

A-1 資料型別函式

A-1-1 取得資料型別函式

✓ 語法

```
TypeName(varName As Object) As String
```

✓ 說明

TypeName() 函式可以用來取得參數 *varName* 的資料型別，傳回值為字串。舉例來說，假設變數 A 的型別為 Integer，那麼 TypeName(A)、TypeName(True)、TypeName(1234567.89)、TypeName("Happy")、TypeName(#1/1/2002#)、TypeName(Chr(65)) 的傳回值分別為 "Integer"、"Boolean"、"Double"、"String"、"Date"、"Char"。

A-1-2 檢查資料型別函式

| 函式 | 說明 |
|---------------------------|---|
| IsNothing(<i>obj</i>) | 檢查參數是否已經初始化，若物件變數沒有指向任何真實物件，傳回 True，否則傳回 False。 |
| IsDBNull(<i>obj</i>) | 檢查參數是否包含任何有效資料，是的話，傳回 True，否則傳回 False。請注意，DBNull 不同於 Nothing 或空字串 ""。 |
| IsError(<i>obj</i>) | 檢查參數是否為錯誤代碼，是的話，傳回 True，否則傳回 False。 |
| IsDate(<i>obj</i>) | 檢查參數是否為 Date 型別或可以轉換成為 Date 型別的字串，是的話，傳回 True，否則傳回 False。 |
| IsNumeric(<i>obj</i>) | 檢查參數是否為數值型別 (Byte、Short、Integer、Long、Single、Double、Decimal)，是的話，傳回 True，否則傳回 False。 |
| IsReference(<i>obj</i>) | 檢查參數是否為 Object 型別，是的話，傳回 True，否則傳回 False。 |
| IsArray(<i>obj</i>) | 檢查參數是否為陣列，是的話，傳回 True，否則傳回 False。 |

A-1-3 轉換資料型別函式

這些函式的參數均為字串運算式或數值運算式，例如 CStr(#1/1/2002#) 會傳回 "2002/1/1"，因為這個函式的功能就是將參數轉換成 String 型別並傳回來。要注意的是若參數超出型別轉換函式的型別所允許的範圍或參數的格式難以辨識，將會產生錯誤。

| 轉換函式 | 傳回值型別 | 說明 |
|----------------------|---------|--|
| CBool(<i>exp</i>) | Boolean | 將參數轉換成 Boolean (布林) 型別並傳回來。 |
| CByte(<i>exp</i>) | Byte | 將參數轉換成 Byte 型別，參數須為 0 ~ 255，小數部分會被四捨五入。 |
| CChar(<i>exp</i>) | Char | 將參數轉換成 Char 型別，參數須為 0 ~ 65,535。 |
| CDate(<i>exp</i>) | Date | 將參數轉換成 Date 型別，參數須為有效的日期時間表示方式。 |
| CDbl(<i>exp</i>) | Double | 將參數轉換成 Double 型別，參數須為 -1.79769313486231E+308 ~ -4.94065645841247E-324 的負數或 4.94065645841247E-324 ~ 1.79769313486231E+308 的正數。 |
| CDec(<i>exp</i>) | Decimal | 將參數轉換成 Decimal 型別，參數須為 +/-79,228,162,514,264,337,593,543,950,335 間的整數、 +/-7.9228162514264337593543950335 間的浮點數或最小的非零值 +/-0.000000000000000000000001。 |
| CInt(<i>exp</i>) | Integer | 將參數轉換成 Integer 型別，參數須為 -2,147,483,648 ~ 2,147,483,647 的整數，小數部分會被四捨五入。 |
| CLng(<i>exp</i>) | Long | 將參數轉換成 Long 型別，參數須為 -9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 的整數，小數部分會被四捨五入。 |
| CObj(<i>exp</i>) | Object | 將參數轉換成 Object 型別，參數須為有效的運算式。 |
| CShort(<i>exp</i>) | Short | 將參數轉換成 Short 型別，參數須為 -32,768 ~ 32,767 的整數，小數部分會被四捨五入。 |

| 轉換函式 | 傳回值型別 | 說明 |
|---|------------------------|--|
| CSng(<i>exp</i>) | Single | 將參數轉換成 Single 型別，參數須為 -3.402823E+38 ~ -1.401298E-45 的負數或 1.401298E-45 ~ 3.402823E+38 的正數。 |
| CStr(<i>exp</i>) | String | 將參數轉換成 String 型別，要注意的是若參數為 Boolean 型別，則傳回的字串為 "True" 或 "False"，若參數為 Date 型別，則傳回值將以系統的短日期格式顯示。 |
| CType | | 這個函式的呼叫形式為 CType(<i>expression</i> , <i>typename</i>)，也就是將第一個參數轉換成第二個參數所指定型別，然後傳回來。 |
| Asc(<i>var</i>) AscW(<i>var</i>) | Integer | 這兩個函式的參數可以是 Char 或 String 型別，傳回值為字元或字串第一個字元的字碼指標，例如 Asc("1") 和 Asc("123") 均會傳回 "1" 的字碼指標 49。 |
| Chr(<i>int</i>) ChrW(<i>int</i>) | Char | 這個函式的參數為 Integer 型別，代表的是字碼指標，傳回值是此字碼指標所代表的字元，例如 Chr(65) 會傳回字元 A。 |
| Hex(<i>num</i>) | String | 這個函式的參數為任意數值型別，傳回值是將參數轉換成十六進位（最多有 8 個十六進位數字），且型別為 String，若參數為 Empty，則傳回 "0"，例如 Hex(128) 會傳回字串 "80"。 |
| Oct(<i>num</i>) | String | 這個函式的參數為任意數值型別，傳回值是將參數轉換成八進位（最多有 11 個八進位數字），且型別為 String，若參數為 Empty，則傳回 "0"，例如 Oct(128) 會傳回字串 "200"。 |
| Str(<i>num</i>) | String | 將數值型別參數轉換成字串傳回來，例如 Str(-1234.567) 會傳回字串 "-1234.567"。 |
| Val(<i>str</i>) | Integer 或 Double | 這個函式的參數為包含數字的字串，作用是將字串內的數字轉換成 Integer 或 Double 型別傳回來，例如 Val("12.34 5") 會傳回 12.345，Val("12 dot34 5") 則會傳回 12。 |
| Fix(<i>num</i>) | 整數 | 傳回參數的整數部分，若參數小於 0，傳回大於等於參數的第一個負整數，例如 Fix(10.8)、Fix(-10.8) 會傳回 10、-10。 |
| Int(<i>num</i>) | 整數 | 傳回參數的整數部分，若參數小於 0，傳回小於等於參數的第一個負整數，例如 Int(10.8)、Int(-10.8) 會傳回 10、-11。 |

