

码栈

开发指南

开发指南

编码开发

日期相关函数

J_Date_Add

计算一个指定的日期和间隔后的新日期，时间格式为:YYYY-MM-DD。

函数

J_Date_Add(日期,要添加的数量,[模式:日])

- 日期：指定日期；
- 要添加的数量：可以是正数也可以是负数；
- 模式：模式:年/Y 月/M 日/D 时/H 分/N 秒/S。

返回值

计算后的日期，格式为：YYYY-MM-DD。

样例

```
Func Example32()
;声明一个名为$d1的局域变量，保存日期格式的文本"2016-03-04 23:00:35"
Local $d1 = "2016-03-04 23:00:35"
;声明一个名为$d2的局域变量，保存日期格式的文本"2016-03-05 06:07:09"
```

```
Local $d2 = "2016-03-05 06:07:09"  
;声明一个名为$d3的局域变量，保存日期格式的文本"2016-03-06 14:50:21"  
Local $d3 = "2016-03-06 14:50:21"  
;声明一个名为$d4的局域变量，保存在日期$d1的基础上后退3天(D)  
Local $d4 = J_Date_Add($d1, 3, "D")  
;$d2和$d3相差多少个小时(H)  
Local $diff = J_Date_Diff($d2, $d3, "H")  
;以系统信息提示框的形式显示出$d4保存的数据  
J_MsgBox($d4)  
;以系统信息提示框的形式显示出$diff保存的数据  
J_MsgBox($diff)  
EndFunc ;==>Example32
```

J_Date_DayOfWeek

获取某日期的星期值，返回1-7分别代表周一到周日。

函数

J_Date_DayOfWeek([日期:系统时间])

说明：默认值为当前时间。

返回值

某日期的星期值。

样例

```
Func Example35()  
;取当天的星期值  
j_msgbox(J_Date_DayOfWeek())  
;获取2017-03-08那天的星期值  
j_msgbox(J_Date_DayOfWeek('2017-03-08'))  
  
EndFunc ;==>Example35
```

J_Date_Diff

返回两个日期的不同，使用指定类型表达。

函数

J_Date_Diff(开始日期,结束日期[,模式:日])

- 开始日期：指定开始日期；
- 结束日期：指定结束日期；
- 模式：年/Y 月/M 日/D 时/H 分/N 秒/S，建议使用英文。

返回值

返回两个日期间隔的数字。

样例

```
Func Example32()  
;声明一个名为$d1的局域变量，保存日期格式的文本"2016-03-04 23:00:35"  
Local $d1 = "2016-03-04 23:00:35"  
;声明一个名为$d2的局域变量，保存日期格式的文本"2016-03-05 06:07:09"  
Local $d2 = "2016-03-05 06:07:09"  
;声明一个名为$d3的局域变量，保存日期格式的文本"2016-03-06 14:50:21"  
Local $d3 = "2016-03-06 14:50:21"  
;声明一个名为$d4的局域变量，保存在日期$d1的基础上后退3天(D)  
Local $d4 = J_Date_Add($d1, 3, "D")  
;$d2和$d3相差多少个小时(H)  
Local $diff = J_Date_Diff($d2, $d3, "H")  
;以系统信息提示框的形式显示出$d4保存的数据  
J_MsgBox($d4)  
;以系统信息提示框的形式显示出$diff保存的数据  
J_MsgBox($diff)  
EndFunc ;==>Example32
```

J_Date_lastDayOfMonth

获取某日期所在月份的最后一天。

函数

J_Date_lastDayOfMonth([日期:系统时间])

说明：默认值为当前日期。

返回值

某日期所在月份的最后一天。

样例

```
Func Example34()  
;取当前月份的最后一天  
j_msgbox(J_Date_lastDayOfMonth())  
  
;取2017-04月的最的一天  
j_msgbox(j_date_lastDayOfMonth('2017-04'))  
j_msgbox(j_date_lastDayOfMonth('2017-04-08'))  
  
EndFunc ;==>Example34
```

J_Date_sysDate

根据格式，返回特定格式的当前系统时间。

函数

J_Date_sysDate([格式:yyyy-mm-dd hh:mm:ss])

返回值

特定格式的当前时间，默认格式是 yyyy-mm-dd hh:mm:ss。

样例

```
Func Example33()  
;取当年月日  
j_msgbox(J_Date_sysDate('yyyy-mm-dd'))  
;取年月日 时分秒  
j_msgbox(J_Date_sysDate('yyyy-mm-dd hh:mi:ss'))  
;另一种格式  
j_msgbox(J_Date_sysDate('yyyy/mm/dd hh:mi:ss'))  
EndFunc ;==>Example33
```

J_Date_trunc

获取日期中的信息，如年、月、日。

函数

J_Date_trunc(日期[模式:日])

模式可选值：日/天/年/月/时/分/秒

返回值

反馈指定日期和模式的信息。

样例

```
Func Example36()  
j_msgbox(j_date_trunc('2015-01-05', '年'))  
j_msgbox(j_date_trunc('2015-01-05 12:34:09', '月'))  
j_msgbox(j_date_trunc('2015-01-05 12:34:09', '天'))  
j_msgbox(j_date_trunc('2015-01-05', '时'))  
j_msgbox(j_date_trunc('2015-01-05 12:34:09', '时'))  
j_msgbox(j_date_trunc('2015-01-05 12:34:09', '分'))  
j_msgbox(j_date_trunc('2015-01-05 12:34:09', '秒'))  
EndFunc ;==>Example36
```

Excel相关

J_Excel_Close

关闭Excel。

函数

J_Excel_Close(Excel对象, [默认保存:是])

返回值

成功返回1，失败返回0。

样例

```
Func Example38()
;新创建一个Excel，可见
Local $excel1 = J_Excel_Create("是")
;在后台打开Excel，不可见
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")
;Excel表格一共有多少行
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")
;Excel中一共有多少列
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")
;拷贝数据，从1行1列到最大行最大列
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")
;粘贴数据
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")
;读取Excel1中新2_sheet中的第2行第3列
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")
;信息框显示出读取的数据
J_MsgBox($string)
;删除Excel1的sheet1
J_Excel_DeleteSheet($excel1, "sheet1")
;将Excel1保存至D:\2.xlsx
J_Excel_Save($excel1, "D:\2.xlsx")
;关闭Excel1，保存
J_Excel_Close($excel1)
;关闭Excel2，不保存
J_Excel_Close($excel2, "否")
EndFunc ;==>Example38g", "D:\5.jpg", 700, 1000)
EndFunc ;==>Example40
```

