

Left 函数

返回 **Variant (String)**，其中包含字符串中从左边算起指定数量的字符。

语法

Left(string, length)

Left 函数的语法有下面的[命名参数](#)：

部分	说明
<i>string</i>	必要参数。 字符串表达式 其中最左边的那些字符将被返回。如果 <i>string</i> 包含 Null ，将返回 Null。
<i>length</i>	必要参数；为 Variant (Long) 。 数值表达式 ，指出将返回多少个字符。如果为 0，返回零长度字符串 ("")。如果大于或等于 <i>string</i> 的字符数，则返回整个字符串。

说明

欲知 *string* 的字符数，使用 **Len** 函数。

注意 **LeftB** 函数作用于包含在字符串中的字节数据。所以 *length* 指定的是字节数，而不是要返回的字符数。

Mid 函数

从字符串中返回指定数目的字符。

Mid(string, start[, length])

参数

string

字符串表达式，从中返回字符。如果 **string** 包含 Null，则返回 Null。

Start

string 中被提取的字符部分的开始位置。如果 **start** 超过了 **string** 中字符的数目，**Mid** 将返回零长度字符串 ("")。

Length

要返回的字符数。如果省略或 **length** 超过文本的字符数（包括 **start** 处的字符），将返回字符串中从 **start** 到字符串结束的所有字符。

说明

要判断 **string** 中字符的数目，可使用 **Len** 函数。

下面的示例利用 **Mid** 函数返回字符串中从第四个字符开始的六个字符：

```
Dim MyVar
```

MyVar = Mid("VB 脚本 is fun!", 4, 6) 'MyVar 包含 "Script"。

注意 MidB 函数与包含在字符串中的字节数据一起使用。其参数不是指定字符数，而是字节数。

Len 函数

返回字符串内字符的数目，或是存储一变量所需的字节数。

Len([string](#) | varname)

参数

string

任意有效的字符串表达式。如果 **string** 参数包含 Null，则返回 Null。

Varname

任意有效的变量名。如果 **varname** 参数包含 Null，则返回 Null。

说明

下面的示例利用 **Len** 函数返回字符串中的字符数目：

```
Dim MyString
```

```
MyString = Len("VBSCRIPT") 'MyString 包含 8。
```

注意 LenB 函数与包含在字符串中的字节数据一起使用。**LenB** 不是返回字符串中的字符数，而是返回用于代表字符串的字节数。

Right 函数

从字符串右边返回指定数目的字符。

Right([string](#), length)

参数

string

字符串表达式，其最右边的字符被返回。如果 **string** 参数中包含 Null，则返回 Null。

Length

数值表达式，指明要返回的字符数目。如果为 0，返回零长度字符串；如果此数大于或等于 **string** 参数中的所有字符数目，则返回整个字符串。

说明

要确定 **string** 参数中的字符数目，使用 **Len** 函数。

下面的示例利用 **Right** 函数从字符串右边返回指定数目的字符：

```
Dim AnyString, MyStr
```

```
AnyString = "Hello World" ' 定义字符串。
```

```
MyStr = Right(AnyString, 1) ' 返回 "d"。
```

```
MyStr = Right(AnyString, 6) ' 返回 " World"。
```

```
MyStr = Right(AnyString, 20) ' 返回 "Hello World"。
```

注意 RightB 函数用于字符串中的字节数据，**length** 参数指定返回的是字节数目，而不是

字符数目。

InStr 函数

返回某字符串在另一字符串中第一次出现的位置。

InStr([*start*,]*string1*, *string2*[, *compare*])

参数

start

可选项。数值表达式，用于设置每次搜索的开始位置。如果省略，将从第一个字符的位置开始搜索。如果 **start** 包含 Null，则会出现错误。如果已指定 **compare**，则必须要有 **start** 参数。

string1

必选项。接受搜索的字符串表达式。

string2

必选项。要搜索的字符串表达式。

compare

可选项。指示在计算子字符串时使用的比较类型的数值。有关数值，请参阅“设置”部分。如果省略，将执行二进制比较。

设置

compare 参数可以有以下值：

常数	值	描述
vbBinaryCompare	0	执行二进制比较。
vbTextCompare	1	执行文本比较。

返回值

InStr 函数返回以下值：

如果	InStr 返回
string1 为零长度	0
string1 为 Null	Null
string2 为零长度	start
string2 为 Null	Null
string2 没有找到	0
在 string1 中找到 string2	找到匹配字符串的位置
start > Len(string2)	0