A-2 數學函式

| 函式 | 說明 |
|---|--|
| 取絕對值 Abs(num) | 傳回參數的絕對值，傳回值的型別與參數相同，參數可以是任何數值，例如 Abs(-123.456) 會傳回 123.456。 |
| 取自然對數之底數的某次方值 Exp(num) | 傳回自然對數之底數 e 的 次方值，e 的值約 2.718282。例如 Exp(2) 會傳回 e 的平方值 7.38905609893065。請注意，參數 num 的值不得超過 709.782712893，否則會產生溢位。 |
| 取 16 進位值 Hex(num) | 這個函式的參數為任意數值型別，傳回值是將參數轉換成十六進位 (最多有 8 個十六進位數字)，且型別為 String，若參數為 Empty，則傳回 "0"，例如 Hex(128) 會傳回字串 "80"。 |
| 取對數值 Log(num) | 傳回參數 num 的自然對數值，傳回值的型別為 Double，參數 num 可以是大於 0 的任何數值。例如 Log(Exp(2)) 會傳回 2。請注意，自然對數是以 e 為底數的對數，e 的值約 2.718282，若要以任意底數 n 來計算數值 x 的對數值，可以利用後式將 x 的自然對數值除以 n 的自然對數值： $\text{Log}_n(x) = \text{Log}(x) / \text{Log}(n)$ |
| 取 8 進位值 Oct(num) | 這個函式的參數為任意數值型別，傳回值是將參數轉換成八進位 (最多有 11 個八進位數字)，且型別為 String，若參數為 Empty，則傳回 "0"，例如 Oct(128) 會傳回字串 "200"。 |
| 取四捨五入值 Round(num) Round(num,int) | 傳回與參數 num 最接近的整數 (即四捨五入)，若要指定四捨五入至哪個小數位數，可以加上第二個參數 int。例如 Round(123.4567)、Round(123.4567, 3) 會分別傳回 123、123.457。 |
| 三角函式 Atan(num) Cos(num) Sin(num) Tan(num) | Atan() 會傳回參數的反正切 (Arctangent) 值，型別為 Double; Cos() 會傳回參數的餘弦 (Cosine) 值; Sin() 會傳回參數的正弦值; Tan() 會傳回角度的正切值。請注意，參數 num 必須為弧度，而不是角度，換言之，如果要計算 Sin30° 的值，必須先將角度轉換成弧度，公式為「弧度 = 角度 × π ÷ 180」，故 Sin30° 的值為 Sin(30 * 3.14158265358979 / 180)，將傳回 0.499998556623632 ≈ 0.5。 |
| 取正負符號 Sign(num) | 傳回參數的正負符號，若 num 大於 0，傳回值為 1；若 num 等於 0，傳回值為 0；若 num 小於 0，傳回值為 -1。 |

| 函式 | 說明 |
|-------------------|---|
| 取平方根 Sqrt(num) | 傳回參數的平方根，型別為 Double，例如 Sqrt(2) 會傳回 2 的平方根為 1.4142135623731。 |
| 產生亂數 Rnd[num] | <p>傳回一個小於 1、大於等於 0 的亂數，參數及傳回值的型別均為 Single。若參數 num 小於 0，那麼每次都會傳回相同的亂數；若沒有提供參數或參數 num 大於 0，那麼會依序產生下一個亂數，此為預設值；若參數 num 等於 0，那麼會傳回最近一次產生的亂數。</p> <p>請注意，在使用 Rnd() 函式產生亂數之前，最好先呼叫 Randomize 陳述式，但不要給任何參數，如此便會以作業系統的時間當作亂數種子來起始亂數產生器，而不會每次都產生相同的亂數。</p> <p>另外告訴您一個技巧，若要隨機產生一個 N 到 M 的整數，可以利用公式 $\text{Int}(\text{Rnd()} * M - N + 1) + N$。</p> |

A-3 日期時間函式

A-3-1 取得/設定目前日期時間函式

| 函式 | 說明 |
|--------------|---|
| Now() | 傳回目前系統的日期和時間，例如 2002/4/8 上午 10:25:36。這是一個唯讀屬性，不是函式。 |
| Today() | 傳回或設定目前系統的日期。這是一個屬性，不是函式。 |
| TimeOfDay() | 傳回或設定目前系統的時間。這是一個屬性，不是函式。 |
| DateString() | 以「yyyy-mm-dd」格式傳回或設定目前系統的日期，例如 2002/04/08。這是一個屬性，不是函式。 |
| TimeString() | 以「hh-mm-ss」格式傳回或設定目前系統的時間，例如 10:25:36。這是一個屬性，不是函式。 |
| Timer() | 以 Double 型別傳回自前一個午夜迄今所經過的秒數和毫秒數，秒數是傳回值的整數部份，毫秒則是小數部份。這是一個唯讀屬性，不是函式。 |

A-3-2 傳回日期時間函式

合併年月日成日期函式 DateSerial()

- ✓ 語法

```
DateSerial(year As Integer, month As Integer, day As Integer) As DateTime
```

- ✓ 說明

將參數所指定的年月日合併成日期傳回來。參數 *year* 的值須是介於 1 ~ 9999 的整數，若是介於 0 ~ 99 的整數，將被解譯為 1930 ~ 2029，若是小於 1 的整數，將自目前年度中減去；參數 *month* 的值須是介於 1 ~ 12 的整數，若是小於 1 的整數，將自目前月份中減去；參數 *day* 的值須是介於 1 ~ 31 的整數，若是小於 1 的整數，將自目前日數中減去。例如 DateSerial(99, 10, 22) 會傳回 #1999/10/22#，DateSerial(99, 10, 22 - 23) 會傳回 #1999/9/29#，DateSerial(99, 10, 22 + 3) 會傳回 #1999/10/25#。

