VBScript 实例 ➔

VBScript 教程

VBScript 是微软公司出品的脚本语言。

VBScript 是ASP (Active Server Pages)默认使用的脚本语言。

你可以在 Internet Explorer 尝试使用 VBScript 。

VBScript 编辑器

VBScript 是一个动态脚本语言。

Internet Explorer 支持 VBScript. 所以你可以在IE浏览器中执行VBScript， 通过我们的 VBScript 在线实例，你可以查看实例的运行结果：

实例(只能在Internet Explorer运行)

```
<html>
<body>

<script type="text/vbscript">
document.write("This is my first VBScript!")
</script>

</body>
</html>
```

尝试一下 »

点击 "尝试一下" 按钮查看运行结果。

什么是 VBScript?

- VBScript 是一种脚本语言
- 脚本语言是一种轻量级的编程语言
- VBScript 是微软的编程语言 Visual Basic 的轻量级的版本

VBScript 实例

通过实例进行学习。使用我们的编辑器，你可以编辑源代码，然后单击 "尝试一下" 按钮来查看结果。

在线实例！

VBScript 参考手册

在本站中你可以找到完整的 VBScript 参考手册。

VBScript 参考手册

VBScript 实例 →

点我分享笔记

VBScript 实例 VBScript 变量

VBScript 用法

HTML <script> 标签被用来向 HTML 中插入 VBScript。

HTML 中的 VBScript

如需在 HTML 中插入 VBScript，脚本必须写在标准的 <script> 和 </script> 标签之间。

在 <script> 标签中，请使用 type 属性来定义脚本语言 "text/vbscript"：

```
<html>
<body>
<script type="text/vbscript">
...
</script>
</body>
</html>
```

IE 将解释和执行 <script> 和 </script> 之间的 VBScript 代码。

VBScript 不应该被用作客户端脚本语言！
在这里，我们使用仅适用于 IE 的 VBScript 的用于学习。

VBScript 输出

当 VBScript 被用在 Web 服务器上的 ASP 页面时，语句 **response.write()** 产生输出。

当我们使用 Internet Explorer 来测试 VBScript，我们使用 **document.write()** 来产生输出：

实例（仅适用于 Internet Explorer）

```
<html>
<body>
<script type="text/vbscript">
document.write("Hello World!")
</script>
</body>
</html>
```

尝试一下 »

在上面的实例中，浏览器输出 "Hello World!" 到 HTML 页面。

VBScript 实例 VBScript 变量

VBScript 用法 VBScript 程序

VBScript 变量

变量是存储信息的"容器"。

尝试一下 - 实例（只适用于 IE）

创建和改变变量

如何创建一个变量，并为它赋值，然后再改变它的值。

在一段文本中插入变量值

如何在一段文本中插入变量值。

创建数组

数组用来存储一系列相关的数据项。本例演示如何创建一个存储名字的数组。

还记得在学校里学过的代数吗？

还记得在学校里学过的代数吗？x=5，y=6，z=x+y

还记得吗？一个字母（比如 x）可以保存一个值（比如 5），并且可以使用上面的信息计算出 z 的值是 11。

这些字母称为**变量**，变量可用于保存值（x=5）或表达式（z=x+y）。

VBScript 变量

与代数相比，VBScript 变量用于保存值或表达式。

变量可以有一个短的名称，如 x，或一个更具描述性的名称，如 carname。

VBScript 变量名称的规则：

- 必须以字母开头
- 不能包含点号（.）
- 不能超过 255 个字符

在 VBScript 中，所有的变量都与类型 *variant* 相关，可存储不同类型的数据。

声明（创建）VBScript 变量

在 VBScript 创建变量通常指"声明"变量。

您可以通过 Dim、Public 或 Private 语句声明 VBScript 变量。如下所示：

```
Dim x
Dim carname
```

现在您已经创建了两个变量。变量的名称是 "x" 和 "carname"。

您也可以在脚本中通过使用它的名称来声明变量。如下所示：

```
carname="Volvo"
```

现在您又创建了一个变量。变量的名称是 "carname"。然后，这个做法不是一个好习惯，因为您可能会在脚本中拼错变量名，那样可能会在脚本运行时引起奇怪的结果。

如果您拼错变量名，比如 "carname" 变量错拼为 "carnime"，脚本会自动创建一个名为 "carnime" 的新变量。为了防止脚本这样做，您可以使用 Option Explicit 语句。如果您使用这个语句，就必须使用 dim、public 或 private 语句来声明所有的变量。

