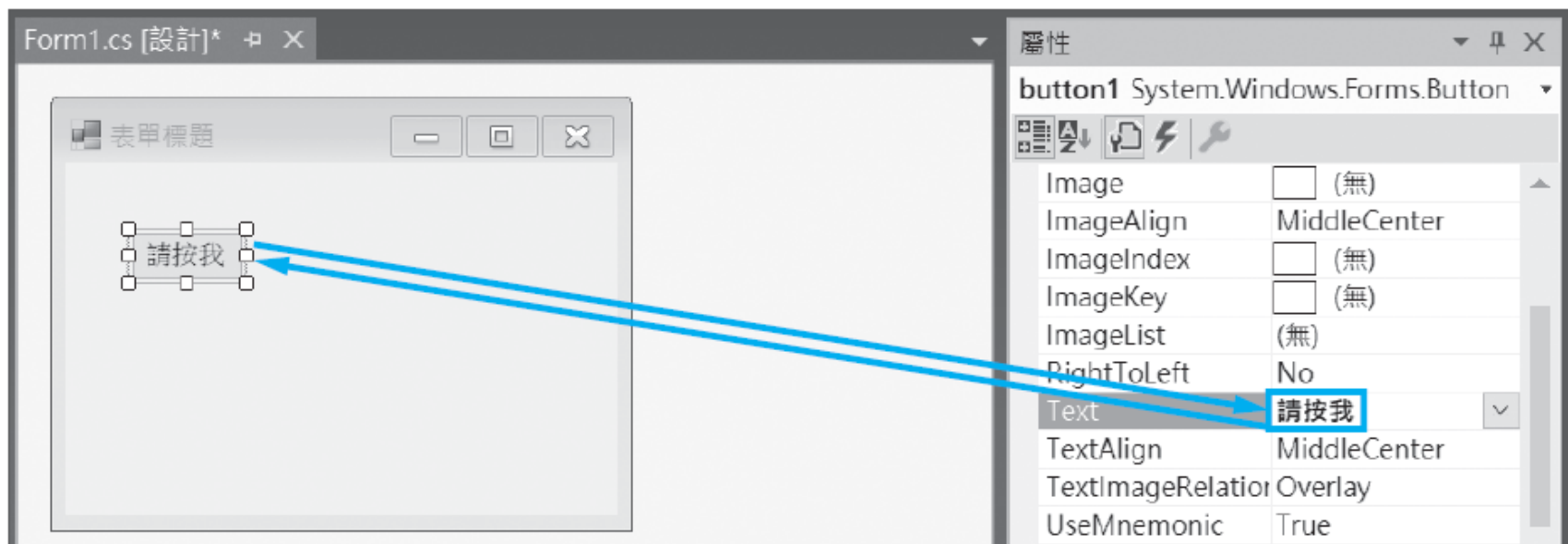


圖 1-37 修改「button1」元件上的文字

- STEP03

- 使用滑鼠點選「請按我」按鈕兩次，進入按鈕元件Click 事件程式編輯視窗，如圖1-38 所示。
 - **button1_Click** 代表事件名稱
 - **button1** 為「請按我」按鈕的元件名稱
 - **Click** 為按下按鈕的事件
 - **button1_Click** 即代表當偵測到button1元件的Click 事件時，會驅動此事件，執行事件內的程式。

- 本範例就會執行「`MessageBox.Show(" 這是我的第一個C# 程式");`」
- 就會顯示一個訊息視窗，內容為「這是我的第一個C# 程式」，因`MessageBox` 為顯示訊息視窗的系統函式，函式內的字串則是顯示在訊息視窗中的字串。



