

Array()

FUNCTION: 返回一个数组

SYNTAX: Array(list)

ARGUMENTS: 字符，数字均可

EXAMPLE: <%

```
Dim myArray()
```

```
For i = 1 to 7
```

```
Redim Preserve myArray(i)
```

```
myArray(i) = WeekdayName(i)
```

```
Next
```

```
%>
```

RESULT: 建立了一个包含 7 个元素的数组 myArray

```
myArray("Sunday","Monday", ... .. "Saturday")
```

CInt()

FUNCTION: 将一个表达式转化为数字类型

SYNTAX: CInt(expression)

ARGUMENTS: 任何有效的字符均可

EXAMPLE: <%

```
f = "234"
```

```
response.write cINT(f) + 2
```

```
%>
```

RESULT: 236

转化字符 "234" 为数字 "234"，如果字符串为空，则返回 0 值

CreateObject()

FUNCTION: 建立和返回一个已注册的 ACTIVEX 组件的实例。

SYNTAX: CreateObject(objName)

ARGUMENTS: objName 是任何一个有效、已注册的 ACTIVEX 组件的名字。

EXAMPLE: <%

```
Set con = Server.CreateObject("ADODB.Connection")
```

```
%>
```

RESULT:

CStr()

FUNCTION: 转化一个表达式为字符串。

SYNTAX: CStr(expression)

ARGUMENTS: expression 是任何有效的表达式。

EXAMPLE: <%

```
s = 3 + 2
```

```
response.write "The result is: " & cStr(s)
```

```
%>
```

RESULT: 转化数字 “ 5 ” 为字符 “ 5 ”

Date()

FUNCTION: 返回当前系统日期。

SYNTAX: Date()

ARGUMENTS: None.

EXAMPLE: <%=Date%>

RESULT: 8/4/99

DateAdd()

FUNCTION: 返回一个被改变了的日期。

SYNTAX: DateAdd(timeinterval,number,date)

ARGUMENTS: timeinterval is the time interval to add; number is amount of time intervals to add; and date is the starting date.

EXAMPLE: <%

currentDate = #8/4/99#

newDate = DateAdd("m",3,currentDate)

response.write newDate

%>

<%

currentDate = #12:34:45 PM#

newDate = DateAdd("h",3,currentDate)

response.write newDate

%>

RESULT: 11/4/99

3:34:45 PM

"m" = "month";

"d" = "day";

If currentDate is in time format then,

"h" = "hour";

"s" = "second";

DateDiff()

FUNCTION: 返回两个日期之间的差值。

SYNTAX: DateDiff(timeinterval,date1,date2 [, firstdayofweek [, firstweekofyear>>])

ARGUMENTS: timeinterval 表示相隔时间的类型，如 “ M表示 ‘月’ ”。

EXAMPLE: <%

fromDate = #8/4/99#

toDate = #1/1/2000#

response.write "There are " & _

DateDiff("d",fromDate,toDate) & _

" days to millenium from 8/4/99."

%>

RESULT: 从 8/4/99 到 2000 年还有 150 天.

Day()

FUNCTION: 返回一个月的第几日 .

SYNTAX: Day(date)

ARGUMENTS: date 是任何有效的日期。

EXAMPLE: <%=Day(#8/4/99#)%>

RESULT: 4

FormatCurrency()

FUNCTION: 返回表达式，此表达式已被格式化为货币值

SYNTAX: FormatCurrency(Expression [, Digit [, LeadingDigit [, Paren [, GroupDigit>>>>])

ARGUMENTS: Digit 指示小数点右侧显示位数的数值。默认值为 -1，指示使用的是计算机的区域设置；LeadingDigit 三态常数，指示是否显示小数值小数点前面的零。

EXAMPLE: <%=FormatCurrency(34.3456)%>

RESULT: \$34.35

FormatDateTime()

FUNCTION: 返回表达式，此表达式已被格式化为日期或时间

SYNTAX: FormatDateTime(Date, [, NamedFormat>)

ARGUMENTS: NamedFormat 指示所使用的日期 / 时间格式的数值，如果省略，则使用 vbGeneralDate.