將字串轉換成日期函式 DateValue()

- ✓ 語法

```
DateValue(str As String) As DateTime
```

- ✓ 說明

將參數轉換成日期傳回來。參數 *str* 是表示由 1 年 1 月 1 日 00:00:00 到 9999 年 12 月 31 日 23:59:59 之日期/時間值的字串運算式，例如 DateValue("February 12, 1972") 會傳回 #1972/2/12#。

合併時分秒成時間函式 TimeSerial()

- ✓ 語法

`TimeSerial(hour As Integer, minute As Integer, second As Integer) As DateTime`

- ✓ 說明

將參數所指定的時分秒合併成時間傳回來。參數 *hour* 的值須是介於 0 ~ 23 的整數，若是小於 1 的整數，將自目前小時數中減去；參數 *minute* 和參數 *second* 的值須是介於 0 ~ 59 的整數，若是小於 1 的整數，將分別自目前分鐘數與秒數中減去，例如 `TimeSerial(12 - 6, -15, 0)` 會傳回上午 05:45:00。

將字串轉換成時間函式 TimeValue()

- ✓ 語法

`TimeValue(str As String) As DateTime`

- ✓ 說明

將參數轉換成時間傳回來。參數 *str* 是表示由 1 年 1 月 1 日 00:00:00 到 9999 年 12 月 31 日 23:59:59 之日期/時間值的字串運算式，例如 `TimeValue("4:35:17 PM")` 會傳回 #下午 04:35:17#。

取得月份名稱函式 MonthName()

`MonthName(month As Integer[, abbreviate As Boolean]) As String`

- ✓ 說明

根據參數 *month* 的值傳回月份的名稱，例如 `MonthName(12)` 會傳回 "十二月" 字串，參數 *month* 的型別為 `Integer`；選擇性參數 *abbreviate* 的型別為 `Boolean`，用來表示月份的名稱能否被縮寫。

取得星期幾名稱函式 WeekdayName()

✓ 語法

```
WeekdayName(weekday As Integer [,abbreviate As Boolean [,dayofweek As _
FirstDayOfWeek]]) As String
```

✓ 說明

根據參數 *weekday* 的值傳回星期幾的名稱，在預設的情況下，1、2~7 會分別傳回 "星期日"、"星期一" ~ "星期六"，參數 *weekday* 的型別為 Integer；選擇性參數 *abbreviate* 的型別為 Boolean，用來表示星期幾的名稱能否被縮寫，而選擇性參數 *firstdayofweek* 的型別為 FirstDayOfWeek 列舉，用來表示一星期的第一天，預設為 FirstDayOfWeek.System (0)，即採取系統設定中所指定的每週第一天，通常是星期日。例如 WeekdayName(5, ,FirstDayOfWeek.Friday) 會傳回 "星期二"，因為第三個參數指定第一天為星期五。

| FirstDayOfWeek 列舉成員 | 值 | 說明 |
|--------------------------|---|----------------------|
| FirstDayOfWeek.System | 0 | 系統設定中所指定的每週第一天 (預設值) |
| FirstDayOfWeek.Sunday | 1 | 星期日 |
| FirstDayOfWeek.Monday | 2 | 星期一 |
| FirstDayOfWeek.Tuesday | 3 | 星期二 |
| FirstDayOfWeek.Wednesday | 4 | 星期三 |
| FirstDayOfWeek.Thursday | 5 | 星期四 |
| FirstDayOfWeek.Friday | 6 | 星期五 |
| FirstDayOfWeek.Saturday | 7 | 星期六 |

A-3-3 計算日期時間函式

日期時間增減函式 DateAdd()

✓ 語法

```
DateAdd(interval As DateInterval, number As Double, dt As DateTime) As DateTime
```

✓ 說明

將參數 *dt* 指定的日期加上 *number* 個 *interval*。舉例來說，DateAdd("m", 1, #11/25/90#) 會傳回 #1990/12/25#，其中 *interval* 為 "m"，表示月份，*number* 為 "1"，*dt* 為指定的日期，這個敘述的意義是將 *dt* 指定的日期加上 1 個月；DateAdd("q", 2, #2001/1/31#) 會傳回 #2001/7/31#，也就是 #2001/1/31# 加上 2 個季。要注意的是 *interval* 屬於 DateInterval 列舉，其成員的值如下：

| 成員 | 值 | 說明 | 成員 | 值 | 說明 |
|-----------------------------|------|-------------|----------------------------|---|-------------|
| DateInterval.Year | yyyy | 年 | DateInterval.Quarter | q | 季 |
| DateInterval.Month | m | 月 | DateInterval. DayOfYear | y | 一年中的 第幾天 |
| DateInterval.Day | d | 日 | DateInterval.Hour | h | 小時 |
| DateInterval. WeekOfYear | ww | 一年中的 第幾週 | DateInterval. DayOfWeek | w | 一週中的 星期幾 |
| DateInterval.Minute | n | 分鐘 | DateInterval.Second | s | 秒數 |

日期時間差函式 DateDiff()

✓ 語法

```
DateDiff(interval As DateInterval, dt1 As DateTime, dt2 As DateTime[, dayofweek As FirstDayOfWeek[, weekofyear As FirstWeekOfYear]]) As Long
```

✓ 說明

傳回 *dt1* 和 *dt2* 兩個日期參數之間的時間數，而且時間單位為參數 *interval*。例如 `DateDiff("m", #12/5/90#, #1/10/90#)` 會傳回 -11，即兩個日期間隔 11 個月（第一個參數指定時間單位為月）；`DateDiff("d", #12/5/90#, #12/15/90#)` 會傳回 10，即兩個日期間隔 10 天（第一個參數指定時間單位為天）。

