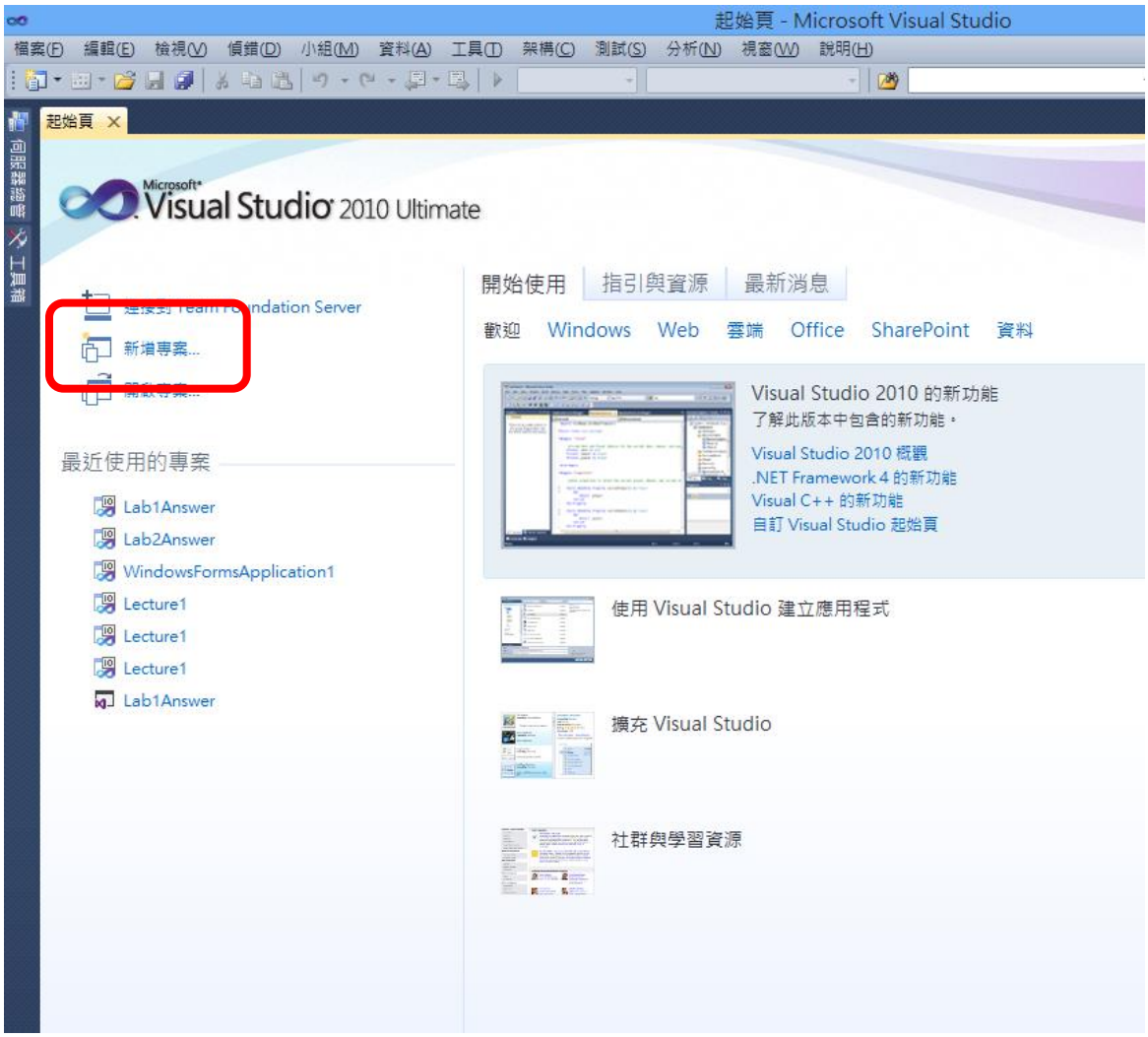
