

CH. 8

陣列 (ARRAY)

HORAZON

C#程式設計

什麼是陣列 (ARRAY)?

- 陣列是一種**資料結構**，用來儲存**多個**相同類型的資料。
- 就像是一個置物櫃，每個格子都有編號，可以放一樣的東西。

為什麼需要陣列？

如果不使用陣列，儲存 5 個學生的成績需要宣告 5 個變數：

`score1` , `score2` , `score3` , `score4` , `score5` ... (太麻煩了！)

使用陣列，只需要一個變數： `scores` 。

宣告與建立陣列

語法：資料型別[] 陣列名稱 = new 資料型別[長度];

```
// 建立一個可以放 5 個整數的陣列
int[] numbers = new int[5];

// 建立一個可以放 3 個字串的陣列
string[] names = new string[3];
```

直接初始化值：

```
// 宣告時直接給值，系統會自動判斷長度為 3
int[] scores = { 80, 90, 75 };

// 另一種寫法
string[] fruits = new string[] { "Apple", "Banana" };
```

存取陣列元素

- 陣列的索引 (Index) 是從 0 開始的。
- 第一個元素的索引是 0，第二個是 1，以此類推。

```
int[] data = { 10, 20, 30 };  
  
Console.WriteLine(data[0]); // 印出 10  
Console.WriteLine(data[2]); // 印出 30  
  
data[1] = 99; // 修改第二個元素的值為 99  
Console.WriteLine(data[1]); // 印出 99
```

[!WARNING]

若存取超過範圍的索引 (例如 `data[3]`)，會發生 `IndexOutOfRangeException` 錯誤。

陣列長度 (LENGTH)

使用 `.Length` 屬性可以取得陣列的長度 (元素個數)。

```
string[] cars = { "Volvo", "BMW", "Ford", "Mazda" };  
Console.WriteLine(cars.Length); // 印出 4
```

這在搭配迴圈使用時非常有用，可以確保不會超出範圍。

使用迴圈走訪陣列 (FOR)

最傳統的方式，利用索引值來存取。

```
int[] numbers = { 2, 4, 6, 8, 10 };  
  
for (int i = 0; i < numbers.Length; i++)  
{  
    Console.WriteLine($"索引 {i} 的值是：{numbers[i]}");  
}
```

- 優點：可以知道目前的索引值，也可以修改陣列內容。
- 缺點：語法稍長。

使用迴圈走訪陣列 (FOREACH)

最簡潔的方式，直接依序取出元素。

```
string[] members = { "Alice", "Bob", "Charlie" };  
  
foreach (string member in members)  
{  
    Console.WriteLine(member);  
}
```

- 優點：語法簡單，不易出錯。
- 缺點：無法知道目前的索引值，且**不能**透過 foreach 修改陣列元素的值 (唯讀)。

常用陣列操作 (ARRAY 類別)

C# 提供了 `Array` 類別來協助處理陣列。

1. 排序 (Sort)

```
int[] nums = { 5, 1, 3, 2, 4 };  
Array.Sort(nums); // 由小到大排序  
// nums 變成 { 1, 2, 3, 4, 5 }
```

2. 搜尋 (IndexOf)

```
int index = Array.IndexOf(nums, 3);  
// 找到 3 在陣列中的位置 (索引值)
```


二維陣列 (MULTIDIMENSIONAL ARRAY)

就像是 Excel 表格，有列 (Row) 和 欄 (Column)。

```
// 宣告一個 2x3 的二維陣列 (2列 3欄)  
int[,] matrix = {  
    { 1, 2, 3 },  
    { 4, 5, 6 }  
};
```

```
Console.WriteLine(matrix[0, 0]); // 印出 1 (第一列第一欄)  
Console.WriteLine(matrix[1, 2]); // 印出 6 (第二列第三欄)
```

常用於地圖、棋盤遊戲等應用。

總結

- 陣列是儲存**固定大小、相同型別**資料的容器。
- 索引從 0 開始。
- 使用 `Length` 取得長度。
- 使用 `for` 或 `foreach` 遍歷資料。
- 善用 `Array.Sort()` 等內建方法來處理資料。