



資料型別的成員

A-1 認識資料型別

A-2 byte 型別的成員

A-3 sbyte 型別的成員

A-4 short 型別的成員

A-5 ushort 型別的成員

A-6 int 型別的成員

A-7 uint 型別的成員

A-8 long 型別的成員

A-9 ulong 型別的成員

A-10 float 型別的成員

A-11 double 型別的成員

A-12 decimal 型別的成員

A-13 bool 型別的成員

A-14 object 型別的成員

A-15 char 型別的成員

A-16 string 型別的成員

A-17 System.DateTime

結構的成員



A-1 認識資料型別

C# 的型別都是由 .NET Framework 的 System 命名空間內的某個結構 (structure) 或某個類別 (class) 所支援，例如 bool 型別是由 System 命名空間內的 Boolean 結構所支援，換句話說，bool 可以視為 System.Boolean 的別名 (alias)，因此，下面兩個敘述的意義是相同的：

```
bool X;  
System.Boolean X;
```

在 .NET Framework 中，結構屬於實值型別，類別屬於參考型別，諸如 byte、short、int、long、sbyte、ushort、uint、ulong、float、double、decimal、bool、char 等實值型別是由 System 命名空間內的 Byte、Int16、Int32、Int64、SByte、UInt16、UInt32、UInt64、Single、Double、Decimal、Boolean、Char 等結構所支援，而諸如 string、object 等參考型別則是由 String、Object 等類別所支援。

既然 C# 的型別是由結構或類別所支援，所以它們就會擁有建構函式 (constructor)、方法 (method)、欄位 (field) 等成員。比方說，C# 的 int 型別是由 System.Int32 結構所支援，而這個結構有一個名稱為 GetType() 的方法可以傳回目前物件的型別，例如 123.GetType() 會傳回 System.Int32。

現在，我們就將內建型別的成員列出來，您可以快速瀏覽，無須熟背，有關這些成員的詳細用法，可以參閱 MSDN 文件。

本附錄除了列出 System 命名空間內的 Byte、Int16、Int32、Int64、SByte、UInt16、UInt32、UInt64、Single、Double、Decimal、Boolean、Char 等結構及 String、Object 等類別的成員，還另外列出 DateTime 結構的成員，雖然 C# 沒有提供日期時間型別，但透過 DateTime 結構的成員，我們可以取得 / 設定系統目前的日期時間或進行日期時間的運算。

A-2 byte 型別的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 byte 型別的最大值 (255)。
MinValue	這個常數是 byte 型別的最小值 (0)。
公有方法 (Public Method)	
CompareTo(obj)	將目前案例與參數 <i>obj</i> 做比較，傳回值為 int 型別，小於 0 表示目前案例小於參數 <i>obj</i> ，等於 0 表示目前案例等於參數 <i>obj</i> ，大於 0 表示目前案例大於參數 <i>obj</i> 。
Equals(obj) Equals(obj1, obj2)	前者是判斷目前案例與參數 <i>obj</i> 是否相等，後者是判斷參數 <i>obj1</i> 和參數 <i>obj2</i> 是否相等，是就傳回 true，否則傳回 false。
ReferenceEquals(obj1, obj2)	判斷參數 <i>obj1</i> 和參數 <i>obj2</i> 是否為相同案例或皆為 null 參考，是就傳回 true，否則傳回 false。
GetHashCode()	傳回目前案例的雜湊碼 (hash code)，型別為 int。
GetType()	傳回目前案例的型別，即 System.Byte。
GetTypeCode()	傳回 byte 型別的 TypeCode，即 TypeCode 列舉的 Byte 成員，值為 6。
Parse(str)	將 string 型別的參數 <i>str</i> 轉換成 byte 型別，然後傳回，例如 System.Byte.Parse("65") 會傳回型別為 byte 的數值 65。
ToString()	將目前案例的值轉換成 string 型別，然後傳回，比方說，假設 byte 型別變數 <i>X</i> 的值為 100，那麼 <i>X.ToString()</i> 會傳回字串 "100"。
TryParse(str, result)	將 string 型別的參數 <i>str</i> 轉換成 byte 型別，若轉換結果包含參數 <i>result</i> 所指定的 byte 值，就傳回 true，否則傳回 false。

■ A-3 sbyte 型別的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 sbyte 型別的最大值 (127)。
MinValue	這個常數是 sbyte 型別的最小值 (-128)。
這個結構的公有方法和 byte 型別相似，不同的是 GetType() 會傳回 System.SByte，GetTypeCode() 會傳回 TypeCode 列舉的 SByte 成員，值為 5。	

■ A-4 short 型別的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 short 型別的最大值 (32767)。
MinValue	這個常數是 short 型別的最小值 (-32768)。
這個結構的公有方法和 byte 型別相似，不同的是 GetType() 會傳回 System.Int16，GetTypeCode() 會傳回 TypeCode 列舉的 Int16 成員，值為 7。	