圖 1-38 使用滑鼠點選「請按我」按鈕兩次

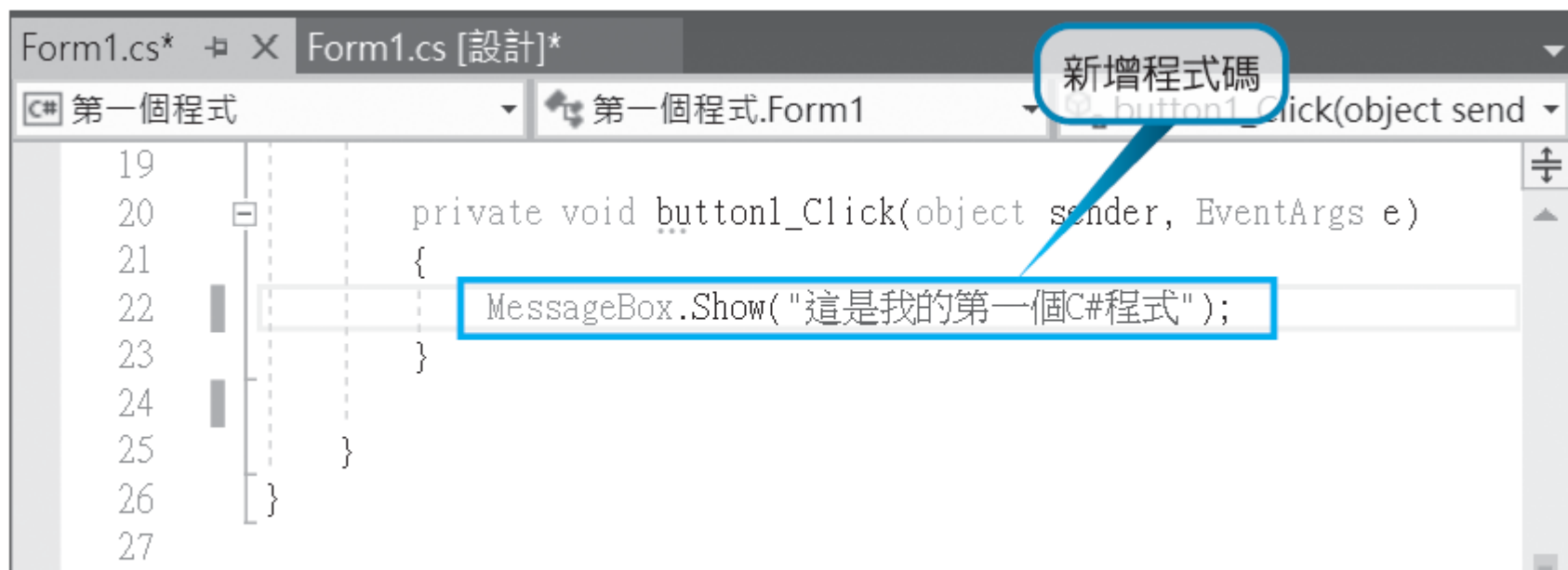



圖 1-39 進入按鈕元件 Click 事件程式編輯視窗

• STEP 04 程式的編譯與執行

一點選「 開始」進行程式的編譯、執行與除錯。

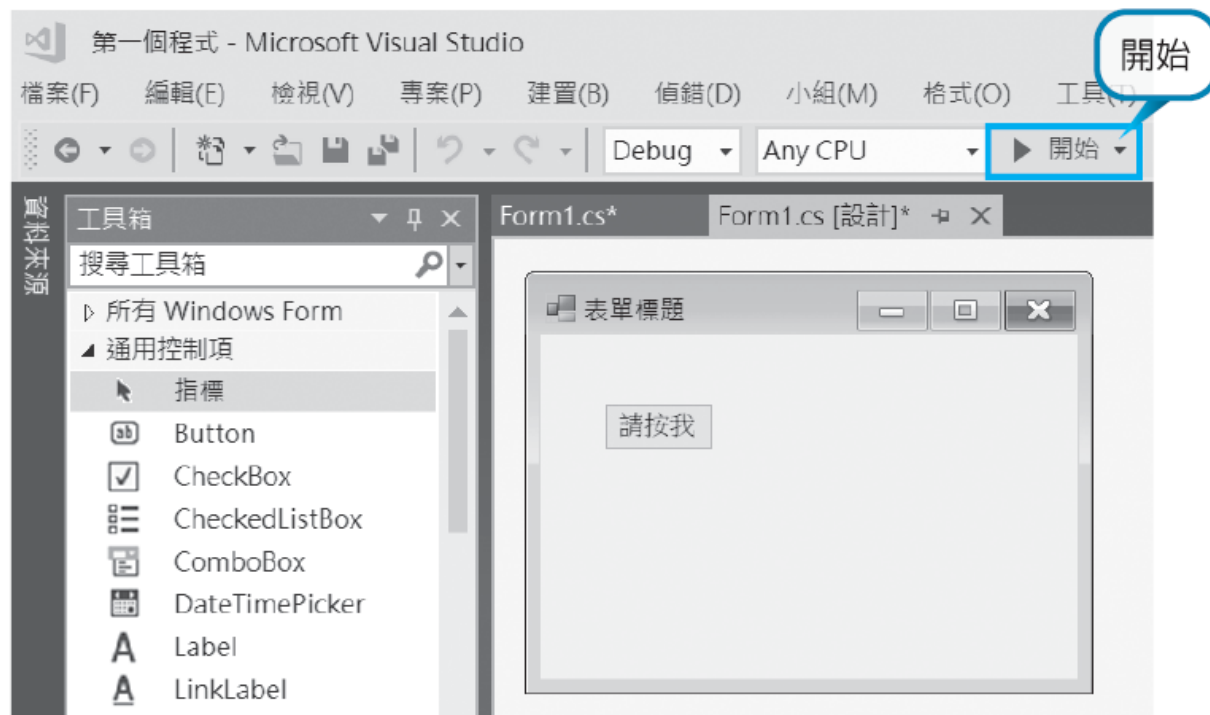
