

# BuildSchool先修課程 day1 & day2

## 環境準備 & C#練習

講師: 邱秉澄(Jimmy)

日期: 2021/03/09

# 下載 Visual Studio

- 下載 Visual Studio 2019 Community (社群)

<https://visualstudio.microsoft.com/zh-hant/downloads/>

# 安裝 Visual Studio

- 安裝選擇

- ◆ Web與雲端

- ASP.NET與網頁程式開發
    - Azure開發

- ◆ 傳統型與行動裝置

- .NET桌面開發
    - 通用Windows平台開發

- ◆ 其他工具組

- 資料儲存和處理
    - Visual Studio擴充功能開發
    - .NET Core跨平台開發

# 安裝 Visual Studio - 1

正在修改 - Visual Studio Community 2019 - 16.5.4

工作負載    個別元件    語言套件    安裝位置

Web 與雲端 (4)

 <b>ASP.NET 與網頁程式開發</b> 使用 ASP.NET Core、ASP.NET、HTML/JavaScript 及容器 (包括 Docker 支援) 建立 Web 應用程式。	<input checked="" type="checkbox"/>	 <b>Azure 開發</b> 用於使用 .NET Core 和 .NET Framework 開發雲端應用程式及建立資源的 Azure SDK、工具及專案，並包含應用程式...	<input checked="" type="checkbox"/>
 <b>Python 開發</b> 對 Python 進行編輯、偵錯、互動式開發及原始檔控制。	<input type="checkbox"/>	 <b>Node.js 開發</b> 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。	<input type="checkbox"/>

傳統型與行動裝置 (5)

 <b>.NET 桌面開發</b> 使用 C#、Visual Basic 及 F#，利用 .NET Core 和 .NET Framework 建置 WPF、Windows Forms 與主控台應用程式...	<input checked="" type="checkbox"/>	 <b>使用 C++ 的桌面開發</b> 使用您選擇的工具 (包括 MSVC、Clang、CMake 或 MSBuild)，建置適用於 Windows 的新式 C++ 應用程式。	<input type="checkbox"/>
 <b>通用 Windows 平台開發</b> 使用 C#、VB 或選用 C++，來建立適用於通用 Windows 平台的應用程式。	<input checked="" type="checkbox"/>	 <b>使用 .NET 進行行動開發</b> 使用 Xamarin 建置適用於 iOS、Android 或 Windows 的跨平台應用程式。	<input type="checkbox"/>

# 安裝 Visual Studio - 2

正在修改 - Visual Studio Community 2019 - 16.3.4

工作負載   個別元件   語言套件   安裝位置

遊戲 (2)

 <p>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。</p> <input type="checkbox"/>	 <p>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。</p> <input type="checkbox"/>
---	--

其他工具組 (6)

 <p>資料儲存和處理 使用 SQL Server、Azure Data Lake 或 Hadoop 連線、開發及測試資料解決方案。</p> <input checked="" type="checkbox"/>	 <p>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 與 F#。</p> <input type="checkbox"/>
 <p>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。</p> <input checked="" type="checkbox"/>	 <p>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。</p> <input type="checkbox"/>
 <p>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。</p> <input type="checkbox"/>	 <p>.NET Core 跨平台開發 使用 .NET Core、ASP.NET Core、HTML/JavaScript 及容器 (包括 Docker 支援)，建置跨平台應用程式。</p> <input checked="" type="checkbox"/>

暫停一下  
來聊聊天，關於未來上課

# 初學者害怕的幾件事

## 1. 害怕老師 or 助教，不敢發問

- 問了會被笑
- 問了會被罵

## 2. 專有名詞越來越多開始暈

- 數學 ->  $\sin$ 、 $\cos$ 、 $\log$ .....
- 程式 -> ASP.NET、.NET Core、JavaScript、Go Lang、Python.....