■ A-5 ushort 型別的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 ushort 型別的最大值 (65535)。
MinValue	這個常數是 ushort 型別的最小值 (0)。
這個結構的公有方法和 byte 型別相似，不同的是 GetType() 會傳回 System.UInt16，GetTypeCode() 會傳回 TypeCode 列舉的 UInt16 成員，值為 8。	

A-6 int 型別的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 int 型別的最大值 (2147483647)。
MinValue	這個常數是 int 型別的最小值 (-2147483648)。
這個結構的公有方法和 byte 型別相似，不同的是 GetType() 會傳回 System.Int32，GetTypeInfo() 會傳回 TypeInfo 列舉的 Int32 成員，值為 9。	

A-7 uint 型別的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 uint 型別的最大值 (4294967295)。
MinValue	這個常數是 uint 型別的最小值 (0)。
這個結構的公有方法和 byte 型別相似，不同的是 GetType() 會傳回 System.UInt32，GetTypeInfo() 會傳回 TypeInfo 列舉的 UInt32 成員，值為 10。	

A-8 long 型別的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 long 型別的最大值 (9223372036854775807)。
MinValue	這個常數是 long 型別的最小值 (-9223372036854775808)。
這個結構的公有方法和 byte 型別相似，不同的是 GetType() 會傳回 System.Int64，GetTypeInfo() 會傳回 TypeInfo 列舉的 Int64 成員，值為 11。	

A-9 ulong 型別的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 <code>ulong</code> 型別的最大值 (18446744073709551615)。
MinValue	這個常數是 <code>ulong</code> 型別的最小值 (0)。
這個結構的公有方法和 <code>byte</code> 型別相似，不同的是 <code>GetType()</code> 會傳回 <code>System.UInt64</code> ， <code>GetTypeInfoCode()</code> 會傳回 <code>TypeInfoCode</code> 列舉的 <code>UInt64</code> 成員，值為 12。	

A-10 float 型別的成員

名稱	說明
公有欄位 (Public Field)	
Epsilon	這個常數是 <code>float</code> 型別的最小非零正值 (1.4E-45)。
MaxValue	這個常數是 <code>float</code> 型別的最大值 (3.402823E38)。
MinValue	這個常數是 <code>float</code> 型別的最小值 (-3.402823E+38)。
NaN	這個常數是 0.0 除以 0 的結果，當運算式的值沒有定義時，就會傳回 NaN (Not a Number)。
NegativeInfinity	這個常數是負浮點數除以 0 的結果，即負無限大。
PositiveInfinity	這個常數是正浮點數除以 0 的結果，即正無限大。
公有方法 (Public Method)	
IsInfinity(exp)	若 <code>exp</code> 的值為正負無限大，就傳回 <code>true</code> ，否則傳回 <code>false</code> 。
InNaN(exp)	若 <code>exp</code> 的值為 NaN，就傳回 <code>true</code> ，否則傳回 <code>false</code> 。
IsNegativeInfinity(exp)	若 <code>exp</code> 的值為負無限大，就傳回 <code>true</code> ，否則傳回 <code>false</code> 。
IsPositiveInfinity(exp)	若 <code>exp</code> 的值為正無限大，就傳回 <code>true</code> ，否則傳回 <code>false</code> 。
其它公有方法和 <code>byte</code> 型別相似，不同的是 <code>GetType()</code> 會傳回 <code>System.Single</code> ， <code>GetTypeInfoCode()</code> 會傳回 <code>TypeInfoCode</code> 列舉的 <code>Single</code> 成員，值為 13。	

A-11 double 型別的成員

名稱	說明
公有欄位 (Public Field)	
Epsilon	這個常數是 double 型別的最小非零正值 (4.94065645841247E-324)。
MaxValue	這個常數是 double 型別的最大值 (1.79769313486232E+308)。
MinValue	這個常數是 double 型別的最小值 (-1.79769313486232E+308)。
NaN	這個常數是 0.0 除以 0 的結果，當運算式的值沒有定義時，就會傳回 NaN (Not a Number)。
NegativeInfinity	這個常數是負浮點數除以 0 的結果，即負無限大。
PositiveInfinity	這個常數是正浮點數除以 0 的結果，即正無限大。
公有方法 (Public Method)	
IsInfinity(<i>exp</i>)	若 <i>exp</i> 的值為正負無限大，就傳回 true，否則傳回 false。
InNaN(<i>exp</i>)	若 <i>exp</i> 的值為 NaN，就傳回 true，否則傳回 false。
IsNegativeInfinity(<i>exp</i>)	若 <i>exp</i> 的值為負無限大，就傳回 true，否則傳回 false。
IsPositiveInfinity(<i>exp</i>)	若 <i>exp</i> 的值為正無限大，就傳回 true，否則傳回 false。
其它公有方法和 byte 型別相似，不同的是 GetType() 會傳回 System.Double，GetTypeCode() 會傳回 TypeCode 列舉的 Double 成員，值為 14。	