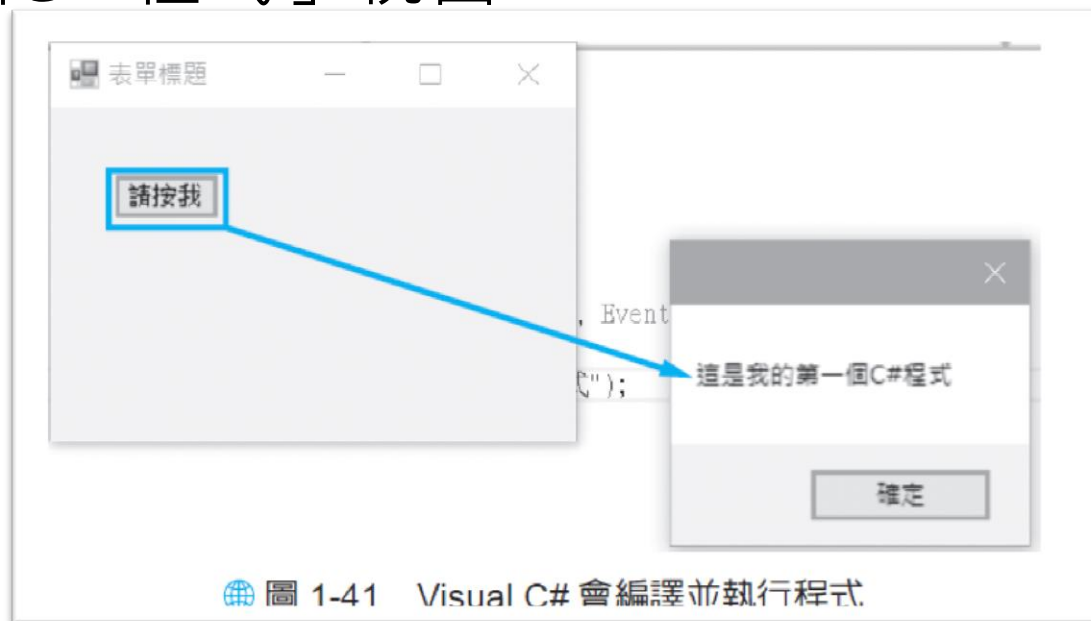


圖 1-40 點選「開始」進行程式的編譯、執行與除錯

- 一點選「▶ 開始 ▾」後，Visual C# 會編譯並執行程式，產生如下視窗，點選「請按我」按鈕，會驅動「button1_Click」事件，產生「這是我的第一個C# 程式」視窗。