EXAMPLE: <%=FormatDateTime("08/4/99", vbLongDate)%>

RESULT: Wednesday, August 04, 1999

FormatNumber()

FUNCTION: 返回表达式，此表达式已被格式化为数值 .

SYNTAX: FormatNumber(Expression [, Digit [, LeadingDigit [, Paren [, GroupDigit>>>>])

ARGUMENTS: Digit 指示小数点右侧显示位数的数值。默认值为 -1，指示使用的是计算机的区域设置。；LeadingDigit i 指示小数点右侧显示位数的数值。默认值为 -1，指示使用的是计算机的区域设置。；Paren 指示小数点右侧显示位数的数值。默认值为 -1，指示使用的是计算机的区域设置。；GroupDigit i 指示小数点右侧显示位数的数值。默认值为 -1，指示使用的是计算机的区域设置。

EXAMPLE: <%=FormatNumber(45.324567, 3)%>

RESULT: 45.325

FormatPercent()

FUNCTION: 返回表达式，此表达式已被格式化为尾随有 % 符号的百分比（乘以 100 后） (%)

SYNTAX: FormatPercent(Expression [, Digit [, LeadingDigit [, Paren [, GroupDigit>>>>])

ARGUMENTS: 同上 .

EXAMPLE: <%=FormatPercent(0.45267, 3)%>
RESULT: 45.267%

Hour()

FUNCTION: 以 24 时返回小时数 .
SYNTAX: Hour(time)
ARGUMENTS:
EXAMPLE: <%=Hour("#4:45:34 PM#")%>
RESULT: 16
(Hour has been converted to 24-hour system)

Instr()

FUNCTION: 返回字符或字符串在另一个字符串中第一次出现的位置 .
SYNTAX: Instr([start, > strToBeSearched, strSearchFor [, compare>)
ARGUMENTS: Start 为搜索的起始值 , strToBeSearched 接受搜索的字符串
strSearchFor 要搜索的字符 compare 比较方式 (详细见 ASP 常数)
EXAMPLE: <%
strText = "This is a test!!"
pos = Instr(strText, "a")
response.write pos
%>
RESULT: 9

InstrRev()

FUNCTION: 同上 , 只是从字符串的最后一个搜索起
SYNTAX: InstrRev([start, > strToBeSearched, strSearchFor [, compare>)
ARGUMENTS: 同上 .
EXAMPLE: <%
strText = "This is a test!!"
pos = InstrRev(strText, "s")
response.write pos
%>
RESULT: 13

Int()

FUNCTION: 返回数值类型 , 不四舍五入。
SYNTAX: Int(number)
ARGUMENTS:
EXAMPLE: <%=INT(32.89)%>
RESULT: 32

IsArray()

FUNCTION: 判断一对象是否为数组 , 返回布尔值 .

SYNTAX: IsArray(name)

ARGUMENTS:

EXAMPLE: <%

strTest = "Test!"

response.write IsArray(strTest)

%>

RESULT: False

IsDate()

FUNCTION: 判断一对象是否为日期，返回布尔值

SYNTAX: IsDate(expression)

ARGUMENTS: expression is any valid expression.

EXAMPLE: <%

strTest = "8/4/99"

response.write IsDate(strTest)

%>

RESULT: True

IsEmpty()

FUNCTION: 判断一对象是否初始化，返回布尔值 .

SYNTAX: IsEmpty(expression)

ARGUMENTS:

EXAMPLE: <%

Dim i

response.write IsEmpty(i)

%>

RESULT: True

IsNull()

FUNCTION: 判断一对象是否为空，返回布尔值 .

SYNTAX: IsNull(expression)

ARGUMENTS:

EXAMPLE: <%

Dim i

response.write IsNull(i)

%>

RESULT: False

IsNumeric()

FUNCTION: 判断一对象是否为数字，返回布尔值 .

SYNTAX: IsNumeric(expression)

ARGUMENTS:

EXAMPLE: <%

i = "345"

```
response.write    IsNumeric(i)
%>
RESULT: True
就算数字加了引号，    ASP  还是认为它是数字。
```

IsObject()

FUNCTION: 判断一对象是否为对象，返回布尔值 .

