

# C# 常用類別

## 一. ArrayList 類別

在 .NET Framework 另外提供一個 ArrayList 類別，讓你可以建立不定長度的陣列，由於此種陣列的資料型別為 object，因此在陣列中的陣列元素允許存放不同型別的資料。ArrayList 類別是包含在 System.Collections 的命名空間中，所以，要先在程式的最開頭撰寫如下敘述引用 System.Collections 命名空間，這樣才可以使用較簡潔的物件名稱來建立 ArrayList 物件類別。

```
using System.Collections;
```

使用上面敘述引用 System.Collections 命名空間後，接著可以使用下列語法來建立 ArrayList 物件：

```
ArrayList 陣列名稱 = new ArrayList();
```

下面介紹 ArrayList 類別常用屬性與方法。

成員名稱	說明
Count 屬性	取得 ArrayList 中陣列元素的總數目。
Item 屬性	設定或取得由索引(Index)所指的陣列元素的內容。
Add 方法	將物件加入到 ArrayList 的最後。
Insert 方法	將物件插入到 ArrayList 所指定陣列索引的位置。
Remove 方法	由 ArrayList 移除第一個符合指定物件的陣列元素。
Clear 方法	清除 ArrayList 中所有的陣列元素。
CopyTo 方法	將 ArrayList 中全部或部分陣列元素複製到另一個一維陣列中。

成員名稱	說明
IndexOf 方法	傳回 ArrayList 中第一個符合指定物件的索引值。若找不到，會傳回一個負值。
Sort 方法	將 ArrayList 中所有元素以遞增方式排序。
Reverse 方法	將 ArrayList 中所有元素反轉。若對 ArrayList 先做 Sort 再做 Reverse 相當於對 ArrayList 做降冪排序。
BinarySearch 方法	使用二分搜尋法由 ArrayList 找尋所指定的物件，若有找到傳回該物件的索引；若沒找到傳回一個負值。使用此方法之前，必須先將 ArrayList 做遞增排序。

## 二. char 字元類別

在 .NET Framework 的 System.Char 對 char 資料型別(即 char 類別)提供下列成員，以方便您對字元作各種處理：

成員名稱	說明
Equals	<p>語法：char.Equals(objectA,objectB)；</p> <p>功能：若物件 A 與物件 B 相等則傳回 true，否則傳回 false。</p> <p>範例：bool result;</p> <pre>char char1='a'; char char2='c'; result=char.Equals(char1,char2);    // 傳回 false Console.WriteLine(result.ToString()); // 顯示 False</pre>
GetNumericValue	<p>語法：char.GetNumericValue(MyChar)；</p> <p>功能：指定字元是否為數字，若是傳回該數字，否則傳回-1。</p> <p>範例：char MyChar = '\u0039'; // Unicode</p> <pre>Console.WriteLine(char.GetNumericValue(MyChar)); //顯示 9 Console.WriteLine(char.GetNumericValue('6'));    //顯示 6 Console.WriteLine(char.GetNumericValue('k'));    //顯示-1</pre>
IsControl	<p>語法：char.IsControl(string s,int index);</p> <p>功能：判斷位於指定字串的指定位置的字元是否為控制字元。</p> <p>範例：string s = 'lung';</p> <pre>bool result2=char.IsControl(s,0); Console.WriteLine(result2); // 顯示 False</pre>