- STEP 05 儲存專案

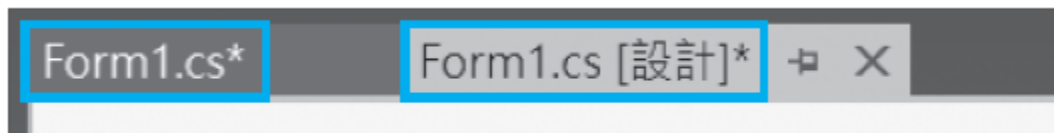
- 點選「檔案→全部儲存」，以儲存專案。

- STEP 06

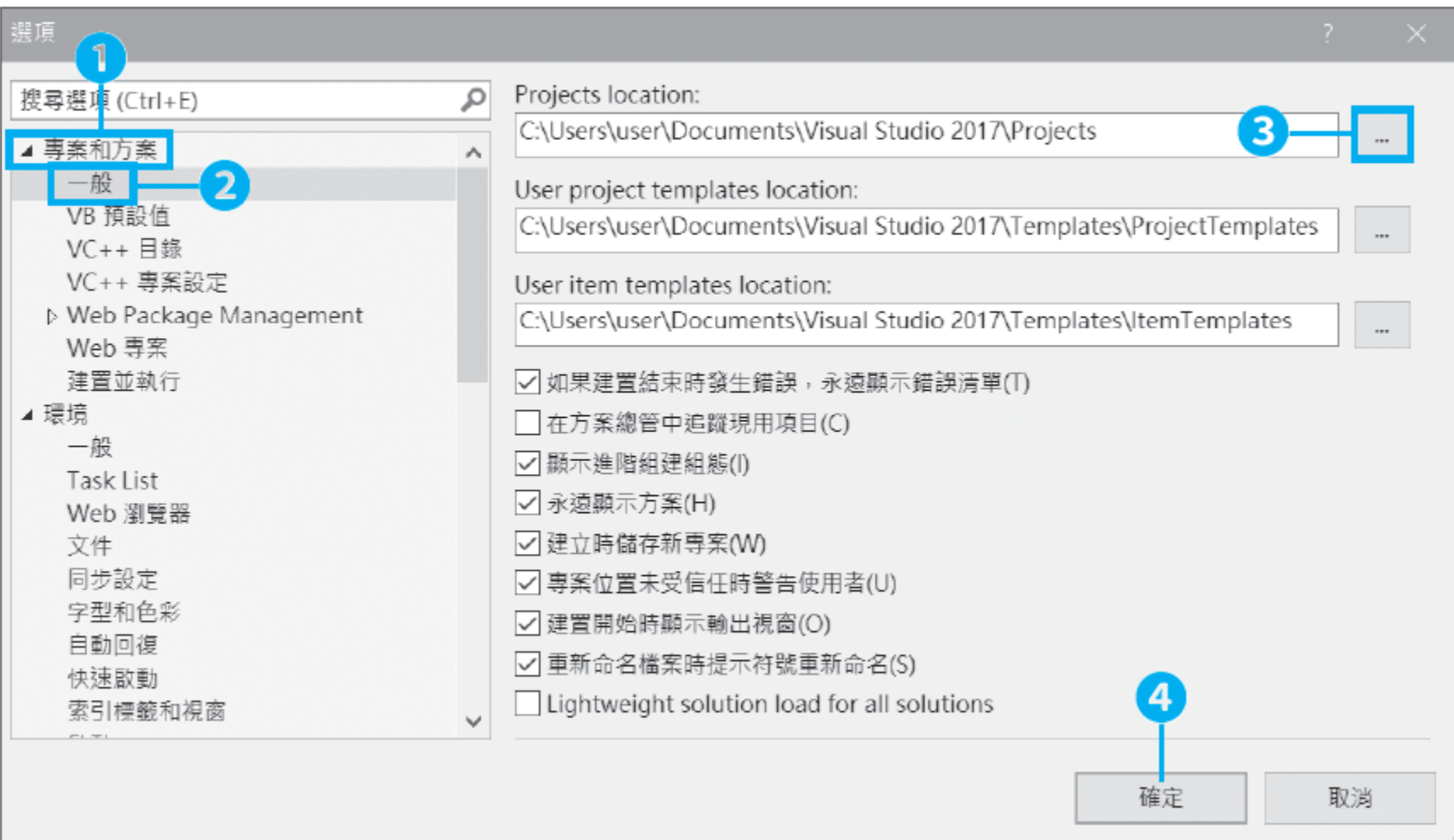
- 若要開啟已經存在的專案，點選「檔案→開啟→專案/ 方案」，選擇要開啟的專案，就可以開啟已經存在的C# 專案。

