

CH. 9

字串 (STRING)

HORAZON

C# 程式設計

什麼是字串 (STRING)?

- 字串是由一連串的字元 (char) 所組成的。
- 在 C# 中，字串 (string) 是參考型別 (Reference Type)。
- 字串具有不可變性 (Immutability)：
 - 一旦建立，其內容就不能被改變。
 - 當你修改字串時，其實是產生了一個新的字串物件。

```
string name = "Horazon";
char firstChar = name[0]; // 可以像陣列一樣用索引存取，取得 'H'
```

字串格式化 (FORMATTING)

將變數插入字串中的方法。

1. 字串插值 (String Interpolation) \$ (推薦)

```
int age = 18;  
string message = $"我的年齡是 {age} 歲";
```

2. string.Format

```
string message = string.Format("我的年齡是 {0} 歲", age);
```

3. 傳統串接 +

```
string message = "我的年齡是 " + age + " 歲";
```

常用字串函數 (1) - 分割與合併

Split (分割)

將字串依照分隔符號切成陣列。

```
string data = "A,B,C,D";
string[] parts = data.Split(',');
// { "A", "B", "C", "D" }
```

Join (合併)

將陣列組合成一個字串。

```
string[] words = { "Hello", "world" };
string sentence = string.Join(" ", words);
// "Hello World"
```

常用字串函數 (2) - 轉換與修改

大小寫轉換

```
string text = "Hello";
Console.WriteLine(text.ToUpper()); // "HELLO"
Console.WriteLine(text.ToLower()); // "hello"
```

取代 (Replace)

```
string text = "I love Java";
string newText = text.Replace("Java", "C#"); // "I love C#"
```

去除空白 (Trim)

```
string input = " user ";
Console.WriteLine(input.Trim()); // "user" (去除前後空白)
```

常用字串函數 (3) - 擷取與搜尋

Substring (子字串)

```
string text = "Hello world";
// 從索引 6 開始，取 5 個字元
string sub = text.Substring(6, 5); // "world"
```

Contains (包含)

```
bool hasWorld = text.Contains("world"); // true
```

IndexOf (搜尋位置)

```
int index = text.IndexOf("world"); // 6
```

字串效能問題

由於字串的**不可變性**，頻繁修改字串會造成效能低落。

```
string s = "";
for (int i = 0; i < 10000; i++)
{
    s += i; // 每次都會產生一個新的字串物件 !
}
```

記憶體中會產生大量的垃圾物件。

STRINGBUILDER 類別

若是需要頻繁修改字串（例如在迴圈中串接），請使用 **StringBuilder**。

StringBuilder 是可變的 (Mutable)，直接在記憶體中修改，不會產生新物件。

```
using System.Text; // 記得引用命名空間

StringBuilder sb = new StringBuilder();
for (int i = 0; i < 10000; i++)
{
    sb.Append(i); // 高效串接
}

string result = sb.ToString(); // 最後再轉回字串
```

STRINGBUILDER 常用方法

- Append(value) : 加到最後面
- AppendLine(value) : 加到最後面並換行
- Insert(index, value) : 插入到指定位置
- Remove(index, length) : 刪除指定範圍
- Replace(old, new) : 取代文字

- `string` 是不可變的，每次修改都會產生新物件。
- `">${變數}` 是最常用的格式化方式。
- 善用 `Split`, `Join`, `Replace`, `Substring` 等內建函數。
- 若需頻繁修改字串，請務必使用 `StringBuilder` 以提升效能。