A-12 decimal 型別的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 decimal 型別的最大值 (79228162514264337593543950335)。
MinusOne	這個常數是 -1。
MinValue	這個常數是 decimal 型別的最小值 (-79228162514264337593543950335)。
One	這個常數是 1。
Zero	這個常數是 0。
公有方法 (Public Method)	
Add(dec1, dec2)	傳回參數 dec1 加上參數 dec2 的結果，型別為 decimal。
Compare(dec1, dec2)	傳回參數 dec1 和參數 dec2 的比較結果，型別為 int，小於 0 表示參數 dec1 比較小，等於 0 表示兩個參數相等，大於 0 表示參數 dec1 比較大。
Divide(dec1, dec2)	傳回參數 dec1 除以參數 dec2 的商數，型別為 decimal。
Equals(dec1, dec2)	傳回參數 dec1 和參數 dec2 是否相等，型別為 bool。
Floor(dec)	傳回比參數小 1 的整數，若參數無小數部分，則傳回該參數，例如 System.Decimal.Floor(12345678901234567.8D) 會傳回 12345678901234567，而 System.Decimal.Floor(-12345678901234567.8D) 會傳回 -12345678901234568。
FromOACurrency(Ing)	將包含 OLE Automation 貨幣值的參數 Ing 轉換成 decimal 型別，然後傳回。
GetBits(dec)	傳回參數 dec 轉換成二進位數字的結果，型別為 int 陣列。
Multiply(dec1, dec2)	傳回參數 dec1 乘以參數 dec2 的結果，型別為 decimal。
Negate(dec)	傳回參數 dec 轉換成負數的結果，型別為 decimal。

名稱	說明
<code>Reminder(dec1, dec2)</code>	傳回參數 <i>dec1</i> 除以參數 <i>dec2</i> 的餘數，型別為 <code>decimal</code> 。
<code>Round(dec, i)</code>	傳回參數 <i>dec</i> 四捨五入至參數 <i>i</i> 所指定的小數點位數，例如 <code>System.Decimal.Round(12345.6789D, 3)</code> 會傳回 <code>12345.679</code> (型別為 <code>decimal</code>)。
<code>Subtract(dec1, dec2)</code>	傳回參數 <i>dec1</i> 減去參數 <i>dec2</i> 的結果，型別為 <code>decimal</code> 。
<code>Truncate(dec)</code>	傳回參數 <i>dec</i> 捨去小數部分的結果，型別為 <code>decimal</code> 。
<code>ToByte(dec)</code> 、 <code>ToDouble(dec)</code> 、 <code>ToInt16(dec)</code> 、 <code>ToInt32(dec)</code> 、 <code>ToInt64(dec)</code> 、 <code>ToOACurrency(dec)</code> 、 <code>ToSByte(dec)</code> 、 <code>ToSingle(dec)</code> 、 <code>ToUInt16(dec)</code> 、 <code>ToUInt32(dec)</code> 、 <code>ToUInt64(dec)</code> 等方法可以將參數 <i>dec</i> 轉換成 <code>byte</code> 、 <code>double</code> 、 <code>short</code> 、 <code>int</code> 、 <code>long</code> 、 <code>long</code> 、 <code>sbyte</code> 、 <code>float</code> 、 <code>ushort</code> 、 <code>uint</code> 、 <code>ulong</code> 型別。	
其它公有方法和 <code>byte</code> 型別相似，不同的是 <code>GetType()</code> 會傳回 <code>System.Decimal</code> ， <code>GetTypeCode()</code> 會傳回 <code>TypeCode</code> 列舉的 <code>Decimal</code> 成員，值為 <code>15</code> 。	
公有運算子與型別轉換 (Public Operator and Type Conversion)	
<code>op_Addition(dec1, dec2)</code>	傳回參數 <i>dec1</i> 加上參數 <i>dec2</i> 的結果，型別為 <code>decimal</code> 。
<code>op_Decrement(dec)</code>	傳回參數 <i>dec</i> 減去 1 的結果，型別為 <code>decimal</code> 。
<code>op_Division(dec1, dec2)</code>	傳回參數 <i>dec1</i> 除以參數 <i>dec2</i> 的商數，型別為 <code>decimal</code> 。
<code>op_Equality(dec1, dec2)</code> 、 <code>op_GreaterThan(dec1, dec2)</code> 、 <code>op_GreaterThanOrEqual(dec1, dec2)</code> 、 <code>op_Inequality(dec1, dec2)</code> 、 <code>op_LessThan(dec1, dec2)</code> 、 <code>op_LessThanOrEqual(dec1, dec2)</code> 等方法會傳回參數 <i>dec1</i> 是否等於、大於、大於等於、不等於、小於、小於等於參數 <i>dec2</i> ，型別為 <code>bool</code> 。	
<code>op_Increment(dec)</code>	傳回參數 <i>dec</i> 加上 1 的結果，型別為 <code>decimal</code> 。
<code>op_Modulus(dec1, dec2)</code>	傳回參數 <i>dec1</i> 除以參數 <i>dec2</i> 的餘數，型別為 <code>decimal</code> 。
<code>op_Multiply(dec1, dec2)</code>	傳回參數 <i>dec1</i> 乘以參數 <i>dec2</i> 的結果，型別為 <code>decimal</code> 。
<code>op_Subtraction(dec1, dec2)</code>	傳回參數 <i>dec1</i> 減去參數 <i>dec2</i> 的結果，型別為 <code>decimal</code> 。
<code>op_UnaryNegation(dec)</code>	傳回參數 <i>dec</i> 轉換成負數的結果，型別為 <code>decimal</code> 。
<code>op_UnaryPlus(dec)</code>	傳回參數 <i>dec</i> 轉換成正數的結果，型別為 <code>decimal</code> 。