说明

下面的示例利用 **InStr** 搜索字符串：

```
Dim SearchString, SearchChar, MyPos
```

```
SearchString = "XXpXXpXXPXXP" ' 要搜索的字符串。
```

```
SearchChar = "P" ' Search for "P".
```

```
MyPos = InStr(4, SearchString, SearchChar, 1) ' 在位置 4 进行的文本比较。返回 6。
```

```
MyPos = InStr(1, SearchString, SearchChar, 0) ' 在位置 1 进行的二进制比较。返回 9。
```

```
MyPos = InStr(SearchString, SearchChar) ' 默认情况下，进行的是二进制比较（省略了最
```

后的参数)。返回 9。

`MyPos = Instr(1, SearchString, "W")` 在位置 1 进行的二进制比较。返回 0 (找不到 "W")。

注意 `InStrB` 函数使用包含在字符串中的字节数据, 所以 `InStrB` 返回的不是一个字符串在另一个字符串中第一次出现的字符位置, 而是字节位置。

LTrim、RTrim 与 Trim 函数

返回不带前导空格 (`LTrim`)、后续空格 (`RTrim`) 或前导与后续空格 (`Trim`) 的字符串副本。

`LTrim(string)`

`RTrim(string)`

`Trim(string)`

`string` 参数是任意有效的字符串表达式。如果 `string` 参数中包含 `Null`, 则返回 `Null`。

说明

下面的示例利用 `LTrim`, `RTrim`, 和 `Trim` 函数分别用来除去字符串开始的空格、尾部空格、开始和尾部空格:

```
Dim MyVar
```

```
MyVar = LTrim(" vbscript ") 'MyVar 包含 "vbscript"。
```

```
MyVar = RTrim(" vbscript ") 'MyVar 包含 " vbscript"。
```

```
MyVar = Trim(" vbscript ") 'MyVar 包含 "vbscript"。
```

Rnd 函数

返回一个随机数。

`Rnd[(number)]`

`number` 参数可以是任意有效的数值表达式。

说明

`Rnd` 函数返回一个小于 1 但大于或等于 0 的值。`number` 的值决定了 `Rnd` 生成随机数的方式:

如果 `number` 为 `Rnd` 生成

小于零 每次都相同的值, 使用 `number` 作为种子。

大于零 序列中的下一个随机数。

等于零 最近生成的数。

省略 序列中的下一个随机数。

因每一次连续调用 `Rnd` 函数时都用序列中的前一个数作为下一个数的种子, 所以对于任何最初给定的种子都会生成相同的数列。

在调用 `Rnd` 之前, 先使用无参数的 `Randomize` 语句初始化随机数生成器, 该生成器具有基于系统计时器的种子。

要产生指定范围的随机整数, 请使用以下公式:

$\text{Int}((\text{upperbound} - \text{lowerbound} + 1) * \text{Rnd} + \text{lowerbound})$

这里， **upperbound** 是此范围的上界，而 **lowerbound** 是此范围内的下界。

注意 要重复随机数的序列，请在使用数值参数调用 **Randomize** 之前，立即用负值参数调用 **Rnd**。使用同样 **number** 值的 **Randomize** 不能重复先前的随机数序列。

Randomize 语句

初始化随机数生成器。

语法

Randomize [*number*]

可选的 *number* 参数是 [Variant](#) 或任何有效的[数值表达式](#)。

说明

Randomize 用 *number* 将 **Rnd** 函数的随机数生成器初始化，该随机数生成器给 *number* 一个新的[种子值](#)。如果省略 *number*，则用系统计时器返回的值作为新的种子值。

如果没有使用 **Randomize**，则（无参数的）**Rnd** 函数使用第一次调用 **Rnd** 函数的种子值。

注意 若想得到重复的随机数序列，在使用具有数值参数的 **Randomize** 之前直接调用具有负参数值的 **Rnd**。使用具有同样 *number* 值的 **Randomize** 是不会得到重复的随机数序列的。