## 3. 找到屬於自己的讀書方式，沒有最好只有最適合

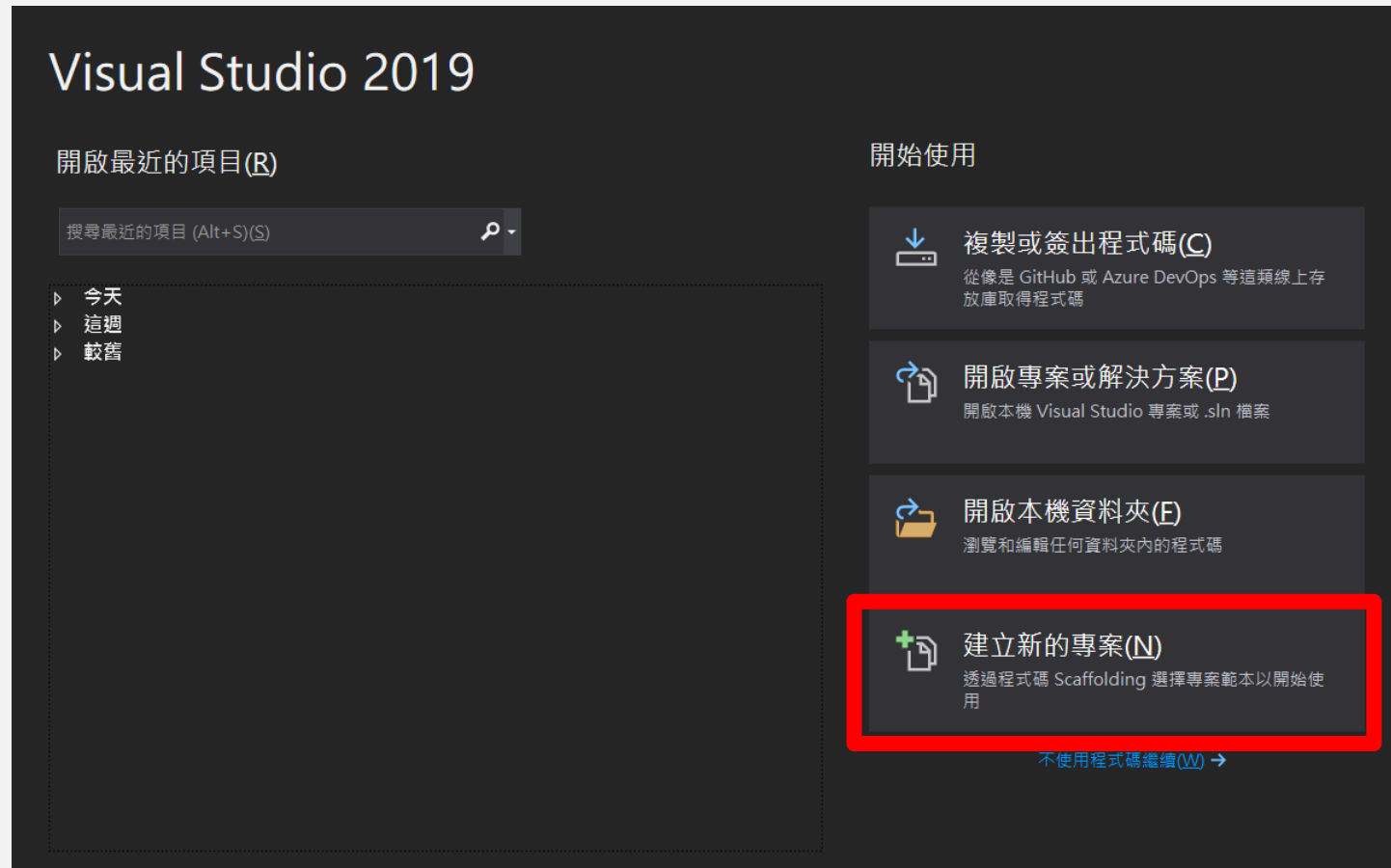
- 寫筆記的方式
- 練習的方式

# 第一個程式 - Hello World

- 新增方案 & 專案
- 方案名稱 BuildSchoolLab
- 專案名稱 Lab1\_HelloWorld
- 專案類型 主控台應用程式(.Net Framework)