SYNTAX: IsObject(expression)

ARGUMENTS:

EXAMPLE: <%

```
Set con = Server.CreateObject("ADODB.Connection")
response.write    IsObject(con)
%>
RESULT: True
```

LBound()

FUNCTION: 返回指定数组维的最小可用下标 .

SYNTAX: Lbound(arrayname [, dimension>)

ARGUMENTS: dimension 指明要返回哪一维下界的整数。 使用 1 表示第一维， 2 表示第二维，以此类推。如果省略 dimension 参数，默认值为 1.

EXAMPLE: <%

```
i = Array("Monday","Tuesday","Wednesday")
response.write    LBound(i)
%>
RESULT: 0
```

LCase()

FUNCTION: 返回字符串的小写形式

SYNTAX: Lcase(string)

ARGUMENTS: string is any valid string expression.

EXAMPLE: <%

```
strTest = "This is a test!"
response.write    LCase(strTest)
%>
RESULT: this is a test!
```

Left()

FUNCTION: 返回字符串左边第 length 个字符以前的字符（含第 length 个字符）.

SYNTAX: Left(string, length)

ARGUMENTS:

EXAMPLE: <%

```
strTest = "This is a test!"
response.write    Left(strTest, 3)
```

```
%>  
RESULT: Thi
```

Len()

```
FUNCTION:  返回字符串的长度 .  
SYNTAX: Len(string | varName)  
ARGUMENTS:  
EXAMPLE:  <%  
strTest  = "This  is a test!"  
response.write  Len(strTest)  
%>  
RESULT: 15
```

LTrim()

```
FUNCTION:  去掉字符串左边的空格 .  
SYNTAX: LTrim(string)  
ARGUMENTS:  
EXAMPLE:  <%  
strTest  = " This  is a test!"  
response.write  LTrim(strTest)  
%>  
RESULT: This  is a test!
```

Mid()

```
FUNCTION:  返回特定长度的字符串 (从 start 开始 ,长度为 length).  
SYNTAX: Mid(string, start [, length>)  
ARGUMENTS:  
EXAMPLE:  <%  
strTest  = "This  is a test! Today  is Monday."  
response.write  Mid(strTest, 17, 5)  
%>  
RESULT: Today
```

Minute()

```
FUNCTION:  返回时间的分钏 .  
SYNTAX: Minute(time)  
ARGUMENTS:  
EXAMPLE:  <%=Minute(#12:45:32    PM#)%>  
RESULT: 45
```

Month()

```
FUNCTION:  返回日期 .  
SYNTAX: Month(date)  
ARGUMENTS:  date is any valid date expression.
```

EXAMPLE: <%=Month(#08/04/99#)%>

RESULT: 8

MonthName()

FUNCTION: Returns a string identifying the specified month.

SYNTAX: MonthName(month, [, Abb>)

ARGUMENTS: month is the numeric representation for a given month; Abb (optional) is a boolean value used to display month abbreviation. True will display the abbreviated month name and False (default) will not show the abbreviation.

EXAMPLE: <%=MonthName(Month(#08/04/99#))%>

RESULT: August

Now()

FUNCTION: Returns the current system date and time.

SYNTAX: Now()

ARGUMENTS: None

EXAMPLE: <%=Now%>

RESULT: 8/4/99 9:30:16 AM

Replace()

FUNCTION: Returns a string in which a specified sub-string has been replaced with another substring a specified number of times.

SYNTAX: Replace(strToBeSearched, strSearchFor, strReplaceWith [, start [, count [, compare>>>])

ARGUMENTS: strToBeSearched is a string expression containing a sub-string to be replaced; strSearchFor is the string expression to search for within strToBeSearched; strReplaceWith is the string expression to replace sub-string strSearchFor; start (optional) is the numeric character position to begin search; count (optional) is a value indicating the comparison constant.

EXAMPLE: <%

strTest = "This is an apple!"

response.write Replace(strTest, "apple", "orange")

%>

RESULT: This is an orange!

Right()

FUNCTION: 返回字符串右边第 length 个字符以前的字符 (含第 length 个字符).