Rnd 函数示例

本示例使用 **Rnd** 函数随机生成一个 1 到 6 的随机整数。

本示例用 **Randomize** 语句初始化随机数生成器。由于忽略了数值参数，所以 **Randomize** 用 **Timer** 函数的返回值作为新的随机数种子值。

```
Dim MyValue
```

```
Randomize ' 对随机数生成器做初始化的动作。
```

```
MyValue = Int((6 * Rnd) + 1) ' 生成 1 到 6 之间的随机数值。
```

Split 函数

描述

返回一个下标从零开始的一维[数组](#)，它包含指定数目的子字符串。

语法

Split(*expression* [, *delimiter* [, *count* [, *compare*]])

Split 函数语法有如下几部分：

部分	描述
<i>expression</i>	必需的。包含子字符串和分隔符的 字符串表达式 。如果

	<i>expression</i> 是一个长度为零的字符串(""), Split 则返回一个空数组, 即没有元素和数据的数组。
<i>delimiter</i>	可选的。用于标识子字符串边界的字符串字符。如果忽略, 则使用空格字符(" ")作为分隔符。如果 <i>delimiter</i> 是一个长度为零的字符串, 则返回的数组仅包含一个元素, 即完整的 <i>expression</i> 字符串。
<i>count</i>	可选的。要返回的子字符串数, -1 表示返回所有的子字符串。
<i>compare</i>	可选的。数字值, 表示判别子字符串时使用的比较方式。关于其值, 请参阅“设置值”部分。

设置值

compare 参数的设置值如下:

常数	值	描述
vbUseCompareOption	-1	用 Option Compare 语句中的设置值执行比较。
vbBinaryCompare	0	执行二进制比较。
vbTextCompare	1	执行文字比较。
vbDatabaseCompare	2	仅用于 Microsoft Access。基于您的数据库的信息执行比较。

Replace 函数

返回字符串, 其中指定数目的某子字符串被替换为另一个子字符串。

Replace(expression, find, replacewith[, compare[, count[, start]])

参数

expression

必选项。字符串表达式包含要替代的子字符串。

Find

必选项。被搜索的子字符串。

Replacewith

必选项。用于替换的子字符串。

Start

可选项。**expression** 中开始搜索子字符串的位置。如果省略, 默认值为 1。在和 **count** 关联时必须用

count

可选项。执行子字符串替换的数目。如果省略, 默认值为 -1, 表示进行所有可能的替换。

在和 *start* 关联时必须用。

Compare

可选项。指示在计算子字符串时使用的比较类型的数值。有关数值，请参阅“设置”部分。如果省略，缺省值为 0，这意味着必须进行二进制比较。

设置

compare 参数可以有以下值：

常数	值	描述
vbBinaryCompare	0	执行二进制比较。
vbTextCompare	1	执行文本比较。

返回值

Replace 返回以下值：

如果	Replace 返回
expression 为零长度	零长度字符串 ("")。
expression 为 Null	错误。
find 为零长度	expression 的副本。
replacewith 为零长度	expression 的副本，其中删除了所有由 find 参数指定的内容。
start > Len(expression)	零长度字符串。
count 为 0	expression 的副本。

说明

Replace 函数的返回值是经过替换（从由 **start** 指定的位置开始到 **expression** 字符串的结尾）后的字符串，而不是原始字符串从开始至结尾的副本。

下面的示例利用 **Replace** 函数返回字符串：

```
Dim MyString
MyString = Replace("XXpXXPXXp", "p", "Y") '二进制比较从字符串左端开始。返回"XXYXXPY"。
MyString = Replace("XXpXXPXXp", "p", "Y", '文本比较从第三个字符开始。返回"YXXYXXY"。 3, , -1, 1)
```

StrComp 函数

返回一个表明字符串比较结果的值。

StrComp(string1, string2[, compare])

参数

string1

必选项。任意有效的字符串表达式。

string2

必选项。任意有效的字符串表达式。

Compare

可选项。指示在计算字符串时使用的比较类型的数值。如果省略，则执行二进制比较。有关数值，请参阅“设置”部分。

设置

compare 参数可以有以下值:

常数	值	描述
vbBinaryCompare	0	执行二进制比较。
vbTextCompare	1	执行文本比较。

返回值

StrComp 函数有以下返回值:

如果	StrComp 返回
string1 小于 string2	-1
string1 等于 string2	0
string1 大于 string2	1
string1 或 string2 为 Null	Null

说明

下面的示例利用 **StrComp** 函数返回字符串比较的结果。如果第三个参数为 1 执行文本比较; 如果第三个参数为 0 或者省略执行二进制比较。

```
Dim MyStr1, MyStr2, MyComp
```

```
MyStr1 = "ABCD": MyStr2 = "abcd" '定义变量。
```

```
MyComp = StrComp(MyStr1, MyStr2, 1) '返回 0。
```

```
MyComp = StrComp(MyStr1, MyStr2, 0) '返回 -1。
```

```
MyComp = StrComp(MyStr2, MyStr1) '返回 1。
```

CInt 函数

返回表达式, 此表达式已被转换为 **Integer** 子类型的 **Variant**。

CInt(expression)

expression 参数是任意有效的表达式。

说明

通常, 可以使用子类型转换函数书写代码, 以显示某些操作的结果应被表示为特定的数据类型, 而不是默认类型。例如, 在出现货币、单精度或双精度运算的情况下, 使用 **CInt** 或 **CLng** 强制执行整数运算。

CInt 函数用于进行从其他数据类型到 **Integer** 子类型的国际公认的格式转换。例如对十进制分隔符 (如千分符) 的识别, 可能取决于系统的区域设置。

如果 *expression* 在 **Integer** 子类型可接受的范围之外, 则发生错误。

下面的示例利用 **CInt** 函数把值转换为 **Integer**:

```
Dim MyDouble, MyInt
```

```
MyDouble = 2345.5678 'MyDouble 是 Double。
```

```
MyInt = CInt(MyDouble) 'MyInt 包含 2346。
```

注意 **CInt** 不同于 **Fix** 和 **Int** 函数删除数值的小数部分, 而是采用四舍五入的方式。当小数部分正好等于 0.5 时, **CInt** 总是将其四舍五入成最接近该数的偶数。例如, 0.5 四舍五入为 0, 以及 1.5 四舍五入为 2。

CStr 函数

返回表达式，该表达式已被转换为 **String** 子类型的 **Variant**。

CStr(expression)

expression 参数是任意有效的表达式。

说明

通常，可以使用子类型转换函数书写代码，以显示某些操作的结果应被表示为特定的数据类型，而不是默认类型。例如，使用 **CStr** 强制将结果表示为 **String**。

CStr 函数用于替代 **Str** 函数来进行从其他数据类型到 **String** 子类型的国际公认的格式转换。例如对十进制分隔符的识别取决于系统的区域设置。

expression 根据下表决定返回的数据：

如果 **expression** 为 **CStr** 返回

Boolean	字符串，包含 True 或 False。
Date	字符串，包含系统的短日期格式日期。
Null	运行时错误。
Empty	零长度字符串 ("")。
Error	字符串，包含跟随有错误号码的单词 Error。
其他数值	字符串，包含此数字。

下面的示例利用 **CStr** 函数把数字转换为 **String**：

```
Dim MyDouble, MyString
```

```
MyDouble = 437.324           'MyDouble 是双精度值。
```

```
MyString = CStr(MyDouble)   'MyString 包含 "437.324"。
```

LCase 函数

返回字符串的小写形式。

LCase(string)

string 参数是任意有效的字符串表达式。如果 **string** 参数中包含 Null，则返回 **Null**。

说明

仅大写字母转换成小写字母；所有小写字母和非字母字符保持不变。

下面的示例利用 **LCase** 函数把大写字母转换为小写字母：

```
Dim MyString
```

```
Dim LCaseString
```

```
MyString = "VBScript"
```

```
LCaseString = LCase(MyString) 'LCaseString 包含 "vbscript"。
```