J_Excel_CopySell

向Excel指定Sheet复制数据。

注意：:sheet名称为空的时候，是向当前激活Sheet中复制。

函数

J_Excel_CopySell(Excel对象, 起始单元格行, 起始单元格列,结束单元格行, 结束单元格列,[Sheet名称])

返回值

成功返回1，失败返回0。

样例

```
Func Example38()  
;新建一个Excel，可见  
Local $excel1 = J_Excel_Create("是")  
;在后台打开Excel，不可见  
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")  
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")  
;Excel表格一共有多少行  
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")  
;Excel中一共有多少列  
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")  
;拷贝数据，从1行1列到最大行最大列  
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")  
;粘贴数据  
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")  
;读取Excel1中新2_sheet中的第2行第3列  
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")  
;信息框显示出读取的数据  
J_MsgBox($string)  
;删除Excel1的sheet1  
J_Excel_DeleteSheet($excel1, "sheet1")  
;将Excel1保存至D:\2.xlsx  
J_Excel_Save($excel1, "D:\2.xlsx")  
;关闭Excel1，保存  
J_Excel_Close($excel1)  
;关闭Excel2，不保存  
J_Excel_Close($excel2, "否")
```



```
EndFunc ;==>Example38
```

J_Excel_Create

新建Excel并返回对象。

函数

J_Excel_Create([是否可见:否])

返回值

Excel对象。

样例

```
Func Example38()  
;新创建一个Excel，可见  
Local $excel1 = J_Excel_Create("是")  
;在后台打开Excel，不可见  
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")  
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")  
;Excel表格一共有多少行  
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")  
;Excel中一共有多少列  
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")  
;拷贝数据，从1行1列到最大行最大列  
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")  
;粘贴数据  
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")  
;读取Excel1中新2_sheet中的第2行第3列  
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")  
;信息框显示出读取的数据  
J_MsgBox($string)  
;删除Excel1的sheet1  
J_Excel_DeleteSheet($excel1, "sheet1")  
;将Excel1保存至D:\2.xlsx  
J_Excel_Save($excel1, "D:\2.xlsx")  
;关闭Excel1，保存  
J_Excel_Close($excel1)  
;关闭Excel2，不保存  
J_Excel_Close($excel2, "否")  
EndFunc ;==>Example38
```

J_Excel_DeleteSheet

删除Excel表格中指定的Sheet。

函数

J_Excel_DeleteSheet(Excel对象,Sheet名称)

返回值

成功返回1，失败返回0。

样例

```
Func Example38()  
;新创建一个Excel，可见  
Local $excel1 = J_Excel_Create("是")  
;在后台打开Excel，不可见  
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")  
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")  
;Excel表格一共有多少行  
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")  
;Excel中一共有多少列  
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")  
;拷贝数据，从1行1列到最大行最大列  
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")  
;粘贴数据  
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")  
;读取Excel1中新2_sheet中的第2行第3列  
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")  
;信息框显示出读取的数据  
J_MsgBox($string)  
;删除Excel1的sheet1  
J_Excel_DeleteSheet($excel1, "sheet1")  
;将Excel1保存至D:\2.xlsx  
J_Excel_Save($excel1, "D:\2.xlsx")  
;关闭Excel1，保存  
J_Excel_Close($excel1)  
;关闭Excel2，不保存  
J_Excel_Close($excel2, "否")  
EndFunc ;==>Example38
```

J_Excel_GetSheetNames

获取Excel表格中指定的Sheet个数和名称。

函数

J_Excel_GetSheetNames(Excel对象,序号=0)

返回值

序号为0的时候，返回sheet的个数。

当序号值大于0的时候，返回sheet所对应的名称。

样例

```
Func Example50()  
;定义一个局域变量$excel,用于保存打开的Excel表格对象。Excel表格打开后处于可见状态  
Local $excel = J_Excel_Open("E:\数据.xlsx", "是")  
;定义一个局域变量，用于保存$excel对象中Sheet的个数  
Local $res = J_Excel_GetSheetNames($excel, 0)  
;创建一个循环，从1到Sheet的最大值，用于遍历每一个Sheet的名字  
For $i = 1 To $res  
;定义一个局域变量$name,用于保存$i个Sheet的名字  
Local $name = J_Excel_GetSheetNames($excel, $i)  
;弹出一个提示框，显示当前是第几个sheet和sheet的名字  
J_MsgBox($name, "第" & $i & "个")  
Next  
EndFunc ;==>Example50
```

J_Excel_MaxCol

获取Excel表格中指定Sheet的数据列数。

注意:sheet名称为空的时候，是获取当前激活Sheet中的列数。

函数

J_Excel_MaxCol(Excel对象,[Sheet名称])

返回值

成功返回Excel表格的列数，失败返回0。

样例

```
Func Example38()  
;新建一个Excel，可见  
Local $excel1 = J_Excel_Create("是")  
;在后台打开Excel，不可见  
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")  
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")  
;Excel表格一共有多少行  
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")  
;Excel中一共有多少列  
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")  
;拷贝数据，从1行1列到最大行最大列  
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")  
;粘贴数据  
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")  
;读取Excel1中新2_sheet中的第2行第3列  
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")  
;信息框显示出读取的数据  
J_MsgBox($string)  
;删除Excel1的sheet1  
J_Excel_DeleteSheet($excel1, "sheet1")  
;将Excel1保存至D:\2.xlsx  
J_Excel_Save($excel1, "D:\2.xlsx")  
;关闭Excel1，保存  
J_Excel_Close($excel1)  
;关闭Excel2，不保存  
J_Excel_Close($excel2, "否")  
EndFunc ;=>Example38
```

J_Excel_MaxCount

获取Excel表格中指定Sheet的数据行数。

注意:sheet名称为空的时候，是获取当前激活Sheet中的行数。

函数

J_Excel_MaxCount(Excel对象,[Sheet名称])

返回值

成功返回Excel表格的行数，失败返回0。

样例

```
Func Example38()  
;新建一个Excel，可见  
Local $excel1 = J_Excel_Create("是")  
;在后台打开Excel，不可见  
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")  
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")  
;Excel表格一共有多少行  
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")  
;Excel中一共有多少列  
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")  
;拷贝数据，从1行1列到最大行最大列  
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")  
;粘贴数据  
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")  
;读取Excel1中新2_sheet中的第2行第3列  
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")  
;信息框显示出读取的数据  
J_MsgBox($string)  
;删除Excel1的sheet1  
J_Excel_DeleteSheet($excel1, "sheet1")  
;将Excel1保存至D:\2.xlsx  
J_Excel_Save($excel1, "D:\2.xlsx")  
;关闭Excel1，保存  
J_Excel_Close($excel1)  
;关闭Excel2，不保存  
J_Excel_Close($excel2, "否")  
EndFunc ;==>Example38
```