參數 *interval* 屬於 `DateInterval` 列舉，其成員的值如 A-10 頁所示；選擇性參數 *dayofweek* 可以省略，用來指定哪一天為一週的第一天，屬於 `FirstDayOfWeek` 列舉，其值有 `FirstDayOfWeek.System` (0)、`FirstDayOfWeek.Sunday` (1)、`FirstDayOfWeek.Monday` (2)、`FirstDayOfWeek.Tuesday` (3)、`FirstDayOfWeek.Wednesday` (4)、`FirstDayOfWeek.Thursday` (5)、`FirstDayOfWeek.Friday` (6)、`FirstDayOfWeek.Saturday` (7) 等八種；選擇性參數 *weekofyear* 可以省略，用來指定哪一週為一年的第一週，屬於 `FirstWeekOfYear` 列舉，其值有 `FirstWeekOfYear.System` (0) 系統預設值、`FirstWeekOfYear.Jan1` (1) 1 月 1 日當週、`FirstWeekOfYear.FirstFourDays` (2) 新年度至少有四天在該週、`FirstWeekOfYear.FirstFullWeek` (3) 新年度的第一個完整星期等四種；例如 `DateDiff("ww", dt1, dt2, 1)` 會傳回兩個日期之間有幾個星期日。

取得日期時間的指定部分函式 `DatePart()`

✓ 語法

```
DatePart(interval As DateInterval, dt As DateTime[, dayofweek As FirstDayOfWeek [, weekofyear As FirstWeekOfYear]]) As Integer
```

✓ 說明

傳回特定日期的指定部分，例如 `DatePart("y", #9/11/92#)` 會傳回天數 255，因為 9 月 11 日為 1992 年的第 255 天；`DatePart("w", #8/22/2000#)` 會傳回 3，表示為星期二。`DatePart()` 函式的參數意義同 `DateDiff()` 函式，請自行參閱。

A-4 字串函式

A-4-1 字串轉換函式

| 函式 | 說明 |
|--------------------------------------|--|
| 轉換成小寫 LCase(<i>str</i>) | 將參數轉換成小寫字母，參數為的型別為 Char 或 String，傳回值的型別與參數相同。例如 LCase("Happy Birthday") 會傳回 "happy birthday"。 |
| 轉換成大寫 UCase(<i>str</i>) | 將參數轉換成大寫字母，參數的型別為 Char 或 String，傳回值的型別與參數相同。 |
| Asc(<i>var</i>)、AscW(<i>var</i>) | 參數的型別為 Char 或 String，傳回值為字元或字串第一個字元的字碼指標，型別為 Integer，例如 Asc("1") 和 Asc("123") 均會傳回 "1" 的字碼指標 49。 |
| Chr(<i>int</i>)、ChrW(<i>int</i>) | 參數的型別為 Integer，代表的是字碼指標，傳回值為此字碼指標所代表的字元，型別為 Char，例如 Chr(65) 會傳回字元 A。 |
| 字串反轉 StrReverse(<i>str</i>) | 將字串參數的字元順序顛倒過來，然後傳回來，例如 StrReverse("Happy") 會傳回字串 "yppaH"。 |

依照指定格式轉換字串函式 StrConv()

- ✓ 語法

```
StrConv(str As String, Conversion As VbStrConv[, LocaleID As Integer]) As String
```

- ✓ 說明

根據參數 *Conversion* 指定的格式將參數 *str* 轉換完畢的結果傳回來。參數 *Conversion* 屬於 Microsoft.VisualBasic.VbStrConv 列舉，其成員如下表所示。例如 StrConv("I Am Happy.", VbStrConv.LowerCase) 會傳回 "i am happy."。

| VbStrConv 列舉成員 | 說明 |
|-------------------------------|--------------------------|
| VbStrConv.None | 不執行轉換。 |
| VbStrConv.LinguisticCasing | 使用大小寫的語言規格，而不是檔案系統（預設值）。 |
| VbStrConv.UpperCase | 將字串轉換成大寫字母。 |
| VbStrConv.LowerCase | 將字串轉換成小寫字母。 |
| VbStrConv.ProperCase | 將字串中每個單字的第一個字母轉換成大寫。 |
| VbStrConv.Wide* | 將字串中的半形字元轉換成全形字元。 |
| VbStrConv.Narrow* | 將字串中的全形字元轉換成半形字元。 |
| VbStrConv.Katakana** | 將字串中的平假名字元轉換成片假名字元。 |
| VbStrConv.Hiragana** | 將字串中的片假名字元轉換成平假名字元。 |
| VbStrConv.SimplifiedChinese* | 將字串中的繁體中文字元轉換成簡體中文。 |
| VbStrConv.TraditionalChinese* | 將字串中的簡體中文字元轉換成繁體中文。 |
| * 適用於亞洲地區設定；** 僅適用於日本。 | |

A-4-2 字串搜尋、比較、篩選與取代函式

字串比較函式 StrComp()

✓ 語法

```
StrComp(str1 As String, str2 As String[, compare AsMicrosoft.VisualBasic.CompareMethod] )
As Integer
```

✓ 說明

傳回表示 *str1* 和 *str2* 兩個字串參數比較結果的值，若 *str1* 小於 *str2*，傳回 -1；若 *str1* 等於 *str2*，傳回 0；若 *str1* 大於 *str2*，傳回 1。選擇性參數 *compare* 的值有 Binary (區分大小寫) 和 Text (不區分大小寫) 兩種，前者為預設值。

篩選字串陣列函式 Filter()

✓ 語法

```
Filter(source() As {Object|String}, match As String[, include As Boolean[, compare As _  
Microsoft.VisualBasic.CompareMethod]]) As String()
```

✓ 說明

從一維字串陣列 *source()* 中篩選包含 *match* 字串的子陣列。若選擇性參數 *include* 為 True (預設值)，表示從字串陣列 *source()* 中篩選包含 *match* 字串的子陣列；若選擇性參數 *include* 為 False，表示從字串陣列 *source()* 中篩選不包含 *match* 字串的子陣列；選擇性參數 *compare* 的值有 Binary (區分大小寫) 和 Text (不區分大小寫) 兩種，前者為預設值，決定了字串的比較方式。舉例來說，假設 *Source()* = {"abc", "abdf", "ghdf", "pqrs"}，那麼 *Filter(Source, "ab", True, CompareMethod.Binary)* 會傳回 {"abc", "abdf"}。

取代字串函式 Replace()