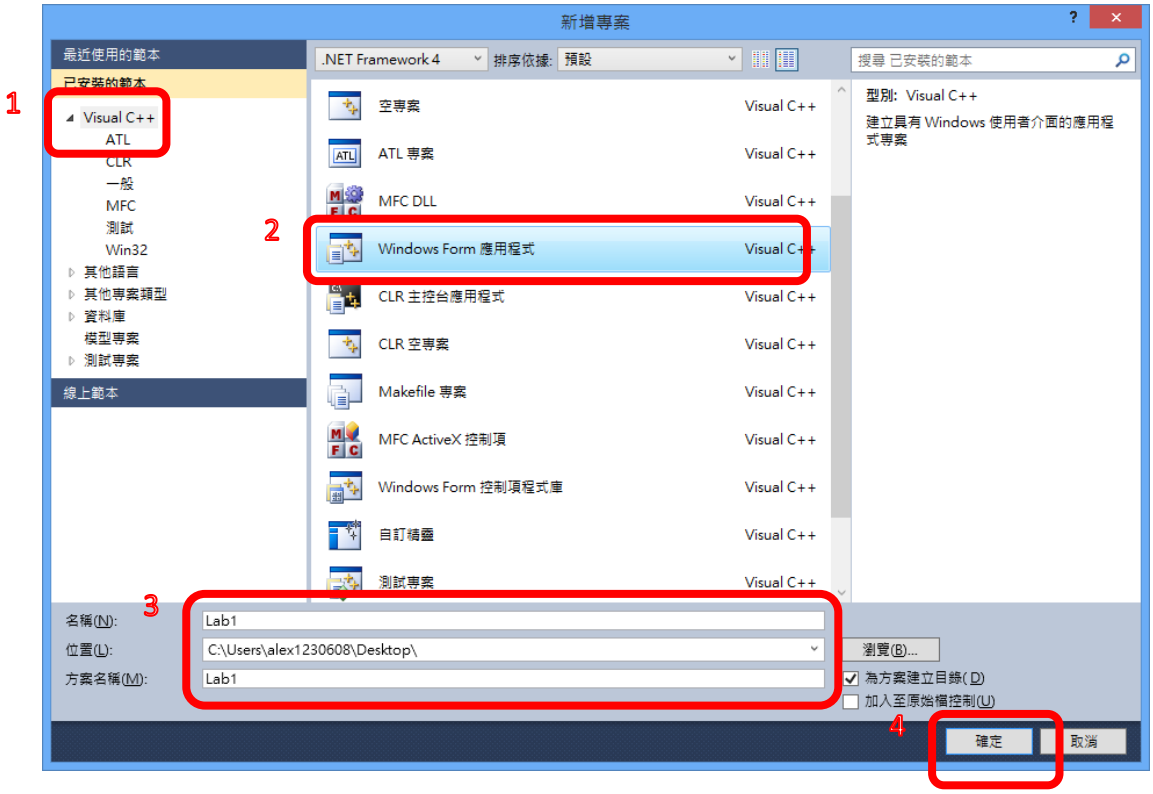
實驗一