J_Excel_Open

打开一个Excel文件并返回对象。

函数

J_Excel_Open(文件路径, [是否可见:否], [只读:否],[文件密码], [设置密码])

返回值

Excel对象。

样例

```
Func Example38()  
;新创建一个Excel , 可见  
Local $excel1 = J_Excel_Create("是")  
;在后台打开Excel , 不可见  
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")  
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")  
;Excel表格一共有多少行  
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")  
;Excel中一共有多少列  
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")  
;拷贝数据 , 从1行1列到最大行最大列  
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")  
;粘贴数据  
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")  
;读取Excel1中新2_sheet中的第2行第3列  
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")  
;信息框显示出读取的数据  
J_MsgBox($string)  
;删除Excel1的sheet1  
J_Excel_DeleteSheet($excel1, "sheet1")  
;将Excel1保存至D:\2.xlsx  
J_Excel_Save($excel1, "D:\2.xlsx")  
;关闭Excel1 , 保存  
J_Excel_Close($excel1)  
;关闭Excel2 , 不保存  
J_Excel_Close($excel2, "否")  
EndFunc ;==>Example38
```

J_Excel_PasteSell

向Excel指定Sheet粘贴数据。

注意：sheet名称为空的时候，是向当前激活Sheet中粘贴。

函数

J_Excel_PasteSell(Excel对象, 单元格行, 单元格列, [Sheet名称])

返回值

成功返回1，失败返回0。

样例

```
Func Example38()  
;新建一个Excel，可见  
Local $excel1 = J_Excel_Create("是")  
;在后台打开Excel，不可见  
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")  
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")  
;Excel表格一共有多少行  
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")  
;Excel中一共有多少列  
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")  
;拷贝数据，从1行1列到最大行最大列  
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")  
;粘贴数据  
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")  
;读取Excel1中新2_sheet中的第2行第3列  
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")  
;信息框显示出读取的数据  
J_MsgBox($string)  
;删除Excel1的sheet1  
J_Excel_DeleteSheet($excel1, "sheet1")  
;将Excel1保存至D:\2.xlsx  
J_Excel_Save($excel1, "D:\2.xlsx")  
;关闭Excel1，保存  
J_Excel_Close($excel1)  
;关闭Excel2，不保存  
J_Excel_Close($excel2, "否")  
EndFunc ;=>Example38
```

J_Excel_Read

向Excel指定Sheet读取数据。

注意:sheet名称为空的时候，是向当前激活Sheet中读取。

函数

J_Excel_Read(Excel对象, 第几行, 第几列, [Sheet名称])

返回值

成功返回相应单元格的数据，失败返回空。

样例

```
Func Example38()  
;新建一个Excel，可见  
Local $excel1 = J_Excel_Create("是")  
;在后台打开Excel，不可见  
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")  
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")  
;Excel表格一共有多少行  
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")  
;Excel中一共有多少列  
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")  
;拷贝数据，从1行1列到最大行最大列  
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")  
;粘贴数据  
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")  
;读取Excel1中新2_sheet中的第2行第3列  
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")  
;信息框显示出读取的数据  
J_MsgBox($string)  
;删除Excel1的sheet1  
J_Excel_DeleteSheet($excel1, "sheet1")  
;将Excel1保存至D:\2.xlsx  
J_Excel_Save($excel1, "D:\2.xlsx")  
;关闭Excel1，保存  
J_Excel_Close($excel1)  
;关闭Excel2，不保存  
J_Excel_Close($excel2, "否")
```



```
EndFunc ;==>Example38
```

J_Excel_Save

保存或者另存当前Excel文件，当不输入另存路径的时候就是保存当前已打开的Excel。

函数

J_Excel_Save(Excel对象,[另存路径])

返回值

成功返回1，失败返回0。

样例

```
Func Example38()  
;新创建一个Excel，可见  
Local $excel1 = J_Excel_Create("是")  
;在后台打开Excel，不可见  
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")  
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")  
;Excel表格一共有多少行  
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")  
;Excel中一共有多少列  
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")  
;拷贝数据，从1行1列到最大行最大列  
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")  
;粘贴数据  
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")  
;读取Excel1中新2_sheet中的第2行第3列  
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")  
;信息框显示出读取的数据  
J_MsgBox($string)  
;删除Excel1的sheet1  
J_Excel_DeleteSheet($excel1, "sheet1")  
;将Excel1保存至D:\2.xlsx  
J_Excel_Save($excel1, "D:\2.xlsx")  
;关闭Excel1，保存  
J_Excel_Close($excel1)  
;关闭Excel2，不保存  
J_Excel_Close($excel2, "否")  
EndFunc ;==>Example38
```

J_Excel_Write

向Excel指定Sheet写入数据(文本格式).

说明:sheet名称为空的时候,是向当前激活Sheet中写入。

函数

J_Excel_Write(Excel对象, 第几行, 第几列, 要写入的数据, [Sheet名称])

返回值

成功返回1, 失败返回0。

样例

```
Func Example38()  
;新建一个Excel, 可见  
Local $excel1 = J_Excel_Create("是")  
;在后台打开Excel, 不可见  
Local $excel2 = J_Excel_Open("D:\1.xlsx", "否")  
J_Excel_Write($excel1, 1, 1, "hello", "new_sheet")  
;Excel表格一共有多少行  
Local $maxrow = J_Excel_MaxCount($excel2, "sheet1")  
;Excel中一共有多少列  
Local $maxcol = J_Excel_MaxCol($excel2, "sheet1")  
;拷贝数据, 从1行1列到最大行最大列  
J_Excel_CopySell($excel2, 1, 1, $maxrow, $maxcol, "sheet1")  
;粘贴数据  
J_Excel_PasteSell($excel1, 1, 1, "new2_sheet")  
;读取Excel1中新2_sheet中的第2行第3列  
Local $string = J_Excel_Read($excel1, 2, 3, "new2_sheet")  
;信息框显示出读取的数据  
J_MsgBox($string)  
;删除Excel1的sheet1  
J_Excel_DeleteSheet($excel1, "sheet1")  
;将Excel1保存至D:\2.xlsx  
J_Excel_Save($excel1, "D:\2.xlsx")  
;关闭Excel1, 保存  
J_Excel_Close($excel1)  
;关闭Excel2, 不保存  
J_Excel_Close($excel2, "否")
```

```
EndFunc ;==>Example38
```