A-13 bool 型別的成員

名稱	說明
公有欄位 (Public Field)	
bool 型別的成員	這個常數是 "False" 字串。
TrueString	這個常數是 "True" 字串。
其它公有方法和 byte 型別相似，不同的是 GetType() 會傳回 System.Boolean，GetTypeCode() 會傳回 TypeCode 列舉的 Boolean 成員，值為 3。	

A-14 object 型別的成員

名稱	說明
公有方法 (Public Method)	
Equals(obj) Equals(obj1, obj2)	前者是判斷目前案例與參數 <i>obj</i> 是否相等，後者是判斷參數 <i>obj1</i> 和參數 <i>obj2</i> 是否相等，是就傳回 true，否則傳回 false。
ReferenceEquals(obj1, obj2)	判斷參數 <i>obj1</i> 和參數 <i>obj2</i> 是否為相同案例或皆為 null 參考，是就傳回 true，否則傳回 false。
GetHashCode()	傳回目前案例的雜湊碼 (hash code)，型別為 int。
GetType()	傳回目前案例的型別，即 System.Object。
ToString()	將目前案例的值轉換成 string 型別。
保護方法 (Protected Method)	
Finalize()	釋放目前案例佔用的資源。
MemberwiseClone()	複製目前案例。

A-15 char 型別的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 char 型別的最大值 (65535)。
MinValue	這個常數是 char 型別的最小值 (0)。
公有方法 (Public Static Method)	
GetNumericValue(chr) GetNumericValue(str, i)	前者可以傳回字元參數 <i>chr</i> 轉換成 double 型別的數值；後者可以傳回字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元轉換成 double 型別的數值。
GetUnicodeCategory(chr) GetUnicodeCategory(str, i)	前者可以傳回字元參數 <i>chr</i> 的 Unicode 分類常數；後者可以傳回字串參數 <i>str</i> 中第 <i>i</i> 個位置之字元的 Unicode 分類常數。例如 Char.GetUnicodeCategory('a') 會傳回 "LowercaseLetter"，Char.GetUnicodeCategory("I Am Happy.", 6) 會傳回 "UppercaseLetter"。
IsControl(chr) IsControl(str, i)	前者可以判斷字元參數 <i>chr</i> 是否為控制字元，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為控制字元，是就傳回 true，否則傳回 false。例如 Char.IsControl(Char(9)) 會傳回 true，因為 Char(9) 為 Tab 字元，而 Char.IsControl("Happy", 3) 會傳回 false。
IsDigit(chr) IsDigit(str, i)	前者可以判斷字元參數 <i>chr</i> 是否為十進位數字，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為十進位數字，是就傳回 true，否則傳回 false。
IsLetter(chr) IsLetter(str, i)	前者可以判斷字元參數 <i>chr</i> 是否為英文字母，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為英文字母，是就傳回 true，否則傳回 false。
IsLetterOrDigit(chr) IsLetterOrDigit(str, i)	前者可以判斷字元參數 <i>chr</i> 是否為英文字母或十進位數字，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為英文字母或十進位數字，是就傳回 true，否則傳回 false。

