



Unity 3D C# 和 Javascript 比較

■ 宋岳儒、陳光奕

雖然本書以 C# 為主，但是，在本節中，我們將提供一些簡單的 C# 和 Javascript 在撰寫腳本上的比較說明。

類別宣告：

C# 屬於物件導向程式語言，而物件導向的基本就是類別 (class)。在 C# 中 **class** 類別名「{ 」和「 }」包起來的範圍屬於一個類別；而在 Javascript 中一個檔案就是一個類別，如圖 G-1 所示。

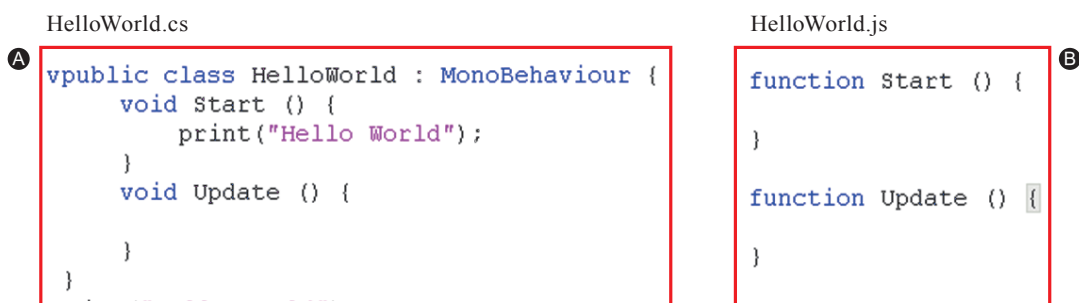


圖 G-1 class 範圍

- A C#: 在「{ }」之間是 HelloWorld 類別的內容
- B Javascript: 整個檔案都是一個類別

型態宣告：

在宣告變數和函式的回傳時，C# 必須要指定其型態，而 Javascript 則可以不指定型態。兩種語言的變數型態宣告如表 G-1。

C#	變數：型態變數名 [= 初始值];	<code>int hp = 10;</code>
Javascript	變數：var 變數名 [: 型態][= 初始值];	<code>var hp : int = 10;</code>

表 G-1 C# 及 Javascript 變數型態

Javascript 的函式回傳型態一樣可以不用指定，在這種情況下，其表示不回傳值給呼叫者，而 C# 在不回傳參數時，是使用型態 void，兩者的函式宣告如表 G-2：

C#	回傳型態 函式名 (參數);	int setHP(int hp)
Javascript	function 函式名 (參數) [回傳型態];	function setHP(int hp):int

表 G-2 C# 及 Java 函式宣告型態

Javascript 還有一個名為「動態型別」的特性，顧名思義就是在執行的時候動態決定變數的型態。一般而言，只要在宣告時，該變數沒有指定型態，程式就會在執行的過程中，根據賦予「值」的型態去變更該變數型態，例如圖 G-2。

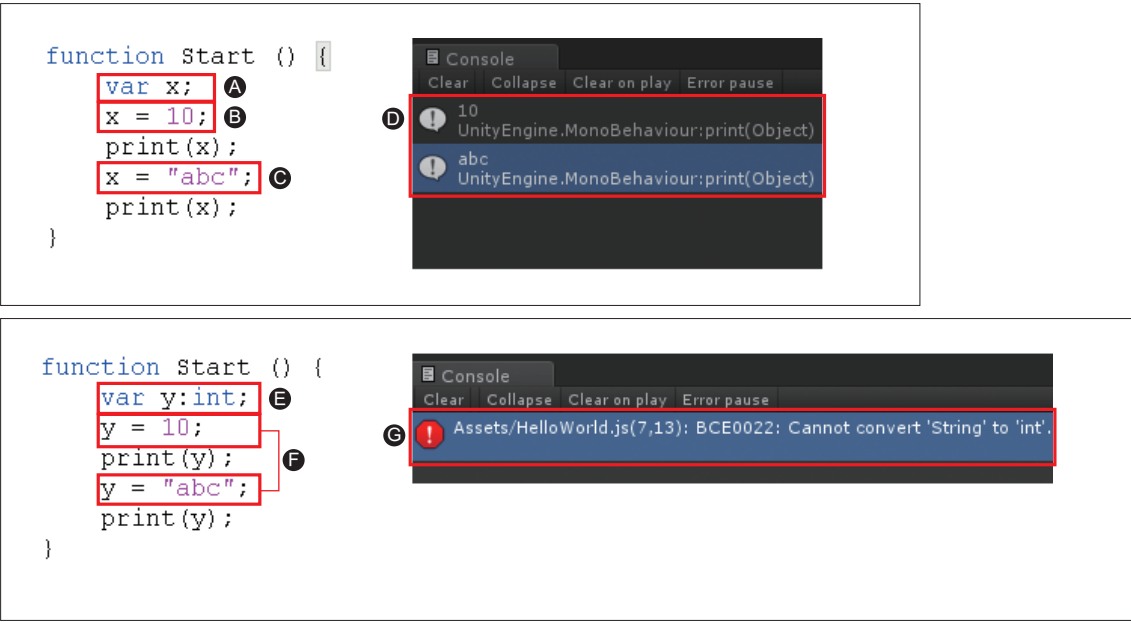


圖 G-2 指定型態和非指定型態的用法差異

- Ⓐ 宣告變數 x 但不指定型態
- Ⓑ 賦予 x 變數整數 10
- Ⓒ 賦予 x 變數字串「abc」
- Ⓓ 執行時輸出的結果是整數 10 和字串「abc」都會正常印出
- Ⓔ 宣告變數 y 並指定為整數 int
- Ⓕ 和前面一樣賦予 y 變數整數 10 和字串「abc」
- Ⓖ 執行時會出現無法轉換 'String' to 'int' 的錯誤

當變數一旦指定型態後，就無法動態變更型態，這時，若是再指定另一個型態，在程式編譯時，就會出錯。雖然動態型別很方便，但是，還是建議在宣告變數時，就要指定型態，因為這種動態型別用法容易搞混變數型態以及裡面存放的東西，最後可能導致執行錯誤而程式當掉。

public 和 private

在類別中，宣告的變數或函式至少要分為 `public` 和 `private` 兩類，`public` 表示該變數或函式可以被類別外部的其它函式使用，而 `private` 則屬於外部函式不可呼叫及使用的變數，在沒有特別指定的情況下，C# 預設為 `private`，而 Javascript 則預設為 `public`。

不同的形態名

C# 和 Javascript 大部分的基本型態名字是相同的，如 `int` 和 `float` 等，兩者不同的部分列在表 G-3。

	布林值	字串
C#	<code>bool</code>	<code>string</code>
Javascript	<code>boolean</code>	<code>String</code>

表 G-3 C# 及 Javascript 形態宣告名字不同列表

創建新元素的 new 函式

在 C# 中創建或使用一個新的物件一定要 `new` 指令，但是 Javascript 則不使用 `new`，以 `Vector3` 這個物件為例，兩種語言的使用方法如下：

● C#: `Vector3 v = new Vector3(1, 1, 1);`

● Javascript: `Vector3 v = Vector3(1, 1, 1);`