文件相关函数

J_File

文件的一些基础操作，动作可选值：

- 文件操作：新建文件/删除文件/复制文件/移动文件/文件改名/获取文件大小；
- 文件夹操作：新建文件夹/删除文件夹/复制文件夹/移动文件夹/文件夹改名/获取文件夹大小。

函数

J_File(文件(夹)路径,动作,值)

返回值

参数	说明
1	成功
0	失败

样例

```
func Example47()  
;文件夹操作：  
local $path = 'D:\file\测试文件'  
local $res = J_File($path,"是否存在")  
;新建  
if $res = 0 then  
J_File($path,"新建文件夹")  
EndIf  
;移动  
J_File($path,"移动文件夹","C:\file\测试文件")  
;复制  
J_File("C:\file\测试文件","复制文件夹","D:\file\测试文件new")  
;改名
```

```
J_File("D:\file\测试文件new","文件夹改名","D:\file\测试文件rename")
;文件夹大小
local $size = J_File("D:\code","获取文件夹大小")
Msgbox(0,"","大小(MB) : " & Round($size / 1024 / 1024))
;删除文件夹
J_File("D:\nw2","删除文件夹")

;文件操作 :
local $filepath = 'D:\file\测试文件rename\11.txt'
$res = J_File($filepath,"是否存在")
;新建
if $res = 0 then
J_File($filepath,"新建文件")
EndIf
;写值
J_File_Write($filepath,'你好 , 码栈','追加')
;获取大小
local $size = J_File($filepath,"获取文件大小")
j_msgbox("文件大小 : 字节" & $size)

;移动:将文件移动到目录 : C:\file\测试文件下
J_File($filepath,"移动文件","C:\file\测试文件")

;复制:将文件复制到目录:D:\file下
J_File($filepath,"复制文件","D:\file")

;改名
J_File($filepath,"文件改名","C:\file\测试文件\22.txt")

;删除文件
J_File($filepath,"删除文件")
endfunc; ==>
```

J_File_MaxCount

获取文件里面一共有多少行。

函数

J_File_MaxCount(文件路径)

返回值

获取文件里面一共有多少行。

样例

```
Func Example31()  
;声明一个名为$filepath的局域变量，创建一个文件选择对话框，用户可以选择文件，返回选择的文件路径，并将此路径保存至  
$filepath变量中  
Local $filepath = J_File_OpenDialog()  
;声明一个名为$max的局域变量，将$filepath文件里面内容行数保存至这个变量中  
Local $max = J_File_MaxCount($filepath)  
;创建一个循环，从1到$max  
For $i = 1 To $max  
;声明一个名为$nr的局域变量，将$filepath文件里面第$i行的内容保存至这个变量中  
Local $nr = J_File_Read($filepath, $i)  
;@crlf的意思是换行符，另起一行  
;通过系统信息提示框显示拼凑的文本  
J_MsgBox("第" & $i & "行" & @CRLF & $nr)  
;下一次循环  
Next  
;当读取文件内容第几行设置为0的时候，此函数会读取文件中的全部文本  
Local $alltext = J_File_Read($filepath, 0)  
J_MsgBox($alltext)  
;向文件尾部追加数据  
J_File_Write($filepath, "追加一条数据", "追加")  
Local $alltext = J_File_Read($filepath, 0)  
;校验追加的数据是否成功  
J_MsgBox($alltext)  
;把文件的数据替换掉  
J_File_Write($filepath, "覆盖过后的数据", "覆盖")  
Local $alltext = J_File_Read($filepath, 0)  
;校验覆盖的数据是否成功  
J_MsgBox($alltext)  
EndFunc ;==>Example31
```

J_File_OpenDialog

用于弹出一个文件选择窗口。

函数

J_File_OpenDialog([标题默认:请选择文件])

返回值

用户选择文件的路径。

样例

```
Func Example31()  
;声明一个名为$filepath的局域变量，创建一个文件选择对话框，用户可以选择文件，返回选择的文件路径，并将此路径保存至  
$filepath变量中  
Local $filepath = J_File_OpenDialog()  
;声明一个名为$max的局域变量，将$filepath文件里面内容行数保存至这个变量中  
Local $max = J_File_MaxCount($filepath)  
;创建一个循环，从1到$max  
For $i = 1 To $max  
;声明一个名为$nr的局域变量，将$filepath文件里面第$i行的内容保存至这个变量中  
Local $nr = J_File_Read($filepath, $i)  
;@crlf的意思是换行符，另起一行  
;通过系统信息提示框显示拼凑的文本  
J_MsgBox("第" & $i & "行" & @CRLF & $nr)  
;下一次循环  
Next  
;当读取文件内容第几行设置为0的时候，此函数会读取文件中的全部文本  
Local $alltext = J_File_Read($filepath, 0)  
J_MsgBox($alltext)  
;向文件尾部追加数据  
J_File_Write($filepath, "追加一条数据", "追加")  
Local $alltext = J_File_Read($filepath, 0)  
;校验追加的数据是否成功  
J_MsgBox($alltext)  
;把文件的数据替换掉  
J_File_Write($filepath, "覆盖过后的数据", "覆盖")  
Local $alltext = J_File_Read($filepath, 0)  
;校验覆盖的数据是否成功  
J_MsgBox($alltext)  
EndFunc ;==>Example31
```

J_File_PathToAllFileName

获得文件夹下面全部的文件名称集合，可以配合J_EXP系列函数使用。

函数

J_File_PathToAllFileName(文件夹路径)

返回值

返回文件夹下所有文件的集合。

样例

```
Func Example46()  
;获得文件夹下所有文件的集合  
Local $file_all = J_File_PathToAllFileName("D:\pic\  
;查看集合文件的个数  
Local $max = UBound($file_all)  
;创建一个For循环，用来遍历全部文件  
For $i = 1 To $max  
;取出当前循环文件的名称  
Local $file = J_EXP($file_all, "", $i)  
;在当前“我的工具”视图中创建这条数据  
J_CreateItem($file)  
Next  
EndFunc ;==>Example46
```

J_File_Read

获取文件里面某一行的数据,当行号为0的时候默认读取全部内容。

函数

J_File_Read(文件路径,第几行)