把 Option Explicit 语句放置于脚本的顶端，如下所示：

```
Option Explicit
Dim carname
carname=some value
```

为变量赋值

您可以为某个变量赋值，如下所示：

```
carname="Volvo"
x=10
```

变量名是在表达式的左侧，需要赋给变量的值在表达式的右侧。现在变量 "carname" 的值是 "Volvo"，变量 "x" 的值是 "10"。

变量的生存期

变量的生存期指的是它可以存在的时长。

当您在一个子程序中声明变量时，变量只能在此程序内进行访问。当退出此程序时，变量也会失效。这样的变量称为本地变量。您可以在不同的子程序中使用名称相同的本地变量，因为每个变量只能在声明它的程序内得到识别。

如果您在子程序以外声明了一个变量，在您的页面上的所有子程序都可以访问它。这类变量的生存期始于它们被声明，止于页面被关闭。

VBScript 数组变量

数组变量用于在一个单一的变量中存储多个值。

在下面的实例中，声明了一个包含 3 个元素的数组：

```
Dim names(2)
```

括号内显示的数字是 2。数组的下标以 0 开始，因此该数组包含 3 个元素。这是容量固定的数组。您可以为数组的每个元素分配数据，如下所示：

```
names(0)="Tove"
names(1)="Jani"
names(2)="Stale"
```

同样地，通过使用特定数组元素的下标号，您可以取回任何元素的值。如下所示：

```
mother=names(0)
```

您可以在一个数组中使用多达 60 个维数。声明多维数组的方法是在括号中用逗号来分隔数字。这里，我们声明了一个包含 5 行 7 列的 2 维数组：

```
Dim table(4,6)
```

为二维数组赋值：

实例（仅适用于 IE）

```
<html>
<body>

<script type="text/vbscript">
Dim x(2,2)
x(0,0)="Volvo"
x(0,1)="BMW"
x(0,2)="Ford"
x(1,0)="Apple"
x(1,1)="Orange"
x(1,2)="Banana"
x(2,0)="Coke"
x(2,1)="Pepsi"
x(2,2)="Sprite"
for i=0 to 2
document.write("<p>")
for j=0 to 2
document.write(x(i,j) & "<br />")
next
document.write("</p>")
next
</script>

</body>
</html>
```

尝试一下 »

VBScript 用法 VBScript 程序

VBScript 变量 VBScript 条件语句

VBScript 程序

VBScript 可使用两种程序：

- 子程序
- 函数程序

VBScript 子程序

子程序：

- 是一系列的语句，被封装在 Sub 和 End Sub 语句内
- 可执行某些操作，但**不会返回**值
- 可带有参数

```
Sub mysub()
some statements
End Sub
```

或者

```
Sub mysub(argument1,argument2)
some statements
End Sub
```

实例（仅适用于 IE）

```
Sub mysub()
document.write("I was written by a sub procedure")
End Sub
```

尝试一下 »

VBScript 函数程序

函数程序

- 是一系列的语句，被封装在 Function 和 End Function 语句内
- 可执行某些操作，并**会返回**值
- 可带有通过程序调用来向其传递的参数。
- 如果没有参数，必须带有空的圆括号 ()
- 通过向函数程序名赋值的方式，可使其返回值

```
Function myfunction()
some statements
myfunction=some value
End Function
```

或者

```
Function myfunction(argument1,argument2)
some statements
myfunction=some value
End Function
```

实例（仅适用于 IE）

```
function myfunction()
myfunction=Date()
end function
```

尝试一下 »

调用程序

这个简单的函数程序被调用来计算两个参数的和：

实例（仅适用于 IE）

```
Function myfunction(a,b)
myfunction=a+b
End Function

document.write(myfunction(5,9))
```

尝试一下 »

函数 "myfunction" 将返回参数 "a" 和参数 "b" 的和。这里返回的是 14。

当您调用程序时，您可以使用 Call 语句，如下所示：

```
Call MyProc(argument)
```

或者，您可以省略 Call 语句，如下所示：

```
MyProc argument
```

← VBScript 变量 VBScript 条件语句 →

点我分享笔记

VBScript 程序 VBScript 循环语句

VBScript 条件语句

条件语句

条件语句用于根据不同的情况执行不同的操作。

在 VBScript 中，我们可以使用四种条件语句：

- **If stat语句ement** - 假如您希望在条件为 true 时执行一系列的代码，可以使用这个语句
- **If...Then...Else 语句** - 假如您希望执行两套代码其中之一，可以使用这个语句
- **If...Then...ElseIf 语句** - 假如您希望选择多套代码之一来执行，可以使用这个语句
- **Select Case 语句** - 假如您希望选择多套代码之一来执行，可以使用这个语句