SYNTAX: Right(string, length)

ARGUMENTS: .

EXAMPLE: <%

strTest = "This is an test!"

response.write Right(strTest, 3)

%>

RESULT: st!

Rnd()

FUNCTION: 产生一个随机数 .
SYNTAX: Rnd [(number) >
ARGUMENTS:
EXAMPLE: <%
Randomize()
response.write RND()
%>
RESULT: 任何一个在 0 到 1 之间的数

Round()

FUNCTION: 返回按指定位数进行四舍五入的数值 .
SYNTAX: Round(expression [, numRight>
ARGUMENTS: numRight 数字表明小数点右边有多少位进行四舍五入。 如果省略, 则

Round 函数返回整数 .

EXAMPLE: <%
i = 32.45678
response.write Round(i)
%>
RESULT: 32

Rtrim()

FUNCTION: 去掉字符串右边的字符串 .
SYNTAX: Rtrim(string)
ARGUMENTS:
EXAMPLE: <%
strTest = "This is a test!! "
response.write RTrim(strTest)
%>
RESULT: This is a test!!

Second()

FUNCTION: 返回秒 .
SYNTAX: Second(time)
ARGUMENTS: .
EXAMPLE: <%=Second(#12:34:28 PM#)%>
RESULT: 28

StrReverse()

FUNCTION: 反排一字符串
SYNTAX: StrReverse(string)
ARGUMENTS:
EXAMPLE: <%

```
strTest = "This is a test!!"
response.write StrReverse(strTest)
%>
RESULT: !!tset a si sihT
```

Time()

FUNCTION: 返回系统时间 .

SYNTAX: Time()

ARGUMENTS: .

EXAMPLE: <%=Time%>

RESULT: 9:58:28 AM

Trim()

FUNCTION: 去掉字符串左右的空格 .

SYNTAX: Trim(string)

ARGUMENTS: string is any valid string expression.

EXAMPLE: <%

```
strTest = " This is a test!! "
```

```
response.write Trim(strTest)
```

%>

RESULT: This is a test!!

UBound()

FUNCTION: 返回指定数组维数的最大可用下标 .

SYNTAX: Ubound(arrayname [, dimension>)

ARGUMENTS: dimension (optional) 指定返回哪一维上界的整数。 1 表示第一维 , 2 表示第二维 , 以此类推。如果省略 dimension 参数 , 则默认值为 1.

EXAMPLE: <%

```
i = Array("Monday","Tuesday","Wednesday")
```

```
response.write UBound(i)
```

%>

RESULT: 2

UCase()

FUNCTION: 返回字符串的大写形式 .

SYNTAX: UCase(string)

ARGUMENTS:

EXAMPLE: <%

```
strTest = "This is a test!!"
```

```
response.write UCase(strTest)
```

%>

RESULT: THIS IS A TEST!!

VarType()

FUNCTION: 返回指示变量类型的值

SYNTAX: VarType(varName)

ARGUMENTS:

EXAMPLE: <%

i = 3

response.write varType(i)

%>

RESULT: 2(数字) 详见 "asp 常数 "

WeekDay()

FUNCTION: 返回在一周的第几天 .

SYNTAX: WeekDay(date [, firstdayofweek>)

ARGUMENTS: .

EXAMPLE: <%

d = #8/4/99#

response.write Weekday(d)

%>

RESULT: 4(星期三)

WeekDayName()

FUNCTION: 返回一周第几天的名字 .

SYNTAX: WeekDayName(weekday [, Abb [, firstdayofweek>>)

ARGUMENTS: Abb 可选。 Boolean 值 , 指明是否缩写表示星期各天的名称。如果省略 , 默认值为 False , 即不缩写星期各天的名称 .firstdayofweek 指明星期第一天的数值

EXAMPLE: <%

d = #8/4/99#

response.write WeekdayName(Weekday(d))

%>

RESULT: Wednesday

Year()

FUNCTION: 返回当前的年份 .

SYNTAX: Year(date)

ARGUMENTS:

EXAMPLE: <%=Year(#8/4/99#)%>

RESULT: 1999

先写这些 , 本人主要还是 php 比较擅长。 O(_)O~