名稱	說明
IsNumber(<i>chr</i>) IsNumber(<i>str</i> , <i>i</i>)	前者可以判斷字元參數 <i>chr</i> 是否為十進位數字或十六進位數字，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為十進位數字或十六進位數字，是就傳回 true ，否則傳回 false 。
IsLower(<i>chr</i>) IsLower(<i>str</i> , <i>i</i>)	前者可以判斷字元參數 <i>chr</i> 是否為小寫字母，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為小寫字母，是就傳回 true ，否則傳回 false 。
IsPunctuation(<i>chr</i>) IsPunctuation(<i>str</i> , <i>i</i>)	前者可以判斷字元參數 <i>chr</i> 是否為標點符號，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為標點符號，是就傳回 true ，否則傳回 false 。
IsSeparator(<i>chr</i>) IsSeparator(<i>str</i> , <i>i</i>)	前者可以判斷字元參數 <i>chr</i> 是否為分隔字元，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為分隔字元，是就傳回 true ，否則傳回 false 。
IsSurrogate(<i>chr</i>) IsSurrogate(<i>str</i> , <i>i</i>)	前者可以判斷字元參數 <i>chr</i> 是否為代理字元，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為代理字元，是就傳回 true ，否則傳回 false 。
IsSymbol(<i>chr</i>) IsSymbol(<i>str</i> , <i>i</i>)	前者可以判斷字元參數 <i>chr</i> 是否為符號字元，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為符號字元，是就傳回 true ，否則傳回 false 。
IsUpper(<i>chr</i>) IsUpper(<i>str</i> , <i>i</i>)	前者可以判斷字元參數 <i>chr</i> 是否為大寫字母，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為大寫字母，是就傳回 true ，否則傳回 false 。
IsWhiteSpace(<i>chr</i>) IsWhiteSpace(<i>str</i> , <i>i</i>)	前者可以判斷字元參數 <i>chr</i> 是否為空白字元，後者可以判斷字串參數 <i>str</i> 中第 <i>i</i> 個位置的字元是否為空白字元，是就傳回 true ，否則傳回 false 。
ToLower(<i>chr</i>)	將字元參數 <i>chr</i> 轉換成對應的小寫字母，然後傳回。
ToUpper(<i>chr</i>)	將字元參數 <i>chr</i> 轉換成對應的大寫字母，然後傳回。
其它公有方法和 byte 型別相似，不同的是 GetType() 會傳回 System.Char ， GetTypeCode() 會傳回 TypeCode 列舉的 Char 成員，值為 4。	

A-16 string 型別的成員

名稱	說明
公有欄位 (Public Field)	
Empty	這個常數是空白字串 ("")。
公有屬性 (Public Property)	
Chars(<i>i</i>)	傳回目前案例位於參數 <i>i</i> 所指定位置的字元，型別為 <code>char</code> 。
Length	傳回目前案例的字串長度 (字元數)，型別為 <code>int</code> 。
公有方法 (Public Method)	
Compare(<i>str1</i> , <i>str2</i>) Compare(<i>str1</i> , <i>str2</i> , <i>boolean</i>) Compare(<i>str1</i> , <i>str2</i> , <i>boolean</i> , <i>CultureInfo</i>) Compare(<i>str1</i> , <i>int1</i> , <i>str2</i> , <i>int2</i> , <i>int3</i>) Compare(<i>str1</i> , <i>int1</i> , <i>str2</i> , <i>int2</i> , <i>int3</i> , <i>boolean</i>) Compare(<i>str1</i> , <i>int1</i> , <i>str2</i> , <i>int2</i> , <i>int3</i> , <i>boolean</i> , <i>CultureInfo</i>)	將兩個字串參數做比較，傳回值為 <code>int</code> 型別，-1 表示 <i>str1</i> 比較小，0 表示 <i>str1</i> 等於 <i>str2</i> ，1 表示 <i>str1</i> 比較大；若還要指定比較時是否忽略大小寫，可以使用第二種形式，也就是再加上布林參數 <i>boolean</i> ；若還要指定比較時須考慮語系或文化，可以使用第三種形式，也就是再加上參數 <i>CultureInfo</i> ；若要指定從 <i>str1</i> 的第 <i>int1</i> 個字元和從 <i>str2</i> 的第 <i>int2</i> 個字元開始做比較 <i>int3</i> 個字元，可以使用第四種形式；若還要指定比較時是否忽略大小寫，可以使用第五種形式，也就是再加上布林參數 <i>boolean</i> ；若還要指定比較時須考慮語系或文化，可以使用第六種形式，也就是再加上參數 <i>CultureInfo</i> 。
CompareOrdinal(<i>str1</i> , <i>str2</i>) CompareOrdinal(<i>str1</i> , <i>int1</i> , <i>str2</i> , <i>int2</i> , <i>int3</i>)	將兩個字串參數做比較，但不考慮語系或文化，傳回值為 <code>int</code> 型別；若要指定從 <i>str1</i> 的第 <i>int1</i> 個字元和從 <i>str2</i> 的第 <i>int2</i> 個字元開始做比較 <i>int3</i> 個字元，可以使用第二種形式。
CompareTo(<i>obj</i>) CompareTo(<i>str</i>)	傳回目前案例與參數比較的結果，傳回值為 <code>int</code> 型別，小於 0 表示目前案例比較小，等於 0 表示目前案例與參數相等，大於 0 表示目前案例比較大。