✓ 語法

```
Replace(str As String, find As String, replacement As String[, start As Integer[, count _  
As Integer[, compare As Microsoft.VisualBasic.CompareMethod]]) As String
```

✓ 說明

將 *str* 字串內的 *find* 子字串替換為 *replacement* 子字串。選擇性參數 *start* 表示從 *str* 字串的第 *start* 個字元開始尋找 *find* 子字串，預設為 1；選擇性 *count* 表示做幾次替換，預設為 -1，表示執行所有可能的替換；選擇性參數 *compare* 的值有 Binary (區分大小寫) 和 Text (不區分大小寫) 兩種，前者為預設值，決定了字串的比較方式。舉例來說，假設 *str* = "Shopping List"，那麼 *Replace(str, "o", "i")* 會傳回 "Shipping List"。

尋找字串函式 InStr()

- ✓ 語法

```
InStr(start As Integer, str1 As String, str2 As String [, compare As _  
Microsoft.VisualBasic.CompareMethod]) As Integer
```

- ✓ 說明

傳回 *str2* 字串在 *str1* 字串中第一次出現的位置。若 *str1* 為空字串 ("") 或 *str2* 沒有出現在 *str1* 或 *start* 大於 *str2* 的長度，就傳回 0；若 *str2* 為空字串 ("")，就傳回 *start*；選擇性參數 *compare* 的值有 Binary (區分大小寫) 和 Text (不區分大小寫) 兩種，前者為預設值，決定了字串的比較方式。例如 InStr(3, "Hello World", "o") 會傳回 5。

尋找字串函式 InStrRev()

- ✓ 語法

```
InStrRev(str1 As String, str2 As Integer[, start As Integer[, compare As _  
Microsoft.VisualBasic.CompareMethod]]) As Integer
```

- ✓ 說明

InStrRev() 函式的用途和 InStr() 函式相似，但改從字串的右邊開始檢查 *str2* 字串在 *str1* 字串中第一次出現的位置，而且 InStrRev() 的參數順序和 InStr() 函式不同。選擇性參數 *start* 用來表示要從第幾個字元開始尋找，預設為 -1，也就是從最後一個字元開始尋找；選擇性參數 *compare* 的值有 Binary (區分大小寫) 和 Text (不區分大小寫) 兩種，前者為預設值，決定了字串的比較方式，例如 InStrRev("Hello World", "o") 會傳回 8。

將字串陣列組成單一字串函式 Join()

- ✓ 語法

```
Join(source() As {Object|String}[, delimiter As String]) As String
```

- ✓ 說明

將字串陣列 *source()* 內的每個字串組成單一字串傳回來，選擇性參數 *delimiter* 表示每個字串之間的分隔符號，預設為空字串 ""。舉例來說，假設 *Source()* = {"Janifer", "Peter", "Jean", "Robert"}，那麼 *Join(Source)* 會傳回 "Janifer Peter Jean Robert"。

將字串分解成子字串陣列函式 Split()

- ✓ 語法

```
Split(str As String[, delimiter As String [,limit As Integer [,compare As _  
Microsoft.VisualBasic.CompareMethod]]]) As String()
```

- ✓ 說明

將組成字串 *str* 的每個子字串放進字串陣列內傳回來。選擇性參數 *delimiter* 表示每個子字串之間的分隔符號，預設值為空字串 ""；選擇性參數 *limit* 是要傳回的子字串數目，預設為 -1，表示傳回所有子字串；選擇性參數 *compare* 的值有 Binary (區分大小寫) 和 Text (不區分大小寫) 兩種，前者為預設值，決定了字串的比較方式。舉例來說，假設 *str* = "I Am Happy."，那麼 *Split(str)* 會傳回字串陣列 {"I", "Am", "Happy."}。

A-4-3 字串操作函式

| 函式 | 說明 |
|--|---|
| 取得字串長度 Len(<i>str</i>) | 傳回字串的字元數目，或儲存一個變數所需的位元組數目，型別為 Integer。舉例來說，假設變數 A 的型別為 Short，那麼 Len(A) 會傳回 2，即儲存一個 Short 變數所需的位元組數目；Len("Hello World") 會傳回 11；Len("程式設計") 會傳回 4。 |
| 由空白字元組成的字串 Space(<i>int</i>) | 傳回由 <i>int</i> 個空白字元所組成的字串， <i>int</i> 的型別為 Integer，傳回值的型別為 String。 |
| 由指定字元組成的字串 StrCup(<i>int</i> , <i>chr</i>) | 傳回由 <i>chr</i> 字元所組成的字串， <i>int</i> 的型別為 Integer， <i>chr</i> 的型別為 Char 或 String，若為 String，則傳回由 <i>int</i> 個該字串中第一個字元所組成的字串。例如 StrDup(3, "I Am Happy.") 會傳回 "III"。 |
| 由字串左邊傳回指定個數字元 Left(<i>str</i> , <i>int</i>) | 從字串 <i>str</i> 左邊傳回 <i>int</i> 個字元，例如 Microsoft.VisualBasic.Left("Happy Birthday", 3) 會傳回 "Hap"。 |
| 由字串右邊傳回指定個數字元 Right(<i>str</i> , <i>int</i>) | 從字串 <i>str</i> 右邊傳回 <i>int</i> 個字元，例如 Microsoft.VisualBasic.Right("Happy Birthday", 3) 會傳回 "day"。 |
| 由字串中指定位置傳回指定個數字元 Mid(<i>str</i> , <i>int1</i> , <i>int2</i>) | 從字串 <i>str</i> 的第 <i>int1</i> 個字元開始傳回 <i>int2</i> 個字元，例如 Mid("Happy Birthday", 3, 5) 會傳回 "ppy B"。 |
| 刪除空白字元 LTrim(<i>str</i>) RTrim(<i>str</i>) Trim(<i>str</i>) | LTrim() 會刪除參數左邊的空白字元，然後將結果傳回 (型別為 String)；RTrim() 會刪除參數右邊的空白字元，然後將結果傳回；Trim() 會刪除參數左右邊的空白字元，然後將結果傳回。舉例來說，假設 <i>str</i> 的值為 " <A> "，那麼 LTrim(<i>str</i>)、RTrim(<i>str</i>)、Trim(<i>str</i>) 的傳回值分別為 "<A> "、" <A>"、"<A>"。 |