UCase 函数

返回字符串的大写形式。

UCase(string)

string 参数是任意有效的字符串表达式。如果 **string** 参数中包含 Null，则返回 **Null**。

说明

只有小写字母被转换成大写字母；所有大写字母和非字母字符均保持不变。

下面的示例利用 **UCase** 函数返回字符串的大写形式：

```
Dim MyWord
```

```
MyWord = UCase("Hello World") ' 返回"HELLO WORLD"。
```

MsgBox 函数

在对话框中显示消息，等待用户单击按钮，并返回一个值指示用户单击的按钮。

MsgBox([prompt](#)[, buttons][, title][, helpfile, context])

参数

prompt

作为消息显示在对话框中的字符串表达式。**prompt** 的最大长度大约是 1024 个字符，这取决于所使用的字符的宽度。如果 **prompt** 中包含多个行，则可在各行之间用回车符 (**Chr**(13))、换行符 (**Chr**(10)) 或回车换行符的组合 (**Chr**(13) & **Chr**(10)) 分隔各行。

Buttons

数值表达式，是表示指定显示按钮的数目和类型、使用的图标样式，默认按钮的标识以及消息框样式的数值的总和。有关数值，请参阅“设置”部分。如果省略，则 **buttons** 的默认值为 0。

Title

显示在对话框标题栏中的字符串表达式。如果省略 **title**，则将应用程序的名称显示在标题栏中。

Helpfile

字符串表达式，用于标识为对话框提供上下文相关帮助的帮助文件。如果已提供 **helpfile**，则必须提供 **context**。在 16 位系统平台上不可用。

Context

数值表达式，用于标识由帮助文件的作者指定给某个帮助主题的上下文编号。如果已提供 **context**，则必须提供 **helpfile**。在 16 位系统平台上不可用。

设置

buttons 参数可以有以下值：

常数	值	描述
vbOKOnly	0	只显示 确定 按钮。
vbOKCancel	1	显示 确定 和 取消 按钮。
vbAbortRetryIgnore	2	显示 放弃 、 重试 和 忽略 按钮。
vbYesNoCancel	3	显示 是 、 否 和 取消 按钮。

vbYesNo	4	显示 是 和 否 按钮。
vbRetryCancel	5	显示 重试 和 取消 按钮。
vbCritical	16	显示 临界信息 图标。
vbQuestion	32	显示 警告查询 图标。
vbExclamation	48	显示 警告消息 图标。
vbInformation	64	显示 信息消息 图标。
vbDefaultButton1	0	第一个按钮为默认按钮。
vbDefaultButton2	256	第二个按钮为默认按钮。
vbDefaultButton3	512	第三个按钮为默认按钮。
vbDefaultButton4	768	第四个按钮为默认按钮。
vbApplicationModal	0	应用程序模式：用户必须响应消息框才能继续在当前应用程序中工作。

vbSystemModal 4096 系统模式：在用户响应消息框前，所有应用程序都被挂起。第一组值 (0 - 5) 用于描述对话框中显示的按钮类型与数目；第二组值 (16, 32, 48, 64) 用于描述图标的样式；第三组值 (0, 256, 512) 用于确定默认按钮；而第四组值 (0, 4096) 则决定消息框的样式。在将这些数字相加以生成 **buttons** 参数值时，只能从每组值中取用一个数字。

返回值

MsgBox 函数有以下返回值：

常数 值 按钮

vbOK	1	确定
vbCancel	2	取消
vbAbort	3	放弃
vbRetry	4	重试
vbIgnore	5	忽略
vbYes	6	是
vbNo	7	否

说明

如果同时提供了 **helpfile** 和 **context**，则用户可以按 **F1** 键以查看与上下文相对应的帮助主题。

如果对话框显示**取消**按钮，则按 **ESC** 键与单击**取消**的效果相同。如果对话框包含**帮助**按钮，则有为对话框提供的上下文相关帮助。但是在单击其他按钮之前，不会返回任何值。

当 Microsoft Internet Explorer 使用 **MsgBox** 函数时，任何对话框的标题总是包含"VBScript"，以便于将其与标准对话框区别开来。

下面的例子演示了 **MsgBox** 函数的用法：

```
Dim MyVar
```

```
MyVar = MsgBox ("Hello World!", 65, "MsgBox Example")
```

'MyVar 包含 1 或 2，这取决于单击的是哪个按钮。