If...Then...Else

在下面的情况中，您可以使用 If...Then...Else 语句：

- 在条件为 true 时，执行某段代码
- 选择两段代码之一来执行

如果在条件为 true 时只执行**一条**语句，可以把代码写为一行：

```
If i=10 Then alert("Hello")
```

在上面的代码中，没有 ..Else.. 语句。我们仅仅让代码在条件为 true 时（当 i=10 时）执行**一项操作。**

如果在条件为 true 时执行**不止一条**语句，那么就必须在一行写一条语句，然后使用关键词 "End If" 来结束这个语句：

```
If i=10 Then
alert("Hello")
i = i+1
End If
```

在上面的代码中，同样没有 ..Else.. 语句。我们仅仅让代码在条件为 true 时执行了**多项操作。**

假如您想要在条件为 true 时执行某条语句，并在条件不为 true 时执行另一条语句，就必须添加关键词 "Else"：

实例（仅适用于 IE）

```
<script type="text/vbscript">
i=hour(time)
If i < 10 Then
document.write("Good morning!")
Else
document.write("Have a nice day!")
End If
</script>
```

尝试一下 »

在上面的代码中，当条件为 true 时会执行第一段代码，当条件不成立时执行第二段代码（当 i 大于 10 时）。

If...Then...ElseIf

如果您想要选择多套代码之一来执行，可以使用 If...Then...ElseIf 语句：

实例（仅适用于 IE）

```
<script type="text/vbscript">
i=hour(time)
If i = 10 Then
document.write("Just started...!")
ElseIf i = 11 Then
document.write("Hungry!")
ElseIf i = 12 Then
document.write("Ah, lunch-time!")
ElseIf i = 16 Then
document.write("Time to go home!")
Else
document.write("Unknown")
End If
</script>
```

尝试一下 »

Select Case

如果您想要选择多套代码之一来执行，可以使用 "Select Case" 语句：

实例（仅适用于 IE）

```
<script type="text/vbscript">
d=weekday(date)
Select Case d
Case 1
document.write("Sleepy Sunday")
Case 2
document.write("Monday again!")
Case 3
document.write("Just Tuesday!")
Case 4
document.write("Wednesday!")
Case 5
document.write("Thursday...")
Case 6
document.write("Finally Friday!")
Case else
document.write("Super Saturday!!!!")
End Select
</script>
```

尝试一下 »

以上代码的工作原理：首先，我们需要一个简单的表达式（常常是一个变量），并且这个表达式会被做一次求值运算。然后，表达式的值会与每个 Case 中的值作比较。如果匹配，被匹配的 Case 所对应的代码会被执行。

VBScript 程序 VBScript 循环语句

VBScript 条件语句 VBScript 总结

VBScript 循环

循环语句

循环语句用于运行相同的代码块指定的次数。Looping statements are used to run the same block of code a specified number of times.

在 VBScript 中，我们可以使用四种循环语句：

- **For...Next 语句** - 运行一段代码指定的次数
- **For Each...Next 语句** - 针对集合中的每个项目或者数组中的每个元素来运行某段代码
- **Do...Loop 语句** - 运行循环，当条件为 true 或者直到条件为 true 时
- **While...Wend 语句** - 不要使用这个语句 - 请使用 Do...Loop 语句代替它

For...Next 循环

请使用 **For...Next** 语句运行一段代码指定的次数。

For 语句规定计数变量（**i**）以及它的初始值和结束值。**Next** 语句会以 1 作为步进值来递增变量（**i**）。

实例

```
<html>
<body>

<script type="text/vbscript">
For i = 0 To 5
document.write("The number is " & i & "<br />")
Next
</script>

</body>
</html>
```

尝试一下 »