如何開始

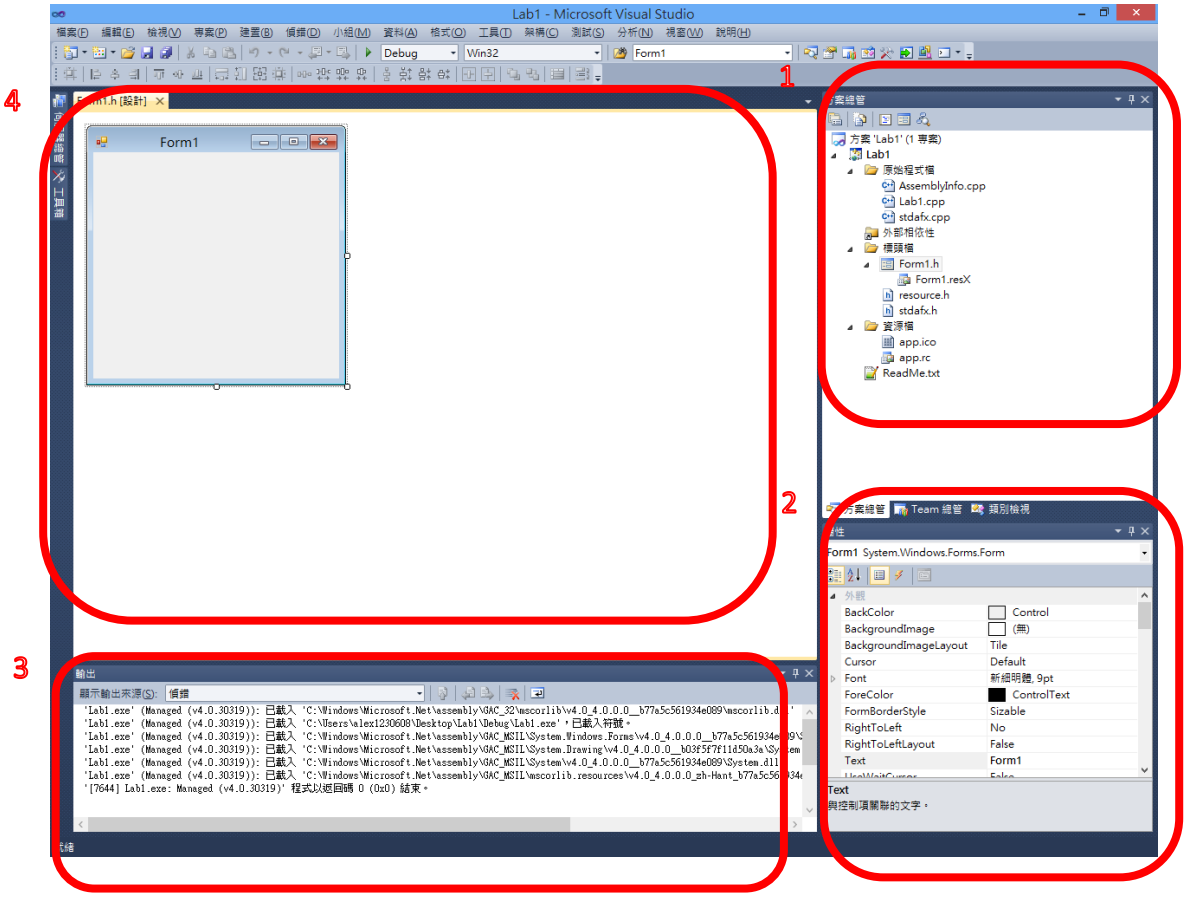
1. 開新專案



2. Visual C++ -> Windows Form 應用程式 -> 方案專案位置與命名 -> 確定



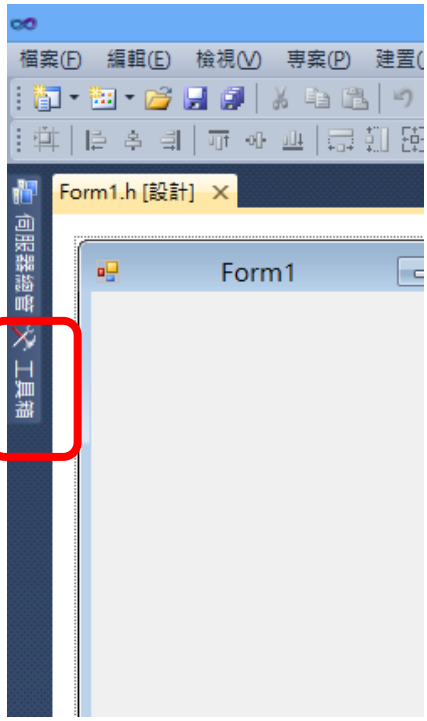
視窗介紹