返回值

返回文件指定行的内容。

样例

```
Func Example31()  
;声明一个名为$filepath的局域变量，创建一个文件选择对话框，用户可以选择文件，返回选择的文件路径，并将此路径保存至  
$filepath变量中  
Local $filepath = J_File_OpenDialog()  
;声明一个名为$max的局域变量，将$filepath文件里面内容行数保存至这个变量中  
Local $max = J_File_MaxCount($filepath)  
;创建一个循环，从1到$max  
For $i = 1 To $max  
;声明一个名为$nr的局域变量，将$filepath文件里面第$i行的内容保存至这个变量中  
Local $nr = J_File_Read($filepath, $i)  
;@crlf的意思是换行符，另起一行
```

```
;通过系统信息提示框显示拼凑的文本
J_MsgBox("第" & $i & "行" & @CRLF & $nr)
;下一次循环
Next
;当读取文件内容第几行设置为0的时候，此函数会读取文件中的全部文本
Local $alltext = J_File_Read($filepath, 0)
J_MsgBox($alltext)
;向文件尾部追加数据
J_File_Write($filepath, "追加一条数据", "追加")
Local $alltext = J_File_Read($filepath, 0)
;校验追加的数据是否成功
J_MsgBox($alltext)
;把文件的数据替换掉
J_File_Write($filepath, "覆盖过后的数据", "覆盖")
Local $alltext = J_File_Read($filepath, 0)
;校验覆盖的数据是否成功
J_MsgBox($alltext)
EndFunc ;==>Example31
```

J_File_Write

写入文件数据，默认追加模式，当文件不存在的时候会自动创建，模式分为：

- 追加：向文件末尾追加要写入的内容；
- 覆盖：把原始文件内容替换为要写入的内容。

函数

J_File_Write(文件路径,写入内容,[模式:追加/覆盖])

返回值

参数	说明
1	成功
0	失败

样例

```
Func Example31()
;声明一个名为$filepath的局域变量，创建一个文件选择对话框，用户可以选择文件，返回选择的文件路径，并将此路径保存至
```



```
$filepath变量中
Local $filepath = J_File_OpenDialog()
;声明一个名为$max的局域变量，将$filepath文件里面内容行数保存至这个变量中
Local $max = J_File_MaxCount($filepath)
;创建一个循环，从1到$max
For $i = 1 To $max
;声明一个名为$nr的局域变量，将$filepath文件里面第$i行的内容保存至这个变量中
Local $nr = J_File_Read($filepath, $i)
;@crlf的意思是换行符，另起一行
;通过系统信息提示框显示拼凑的文本
J_MsgBox("第" & $i & "行" & @CRLF & $nr)
;下一次循环
Next
;当读取文件内容第几行设置为0的时候，此函数会读取文件中的全部文本
Local $alltext = J_File_Read($filepath, 0)
J_MsgBox($alltext)
;向文件尾部追加数据
J_File_Write($filepath, "追加一条数据", "追加")
Local $alltext = J_File_Read($filepath, 0)
;校验追加的数据是否成功
J_MsgBox($alltext)
;把文件的数据替换掉
J_File_Write($filepath, "覆盖过后的数据", "覆盖")
Local $alltext = J_File_Read($filepath, 0)
;校验覆盖的数据是否成功
J_MsgBox($alltext)
EndFunc ;==>Example31
5.5 J_File_Create(文件路径)
作用：创建一个文件。
返回值：返回1
例子：
Func Example31_1()
J_File_Create("D:\1.txt")
EndFunc ;==>Example31_1
```

浏览器相关函数

J_EXP_MaxCount

返回满足此正则表达式匹配的个数。

函数

J_EXP_MaxCount(总字符串,正则表达式)

返回值

返回满足此正则表达式匹配的个数。

样例

```
Func Example28()  
;声明一个名为$ie的局域变量，将通过浏览器窗口标题“淘宝网”获取浏览器对象保存这个变量中  
Local $ie = J_IE_Catch("淘宝网", "标题")  
;注意J_IE_EXP_MaxCount必须要配合控件录制工具使用  
;声明 $max的一个局域变量，在$ie变量中对应的浏览器页面中检索，对应的控件相同属性的在页面中一共有多少个，并保存至$max变量中  
;一般情况下用于循环或者判断  
Local $max = J_IE_EXP_MaxCount($ie, "淘宝网", "搜索框exp")  
;创建一个循环，从1开始$max中保存的数字截止，注意，如果没写step的话步长默认为1  
For $i = 1 To $max  
;声明 $result的一个局域变量，在$ie变量中对应的浏览器页面对应的第$i个控件执行输入hello的操作，并将结果保存至$result变量中(执行成功1，执行失败0)  
Local $result = J_IE_EXP("淘宝网", "搜索框exp", $i, "输入", "hello", $ie)  
;下一次循环  
Next  
EndFunc ;=>Example28
```

J_IE

此函数需要配合码栈控件抓取工具使用，对IE控件以及Windows控件进行一系列操作。

- IE操作

- 点击、模拟点击、弹出框点击、模拟弹出框点击、模拟输入、输入；
- 文件、保存文件、下拉、模糊下拉、正则下拉、获取文本、获取代码、获取坐标、获取对象、获取值、勾选、取消勾选、等待加载、是否勾选；
- 注意获取坐标返回的是一个长度为5的坐标数组[0]全部[1]x1[2]y1[3]x2[4]y2。

- WIN32操作

- 点击、模拟点击、输入、模拟输入、移动、检测窗口、关闭窗口、强行关闭窗口、获取句柄。

说明:如果是点击按钮后会有弹出框，那么请用弹出框点击。

函数

J_IE (MAP名称,控件名称,操作,[值],[目标窗口])

返回值

- 当操作为：点击、模拟点击、弹出框点击、模拟弹出框点击、模拟输入、输入、文件、下拉、模糊下拉、正则下拉、勾选、取消勾选、检测窗口、关闭窗口、强行关闭窗口的时候，成功返回1，失败返回0；
- 当操作为：获取文本的时候返回值为控件文本；
- 当操作位：获取代码的时候返回值为控件内部代码。

样例

Func Example22()

;新建一个IE窗口并打开www.crm.admin.com链接,声明一个名为\$ie的局域变量，装入这个新创建浏览器的对象

Local \$ie = J_IE_Create("www.crm.admin.com")

;普通输入操作，成功返回1，失败返回0

Local \$res1 = J_IE("CRM", "登录输入框", "输入", "镖头", \$ie)

;模拟鼠标键盘进行操作，成功返回1，失败返回0

Local \$res2 = J_IE("CRM", "密码输入框", "模拟输入", "镖头", \$ie)

;普通点击按钮操作，因为不需要输入数据，所以值参数用 "" 表示不要输入，成功返回1，失败返回0

Local \$res3 = J_IE("CRM", "登录按钮", "点击", "", \$ie)