名稱	說明
Concat(obj) Concat (obj, obj) Concat(obj, obj, obj) Concat(str, str) Concat(str, str, str) Concat(str, str, str, str) Concat(params obj[]) Concat(params str[])	將字串參數或物件的值連結成一個字串，傳回值為 string 型別。
Clone()	傳回指向目前案例的參考，型別為 object 。
Copy(str)	根據參數的值建立一個新的 string 案例，型別為 string 。
CopyTo(int1, chr[], int2, int3)	將目前案例從 <i>int1</i> 指定的位置開始複製 <i>int3</i> 個字元到字元陣列參數 <i>chr[]</i> 內 <i>int2</i> 指定的位置。
EndsWith(str)	判斷目前案例是否以參數字串結尾，型別為 bool 。
Equals(obj) Equals (str) Equals(str, str)	傳回目前案例與參數是否相等的結果，第三種形式則是傳回兩個字串參數是否相等的結果，型別均為 bool 。
Format(str, obj) Format(str, params obj[]) Format(str, obj, obj) Format(str, obj, obj, obj)	根據第一個字串參數指定的格式將其它參數加以格式化。
GetEnumerator()	將目前案例轉換為 CharEnumerator 型別。
IndexOf(chr) IndexOf (str) IndexOf(chr, int1) IndexOf(str, int1) IndexOf(chr, int1, int2) IndexOf (str, int1, int2)	傳回參數指定的字元或字串首次出現在目前案例的第幾個位置，型別為 int ；若還要指定從目前案例的第幾個位置開始比對，可以如第三、四種形式加上 <i>int1</i> 參數；若還要指定從目前案例的第幾個位置開始比對幾個字元，可以如第五、六種形式加上 <i>int1</i> 參數及 <i>int2</i> 參數。
Insert(i, str)	在目前案例的第 <i>i</i> 個位置插入參數 <i>str</i> ，型別為 string 。

名稱	說明
IndexOfAny(<i>chr</i>[]) IndexOfAny(<i>chr</i>[], <i>int1</i>) IndexOfAny(<i>chr</i>[], <i>int1</i>, <i>int2</i>)	傳回字元陣列參數 <i>chr</i> [] 內任意字元首次出現在目前案例的第幾個位置，型別為 int ；若還要指定從目前案例的第幾個位置開始比對，可以如第二種形式加上參數 <i>int1</i> ；若還要指定從目前案例的第幾個位置開始比對幾個字元，可以如第三種形式加上參數 <i>int1</i> 及參數 <i>int2</i> 。
Intern(<i>str</i>)	傳回指向字串參數的系統參考 (Reference)，型別為 string 。
IsInterned(<i>str</i>)	傳回指向字串參數的參考 (Reference)，型別為 string 。
Join(<i>str</i>, <i>str</i>[])	將第二個字串陣列參數內的每個元素值以第一個參數指定的字串連結在一起，成為一個字串，然後將結果傳回。
LastIndexOf(<i>chr</i>) LastIndexOf(<i>str</i>) LastIndexOf(<i>chr</i>, <i>int1</i>) LastIndexOf(<i>str</i>, <i>int1</i>) LastIndexOf(<i>chr</i>, <i>int1</i>, <i>int2</i>) LastIndexOf(<i>str</i>, <i>int1</i>, <i>int2</i>)	傳回參數指定的字元或字串最後一次出現在目前案例的第幾個位置，型別為 int ；若還要指定從目前案例的第幾個位置開始比對，可以如第三、四種形式加上參數 <i>int1</i> ；若還要指定從目前案例的第幾個位置開始比對幾個字元，可以如第五、六種形式加上參數 <i>int1</i> 及參數 <i>int2</i> 。
LastIndexOfAny(<i>chr</i>[]) LastIndexOfAny(<i>chr</i>[], <i>int1</i>) LastIndexOfAny(<i>chr</i>[], <i>int1</i>, <i>int2</i>)	傳回字元陣列參數 <i>chr</i> [] 內任意字元最後一次出現在目前案例的第幾個位置，型別為 int ；若還要指定從目前案例的第幾個位置開始比對，可以如第二種形式加上參數 <i>int1</i> ；若還要指定從目前案例的第幾個位置開始比對幾個字元，可以如第三種形式加上參數 <i>int1</i> 及參數 <i>int2</i> 。
PadLeft(<i>int1</i>) PadLeft(<i>int1</i>, <i>chr</i>)	將目前案例向右對齊，長度維持為 <i>int1</i> ，不足的部分在字串的左邊補上空白字元或 <i>chr</i> 字元，型別為 string 。
PadRight(<i>int1</i>) PadRight(<i>int1</i>, <i>chr</i>)	將目前案例向左對齊，長度維持為 <i>int1</i> ，不足的部分在字串的右邊補上空白字元或 <i>chr</i> 字元，型別為 string 。
Remove(<i>int1</i>, <i>int2</i>)	將目前案例的第 <i>int1</i> 個位置刪除 <i>int2</i> 個字元，型別為 string 。