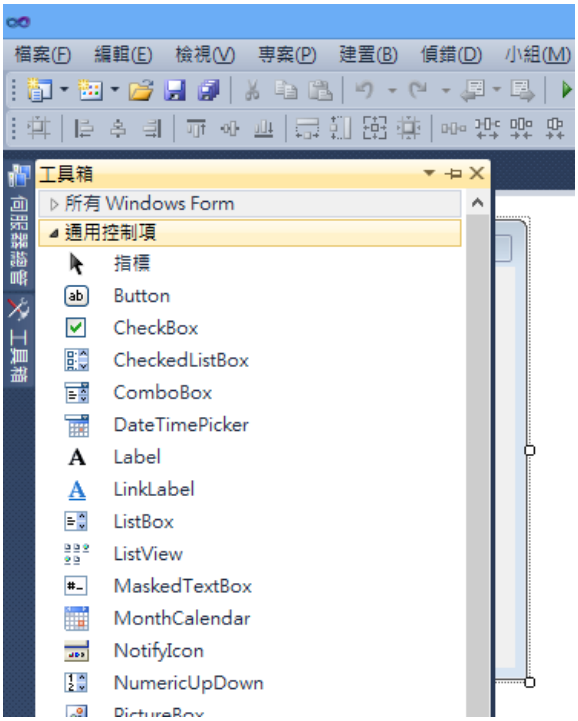


1. 方案總管：多個檔案的 coding 時，常需要使用此窗格找尋檔案
2. 屬性：可對元件進行屬性的設定
3. 輸出：偵錯時常可從此處看到錯誤的原因，是 Debug 時常需要檢視的是地方
4. 主要工作區：開啟檔案後，會於此區以芬頁的方式顯示