A-5 格式化函式

日期時間格式化函式 FormatDateTime()

- ✓ 語法

```
FormatDateTime(exp As DateTime[, namedformat As DateFormat]) As String
```

- ✓ 說明

依照參數 *namedformat* 指定的格式將參數 *exp* 的日期時間格式化，選擇性參數 *namedformat* 屬於 DateFormat 列舉，其成員如下：

| DateFormat 列舉成員 | 說明 |
|------------------------|---|
| DateFormat.GeneralDate | 顯示日期和 (或) 時間。若有日期部份，則會顯示為簡短日期；若有時間部份，則會顯示為完整時間；若兩者皆有，則全會顯示。 |
| DateFormat.LongDate | 使用電腦的地區選項中所指定的完整日期格式來顯示日期。 |
| DateFormat.ShortDate | 使用電腦的地區選項中所指定的簡短日期格式來顯示日期。 |
| DateFormat.LongTime | 使用電腦的地區選項中所指定的時間格式來顯示時間。 |
| DateFormat.ShortTime | 使用 24 小時制 (hh:mm) 來顯示時間。 |

例如 FormatDateTime("#2/14/2002#", DateFormat.LongDate) 會傳回 "2002 年 2 月 14 日"；
FormatDateTime("#2/14/2002#", DateFormat.ShortDate) 會傳回 "2002/2/14"。

貨幣值格式化函式 FormatCurrency()

- ✓ 語法

```
FormatCurrency(exp As Object [, numdigitsafterdecimal As Integer [, leadingdigit As _  
TriState [, negparen As TriState [, groupdigits As TriState]]]) As String
```

✓ 說明

將參數 *exp* 設定成指定的貨幣值格式，此貨幣值使用控制台中地區選項所定義之貨幣符號。選擇性參數 *numdigitsafterdecimal* 表示要顯示小數點後面幾位數，-1 為採取系統預設值；選擇性參數 *leadingdigit* 表示是否要在小於 1 的值前面加上 0，屬於 *TriState* 列舉，其成員如下表所示；選擇性參數 *negparen* 表示是否要將負數用括號括起來，屬於 *TriState* 列舉；選擇性參數 *groupdigits* 表示要加上分隔符號來群組數字，屬於 *TriState* 列舉，例如 `FormatCurrency(123456.789, 2)` 會傳回 NT\$123,456.79。

| TriState 列舉成員 | 值 |
|----------------------------------|---------|
| <code>TriState.True</code> | True |
| <code>TriState.False</code> | False |
| <code>TriState.UseDefault</code> | 地區選項的設定 |

數值格式化函式 `FormatNumber()`

✓ 語法

```
FormatNumber(exp As Object [, numdigitsafterdecimal As Integer [, leadingdigit As _
    TriState [, negparen As TriState [, groupdigits As TriState]]]) As String
```

✓ 說明

將參數 *exp* 設定成指定的數值格式。選擇性參數 *numdigitsafterdecimal* 表示要顯示小數點後面幾位數，-1 為採取系統預設值；選擇性參數 *leadingdigit* 表示是否要在小於 1 的值前面加上 0，屬於 *TriState* 列舉，其成員如上表所示；選擇性參數 *negparen* 表示是否要將負數用括號括起來，屬於 *TriState* 列舉；選擇性參數 *groupdigits* 表示要加上分隔符號來群組數字，屬於 *TriState* 列舉。例如 `FormatNumber(123456.789, 1, , TriState.True)` 會傳回 123,456.8。

百分比格式化函式 FormatPercent()

- ✓ 語法

```
FormatPercent(exp As Object [, numdigitsafterdecimal As Integer [, leadingdigit As _
TriState [, negparen As TriState [, groupdigits As TriState]]]) As String
```

- ✓ 說明

將參數 *exp* 設定成指定的百分比格式（乘以 100）。選擇性參數 *numdigitsafterdecimal* 表示要顯示小數點後面幾位數，-1 為採取系統預設值；選擇性參數 *leadingdigit* 表示是否要在小於 1 的值前面加上 0，屬於 TriState 列舉，其成員如下表所示；選擇性參數 *negparen* 表示是否要將負數用括號括起來，屬於 TriState 列舉；選擇性參數 *groupdigits* 表示要加上分隔符號來群組數字，屬於 TriState 列舉，例如 FormatPercent(123.456, 2) 會傳回 12,345.60%。

| TriState 列舉成員 | 值 |
|---------------------|---------|
| TriState.True | True |
| TriState.False | False |
| TriState.UseDefault | 地區選項的設定 |

格式化函式 Format()

- ✓ 語法

```
Format(exp As Object[, style As String]) As String
```

- ✓ 說明

根據 *style* 指定的格式將 *exp* 格式化。選擇性參數 *style* 為任何預先定義的格式化運算式或使用者自訂的格式化運算式，如以下各表所示。

| 預先定義的日期/時間格式 | 說明 |
|------------------|---|
| General Date 或 G | 顯示日期/時間，例如 Format(Now, "General Date") 會傳回系統目前的日期/時間，好比是 2002/1/31 下午 12:26:03。若沒有分數部分，則只會顯示日期；若沒有整數部分，則只會顯示時間。 |
| Long Date 或 D | 以系統的完整日期格式顯示日期，例如 Format(Now, "Long Date") 會傳回系統目前的長日期，好比是 2002 年 1 月 31 日。 |
| Medium Date | 以主應用程式版本的中等日期格式顯示日期，例如 Format(Now, "Medium Date") 會傳回系統目前的中等日期，好比是 2002 年 1 月 31 日。 |
| Short Date 或 d | 以系統的簡短日期格式顯示日期，例如 Format(Now, "Short Date") 會傳回系統目前的簡短日期，好比是 2002/1/31。 |
| Long Time 或 T | 以系統的完整時間格式顯示時間，例如 Format(Now, "Long Time") 會傳回系統目前的長時間，好比是上午 12:00:00。 |
| Medium Time | 以時、分及上午下午顯示 12 小時時間格式的時間，例如 Format(Now, "Medium Time") 會傳回系統目前的中等時間，好比是上午 12:00:00。 |
| Short Time 或 t | 以 24 小時的時間格式顯示時間，例如 Format(Now, "Short Time") 會傳回系統目前的簡短時間，好比是 17:45。 |
| f | 根據控制台的地區選項設定的格式來顯示完整日期及簡短時間。 |
| F | 根據控制台的地區選項設定的格式來顯示完整日期及完整時間。 |
| g | 根據控制台的地區選項設定的格式來顯示簡短日期及簡短時間。 |
| M、m | 顯示幾月幾日。 |
| R、r | 根據格林威治標準時間 (GMT) 來格式化日期及時間。 |
| s | 根據可排序的索引來格式化日期及時間。 |
| u | 根據 GMT 可排序的索引來格式化日期及時間。 |
| U | 根據 GMT 以完整日期及完整時間來格式化日期及時間。 |
| y | 根據年份及月份來格式化日期。 |