名稱	說明
Replace(<i>chr1</i> , <i>chr2</i>) Replace(<i>str1</i> , <i>str2</i>)	將目前案例的所有 <i>chr1</i> 、 <i>str1</i> 分別以 <i>chr2</i> 、 <i>str2</i> 取代，型別為 string 。
Split(<i>chr</i> []) Split(<i>chr</i> [], <i>int1</i>)	將目前案例中以字元陣列參數內任意字元分隔開來的子字串放入一個字串陣列，傳回值型別為 string []，若要限制傳回之字串陣列的元素個數，可以加上第二個參數 <i>int1</i> 。
StartsWith(<i>str</i>)	判斷目前案例是否以參數字串開頭，型別為 bool 。
SubString(<i>int1</i>) SubString(<i>int1</i> , <i>int2</i>)	傳回目前案例從第 <i>int1</i> 個位置開始的子字串，若要限制子字串的字元個數，可以加上參數 <i>int2</i> 。
ToCharArray() ToCharArray(<i>int1</i> , <i>int2</i>)	將目前案例的每個字元放入字元陣列，或將目前案例第 <i>int1</i> 個位置起的 <i>int2</i> 個字元放入字元陣列，然後將結果傳回。
ToLower()	將目前案例轉換成小寫字母，型別為 string 。
ToUpper()	將目前案例轉換成大寫字母，型別為 string 。
Trim()、Trim(<i>chr</i> [])	從目前案例移除空白字元或字元陣列參數 <i>chr</i> [] 內所指定的任意字元，然後將結果傳回，型別為 string 。
TrimEnd(<i>chr</i> [])	從目前案例的結尾開始移除字元陣列參數 <i>chr</i> [] 內所指定的任意字元，然後將結果傳回，型別為 string 。
TrimStart(<i>chr</i> [])	從目前案例的開頭開始移除字元陣列參數 <i>chr</i> [] 內所指定的任意字元，然後將結果傳回，型別為 string 。
其它公有方法和 byte 型別相似，不同的是 GetType() 會傳回 System.String ， GetTypeCode() 會傳回 TypeCode 列舉的 String 成員，值為 18。	
公有運算子 (Public Operator)	
op_Equality(<i>str1</i> , <i>str2</i>)	傳回兩個參數是否相等，型別為 bool 。
op_Inequality(<i>str1</i> , <i>str2</i>)	傳回兩個參數是否不相等，型別為 bool 。

A-17 System.DateTime 結構的成員

名稱	說明
公有欄位 (Public Field)	
MaxValue	這個常數是 DateTime 結構的最大值 (11:59:59 PM, 12/31/9999)。
MinValue	這個常數是 DateTime 結構的最小值 (12:00:00 AM, 1/1/0001)。
公有屬性 (Public Property)	
DayOfWeek	傳回目前案例為星期幾，屬於 DayOfWeek 列舉，其中 0、1、2 ~ 6 分別表示星期日 (Sunday)、星期一 (Monday)、星期二 (Tuesday) ~ 六 (Saturday)。
DayOfYear	傳回目前案例為一年中的第幾天，例如 1/5/2020 是 2020 年的第 5 天。
Now	傳回系統目前的日期時間，型別為 DateTime 結構。
Ticks	傳回目前案例與 1/1/0001 12:00:00AM 間隔幾個千萬分之一秒 (10^{-7} 秒、100ns)，型別為 long，換句話說，1Tick 等於 10^{-7} 秒或 100ns，而 1ns 等於 10^{-9} 秒，例如 1/1/0001 12:00:01AM 與 1/1/0001 12:00:00AM 間隔 1 秒，即 10^7 個千萬分之一秒 (10^7 Ticks)。
TimeOfDay	傳回從凌晨零點零分零秒 (12:00:00 AM) 到目前案例所經過的時間，型別為 TimeSpan 結構。
Today	傳回系統目前的日期，型別為 DateTime 結構。
UtcNow	以 UTC 格式傳回系統目前的日期時間，型別為 DateTime 結構。
其它屬性還有 Date、Day、Hour、Millisecond、Minute、Month、Second、Year 可以分別傳回目前案例的年月日、日、小時、毫秒 (千分之一秒、 10^{-3} 秒、ms)、分鐘、月份、秒、年份。	

名稱	說明
公有方法 (Public Method)	
Add(ts)	傳回目前案例加上參數 ts (時間間隔) 的結果，參數的型別為 TimeSpan ，傳回值型別為 DateTime ， TimeSpan 指的是一段時間，單位為 Tick ，1 Tick = 100ns。
AddDays(dbl)	傳回目前案例加上參數 dbl (天數) 的結果，參數的型別為 double ，傳回值的型別為 DateTime 。
AddHours(dbl)	傳回目前案例加上參數 dbl (小時數) 的結果，參數的型別為 double ，傳回值的型別為 DateTime 。
AddMilliseconds(dbl)	傳回目前案例加上參數 dbl (毫秒數) 的結果，參數的型別為 double ，傳回值的型別 DateTime 。
AddMinutes(dbl)	傳回目前案例加上參數 dbl (分鐘數) 的結果，參數的型別為 double ，傳回值的型別為 DateTime 。
AddMonths(int1)	傳回目前案例加上參數 int1 (月數) 的結果，參數的型別為 int ，傳回值的型別為 DateTime 。
AddSeconds(dbl)	傳回目前案例加上參數 dbl (秒數) 的結果，參數的型別為 double ，傳回值的型別為 DateTime 。
AddTicks lng)	傳回目前案例加上參數 (Tick 數) 的結果，參數的型別為 long ，傳回值的型別為 DateTime 。
AddYears(int1)	傳回目前案例加上參數 int1 (年數) 的結果，參數的型別為 int ，傳回值的型別為 DateTime 。
Compare(dt1, dt2)	將兩個型別為 DateTime 的參數做比較，傳回值為 int 型別，小於 0 表示第一個參數小於第二個參數，等於 0 表示第一個參數等於第二個參數，大於 1 表示第一個參數大於第二個參數。
CompareTo(obj)	將目前案例與參數 obj 做比較，傳回值為 int 型別，小於 0 表示目前案例小於參數 obj ，等於 0 表示目前案例等於參數，大於 0 表示目前案例大於參數 obj 。
Equals(obj) Equals(dt1, dt2)	前者可以判斷目前案例與參數是否相等，後者可以判斷兩個參數是否相等，是就傳回 true ，否則傳回 false 。