# 第一個程式 - Hello World - 1



# 第一個程式 - Hello World - 2



# 第一個程式 - Hello World - 3

設定新的專案

主控台應用程式 (.NET Framework) C# Windows 主控台

專案名稱(N)

Lab1\_HelloWorld

位置(L)

C:\Users\USER\Desktop\BS2020\

解決方案名稱(M) ⓘ

BuildSchoolLab

架構(E)

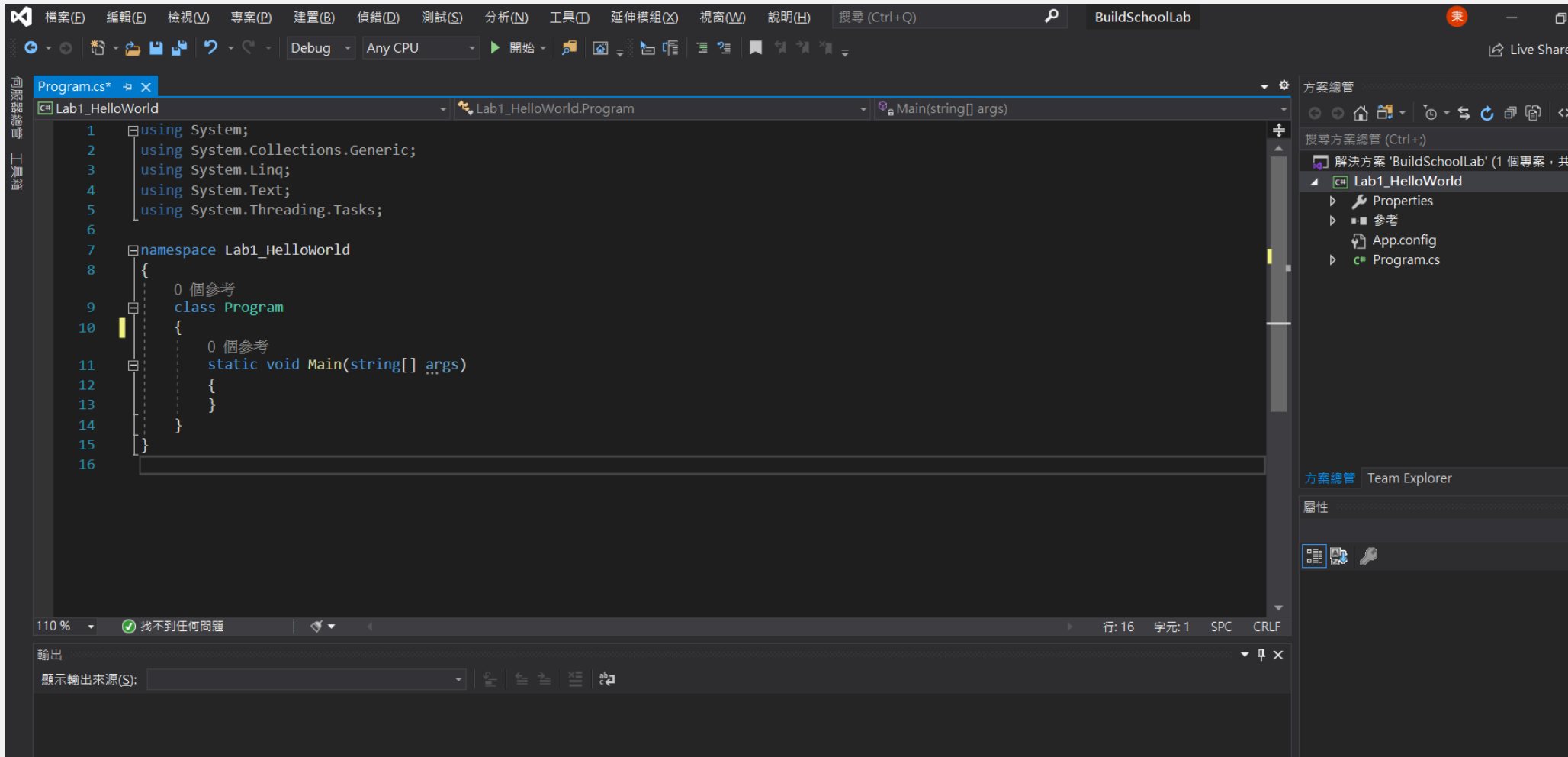
.NET Framework 4.7.2

上一步(B) 建立(C)