充電時間

- 1. 在檔案名稱後緊接星號(*)，如下圖，表示該檔案已經被修改，需要儲存，可以點選「檔案→全部儲存」。



- 2. 更改專案預設儲存位置
 - (1) 點選「工具→選項」，再點選「**①**專案與方案→**②**一般」。
 - (2) 修改專案位置，**③**點選右邊瀏覽資料夾圖示「」，設定專案資料夾，**④**最後按下「確定」。



1-2-6 說明文件資源的獲得

- 在Visual C# 中選擇元件或程式碼中關鍵字，按鍵盤「F1」功能鍵，會自動開啟瀏覽器到MSDN 找出相關說明網頁。
- 以「Button」元件為例，點選工具箱中「Button」元件，按下鍵盤「F1」功能鍵。

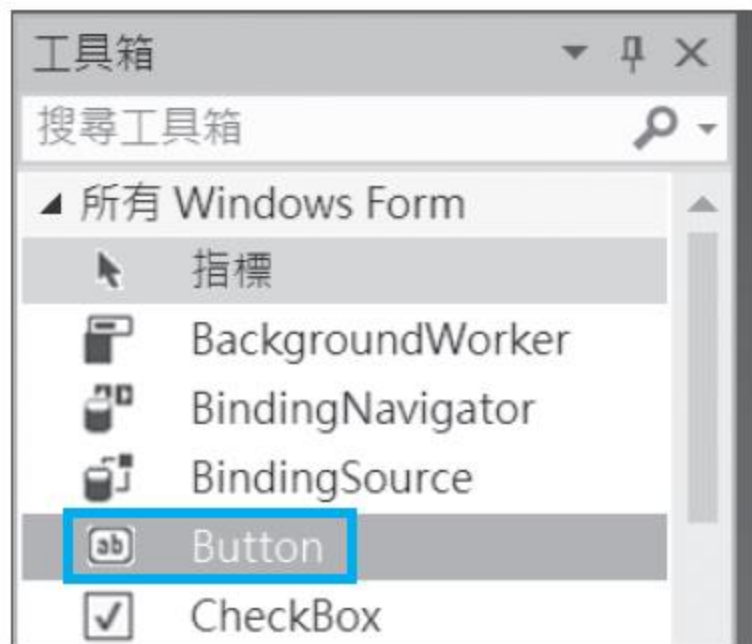


圖 1-42 點選工具箱中「Button」元件
並按下鍵盤「F1」功能鍵

- 會自動打開瀏覽器，找尋「Button」元件相關說明網頁。

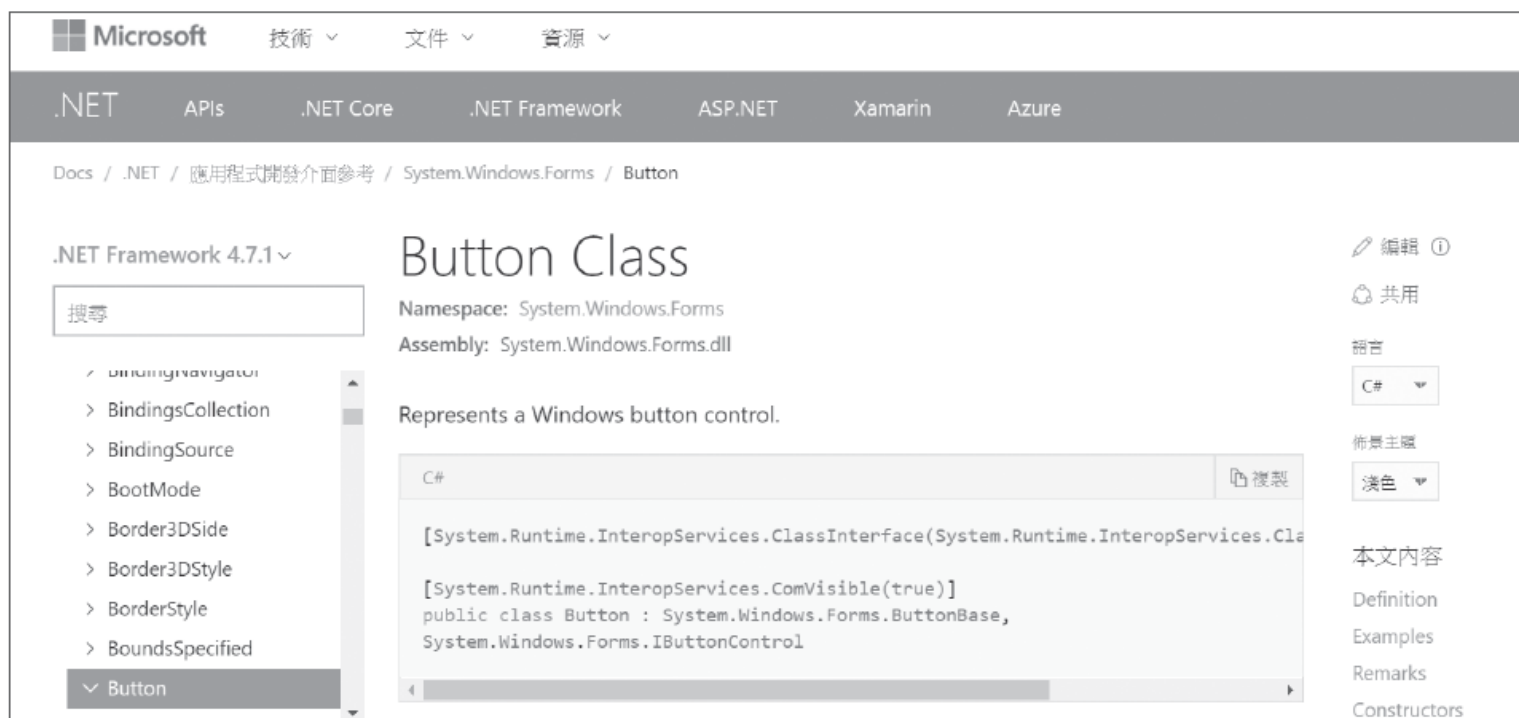



圖 1-43 自動開啓瀏覽器搜尋相關說明網頁

1-3 物件導向程式設計的基礎概念

- 物件
 - C# 有許多內建好的物件，如表單(Form)、按鈕(Button)、文字輸入(TextBox) 與標籤(Label) 等
 - 使用物件的好處是使用者不需要知道物件實作的細節，只要知道物件提供的屬性、事件與方法，經由修改屬性、呼叫事件與方法達成想要的功能。

- 屬性
 - 屬性是物件的外觀與功能設定，如物件的大小、位置、顯示文字與顏色等，C# 對於不同物件提供不同的屬性設定，讓物件符合程式設計的需要。

 表 1-6 設定格式

| 屬性設定格式 | 範例 |
|-------------|----------------------|
| 物件.屬性 = 屬性值 | label1.Text=" 我的標籤 " |

- **事件**

- 事件為使用者與程式互動的基礎，如按下 Button 物件、外部鍵盤輸入、滑鼠移動等都會驅動事件，當發生此事件就會呼叫對應的事件函式處理，該事件內可允許使用者撰寫程式，經由程式的執行達成所需的功能。