;模拟鼠标点击按钮操作，因为不需要输入数据，所以值参数用 "" 表示不要输入，成功返回1，失败返回0

Local \$res4 = J_IE("CRM", "刷新", "模拟点击", "", \$ie)

;在显示区域读取里面可见文本，返回文本

Local \$res5 = J_IE("CRM", "显示区域", "获取文本", "", \$ie)

;获取文本框用户输入的信息，返回文本

Local \$res6 = J_IE("CRM", "输入框", "获取值", "", \$ie)

;对于页面复选框进行勾选操作，成功返回1，失败返回0

Local \$res7 = J_IE("CRM", "复选框", "勾选", "", \$ie)

;获取页面勾选框的状态，勾选状态返回1，未勾选返回0

Local \$res8 = J_IE("CRM", "复选框", "是否勾选", "", \$ie)

;下拉列表操作，成功返回1，失败返回0

Local \$res9 = J_IE("CRM", "小二选择", "下拉", "镖头", \$ie)

;通过部分关键字进行下拉列表动作，成功返回1，失败返回0

Local \$res10 = J_IE("CRM", "小二选择", "模糊下拉", "镖", \$ie)

;上传文件控件操作，成功返回1失败返回0

Local \$res11 = J_IE("CRM", "文件按钮", "文件", "D:\pic.jpg", \$ie)

;等待刷新区域的刷新，刷新完毕返回1

Local \$res12 = J_IE("CRM", "刷新区域", "等待加载", "", \$ie)

;点击页面上的控件会弹出提示框是否确认信息，本功能会自动化点击弹出框的确认按钮，成功返回1，失败返回0

Local \$res13 = J_IE("CRM", "提交", "弹出框点击", "", \$ie)

;模拟鼠标点击页面上的控件会弹出提示框是否确认信息，本功能会自动化点击弹出框的确认按钮，成功返回1，失败返回0

Local \$res14 = J_IE("CRM", "提交", "模拟弹出框点击", "", \$ie)

;对WIN控件输入，不需要指定窗体，成功返回1，失败返回0

Local \$res15 = J_IE("旺旺", "旺旺输入框", "输入", "你好")

;对WIN控件模拟键盘输入操作，不需要指定窗体，成功返回1，失败返回0

Local \$res16 = J_IE("旺旺", "旺旺输入框", "模拟输入", "你好")

;对WIN控件点击操作，不需要指定窗体，成功返回1，失败返回0

Local \$res17 = J_IE("旺旺", "发送按钮", "点击", "")

;对WIN控件模拟鼠标点击操作，不需要指定窗体，成功返回1，失败返回0

Local \$res18 = J_IE("旺旺", "发送按钮", "模拟点击", "")

;获取WIN控件句柄，成功返回控件句柄，失败返回0

Local \$res19 = J_IE("旺旺", "旺旺控件", "获取句柄", "")

```
;关闭WIN控件所在的窗口，成功返回1，失败返回0
Local $res20 = J_IE("旺旺", "旺旺控件", "关闭窗口", "")

;J_IE和J_IE_EXP的区别，前者是对页面唯一的控件进行操作，第二个是对页面多个相同的控件指定位置进行操作，其它功能完全相同
;比如下面两个功能是相同的
;普通输入操作，成功返回1，失败返回0
Local $res1 = J_IE("CRM", "登录输入框", "输入", "镖头", $ie)
Local $res1 = J_IE_EXP("CRM", "登录输入框", 1, "输入", "镖头", $ie);对第一个此属性的控件进行操作

;声明 $max的一个局域变量，在$ie变量中对应的浏览器页面中检索，对应的控件相同属性的在页面中一共有多少个，并保存至$max变量中
;一般情况下用于循环或者判断
Local $max = J_IE_EXP_MaxCount($ie, "CRM", "登录输入框")
EndFunc ;==>Example22
```

J_IE_Back

浏览器执行后退动作。

函数

J_IE_Back(浏览器对象)

返回值

参数	说明
1	成功
0	失败

样例

```
Func Example43()
;声明一个居于变量，保存通过链接地址定位的浏览器对象
Local $ie = J_IE_Catch("taobao.com", "地址")
;在浏览器中实现点击女装类目操作
J_IE("淘宝网", "女装类目", "点击", "", $ie)
;页面刷新后实现，浏览器后退功能
J_IE_Back($ie)
EndFunc ;==>Example43
```

J_IE_Catch

捕获已打开的浏览器对象，模式为：标题、地址、子窗口。

函数

J_IE_Catch(字符串,抓取模式)

返回值

成功返回浏览器对象，失败返回0。

样例

```
Func Example27()  
;声明一个名为$ie的局域变量，将通过浏览器窗口标题“淘宝网”获取浏览器对象保存这个变量中  
Local $ie = J_IE_Catch("淘宝网", "标题")  
;声明一个名为$ie2的局域变量，将通过浏览器URL地址www.taobao.com获取浏览器对象保存这个变量中  
Local $ie2 = J_IE_Catch("www.taobao.com", "地址")  
;如果要抓chrome浏览器对象:注意，catch的窗口必须是通过程序打开出来的chrome窗口，可以用J_IE_Create  
local $ie3 = J_IE_Catch("www.taobao.com", "地址", "Chrome")  
EndFunc ;==>Example27
```

J_IE_Close

可以通过IE窗体的对象、标题、地址、甚至是句柄关闭IE窗口。

函数

J_IE_Close (窗口对象/窗口标题/窗口URL/窗口句柄)

返回值

参数	说明
----	----

1	成功
0	失败

样例

```
Func Example23()  
;新建一个IE窗口并打开www.taobao.com链接,声明一个名为$ie的局域变量,装入这个新创建浏览器的对象  
Local $ie = J_IE_Create("www.taobao.com")  
;通过变量$ie中保存的浏览器对象进行关闭浏览器操作  
J_IE_Close($ie)  
;新建一个IE窗口并打开ju.taobao.com链接,声明一个名为$ie的局域变量,装入这个新创建浏览器的对象  
Local $ie = J_IE_Create("ju.taobao.com")  
;通过浏览器窗口标题"聚划算"进行关闭浏览器操作  
J_IE_Close("聚划算")  
EndFunc ;==>Example23
```

J_IE_Create

创建一个IE窗口，返回一个窗口的对象。

函数

J_IE_Create (链接地址,[尝试抓取:否],[是否等待:是],[浏览器类型])