專案名稱

方案名稱

# 第一個程式 - Hello World - 4 (Done)



# 第一程式 - HelloWorld - 筆記

- Console

- Console.Write()
- Console.WriteLine()
- Console.ReadLine()

# 第二個程式 - 宣告

- 新增專案
- 目標方案名稱 BuildSchoolLab
- 專案名稱 Lab2\_Declare
- 專案類型 主控台應用程式(.Net Framework)

# 第二個程式 - 宣告 - 1

```
Program.cs x
Lab2_Declare Lab2_Declare.Program Main(string[] args)
6
7 namespace Lab2_Declare
8 {
9     0 個參考
10    class Program
11    {
12        0 個參考
13        static void Main(string[] args)
14        {
15            string A = "Hello";
16            int B = 123;
17            int[] C = { 1, 2, 3 };
18
19            Console.WriteLine(A);
20            Console.WriteLine(B);
21            Console.WriteLine(C);
22
23            Console.ReadLine();
24        }
25    }
26 }
```

型別 變數名稱 指派運算子 值

int x = 10;

# 第二個程式 - 宣告 - 筆記

- 宣告
  - 顯示型別
    - string
    - int
    - array
    - ...
  - 隱含型別
    - var (一開始一定要指派值)



# 第三個程式 - 判斷

- 新增專案
- 目標方案名稱 BuildSchoolLab
- 專案名稱 Lab3\_IfElse
- 專案類型 主控台應用程式(.Net Framework)

# 第三個程式 – 判斷 – 筆記

- if...else...

```
if(條件)
{
    // true
}
else
{
    // false
}
```

- 比較運算子

- ==
- >
- <
- >=
- <=
- !=

# 第四個程式 – 陣列與迴圈

- 新增專案
- 目標方案名稱 BuildSchoolLab
- 專案名稱 Lab4\_ArrayAndForloop
- 專案類型 主控台應用程式(.Net Framework)

## 第四個程式 – 陣列

```
//int陣列  
int[] numbers = { 1, 2, 3, 4, 5 };  
  
//string陣列  
string[] words = { "Do", "Re", "Mi", "Fa", "So" };
```

之前的宣告只能放一個值，  
但是陣列的話就可以放多個值

注意，多個值但須注意型別要一致

## 第四個程式 – 陣列索引

```
//取值透過索引(位置)取出  
Console.WriteLine(numbers[4]);  
Console.WriteLine(words[0]);
```

當我們要用到陣列中某個值時，需要透過索引將值取出，也就是要透過位置將值取出

注意，在程式的世界中所有的索引都將從0開始

## 第四個程式 - 迴圈

```
//連續取出時我們會用到迴圈
for (int i = 0; i < 5; i = i + 1)
{
    Console.WriteLine(words[i]);
}
```

要清楚它的使用時機非常關鍵，以下為可能使用迴圈的情境

1. 具有規律性
2. 有同一件事情需要重複做

# 第四個程式 – 迴圈 – 筆記

- for

```
for(宣告計數器; 一直做的條件; 一圈做完後計數器+1)
{
    ...
}
```

# 第五個程式 – 更多迴圈

- 新增專案
- 目標方案名稱 BuildSchoolLab
- 專案名稱 Lab5\_Loop
- 專案類型 主控台應用程式(.Net Framework)



# 第五個程式 – 更多迴圈 – 筆記

- foreach

foreach(型別 變數 in 陣列)

- while 先判斷再做

while(條件)

- do...while 先做再判斷

do {

...