成員名稱	說明
IsDigit	<p>語法：char.IsDigit(string s,int index);</p> <p>功能：判斷位於指定字串的指定位置的字元是否為十進位數。</p> <p>範例：string s="a1c3";  bool result2=char.IsDigit(s,1); // 檢查 '1' 字元  Console.WriteLine(result2); // 顯示 True</p>
IsLetter	<p>語法：char.IsLetter(string s,int index);</p> <p>功能：判斷位於指定字串的指定位置的字元是否為字母字元。</p> <p>範例：string s="a1c3";  bool result3=char.IsLetter(s,2); //檢查 'c' 字元  Console.WriteLine(result3); // 顯示 True</p>
IsLetterOrDigit	<p>語法：char.IsLetterOrDigit(string s,int index); ;</p> <p>功能：判斷位於指定字串的指定位置的字元是否為字母或數字字元。</p> <p>範例：string s="a1c3";  bool result3=char.IsLetter(s,2); //檢查 'c' 字元  Console.WriteLine(result3); // 顯示 True</p>
IsLower	<p>語法：char.IsLower(string s,int index);</p> <p>功能：判斷位於指定字串的指定位置的字元是否為字母或數字字元。</p> <p>範例：string s="A1c3";  bool result3=char.IsLower(s,0); //檢查 'A' 字元  Console.WriteLine(result3); // 顯示 False</p>
IsNumber	<p>語法：char.IsNumber(string s,int index);</p> <p>功能：判斷位於指定字串的指定位置的字元是否為十進或十六進制數字字元。</p> <p>範例：string s="a1c3";  bool result3=char.IsNumber(s,3); //檢查 '3' 字元  Console.WriteLine(result3); // 顯示 True</p>
IsPunctuation	<p>語法：char.IsPunctuation(string s,int index);</p> <p>功能：判斷位於指定字串的指定位置的字元是否為標點符號字元。</p> <p>範例：string s="a1,c3";  bool result3=char.IsPunctuation(s,2); //檢查 ',' 字元  Console.WriteLine(result3); // 顯示 True</p>

成員名稱	說明
IsSeperator	<p>語法：char.IsSeperator(string s,int index);</p> <p>功能：判斷位於指定字串的指定位置的字元是否為分隔符號字元，如空白鍵即是分隔符號。</p> <p>範例：string s ="Good Day";  bool result3=char.IsSeperator(s,4); //檢查 ' ' 字元  Console.WriteLine(result3); // 顯示 True</p>
IsSymbol	<p>語法：char.IsSymbol(string s,int index);</p> <p>功能：判斷位於指定字串的指定位置的字元是否為符號字元。</p> <p>範例：string s ="Good+Day!";  bool result3=char.IsSymbol(s,4); // 檢查 '+' 字元  Console.WriteLine(result3); // 顯示 True</p>
IsUpper	<p>語法：bool char.IsUpper(string s,int index);</p> <p>功能：判斷位於指定字串的指定位置的字元是否為大寫字元。</p> <p>範例：string s ="Good+Day!";  bool result3=char.IsUpper(s,5); //檢查 'D' 字元  Console.WriteLine(result3); // 顯示 True</p>
IsWhiteSpace	<p>語法：bool char.IsWhiteSpace(string s,int index);</p> <p>功能：判斷位於指定字串的指定位置的字元是否為空白字元。</p> <p>範例：string s ="Good Day!";  bool result3=char.IsWhiteSpace(s,4); //檢查 ' ' 字元  Console.WriteLine(result3); // 顯示 True</p>
Max Value	<p>語法：DataType.MaxValue ;</p> <p>功能：取得該資料型別的最大值。</p> <p>範例：Console.WriteLine(int.MaxValue); // 2,147,483,647  Console.WriteLine((int)char.MaxValue); // 65,535  Console.WriteLine(float.MaxValue); // 3.402823E+38</p>
Min Value	<p>語法：DataType.MinValue ;</p> <p>功能：取得該資料型別的最小值。</p> <p>範例：Console.WriteLine(int.MinValue); // -2,147,483,648  Console.WriteLine((int)char.MinValue); // 0  Console.WriteLine(float.MinValue); // -3.402823E+38</p>

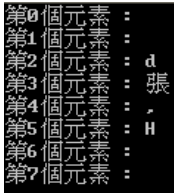
成員名稱	說明
Parse	語法：char.Parse(string s); 功能：將指定的字串轉換成它對等的 Unicode 字元。 範例：Console.WriteLine(char.Parse("A")); // "A"
ToLower	語法：char.ToLower(char c); 功能：將 Unicode 字元值轉換成它的對等小寫字元。 範例：Console.WriteLine(char.ToLower('A')); // 'a' char ch='B'; Console.WriteLine(char.ToLower(ch)); // 'b'
ToUpper	語法：char.ToUpper(char c); 功能：將 Unicode 字元值轉換成它的對等大寫字元。 範例：Console.WriteLine(char.ToUpper('a')); // 'A' char ch='b'; Console.WriteLine(char.ToUpper(ch)); // 'B'
ToString	語法：char.ToString(char c); 功能：將 Unicode 字元值轉換成它的對等的字串。 範例：char c='B'; Console.WriteLine(char.ToString(c)); // "B" Console.WriteLine(c.ToString()); // "B"