返回值

返回浏览器对象，用于后续操作。

样例

```
Func Example21()  
;默认新建一个IE窗口并打开www.taobao.com链接,声明一个名为$ie的局域变量,装入这个新创建浏览器的对象  
Local $ie = J_IE_Create("www.taobao.com")  
  
;如果要用chrome打开  
;Local $ie = J_IE_Create("www.taobao.com",default,default,default,'Chrome')  
  
;左侧文件名 控件名 操作类型 内容 浏览器对象  
;声明 $res1的一个局域变量,实现对于录制的控件进行输入文本“连衣裙”操作,并将执行的结果保存在这个变量中
```

```
Local $res1 = J_IE("淘宝网", "输入框", "输入", "连衣裙", $ie)
;判断操作是否成功，如果输入操作没成功，$res1的值为0，操作成功$res1的值就是1
If $res1 = 0 Then
;信息提示窗口显示提示信息给用户"连衣裙输入失败"
J_MsgBox("连衣裙输入失败")
;结束本次判断
EndIf
;声明 $res1的一个局域变量，实现对于录制的控件进行对控件“搜索”按钮进行点击操作，并将执行的结果保存在这个变量中
Local $res2 = J_IE("淘宝网", "搜索", "点击", "", $ie)
;判断操作是否成功，如果输入操作没成功，$res2的值为0，操作成功$res2的值就是1
If $res2 = 0 Then
;信息提示窗口显示提示信息给用户"搜索点击失败"
J_MsgBox("搜索点击失败")
;结束本次判断
EndIf
;声明 $text的一个局域变量，将$ie对应的浏览器对象获取页面所有可见文本，并将其保存至此变量中
Local $text = J_IE_Text($ie)
;信息提示窗口显示出变量$text中保存的全部文本
J_MsgBox($text)
EndFunc ;==>Example21
```

J_IE_Download

模拟浏览器中点击下载保存动作，将文件保存到指定的地方。文件名可以特别指定，指定时，要注意文件类型。

函数

J_IE_Download(\$url, \$path, \$fileName = " ")

返回值

参数	说明
1	成功
0	失败

异步下载，不等待文件下载完就返回结果。

样例

```
func downloadTest()  
;需先登录商家后台  
Local $url = "https://trade.taobao.com/trade/itemlist/export_by_tfs.do?f_p=TB1EfhdOpXXXXXQXVXXUXJ7WFXX-  
a9722f6c478344a5124fb76750a88441-orders&apply_time=2016-11-18+14%3A08%3A51&start_time=2016-08-  
20+14%3A08%3A51&end_time=2016-11-18+14%3A08%3A51&order_status=%BD%BB%D2%D7%B9%D8%B1%D5"  
  
local $result = J_IE_Download($url, 'd:/nw/')  
if $result = 0 then  
local $msg = J_GetErrorMsg()  
J_MsgBox($msg)  
endif  
EndFunc ;==>downloadTest
```

J_IE_DragAndDrop

拖拽到指定位置，规则如下：1、拖拽的起始位置为控件的中心点；2、偏移量是相对于此控件的中心点来说的，如果向左拖，x轴偏移值应该是负数，如果向上拖，y轴偏移值应该是负数。

函数

J_IE_DragAndDrop(浏览器对象，MAP对象，控件对象，x偏移量，y偏移量)

返回值

无

样例

```
func test()  
local $ie = J_IE_Catch("转账", "标题", "chrome")  
local $array = J_IE("支付宝", "验>>", "获取坐标", "", $ie)  
local $x1 = $array[1]  
local $x2 = $array[3]  
local $array2 = J_IE("支付宝", "验证域", "获取坐标", "", $ie)  
local $lx2 = $array2[3]  
;要拖的长度：  
local $xoffset = $lx2 - ($x2 + $x1) / 2  
j_log($xoffset)  
  
local $result = J_IE_DragAndDrop($ie, "支付宝", "验>>", $xoffset)  
if $result = 0 then
```



```
local $errorMsg = J_GetErrorMsg()
J_MsgBox($errorMsg)
endif
endfunc ;==>test
```

J_IE_EXP

对正则IE控件进行一系列操作：

- 操作：点击、模拟点击、弹出框点击、模拟弹出框点击、模拟输入、输入；
- 操作：文件、下拉、模糊下拉、正则下拉、获取文本、获取代码；
- 操作：获取对象、获取值、勾选、取消勾选、等待加载。

如果是WIN32控件，操作：检测窗口、关闭窗口、强行关闭窗口。

说明：如果是点击按钮后会有弹出框，那么请用弹出框点击。

函数

J_IE_EXP (MAP名称,控件名称,位置=1,操作,[值],[目标窗口])

返回值

当操作为：点击、模拟点击、弹出框点击、模拟弹出框点击、模拟输入、输入、文件、下拉、模糊下拉、正则下拉、勾选、取消勾选、检测窗口、关闭窗口、强行关闭窗口的时候，成功返回1，失败返回0；当操作为：获取文本的时候返回值为控件文本；当操作位：获取代码的时候返回值为控件内部代码。

样例

```
Func Example28()
;声明一个名为$ie的局域变量，将通过浏览器窗口标题“淘宝网”获取浏览器对象保存这个变量中
Local $ie = J_IE_Catch("淘宝网", "标题")
;注意J_IE_EXP_MaxCount必须要配合控件录制工具使用
;声明 $max的一个局域变量，在$ie变量中对应的浏览器页面中检索，对应的控件相同属性的在页面中一共有多少个，并保存至$max变量中
;一般情况下用于循环或者判断
Local $max = J_IE_EXP_MaxCount($ie, "淘宝网", "搜索框exp")
;创建一个循环，从1开始$max中保存的数字截止，注意，如果没写step的话步长默认为1
For $i = 1 To $max
;声明 $result的一个局域变量，在$ie变量中对应的浏览器页面对应的第$i个控件执行输入hello的操作，并将结果保存至
```

```
$result变量中(执行成功1，执行失败0)  
Local $result = J_IE_EXP("淘宝网", "搜索框exp", $i, "输入", "hello", $ie)  
;下一次循环  
Next  
EndFunc ;==>Example28
```

J_IE_GetScrollTop

获取浏览器滚动条高度。

函数

J_IE_GetScrollTop(浏览器对象)

返回值

获取滚动条高度。

样例

```
func test()  
local $ie = j_ie_create("www.taobao.com",default,default,default,"chrome")  
;滚一屏  
J_IE_ScrollPage($ie)  
local $top = J_IE_GetScrollTop($ie)  
j_msgbox($top)  
endfunc
```