名稱	說明
<code>DaysInMonth(int1, int2)</code>	參數 <i>int1</i> 為年份，參數 <i>int2</i> 為月份，傳回值為 <i>int1</i> 年份之 <i>int2</i> 那個月份的天數 (int 型別)，例如 <code>DateTime.DaysInMonth(2016, 2)</code> 會傳回 29。
<code>FromFileTime(Ing)</code>	將型別為 long 的檔案日期參數 <i>Ing</i> 轉換成 <code>DateTime</code> 型別。
<code>FromOAdate(dbl)</code>	將 OLE Automation Date 參數 <i>dbl</i> 轉換成 <code>DateTime</code> 型別。
<code>GetDateTimeFormats(IFormatProvider)</code>	將目前案例依照指定格式轉換成 string 型別。
<code>IsLeapYear(int1)</code>	判斷整數參數 <i>int1</i> 是否為西元閏年，傳回值為 bool 型別。
<code>Parse(str)</code>	將參數 <i>str</i> 轉換成 <code>DateTime</code> 型別。
<code>ParseExact(str1, str2, IFormatProvider)</code>	將 string 型別的參數依照指定格式轉換成 <code>DateTime</code> 型別。
<code>Subtract(dt)</code> <code>Subtract(ts)</code>	前者可以傳回目前案例減去日期時間參數 <i>dt</i> 的時間間隔，型別為 <code>TimeSpan</code> ；後者可以傳回目前案例減去時間間隔參數 <i>ts</i> 的日期時間，型別為 <code>DateTime</code> 。
<code>ToFileTime()</code>	將目前案例轉換為本機系統檔案時間，型別為 long。
<code>ToLongDateString()</code>	將目前案例的日期轉換為長日期字串，型別為 string。
<code>ToLongTimeString()</code>	將目前案例的時間轉換為長時間字串，型別為 string。
<code>ToOAdate()</code>	將目前案例轉換為 OLE Automation Date，型別為 double。
<code>ToShortDateString()</code>	將目前案例的日期轉換為短日期字串，型別為 string。
<code>ToShortTimeString()</code>	將目前案例的時間轉換為短時間字串，型別為 string。
<code>ToUniversalTime()</code>	將目前案例轉換為 UTC 格式，型別為 string。
其它公有方法和 byte 型別相似，不同的是 <code>GetType()</code> 會傳回 <code>System.DateTime</code> ， <code>GetTypeCode()</code> 會傳回 <code>TypeCode</code> 列舉的 <code>DateTime</code> 成員，值為 16。	

名稱	說明
Public Static Operator (公有運算子)	
op_Addition(dt, ts)	傳回第一個 DateTime 參數加上第二個 TimeSpan 時間間隔參數的結果，型別為 DateTime ，一般是使用 Add() 方法取代此運算子。
op_Equality(dt1, dt2)	傳回第一個 DateTime 參數是否等於第二個 DateTime 參數，型別為 bool ，一般是使用 Equals() 方法取代此運算子。
op_GreaterThan(dt1, dt2)	傳回第一個 DateTime 參數是否大於第二個 DateTime 參數，型別為 bool ，一般是使用 Compare() 方法取代此運算子。
op_GreaterThanOrEqual (dt1, dt2)	傳回第一個 DateTime 參數是否大於等於第二個 DateTime 參數，型別為 bool ，一般是使用 Compare() 方法取代此運算子。
op_Inequality(dt1, dt2)	傳回第一個 DateTime 參數是否不等於第二個 DateTime 參數，型別為 bool ，一般是使用 Equals() 方法取代此運算子。
op_LessThan(dt1, dt2)	傳回第一個 DateTime 參數是否小於第二個 DateTime 參數，型別為 bool ，一般是使用 Compare() 方法取代此運算子。
op_LessThanOrEqual (dt1, dt2)	傳回第一個 DateTime 參數是否小於等於第二個 DateTime 參數，型別為 bool ，一般是使用 Compare() 方法取代此運算子。
op_Subtraction(dt1, dt2) op_Subtraction (dt, ts)	前者可以傳回第一個 DateTime 參數減去第二個 DateTime 參數的時間間隔，型別為 TimeSpan ；後者可以傳回第一個 DateTime 參數減去 TimeSpan 時間間隔參數的日期時間，型別為 DateTime ，一般是使用 Subtract() 方法取代此運算子。