### 三. string 字串類別

C# 對字串的處理提供 System.String 基底類別。下表中各範例假設都已對整數變數 n，字串變數 str1、str2、str3 和 str4 做過宣告：

成員名稱	說明
Length 屬性	語法：n=str1.Length; 功能：傳回 str1 字串的長度。 範例：str1="C#.NET 上市"; n=str1.Length; // n 值為 8
ToString 方法	語法：str1=n.ToString(); 功能：將數值轉成字串。 範例：n=1234; str1=n.ToString(); // str1 ⇐ "1234"

成員名稱	說明
Compare 方法	<p>語法：n=String.Compare(str1,str2);</p> <p>功能：str1 字串和 str2 字串作比較，若：</p> <ul style="list-style-type: none"> <li>① str1&gt;str2 傳回 1</li> <li>② str1=str2 傳回 0</li> <li>③ str1&lt;str2 傳回-1</li> </ul> <p>範例：str1="Visual Basic" ;  str2 ="C#.NET" ;  n=String.Compare(str1,str2); // n 值為 1</p>
CompareTo 方法	<p>語法：n=str1.CompareTo(str2);</p> <p>功能：str1 字串和 str2 字串作比較，若：</p> <ul style="list-style-type: none"> <li>① str1&gt;str2 傳回 1</li> <li>② str1=str2 傳回 0</li> <li>③ str1&lt;str2 傳回-1</li> </ul> <p>範例：str1 ="Visual Basic.NET" ;  str2 = "C#.NET" ;  n = str1.CompareTo(str2); //n =1</p>
ToUpper 方法 ToLower 方法	<p>語法：str1.ToUpper(); / str1.ToLower();</p> <p>功能：將 str1 字串變數中所有字母以大 / 小寫顯示。</p> <p>範例：str1 ="Visual Basic.NET" ;  str2=str1.ToUpper(); //str2 ⇨ "VISUAL BASIC.NET"  str2=str1.ToLower(); // str2 ⇨ "visual basic.net"</p>
Copy 方法	<p>語法：str1=String.Copy(str2);</p> <p>功能：將 str2 字串複製給 str1 字串變數。</p> <p>範例：str1 ="Hello" ;  str2="World !" ;  str1=String.Copy(str2); //str1 ⇨ "Hello"</p>

成員名稱	說明
CopyTo 方法	<p>語法：str1.CopyTo(n1,arrayA,n2,n3)；</p> <p>功能：將 str1 字串的第 n1 個位置開始複製 n3 個字元放到 arrayA 字元陣列，由第 n2 個位置開始放起。</p> <p>範例：char[ ] arrayA= new char[8]；  str1 = "David 張,How are you!"；  str1.CopyTo(4, arrayA, 2, 4)；  for (n = 0 ; n&lt;arrayA.Length ; n++){  str2="第" + n + "個元素："；  Console.WriteLine(str2+ arrayA[n])；  }</p> <p>【結果】</p> 
StartsWith 方法 EndsWith 方法	<p>語法：bool a =str1.StartsWith(str2)；  bool a =str1.EndsWith(str2)；</p> <p>功能：若 str2 字串是 str1 字串的開頭 / 結尾字串，若是傳回 true；否則傳回 false。</p> <p>範例：str1 = "Hello World !"；  str2 = "World"；  Console.WriteLine(str1.StartsWith(str2).ToString()) ;//False  Console.WriteLine(str1.EndsWith(str2).ToString()) ;//False</p>
Concat 方法	<p>語法：str3.Concat(str1,str2)；  String.Concat(str1,str2)；</p> <p>功能：將 str1 字串和 str2 頭尾串接放入 str3。</p> <p>範例：str1 = "Hello"；  str2="World !"；  str3.Concat(str1,str2)； //str3 ⇨ "Hello World !"；  Console.WriteLine(String.Concat(str1,str2))；</p>