- 方法

- 方法也是一種函式，可以是內建函式，也可以是自訂函式，與事件的差異在於，事件是由外部鍵盤、滑鼠等所驅動，而方法經由程式呼叫使用，如Math 物件提供Sin 函式，可以計算數學中三角函式的Sin 值。

範例1-2 簡單加法計算機

- 寫一個程式允許使用者輸入兩個數字，點選「相加」按鈕，執行結果顯示於表單上。
- 程式結果預覽
 - 於textBox1 輸入5，textBox2 輸入11，按下「相加」按鈕，結果顯示在label3。

ch1\簡單加法計算機

The screenshot shows a Windows application window titled 'Form1'. Inside the window, there is a simple addition calculator interface. It consists of two text boxes for input, a plus sign, a button labeled '相加' (Add), an equals sign, and a label for the result. The first text box contains the number '5', the second text box contains '11', and the result label displays '16'. A blue callout bubble points to the '16' in the result label, with the text 'label3' inside it.

表單配置

Form1

label1

label2

label3

textBox1

button1

textBox2



Form1

Label1

Button1

Label2

label3

屬性設定

| 物件 | 屬性 | 設定值 |
|---------|------|-----|
| label1 | Text | + |
| label2 | Text | = |
| button1 | Text | 相加 |

STEP 01 點選 label1，在屬性視窗設定屬性 Text 為「+」。

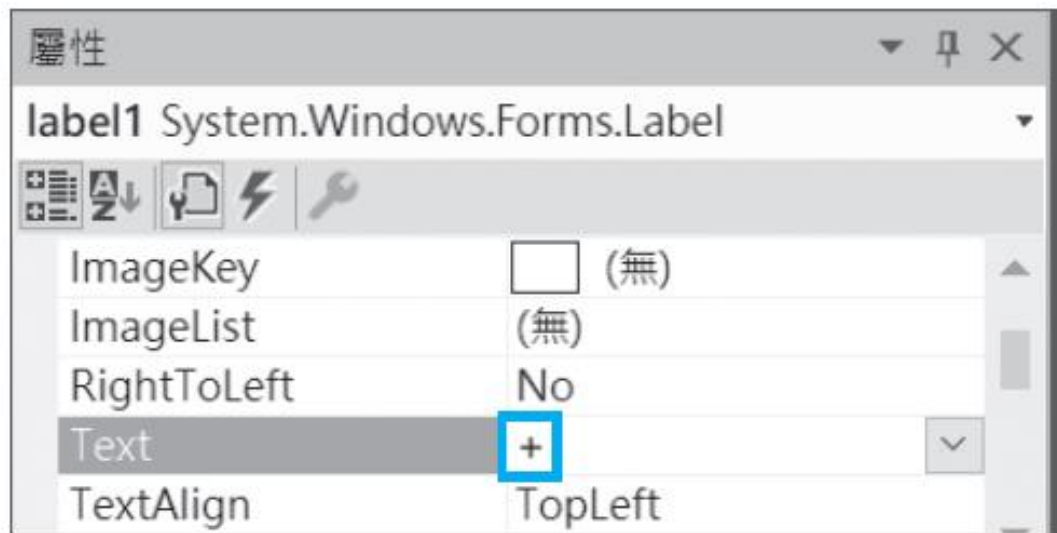


圖 1-44 設定 label1 的 Text 屬性

STEP 02 點選 label2，在屬性視窗設定屬性 Text 為「=」。

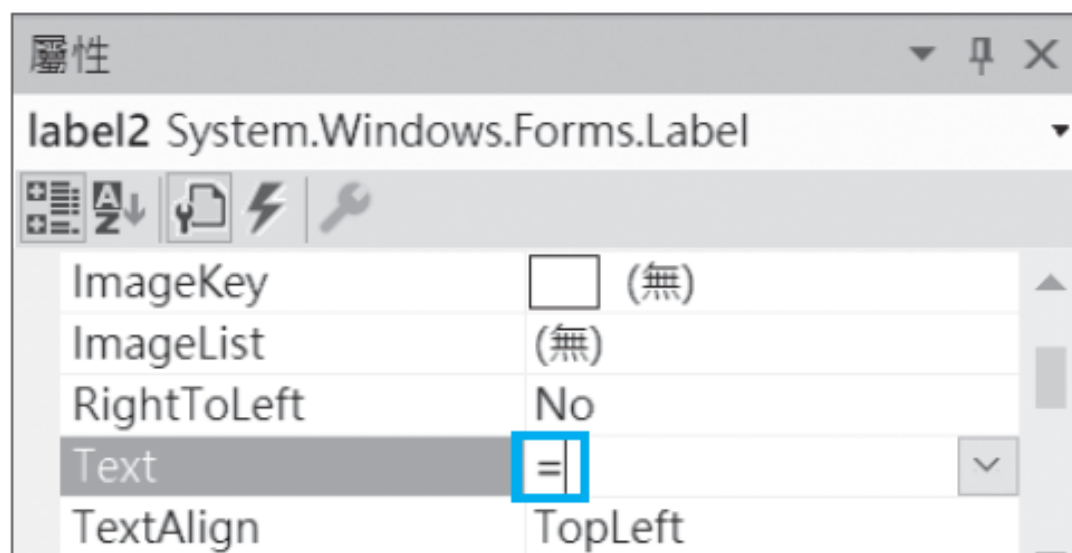


圖 1-45 設定 label2 的 Text 屬性

STEP 03 點選 button1，在屬性視窗設定屬性 Text 為「相加」。

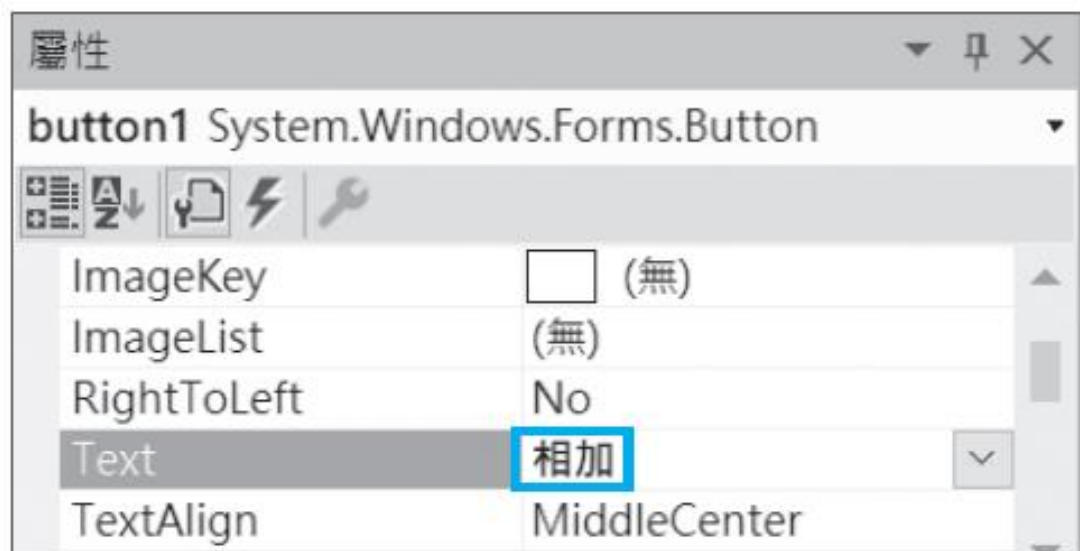


圖 1-46 設定 label3 的 Text 屬性

| 行號 | 程式碼 |
|----|--|
| 20 | private void button1_Click(object sender, EventArgs e) |
| 21 | { |
| 22 | int sum = Convert.ToInt32(textBox1.Text) + |
| 23 | Convert.ToInt32(textBox2.Text); |
| 24 | label3.Text = sum.ToString(); |
| | } |

程式解說

- 利用`Convert.ToInt32`函式將textBox1與textBox2所輸入字串轉成數值，再將數值相加的結果儲存入變數sum
- 將變數sum轉換成字串設定到label3的Text屬性，也就是當按下「相加」按鈕，會驅動button1_Click事件，將textBox1與textBox2相加，結果顯示於label3。