} while(條件)

# 第六個程式 – 方法

- 新增專案
- 目標方案名稱 BuildSchoolLab
- 專案名稱 Lab6\_Method
- 專案類型 主控台應用程式(.Net Framework)

# 第六個程式 - 方法

- 方法寫在類別區塊內
- 方法分為"有回傳值"與"無回傳值"方法
- 方法可以傳入參數

## 無回傳值

```
0 個參考
static void Main(string[] args)
{
    SaySomething("Hello");

    Console.ReadLine();
}

1 個參考
static void SaySomething(string word)
{
    Console.WriteLine(word);
}
```

## 有回傳值

```
0 個參考
static void Main(string[] args)
{
    //SaySomething("Hello");
    int answer = Add(10, 20);
    Console.WriteLine(answer);

    Console.ReadLine();
}

無回傳值
#region 有回傳值
1 個參考
static int Add(int x, int y)
{
    int ans = x + y;
    return ans;
}
#endregion
```

# 第七個程式 – 物件

- 新增專案
- 目標方案名稱 BuildSchoolLab
- 專案名稱 Lab7\_ObjectDesign
- 專案類型 主控台應用程式(.Net Framework)

# 第七個程式 – 物件 – 先思考模型

我現在要設計一家糖果店的模型，先想一想糖果店應該要有什麼?(先簡單一點)

1. 店名
2. 店長
3. 糖果(很多種糖果)

糖果店應該要有什麼方法?

1. 顯示糖果店基本資訊

# 第七個程式 - 物件 - 定義模型(欄位)

0 個參考

```
public class CandyStore
```

```
{
```

0 個參考

```
public string Name { get; set; }
```

0 個參考

```
public string Owner { get; set; }
```

0 個參考

```
public string[] Candies { get; set; }
```

```
}
```

# 第七個程式 - 物件 - 定義模型(方法)

```
0 個參考
public class CandyStore
{
    1 個參考
    public string Name { get; set; }
    1 個參考
    public string Owner { get; set; }
    1 個參考
    public string[] Candies { get; set; }

    0 個參考
    public void ShowInfo()
    {
        Console.WriteLine($"糖果店: {Name}");
        Console.WriteLine($"店長: {Owner}");
        Console.WriteLine("糖果清單");
        foreach(string candy in Candies)
        {
            Console.WriteLine(candy);
        }
    }
}
```

# 第七個程式 – 物件 – Main使用設計好的物件

0 個參考

```
static void Main(string[] args)
{
    CandyStore store_A = new CandyStore();
    store_A.Name = "Build店";
    store_A.Owner = "Dann";
    store_A.Candies = new string[] { "棒棒糖", "軟糖" };

    store_A.ShowInfo();

    Console.WriteLine("-----");
    CandyStore store_B = new CandyStore();
    store_B.Name = "School店";
    store_B.Owner = "Jimmy";
    store_B.Candies = new string[] { "可樂糖", "沙士糖", "酸梅糖" };

    store_B.ShowInfo();

    Console.ReadLine();
}
```



# 回家作業



Homework.exe

請試著用迴圈寫出99乘法表

可以上網找有什麼語法可以用，不是沒有教就不去找

===排版好看版===

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

===轉個方向版===

1*1=1	2*1=2	3*1=3	4*1=4	5*1=5	6*1=6	7*1=7	8*1=8	9*1=9
1*2=2	2*2=4	3*2=6	4*2=8	5*2=10	6*2=12	7*2=14	8*2=16	9*2=18
1*3=3	2*3=6	3*3=9	4*3=12	5*3=15	6*3=18	7*3=21	8*3=24	9*3=27
1*4=4	2*4=8	3*4=12	4*4=16	5*4=20	6*4=24	7*4=28	8*4=32	9*4=36
1*5=5	2*5=10	3*5=15	4*5=20	5*5=25	6*5=30	7*5=35	8*5=40	9*5=45
1*6=6	2*6=12	3*6=18	4*6=24	5*6=30	6*6=36	7*6=42	8*6=48	9*6=54
1*7=7	2*7=14	3*7=21	4*7=28	5*7=35	6*7=42	7*7=49	8*7=56	9*7=63
1*8=8	2*8=16	3*8=24	4*8=32	5*8=40	6*8=48	7*8=56	8*8=64	9*8=72
1*9=9	2*9=18	3*9=27	4*9=36	5*9=45	6*9=54	7*9=63	8*9=72	9*9=81



Build School