成員名稱	說明
Equals 方法	<p>語法：str1.Equals(str2)；</p> <p>功能：檢查 str1 字串是否和 str2 字串相等。</p> <p>範例：if(str1.Equals(str2)) //判斷 str1 是否等於 str2  if (str1==str2) //判斷 str1 是否等於 str2, 同上面敘述</p>
Insert 方法	<p>語法：str1.Insert(n,str2)；</p> <p>功能：將 str2 字串插在 str1 字串的第 n 個位置(由 1 算起)。</p> <p>範例：str1="0123456789";  str2="abc" ;  str3=str1.Insert(5,str2) ; //str3 ⇐ "01234abc56789"</p>
PadLeft 方法 PadRight 方法	<p>語法：str1.PadLeft(n, "*"); 或 str1.PadRight(n, "*");</p> <p>功能：str1 字串靠左 / 靠右對齊，以指定的字元補足使其長度為 n。</p> <p>範例：str1 ="abc";  str2=str1.PadLeft(5, "*"); // str2 ⇐ "***abc"  str2=str1.PadRight(5, "*"); // str2 ⇐ "abc**"</p>
IndexOf 方法 LastIndexOf 方法	<p>語法：n=str1.IndexOf(str2)；  n=str1.LastIndexOf(str2)；</p> <p>功能：IndexOf 由 str1 字串找出 str2 字串第一次出現位置，位置由最前面 [0] 開始算起；若找不到傳回值為-1。LastIndexOf 由 str1 字串找出 str2 字串最後出現的位置，位置由最前面 [0] 開始算起。</p> <p>範例：str1 ="C#.NET ASP.NET C#.NET"；  str2=".NET"；  n=str1.IndexOf("ASP")；//傳回 7  n=str1.LastIndexOf(str2)；// 傳回 17  n=str1.IndexOf("O")；//傳回-1 (找不到)</p>
ToCharArray 方法	<p>語法：str1.ToCharArray(n1,n2)；</p> <p>功能：將 str1 字串放入字元陣列，由字元陣列的第 n1 個元素開始取出 n2 個元素。</p> <p>範例：str1="0123456789"；  str2=str1.ToCharArray(2,4)；// str2 ⇐ "2345"</p>



成員名稱	說明
Join 方法	<p>語法：string.Join(分隔字元或符號, 陣列名稱);</p> <p>功能：會複製陣列內型別為 string 的字串，並透過指定的分隔字元或符號來組合成字串</p> <p>範例：str1="Time to say goodbye";  string[ ] myArray = new string[4];  myArray[0]= "Time"; myArray[1]= "to";  myArray[2]= "say"; myArray[3]= "goodbye"  Console.WriteLine(string.Join(" ",myArray));</p> <p>結果：Time*to*say*goodbye</p>
Split 方法	<p>語法：str1.Split(分隔字元或符號);</p> <p>功能：按照字串中所指定的字元作分隔符號，將字串隔開，再將這些分隔的字串依序放入指定的字串陣列中。</p> <p>範例：str1="Time to say goodbye";  string[ ] myArray = new string[4];  myArray=str1.Split(' ') //以空白當分隔符號</p> <p>結果：myArray[0]= "Time" myArray[1]= "to"  myArray[2]= "say" myArray[3]= "goodbye"</p>
Remove 方法	<p>語法：str1.Remove(n1,n2);</p> <p>功能：將 str1 字串的第 n1 個字元開始往後刪除 n2 個字元。</p> <p>範例：str1="0123456789";  str2=str1.Remove(2,3); //str2⇐"0156789"</p>
Replace 方法	<p>語法：str1.Replace(str2,str3);</p> <p>功能：由 str1 字串中找出 str2 字串以 str3 字串取代。若沒找到符合的字串，則不做取代維持原字串。</p> <p>範例：str1="0123456789";  str2="456";  str3="abcd";  str4=str1.Replace(str2,str3); //str4⇐"0123abcd789"</p>
SubString 方法	<p>語法：str1.Substring(n1,n2);</p> <p>功能：由 str1 字串第 n1 個字元開始取 n2 個字元。</p> <p>範例：str1="How are you 李先生";  str2=str1.Substring(8,4); //str2⇐ "you 李"</p>

成員名稱	說明
TrimStart 方法 TrimEnd 方法 Trim 方法	語法：str1.TrimStart() / str1.TrimEnd() / str1.Trim() 功能：將字串最前面 / 最後面 / 前後的空白去除。 範例：str1=" Happy "; // 前後各留三個空白 str2=str1.TrimStart(); //str2⇐ "Happy " str2=str1.TrimEnd(); //str2⇐ " Happy" str2=str1.Trim(); //str2⇐ "Happy"