| 預先定義的 數值格式 | 說明 |
|----------------|---|
| General Number | 顯示的數值沒有千分位符號，例如 <code>Format(12345.6789, "General Number")</code> 會傳回 12345.6789。 |
| Currency | 顯示的數值有千分位符號，小數點右邊會顯示兩位數，例如 <code>Format(12345.6789, "Currency")</code> 會傳回 NT\$12,345.68。 |
| Fixed | 小數點左邊至少一位數，右邊有兩位數，例如 <code>Format(12345.6789, "Fixed")</code> 會傳回 12345.68。 |
| Standard | 顯示的數值有千分位符號，小數點左邊至少一位數，右邊有兩位數，例如 <code>Format(12345.6789, "Standard")</code> 會傳回 12,345.68。 |
| Percent | 以百分比格式顯示，小數點右邊有兩位數，例如 <code>Format(12345.6789, "Percent")</code> 會傳回 1234567.89%。 |
| Scientific | 使用標準科學符號，例如 <code>Format(12345.6789, "Scientific")</code> 會傳回 1.23E+04。 |
| D 或 d | 將數值顯示為含有小數點格式的數值（十進位），只有整數數值型別才支援此項，例如 <code>Format(4096, "D")</code> 會傳回 4096。 |
| X 或 x | 將數值顯示為含有十六進位格式的數值，只有整數數值型別才支援此項，例如 <code>Format(4096, "X")</code> 會傳回 1000。 |
| Yes/No | 若數字為 0，顯示 No，否則顯示 Yes，例如 <code>Format(12345.6789, "Yes/No")</code> 會傳回 Yes。 |
| True/False | 若數字為 0，顯示 False，否則顯示 True，例如 <code>Format(12345.6789, "True/False")</code> 會傳回 True。 |
| On/Off | 若數字為 0，顯示 Off，否則顯示 On，例如 <code>Format(12345.6789, "On/Off")</code> 會傳回 On。 |

| 使用者自訂的 數值格式字元 | 說明 |
|------------------|--|
| 0 | 0 為預留位數，顯示對應位數的數值或零；若運算式在格式字串中 0 的位置有數字存在，就顯示出來，否則以零顯示；若運算式中數值的位數少於零的位數（無論是小數點的左方或右方），那麼就把前面或後面的零補足；若運算式中數值的小數點右方位數多於零的位數，那麼就只顯示到有零的位數；若運算式中數值的小數點左方位數多於零的位數，那麼多出的部份還是要顯示出來。 |
| # | # 為預留位數，顯示對應位數或不顯示；若運算式在格式字串中 # 的位置有數字存在，就顯示出來，否則不顯示；# 的運作和 0 預留位數大致相同，不同之處只有在當運算式中數值的位數少於 # 的位數（無論是小數點左方或右方）時，不會把前面或後面的零顯示出來。 |
| . , % | . 為預留小數點；, 為預留千分位符號；% 為預留百分比符號，運算式會乘上 100；例如，Format(123456.789, "##,##0.00") 會傳回 123,456.79；Format(123456.789, "###0.0") 會傳回 123456.8；Format(1.23456, "0.00%") 會傳回 123.46%。 |
| :/ | : 為時間分隔符號；/ 為日期分隔符號。 |
| E- E+ e- e+ | E-、E+、e-、e+ 均為科學符號，E- 或 e- 是用減號來表示負的指數，E+ 或 e+ 是用加號來表示正的指數。例如 Format(123456.789, "0.00E+00") 會傳回 1.23E+5。 |
| - + \$ () | -、+、\$、(、) 顯示一個文字字元，若要顯示所列出之字元之外的字元，可以用 \ 字元表示或以 "" 將之括起來。 |
| \ | 顯示格式化字串中的下一個字元。若要顯示和常值字元一樣具有特殊意義的字元，請在字元前面加上反斜線 (\)。若要顯示反斜線，請使用兩個反斜線 (\\)。無法做為常值字元顯示的字元包括日期格式及時間格式化字元 (a、c、d、h、m、n、p、q、s、t、w、y、/ 及 :)、數值格式化字元 (#、0、%、E、e、逗號、句點) 及字元 (@、&、<、>、!)。 |
| "" | 將雙引號 "" 內的字串顯示出來。 |