工具箱介紹

於方案總管開啟 Form1.h 檔時(點兩下)，顯示的會是設計頁，與此種頁面下左側常可看到工具箱選項，開啟即可進行 GUI 介面的設計。

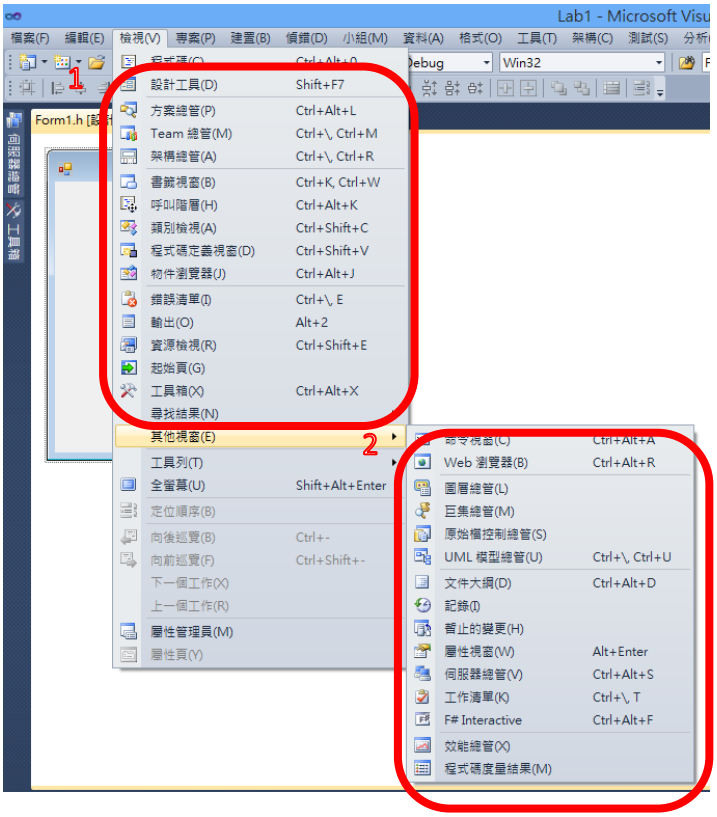




包括工具箱的所有窗格名稱，請盡量記起來，方便不見時，能用以下方法找到

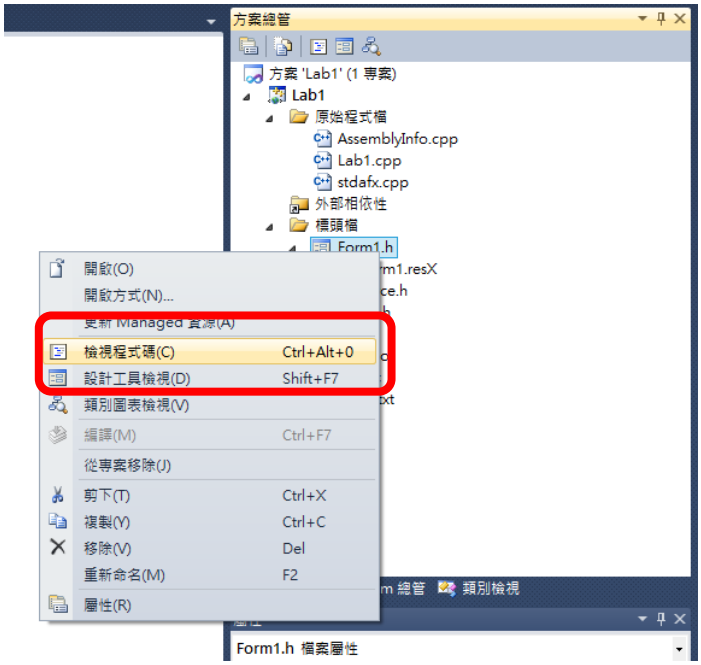
1. 檢視 -> 點選需要的窗格
   (包含:方案總管、輸出、工具箱)

2. 檢視 -> 其他視窗 -> 點選需要的窗格
   (包含：屬性視窗)



開始 Coding

1. 打開程式編輯區：於方案總管欲編輯的檔案上(Form1.h)按滑鼠右鍵，選擇檢視程式碼即可看到該檔案程式編輯區



2. 程式編輯區介紹：編輯如 Form1.h 的 GUI 相關程式時，請勿、切勿、絕對不要更改以下未標記的區域，否則很難改回能夠執行的程式，而成為無藥可救的屍體。

```
#pragma once
namespace Lab1 {
	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;
	/// <summary>
	/// Form1 的摘要
	/// </summary>
	public ref class Form1 : public System::Windows::Forms::Form
	{
	public:
		Form1(void)
		{
```

```
            InitializeComponent();
            //
            //TODO: 在此加入建構函式程式碼
            //
        }

    protected:
        /// <summary>
        /// 清除任何使用中的資源。
        /// </summary>
        ~Form1()
        {
            if (components)
            {
                delete components;
            }
        }

    private:
        /// <summary>
        /// 設計工具所需的變數。
        /// </summary>
        System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
        /// <summary>
        /// 此為設計工具支援所需的方法 - 請勿使用程式碼編輯器
        /// 修改這個方法的內容。
        /// </summary>
        void InitializeComponent(void)
        {
            this->components = (gcnew System::ComponentModel::Container());
            this->Size = System::Drawing::Size(300, 300);
            this->Text = L"Form1";
            this->Padding = System::Windows::Forms::Padding(0);
            this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

        }
```

Constructor
Form 剛建立時會執行的區塊

自訂變數區
可設置一些程式需要的變數
(如：int countOfClick, …)