#### 四. DateTime 日期時間類別

.NET Framework 類別庫中的 DateTime 類別可用來設定日期與時間。下表僅列出 DateTime 部分屬性。使用 DateTime 類別中某些屬性時，必需使用 new 建立 DateTime 物件實體才可使用。以下範例中 dateTime1 是使用 DateTime 類別所建立的日期時間物件實體，此日期時間物件為西元 2017 年 1 月 23 日上午 4 時 5 分 6 秒 7 毫秒。

```
DateTime dateTime1 = new DateTime(2017,1,23,4,5,6,7);
```

成員名稱	說明
Now	語法：DateTime.Now 功能：取得這部電腦上當地目前日期和時間，使用 Now 屬性不需要建立 DateTime 類別的物件實體。 範例：DateTime.Now 傳回 DateTime 型別 傳回值為 2017/9/30 下午 07:40:50
Today	語法：DateTime.Today 功能：取得目前的日期，使用 Today 屬性不需要建立 DateTime 類別的物件實體。 範例：DateTime.Today 傳回 DateTime 型別 傳回值為 2017/9/30 上午 12:00:00
Date	語法：dateTime1.Date 功能：取得 DateTime 物件的日期。 範例：dateTime1.Date 傳回 DateTime 型別 傳回值為 2017/1/23 上午 12:00:00
Year	語法：dateTime1.Year 功能：取得 DateTime 物件的年份，為 Int32 數值型別。 範例：dateTime1.Year 傳回值為 2017

成員名稱	說明
Month	語法：dateTime1.Month 功能：取得 DateTime 物件的月份，為 Int32 數值型別。 範例：dateTime1.Month 傳回值為 1，為 Int32 型別
Day	語法：dateTime1.Day 功能：取得 DateTime 物件的日期，為 Int32 數值型別。 範例：dateTime1.Month 傳回值為 23
Hour	語法：dateTime1.Hour 功能：取得 DateTime 物件的小時，為 Int32 數值型別。 範例：dateTime1.Hour 傳回值為 4
Minute	語法：dateTime1.Minute 功能：取得 DateTime 物件的分鐘，為 Int32 數值型別。 範例：dateTime1.Minute 傳回值為 5
Second	語法：dateTime1.Second 功能：取得 DateTime 物件的秒數，為 Int32 數值型別。 範例：dateTime1.Second 傳回值為 6
Millisecond	語法：dateTime1.Day 功能：取得 DateTime 物件的毫秒，為 Int32 數值型別。 範例：dateTime1.Millisecond 傳回值為 7

## 五. Math 數學類別

.NET Framework 類別程式庫中的 Math(數學)類別，提供許多有關三角函數、對數以及其它常見的數學函數與常數。

成員名稱	說明
Math.Sign	語法：Math.Sign(n) 功能：判斷 n 數值的正負。 若 $n > 0$ ，則傳回值為 1。 若 $n = 0$ ，則傳回值為 0。 若 $n < 0$ ，則傳回值為-1。 範例：Math.Sign(12) 傳回值為 1。 Math.Sign(-12) 傳回值為-1。 Math.Sign(0) 傳回值為 0。