Step 关键词

通过 **Step** 关键词，您可以规定计数变量递增或递减的步进值。

在下面的实例中，计数变量（**i**）每次循环的递增步进值为 2。

```
For i=2 To 10 Step 2
some code
Next
```

如果要递减计数变量，您就必须使用负的 **Step** 值。并且必须规定小于开始值的结束值。

在下面的实例中，计数变量（**i**）每次循环的递减步进值为 2。

```
For i=10 To 2 Step -2
some code
Next
```

退出 For...Next

您可以通过 Exit For 关键词退出 For...Next 语句。

```
For i=1 To 10
If i=5 Then Exit For
some code
Next
```

For Each...Next 循环

For Each...Next 针对集合中的每个项目或者数组中的每个元素来重复运行某段代码。

实例

```
<html>
<body>

<script type="text/vbscript">
Dim cars(2)
cars(0)="Volvo"
cars(1)="Saab"
cars(2)="BMW"

For Each x In cars
document.write(x & "<br />")
Next
</script>

</body>
</html>
```

尝试一下 »

Do...Loop

如果你不知道重复多少次，可以使用 Do...Loop 语句。

Do...Loop 语句重复执行某段代码直到条件是 true 或条件变成 true。

重复执行代码直到条件是 true

您可以使用 While 关键字来检查 Do... Loop 语句的条件。

```
Do While i>10
some code
Loop
```

如果 **i** 等于 9，上述循环内的代码将终止执行。

```
Do
some code
Loop While i>10
```

这个循环内的代码将被执行至少一次，即使 **i** 小于 10。

重复执行代码直到条件变成 true

您可以使用 Until 关键字来检查 Do...Loop 语句的条件。

```
Do Until i=10
some code
Loop
```

如果 **i** 等于 10，上述循环内的代码将终止执行。

```
Do
some code
Loop Until i=10
```

这个循环内的代码将被执行至少一次，即使 **i** 等于 10。

退出 Do...Loop

您可以通过 Exit Do 关键词退出 Do...Loop 语句。

```
Do Until i=10
i=i-1
If i<10 Then Exit Do
Loop
```

这个循环内的代码，只要 **i** 不为 10 且 **i** 大于 10 时都将被执行。

更多实例（仅适用于 IE）

循环遍历标题

如何循环遍历 html 中的六个标题。

Do...While loop

如何做一个简单的 **Do...While** 循环。

VBScript 条件语句 VBScript 总结

VBScript 循环语句 VBScript CDate 函数

您已经学习了 VBScript，下一步呢？

VBScript 总结

本教程已经向您讲解了 VBScript。

您已经学习了有关变量和函数，以及如何如何根据不同的情况运行不同的脚本。

如需了解更多有关 VBScript 的信息，请参阅我们的 VBScript 实例和我们的 VBScript 参考手册。

现在您已经学习了 VBScript，下一步呢？

下一步应该是学习 ASP。

HTML 文件中的脚本是在客户端（在浏览器）执行的，ASP 文件中的脚本是在服务器上执行的。

通过 ASP，您可以动态地编辑、改变或添加网页的任何内容，响应 HTML 表单提交的数据，访问任何数据或数据库并向浏览器返回结果，为不同的用户定制更有用的页面等等。

由于 ASP 文件被作为纯粹的 HTML 返回浏览器，，因此我们可以在任何浏览器中查看 ASP。

如果您想要学习更多有关 ASP 的知识，请访问我们的 ASP 教程。

VBScript 循环语句 VBScript CDate 函数

VBScript 教程 VBScript 用法

VBScript 实例

基础

使用VBScript写文本

使用HTML标签格式化文本

head部分中的函数

body部分中的脚本

变量

创建一个变量

在一段文本中插入变量的值

创建数组

程序

子程序

函数程序

条件语句

If...then..else 语句

If...then..elseif 语句

Select case 语句

随机链接

循环

For..next 循环

循环输出HTML标题

For..each 循环

Do...While 循环

日期和时间函数

显示日期和时间

显示星期

显示月份

显示当前的月份和星期

倒记秒数到3000年

向日期增加一个时间间隔

格式化日期和时间

这是一个日期吗？

其他内建函数

大写或小写字符？

删除字符串端部的空格

颠倒字符顺序

对一个数字进行取整数操作

返回随机数

返回0-99之间的随机数

从一段字符串的左侧或右侧返回指定数目的字符

替换字符串中的某些字符

从一段字符串返回指定数目的字符

VBScript 教程 VBScript 用法

点我分享笔记

VBScript VarType 函数 VBScript 关键字

VBScript 函数

本页列出了所有内建的 VBScript 函数，主要分为以下几类：

- Date/Time 函数
- Conversion 函数
- Format 函数
- Math 函数
- Array 函数
- String 函数
- 其他函数