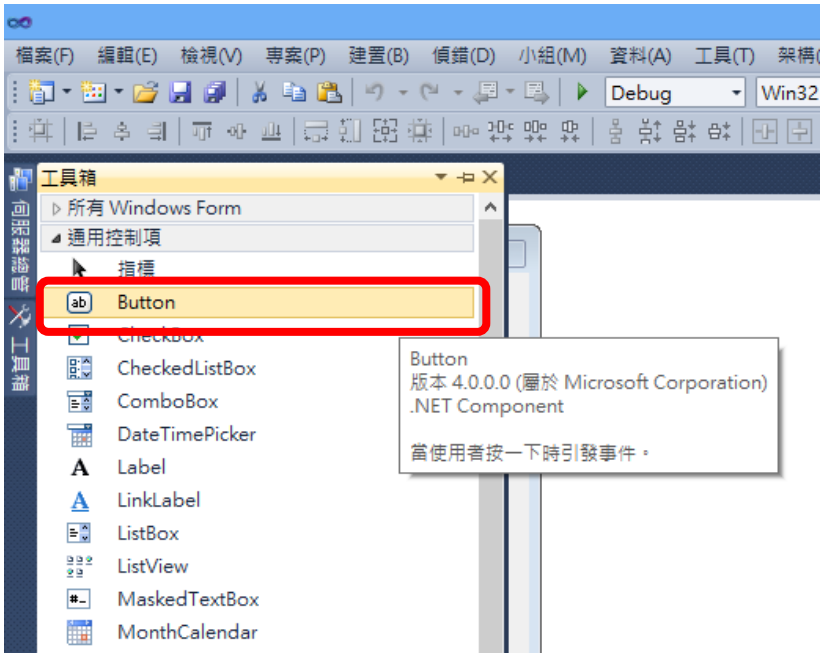
```
#pragma endregion
    };
}
```

除了上面標記的兩塊區域可以編輯之外，還可自行增加函數定義；或是透過 GUI 設計頁面所設定的 Event 的 Handler(如：Click)，都可以自行定義 function 內部

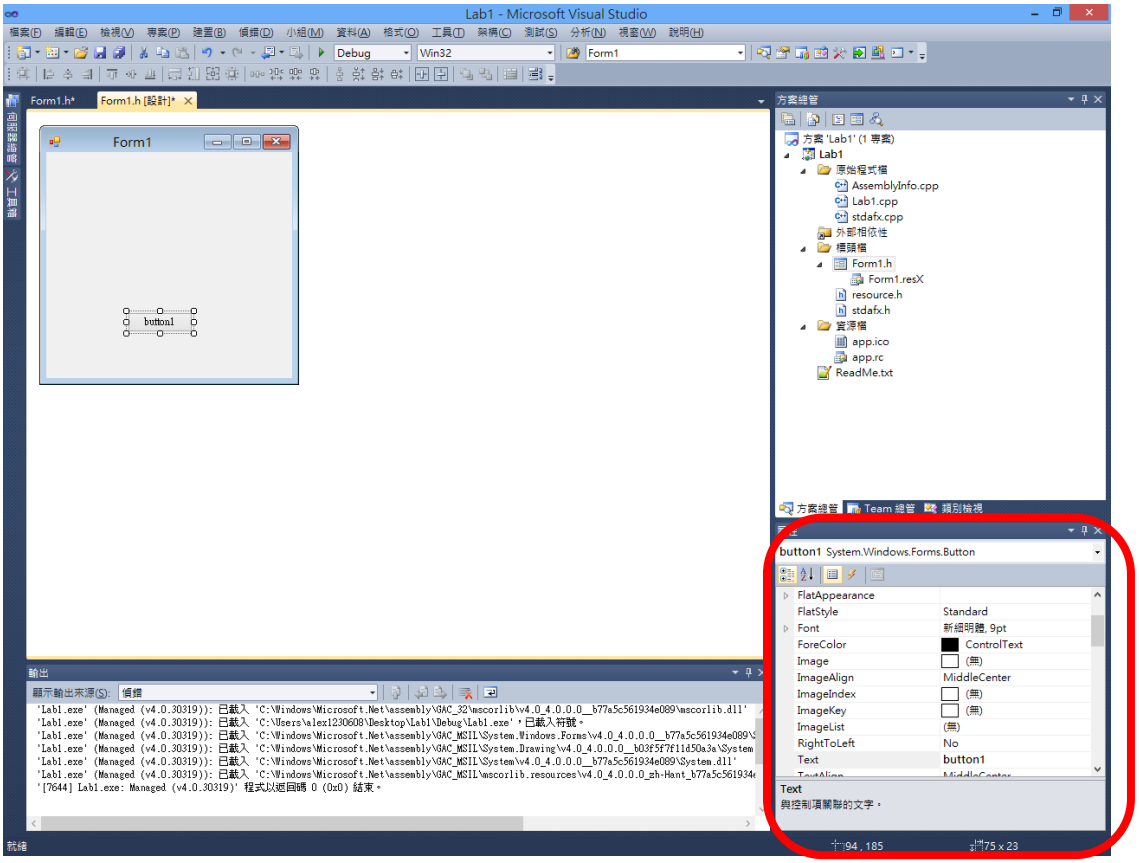
Event 介紹(新增)