| 使用者自訂的日期時間格式化字元 | 說明 |
|---------------------|--|
| M、MM、MMM、MMMM | M 是以沒有零開頭的數字來顯示月 (1 ~ 12)；MM 是以有零開頭的數字來顯示月 (01 ~ 12)；MMM 是以簡寫來表示月；MMMM 是以全名來表示月。例如 Format(#1/3/2002#, "MM") 會傳回 "01"、Format(#1/3/2002#, "MMMM") 會傳回 "一月"。 |
| d、dd、ddd、dddd | d 是以沒有零開頭的數字來顯示天；dd 是以有零開頭的數字來顯示天；ddd 是以簡寫來表示天；dddd 是以全名來表示天。例如 Format(#1/3/2002#, "dddd") 會傳回 "星期四"、Format(#1/3/2002#, "ddd") 會傳回 "星期四"、Format(#1/3/2002#, "dd") 會傳回 "03"。 |
| gg | 顯示紀年，例如 Format(#1/3/2002#, "gg") 會傳回 "西元"。 |
| H、HH、h、hh、m、mm、s、ss | H 是以 24 小時制、沒有零開頭的數字將小時顯示成數字 (0 ~ 23)；HH 是以 24 小時制、有零開頭的數字來顯示小時 (0 ~ 23)；h 是以 12 小時制、沒有零開頭的數字將小時顯示成數字；hh 是以 12 小時制、有零開頭的數字來顯示小時；m 是以沒有零開頭的數字來顯示分 (0 ~ 59)；mm 是以有零開頭的數字來顯示分 (00 ~ 59)；s 是以沒有零開頭的數字來顯示秒 (0 ~ 59)；ss 是以有零開頭的數字來顯示分 (00 ~ 59)；例如 Format(#02:31:50 PM#, "HH:mm:ss") 會傳回 14:31:50、Format(#02:31:50 PM#, "hh:mm:ss") 會傳回 02:31:50 |
| f | 顯示秒數的小數部份，例如 ff 會顯示至百分之一秒，ffff 會顯示至萬分之一秒，最多可以使用 7 個 f 符號。 |
| t | 使用 12 小時制，同時在正午之前的任何小時加上大寫 A，在正午與 11:59 P.M 之前的任何小時加上大寫 P。 |
| tt | 使用 12 小時制，同時在正午之前的任何小時加上大寫 AM；在正午與 11:59 P.M 之前的任何小時加上大寫 PM。 |
| y、yy、yyy、yyyy | y 是一位數來表示年 (0 ~ 9)；yy 是以兩位數來表示年 (00 ~ 99)；yyy 和 yyyy 是以四位數來表示年；例如 Format(#1/3/2002#, "yyyy-MMMM-dd") 會傳回 2002-一月-03。 |

| 使用者自訂的日期時間格式字元 | 說明 |
|----------------|---|
| :z、zz、zzz | :z 是顯示時區時差，而且不需要加上前置零 (例如 -8)；zz 是顯示時區時差，而且要加上前置零 (例如 -08)；zzz 是顯示完整時區時差 (例如 -08:00)。 |
| :/% | : 是時間分隔符號；/ 是日期分隔符號；% 是用來指示應將後面的字元當作單一字母格式讀取，而不管是否還跟著任何字母。 |

我們來看一些例子：

```

Format(#3/8/2005 3:20:48 AM#, "M/d/yyyy H:mm") '傳回 3/8/2005 3:20
Format(#3/8/2005 3:20:48 AM#, "HH:mm:ss")      '傳回 03:20:48
Format(#3/8/2005 3:20:48 AM#, "hh:mm tt")      '傳回 03:20 上午
Format(#3/8/2005 3:20:48 AM#, "d-MMMM-yy")     '傳回 8-三月-05
Format(#3/8/2005 3:20:48 AM#, "M/d/yy")       '傳回 3/8/05
Format(5459.4, "##,##0.00")                    '傳回 5,459.40
Format(334.9, "###0.00")                       '傳回 334.90
Format(5, "0.00%")                             '傳回 500.00%
Format(5, "$#,##0.00")                         '傳回$5.00
Format(Now(), "ff")                            '傳回目前時間的百分之一秒

```


A-6 陣列與集合函式

A-6-1 陣列處理函式

| 函式 | 說明 |
|--|--|
| 陣列判斷函式 IsArray(<i>varName</i>) | 若參數 <i>varName</i> 指向一個陣列，就傳回 True，否則傳回 False。 |
| 取得陣列最小註標函式 LBound(<i>arrName</i> , <i>arrRank</i>) | 傳回陣列某個維度的最小註標 (Integer 型別)，第一個參數 <i>arrName</i> 為陣列的名稱，第二個參數 <i>arrRank</i> 為陣列的第幾個維度，1 表示第一維，2 表示第二維，...餘此類推，若第二個參數省略不寫，表示為預設值 1。舉例來說，LBound(A, 2) 將傳回 A 陣列第二維的最小註標。由於 VB.NET 不允許變更陣列的最小註標，一律從 0 開始，故 LBound() 函式的傳回值恆為 0。 |
| 取得陣列最大註標函式 UBound(<i>arrName</i> , <i>arrRank</i>) | 傳回陣列某個維度的最大註標 (Integer 型別)，第一個參數 <i>arrName</i> 為陣列的名稱，第二個參數 <i>arrRank</i> 為陣列的第幾個維度，1 表示第一維，2 表示第二維，...餘此類推，若第二個參數省略不寫，表示為預設值 1。舉例來說，假設 Dim A(3, 5) As Short，UBound(A) 將傳回 A 陣列第一維的最大註標 3，UBound(A, 2) 將傳回 A 陣列第二維的最大註標 5。 |
| ReDim [Preserve] <i>arrName</i> [(<i>boundlist</i>)] | 這個陳述式可以重新配置 <i>arrName</i> 陣列某個維度的上限為 <i>boundlist</i> ，若要保留陣列內原來的元素值，必須加上 Preserve 關鍵字。 |
| Erase <i>arrList</i> | 這個陳述式可以釋放陣列變數並移除其元素所佔用的記憶體位置，舉例來說，若要釋放陣列 A(10) 和 B(5, 8)，可以寫成 Erase A, B。 |

A-6-2 集合處理函式

將物件加入集合函式 Add()

- ✓ 語法

```
Add(item As Object[, key As String[, {before|after } As Object = Nothing]])
```

- ✓ 說明

將物件 *item* 加入集合內。若要指派唯一的鍵值給加入集合的物件，可以使用選擇性參數 *key*；若要將物件插入集合內第 *before* 個物件的前面，可以加上這個選擇性參數；若要將物件插入集合內第 *after* 個物件的後面，可以加上這個選擇性參數。

將物件自集合移除函式 Remove()

- ✓ 語法

```
Remove(key As String|index As Integer)
```

- ✓ 說明

自集合移除鍵值為 *key* 或註標為 *index* 的物件。

參考集合內的項目屬性 Item()

- ✓ 語法

```
Item(index As Integer|Object) As Object
```

- ✓ 說明

這個唯讀屬性可以傳回集合內鍵值或註標為 *index* 的項目。



休息一下吧！