CH. 4

變數宣告與使用

HORAZON

C#程式設計

程式學習地圖 (初步、核心邏輯)

基本結構

變數宣告

循序

選擇 (IF, SWITCH)

迴圈 (FOR, WHILE, DO-WHILE)

符號的稱呼方式

括號 (BRACKETS)

常用於範圍界定、優先權處理

```
{ }   大括號   Curly Brackets     (程式區塊 Scope)
[ ]   中括號   Square Brackets    (陣列 Array)
( )   小括號   Parentheses        (函式、運算優先權)
```

引號 (QUOTATION MARKS)

常用於文字資料的定義

```
" "   雙引號   Double Quote       (字串 String)
' '   單引號   Single Quote       (字元 Char)
```

小嘗試

我們先嘗試修改第一個程式碼

```
Console.WriteLine("Hello, World!");
```

改成中文與加上自己的名字

```
Console.WriteLine("你好, Horazon!");
```

為何需要變數

```
Console.WriteLine("你好, Horazon!");
```

如果使用者名稱改變了，我們就要修改程式。

```
Console.WriteLine("你好, 張仕明!");
```

在程式使用中，如果都是一成不變的輸出文字，是無法發揮電腦功能的。
我們要製作一個可以修改的 **容器** 來儲存資料

什麼是變數 (VARIABLES)？

- **定義**：變數是電腦記憶體中用來儲存資料的「容器」。
- **概念**：你可以把它想像成一個貼有**標籤**的**盒子**。
  - **標籤** = 變數名稱 (Name)
  - **盒子大小/種類** = 資料型別 (Data Type)
  - **內容物** = 儲存的值 (Value)

變數的宣告與賦值

在 C# 中，使用變數前必須先宣告其型別。

基本語法：

資料型別 變數名稱 = 初始值;

```
// 範例：儲存玩家分數
int score = 100;

// 範例：儲存玩家名稱
string playerName = "Gemini";

// 範例：儲存身高
double height = 175.5;
```

常用資料型別

型別	說明	範例
int	整數	20, -2
double	雙精度浮點數	3.21, -0.5
float	單精度浮點數	10.0f, 5f
bool	真假值/布林	true,false
string	字串	"你好","Hello"

命名規則 (CAMEL CASE)

好的命名能讓程式更好讀：

✅ userAge, totalPrice (小駝峰式)

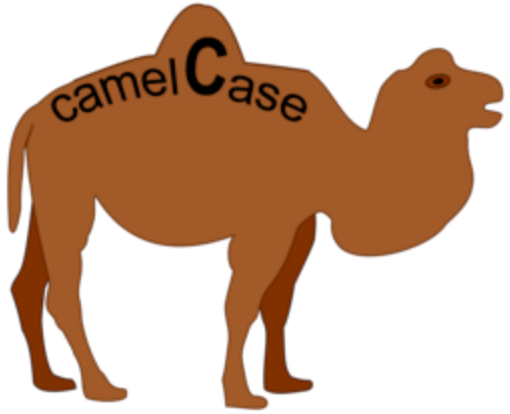
✅ itemNo1, itemNo2 (小駝峰式 後面使用數字)

⚠️ MyName, UnitLife (大駝峰式Pascal Case，可以使用但不建議)

⚠️ 年齡, 身高 (中文可使用，但不建議)

❌ 123name (不能數字開頭)

❌ my name (不能有空格)



使用變數

在這個例子中，我們宣告變數、賦值、並使用它們
請嘗試修改成自己的名字，並且將分數改為100分

```
using System;

var myName = "小明";
int score = 95;

Console.WriteLine(myName+" 的分數是 "+score+" 分");
```

輸出技巧

```
using System;

var myName = "小明";
int score = 95;

Console.WriteLine(myName+" 的分數是 "+score+" 分");
```

這段程式碼中，輸出部分使用 + 來連接文字與變數，

經常使用後會有點麻煩，我們可以改為使用 **$ (format)**

```
Console.WriteLine($"{myName} 的分數是 {score} 分");
```

變數重新賦值

變數就像一個**盒子**，在程式執行的過程中，你可以隨時更換盒子裡裝的資料。

- **第一次賦值**：稱為「初始化」(Initialization)。
- **重新賦值**：用新的值「覆蓋」舊的值。

當你要改變變數的值時，**不需要**再次寫出型別（如 `int` 或 `string`）。

```
int score = 100;     // 建立變數並給予初始值 100
score = 80;          // 重新賦值為 80 (舊的 100 消失了)
score = 50;          // 再次更換為 50
```

常數 (CONSTANT)

常數(const)使用規則

- 必須在宣告的時候賦予初始值
- 之後不能再改變它的數值

```
const double Pi = 3.14159;
const int MaxSpeed = 120;
const string AppName = "我的應用程式";
```

常見錯誤：重複宣告

已使用過的變數名稱不能再次使用，就算是不同型態也不行

```
int score = 90;
double score = 90.5;     //此行錯誤❌
```

以上程式碼會發生錯誤

常見錯誤：賦值型態錯誤

你必須賦予相同型態的數值，否則會發生錯誤

```
int score = 90.5;                //此行錯誤❌

string myName = "Horazon";
myName = 50;                     //此行錯誤❌
```

常見問題：常數問題

沒有給予常數初始值

```
double PI;                  //此行錯誤❌
```

對常數重新賦值

```
string myName = "Horazon";
myName = "Horazon2";     //此行錯誤❌
```

常見問題：溢位例外

每一種型態都有其極限，有興趣可以自行研究