Date/Time 函数

函数	描述
CDate	把有效的日期和时间表达式转换为日期（Date）类型。
Date	返回当前的系统日期。
DateAdd	返回已添加指定时间间隔的日期。
DateDiff	返回两个日期之间的时间间隔数。
DatePart	返回给定日期的指定部分。
DateSerial	返回指定年、月、日的日期。
DateValue	返回日期。
Day	返回代表一月中的一天的数字（介于并包括 1 到 31 之间）。
FormatDateTime	返回格式化为日期或时间的表达式。
Hour	返回代表一天中的小时的数字（介于并包括 0 到 23 之间）。
IsDate	返回指示计算表达式能否转换为日期的布尔值。
Minute	返回一个数字，代表小时的分钟（介于并包括 0 到 59 之间）。
Month	返回一个数字，代表年的月份（介于并包括 1 到 12 之间）。
MonthName	返回指定月份的名称。
Now	返回当前的系统日期和时间。
Second	返回一个数字，代表分钟的秒（介于并包括 0 到 59 之间）。

Time	返回当前的系统时间。
Timer	返回自 12:00 AM 以来的秒数。
TimeSerial	返回特定小时、分钟和秒的时间。
TimeValue	返回时间。
Weekday	返回一个数字，代表一周的天数（介于并包括 1 到 7 之间）。
WeekdayName	返回一周中指定的一天的星期名。
Year	返回一个数字，代表年份。

Conversion 函数

Top

函数	描述
Asc	把字符串中的首字母转换为 ANSI 字符代码。
CBool	把表达式转换为布尔（Boolean）类型。
CByte	把表达式转换为字节（Byte）类型。
CCur	把表达式转换为货币（Currency）类型。
CDate	把有效的日期和时间表达式转换为日期（Date）类型。
CDbl	把表达式转换为双精度（Double）类型。
Chr	把指定的 ANSI 字符代码转换为字符。
CInt	把表达式转换为整数（Integer）类型。
CLng	把表达式转换为长整型（Long）类型。
CSng	把表达式转换为单精度（Single）类型。
CStr	把表达式转换为字符串（String）类型。
Hex	返回指定数字的十六进制值。
Oct	返回指定数字的八进制值。

Format 函数

Top

函数	描述
FormatCurrency	返回作为货币值进行格式化的表达式。
FormatDateTime	返回作为日期或时间进行格式化的表达式。
FormatNumber	返回作为数字进行格式化的表达式。
FormatPercent	返回作为百分数进行格式化的表达式。

Math 函数

Top

函数	描述
Abs	返回指定数字的绝对值。
Atn	返回指定数字的反正切。
Cos	返回指定数字（角度）的余弦。
Exp	返回 *e*（自然对数的底）的幂次方。
Hex	返回指定数字的十六进制值。
Int	返回指定数字的整数部分。
Fix	返回指定数字的整数部分。
Log	返回指定数字的自然对数。
Oct	返回指定数字的八进制值。
Rnd	返回小于1但大于或等于0的一个随机数。
Sgn	返回可指示指定数字的符号的一个整数。
Sin	返回指定数字（角度）的正弦。
Sqr	返回指定数字的平方根。
Tan	返回指定数字（角度）的正切。

Array 函数

Top

函数	描述

Array	返回一个包含数组的变量。
Filter	返回下标从零开始的数组，其中包含基于特定过滤条件的字符串数组的子集。
IsArray	返回一个指示指定的变量是否为数组的布尔值。
Join	返回一个由数组中若干子字符串组成的字符串。
LBound	返回指示数组维数的最小下标。
Split	返回下标从零开始的一维数组，包含指定数量的子字符串。
UBound	返回指示数组维数的最大下标。

String 函数

Top

函数	描述
InStr	返回字符串在另一字符串中首次出现的位置。搜索从字符串的第一个字符开始。
InStrRev	返回字符串在另一字符串中首次出现的位置。搜索从字符串的最末字符开始。
LCase	把指定字符串转换为小写。
Left	从字符串的左侧返回指定数量的字符。
Len	返回字符串中的字符数量。
LTrim	删除字符串左侧的空格。
RTrim	删除字符串右侧的空格。
Trim	删除字符串左侧和右侧的空格。
Mid	从字符串中返回指定数量的字符。
Replace	使用另一个字符串替换字符串的指定部分指定的次数。
Right	从字符串的右侧返回指定数量的字符。
Space	返回由指定数量的空格组成的字符串。
StrComp	比较两个字符串，返回代表比较结果的一个值。
String	返回包含指定长度的重复字符的字符串。