GUI 介面常需要與使用者互動，因此常常需要因應不同的動作，程式必須產生不同的應變。
而使用者的動作就是 Event
程式所要產生的應變就是 Handler

1. Event 來源的 GUI 元件：以 Button 為例，於工具箱中找到 Button，移至 Form1 上。

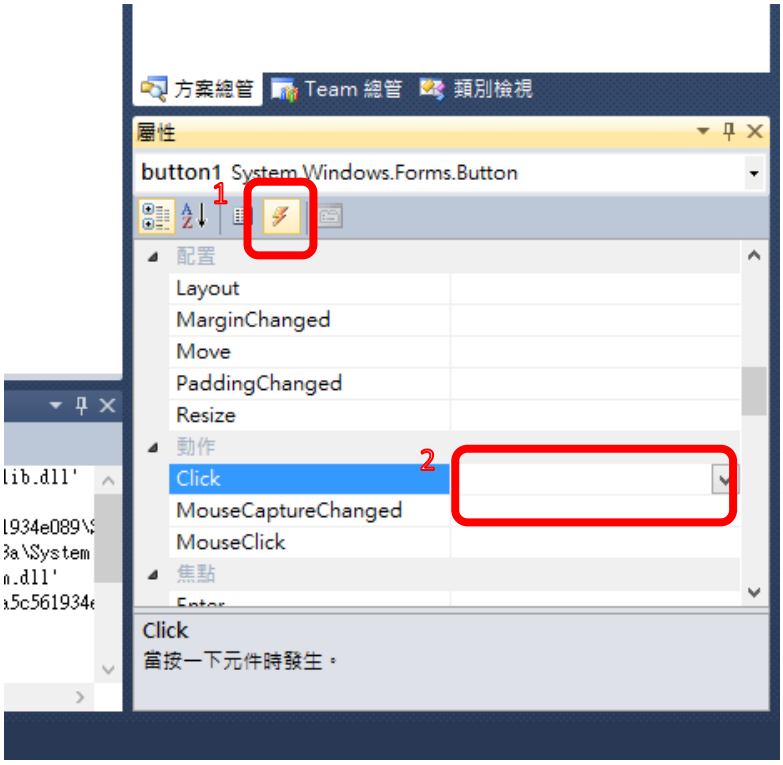


2. 點選設計頁上的 Button，即可於屬性窗格上，看到該 Button 的屬性設定

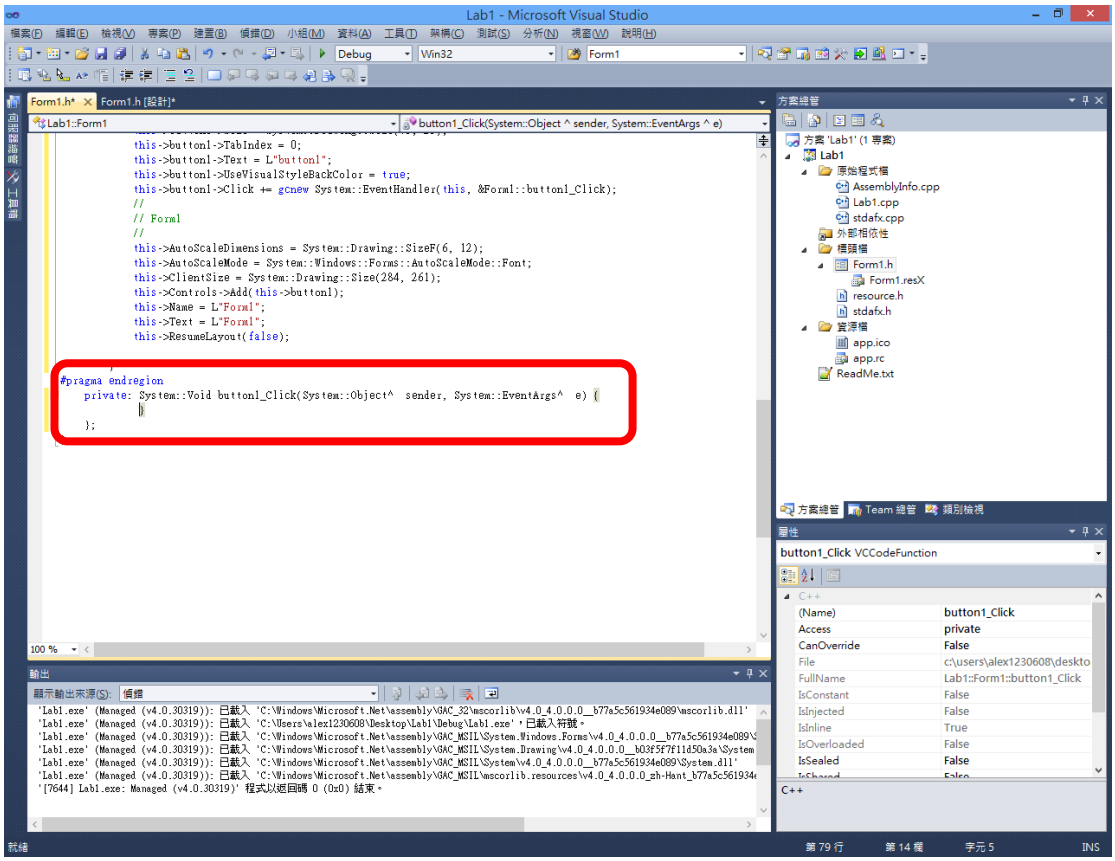


3. 新增事件 Event：選擇閃電按鈕

-> 於欲產生的應變措施的事件的右邊空白處點兩下



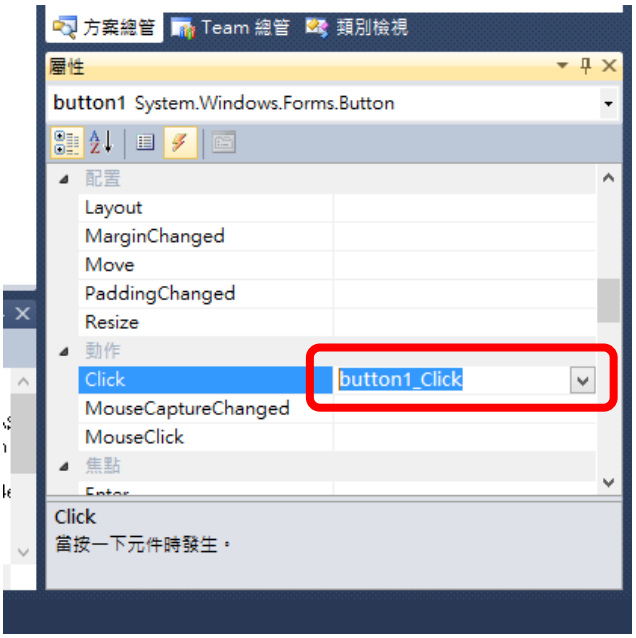
4. 此時也會於主要工作區，也會跳出該事件 Event 的 Handler 定義，我們就可以在此區打上我們要做的事件處理



Event 介紹(刪除)

刪除 Event 常會有人刪除後就無法 Compile 成功，正確步驟如下

1. Delete 整個 EventHandler 的 Function 定義(即上面那張圖紅色區域)
2. (上面步驟大家都很直覺的會去刪除，但常漏掉第 2 步驟，而造成 Compile 不過。)於設計頁上，選擇該 Event 來源的 GUI 元件，找到屬性中的事件頁(閃電)，找到對應的 Event，將右側文字刪除，如此一來才真的取消此事件的設定。



注意事項

1. 於專案中，任何名稱，如檔案的命名等，請都遵守寫程式時的變數名稱命名原則。(如：開頭不用數字、不用中文…等等)
2. 除非對 Visual 的 GUI 設計很熟，否則不要直接複製元件。(如：不要複製 Button)