成員名稱	說明
Math.Floor	語法： <code>Math.Floor(n)</code> 功能：傳回小於或等於 <code>n</code> 的最大整數。 範例： <code>Math.Floor(4.4)</code> 傳回值為 4
Math.Ceiling	語法： <code>Math.Ceiling(n)</code> 功能：傳回大於或等於 <code>n</code> 的最小整數。 範例： <code>Math.Ceiling(4.4)</code> 傳回值為 5
Math.Round	語法： <code>Math.Round(n)</code> 功能：傳回數值 <code>n</code> 的整數部份，而小數第一位四捨六入。 範例： <code>Math.Round(4.5)</code> 傳回值為 4 <code>Math.Round(5.5)</code> 傳回值為 6
Math.Abs	語法： <code>Math.Abs(n)</code> 功能：傳回同樣數值型別的絕對值。 範例： <code>Math.Abs(5.3)</code> ，傳回值為 5.3 <code>Math.Abs(-5.3)</code> ，傳回值為 5.3
Math.Sqrt	語法： <code>Math.Sqrt(n)</code> 功能：傳回數值 <code>n</code> 的平方根。 範例： <code>Math.Sqrt(0.49)</code> ，傳回值為 0.7
Math.Pow	語法： <code>Math.Pow(x,y)</code> 功能：傳回數值 <code>x</code> 的 <code>y</code> 次方，即 $x^y$ 。 範例： <code>Math.Pow(5, 3)</code> ，傳回值為 125
Math.Sin	語法： <code>Math.Sin( r )</code> 功能：傳回數值 <code>r</code> 的正弦函數值。 <code>r</code> 是一個 radians 弧度量。 範例：若 <code>angle = 30 * (Math.PI / 180)</code> 即角度(Degree)為 30°，則 <code>Math.Sin(angle)</code> ，傳回值為 0.5
Math.Cos	語法： <code>Math.Cos( r )</code> 功能：傳回數值 <code>r</code> 的餘弦函數值。 範例： <code>r = 30 * (Math.PI / 180)</code> <code>Math.Cos(r)</code> ，傳回值為 0.866025403784439
Math.Tan	語法： <code>Math.Tan( r )</code> 功能：傳回數值 <code>r</code> 的正切函數值。 範例： <code>Math.Tan(45 * (Math.PI / 180))</code> ，傳回值為 1。

成員名稱	說明
Math.Asin	語法：Math.Asin(x) 功能：傳回數值 x 的反正弦函數值。
Math.Acos	語法：Math.Acos(x) 功能：傳回數值 x 的反餘弦函數值。
Math.Atan	語法：Math.Atan(x) 功能：傳回數值 x 的反正切函數值。
Math.Atan2	語法：Math.Atan2(x,y) 功能：傳回 $\tan^{-1}(y/x)$ 數值。
Math.Sinh	語法：Math.Sinh(r) 功能：傳回數值 r 的雙曲線正弦函數值。
Math.Cosh	語法：Math.Cosh(r) 功能：傳回數值 r 的雙曲線餘弦函數值。
Math.Tanh	語法：Math.Tanh(r) 功能：傳回數值 r 的雙曲線正切函數值。
Math.Exp	語法：Math.Exp(x) 功能：傳回 $e^x$ 值。E 為自然對數，大小約 2.71828182845905
Math.Log	語法：Math.Log(x) 功能：傳回 $\log_e x$ 值。 $\log_e x$ 在數學上經常用 $\ln x$ 表示。
Math.Log10	語法：Math.Log10(x) 功能：傳回 $\log_{10} x$ 值。
Math.Max	語法：Math.Max(n1,n2) 功能：比較 n1、n2 的大小，傳回較大值。 範例：Math.Max(4,10) 傳回值為 10
Math.Min	語法：Math.Min(n1,n2) 功能：比較 n1、n2 的大小，傳回較小值。 範例：Math.Min(4,10) 傳回值為 4
Math.PI	語法：Math.PI (數學屬性) 功能：Math.PI = 3.14159265358979，即為 $\pi$ 的值
Math.E	語法：Math.E (數學屬性) 功能：Math.E = 2.71828182845905，即為 e 的值。

## 六. Random 亂數類別

亂數的應用隨著電腦的普及，更加顯示其重要性。在 .NET Framework 類別程式庫中提供了 **Random** 類別，此類別為 **System.Random** 型別，其功能可用來產生亂數。使用 **Random** 類別中的方法時，必須建立 **Random** 物件實體才可使用。以下範例中 **rnd1** 是使用 **Random** 類別所建立的亂數物件實體。

```
Random rnd1 = new Random();
```

**Random** 類別沒有屬性，此類別常用的方法如下表說明：

成員名稱	說明
Next	語法： <b>rnd1.Next()</b> // 傳回大於等 0 的亂數，且小於 <b>int</b> 型別的最大值 <b>rnd1.Next(n1)</b> // 傳回 0 到 <b>n1-1</b> 的亂數值 <b>rnd1.Next(n1, n2)</b> // 傳回 <b>n1</b> 到 <b>n2-1</b> 的亂數值
NextDouble	語法： <b>rnd1.NextDouble()</b> 功能：傳回 0.0 到 1.0 之間的亂數。