StrReverse	反转字符串。
UCase	把指定的字符串转换为大写。

其他函数

Top

函数	描述
CreateObject	创建指定类型的对象。
Eval	计算表达式，并返回结果。
GetLocale	返回当前的 locale ID。
GetObject	返回对文件中 automation 对象的引用。
GetRef	允许您把 VBScript 子程序连接到页面上的一个 DHTML 事件。
InputBox	显示对话框，用户可在其中输入文本，并/或点击按钮，然后返回内容。
IsEmpty	返回一个布尔值，指示指定的变量是否已被初始化。
IsNull	返回一个布尔值，指示指定的表达式是否包含无效数据（Null）。
IsNumeric	返回一个布尔值，指示指定的表达式是否可作为数字来计算。
IsObject	返回一个布尔值，指示指定的表达式是否是一个 automation 对象。
LoadPicture	返回一个图片对象。仅用于 32 位平台。
MsgBox	显示消息框，等待用户点击按钮，并返回指示用户点击了哪个按钮的值。
RGB	返回一个表示 RGB 颜色值的数字。
Round	对数字进行四舍五入。
ScriptEngine	返回使用中的脚本语言。
ScriptEngineBuildVersion	返回使用中的脚本引擎的内部版本号。
ScriptEngineMajorVersion	返回使用中的脚本引擎的主版本号。
ScriptEngineMinorVersion	返回使用中的脚本引擎的次版本号。
SetLocale	设置 locale ID，并返回之前的 locale ID。

TypeName	返回指定变量的子类型。
VarType	返回指示变量子类型的值。

VBScript VarType 函数 VBScript 关键字

VBScript 函数

VBScript 关键字

VBScript 关键字

关键字	描述
Empty	用于指示一个未初始化的变量值。当第一次创建变量时或变量值显式设置为空时，变量值未初始化且变量为被赋值。 实例： Dim x 'x 变量未初始化！ x="ff" 'x 变量不再是未初始化 x=Empty 'x 变量未初始化！ **注意：**这和 Null 不一样！！
IsEmpty	用于测试一个变量是否未初始化。 实例：If (IsEmpty(x)) 'x 变量未初始化？
Nothing	用于指示一个未初始化的对象值，或者把对象变量从对象分离用于释放系统资源。 实例: Set myObject=Nothing
Is Nothing	用于测试一个值是否是初始化的对象。 实例：If (myObject Is Nothing) '它是否未设置？ **注意：**如果您把一个值与 Nothing 作比较，您将不会得到正确的结果！实例: If (myObject = Nothing) '总是错误！
Null	用于指示变量不包含有效数据。 Null 把值设置为"无效"，Empty 则表示值"未设置"。 **注意：**这不同于 Empty 或 Nothing！！ 实例：x=Null 'x 不包含有效数据
IsNull	用于测试一个值是否包含无效数据。 实例: if (IsNull(x)) 'x 是无效的？
True	用于指示一个布尔条件是正确的（True 为 -1）
False	用于指示一个布尔条件是不正确的（False 为 0）

VBScript 函数

2 篇笔记

写笔记

注意： Is Nothing 必须用于判断已经显示赋值的对象变量，否者会报错。

错误的使用：

```
<%
Dim obj
Response.Write "Obj Is Nothing: " & (obj is Nothing) & "<br/>"
%>
```

报错：

```
Microsoft VBScript 运行时错误 错误 '800a01a8'
缺少对象
```

正确的使用：

```
<%
Dim obj
Set obj = Nothing
Response.Write "Obj Is Nothing: " & (obj is Nothing) & "<br/>"
%>
```

结果：

```
Obj Is Nothing: True
```

末苏 9个月前 (06-28)

注意：IsNull 和 IsEmpty 判断可以位于 Dim 定义前使用，哪怕使用 option explicit 头部申明时也不会出错。

```
<%
option explicit
Response.Write "Obj Is Empty: " & IsEmpty(obj) & "<br/>"
Response.Write "Obj Is Null: " & IsNull(obj) & "<br/>"
Dim obj
%>
```

结果：

```
Obj Is Empty: True
Obj Is Null: False
```

末苏 9个月前 (06-28)