J_IE_Html

获得IE浏览器上的全部Html代码。

函数

J_IE_Html(IE对象)

返回值

IE浏览器上的全部Html代码。

样例

```
Func Example26()  
;新建一个IE窗口并打开www.crm.admin.com链接,声明一个名为$ie的局域变量,装入这个新创建浏览器的对象  
Local $ie = J_IE_Create("www.taobao.com")  
;声明 $html的一个局域变量,获取$ie对应的浏览器对象获取页面所有渲染的代码,并将其保存至此变量中  
Local $html = J_IE_Html($ie)  
;信息提示窗口显示出变量$text中保存的全部文本  
J_MsgBox($html)  
EndFunc ;==>Example26
```

J_IE_Link

通过已知浏览器打开一个链接。

函数

J_IE_Link(IE对象,链接地址,[是否等待:是])

返回值

浏览器对象

样例

```
Func Example24()  
;新建一个IE窗口并打开www.taobao.com链接,声明一个名为$ie的局域变量,装入这个新创建浏览器的对象  
Local $ie = J_IE_Create("www.taobao.com")  
;通过变量$ie中保存的浏览器对象所对应的浏览器打开新的链接地址ju.taobao.com  
J_IE_Link($ie, "ju.taobao.com")  
EndFunc ;==>Example24
```

J_IE_ScrollPage

让浏览器滚动一屏。

函数

J_IE_ScrollPage(浏览器对象)

返回值

参数	说明
1	其他
0	滚动条到底或滚动失败

样例

```
func test()
local $ie = J_IE_Catch("淘宝", "标题", 'chrome')
local $result
Do
$value = J_IE_ScrollPage($ie)
Until($value = 0)
endfunc ;==>test
```

J_IE_SetScrollTop

设置浏览器滚动条高度。

函数

J_IE_SetScrollTop(浏览器对象，滚动条高度)

返回值

参数	说明
----	----

1	成功
0	失败

样例

```
func test()
local $ie = J_IE_Catch("淘宝", "标题", 'chrome')
J_IE_SetScrollTop($ie, 300)
endfunc
```

J_IE_Tab_Debug

调试IE表格,一般情况下用于测试。

函数

J_IE_Tab_Debug(表格对象)

返回值

无。

样例

```
Func Example30()
;声明一个名为$ie的局域变量，将通过浏览器窗口标题 “淘宝TSP” 获取浏览器对象保存这个变量中
Local $ie = J_IE_Catch("淘宝TSP", "标题")
;声明一个名为$tab的局域变量，在变量$ie对应的浏览器页面中通过表格可见文本的正则表达式定位表格，并将表格文本对象保存至变量$tab中
Local $tab = J_IE_Tab_FindByText($ie, "买家真实姓名", "文本")
;声明一个名为$tab2的局域变量，在变量$ie对应的浏览器页面中通过表格的内嵌HTML代码的正则表达式定位表格，并将表格html代码对象保存至变量$tab中
Local $tab2 = J_IE_Tab_FindByHtml($ie, "<TD class=ktitle a", "html")
;弹出一个弹出框，显示$tab中对应的表格对象
J_IE_Tab_Debug($tab)
;弹出一个弹出框，显示$tab2中对应的表格对象
J_IE_Tab_Debug($tab2)
;声明一个名为$maxcol的局域变量，获取$tab变量中所对应的表格的列数，并将其保存至$maxcol中
Local $maxcol = J_IE_Tab_MaxCol($tab)
```

```
;声明一个名为$maxcount的局域变量，获取$tab变量中所对应的表格的行数，并将其保存至$maxcount中
Local $maxcount = J_IE_Tab_MaxCount($tab)
;创建一个For循环，从1循环到$maxcount
For $c = 1 To $maxcount
;在第一个循环下创建一个循环，从1循环到$maxcol
For $w = 1 To $maxcol
;声明一个名为$text的局域变量，将$tab表格中对应的$c行$w列的数据保存其中
Local $text = J_IE_Tab_Read($tab, $c, $w)
;通过系统信息提示框的方式显示出$text中的数据
J_MsgBox($text)
;下一次循环，对应For $w = 1 To $maxcol
Next
;在上述循环执行完毕之后才能执行到这里
;下一次循环，对应For $c = 1 To $maxcount
Next
EndFunc ;==>Example30
```

J_IE_Tab_FindByHtml

通过html正则表达式获取表格对象，模式：文本,html。

函数

J_IE_Tab_FindByHtml(IE对象,正则表达式,[模式:文本])

返回值

成功返回表格对象，失败返回0。

样例

```
Func Example30()
;声明一个名为$ie的局域变量，将通过浏览器窗口标题“淘宝TSP”获取浏览器对象保存这个变量中
Local $ie = J_IE_Catch("淘宝TSP", "标题")
;声明一个名为$tab的局域变量，在变量$ie对应的浏览器页面中通过表格可见文本的正则表达式定位表格，并将表格文本对象保存至变量$tab中
Local $tab = J_IE_Tab_FindByText($ie, "买家真实姓名", "文本")
;声明一个名为$tab2的局域变量，在变量$ie对应的浏览器页面中通过表格的内嵌HTML代码的正则表达式定位表格，并将表格html代码对象保存至变量$tab中
Local $tab2 = J_IE_Tab_FindByHtml($ie, "<TD class=ktile a", "html")
;弹出一个弹出框，显示$tab中对应的表格对象
J_IE_Tab_Debug($tab)
;弹出一个弹出框，显示$tab2中对应的表格对象
```

```
J_IE_Tab_Debug($tab2)
;声明一个名为$maxcol的局域变量，获取$tab变量中所对应的表格的列数，并将其保存至$maxcol中
Local $maxcol = J_IE_Tab_MaxCol($tab)
;声明一个名为$maxcount的局域变量，获取$tab变量中所对应的表格的行数，并将其保存至$maxcount中
Local $maxcount = J_IE_Tab_MaxCount($tab)
;创建一个For循环，从1循环到$maxcount
For $c = 1 To $maxcount
;在第一个循环下创建一个循环，从1循环到$maxcol
For $w = 1 To $maxcol
;声明一个名为$text的局域变量，将$tab表格中对应的$c行$w列的数据保存其中
Local $text = J_IE_Tab_Read($tab, $c, $w)
;通过系统信息提示框的方式显示出$text中的数据
J_MsgBox($text)
;下一次循环，对应For $w = 1 To $maxcol
Next
;在上述循环执行完毕之后才能执行到这里
;下一次循环，对应For $c = 1 To $maxcount
Next
EndFunc ;==>Example30
```

J_IE_Tab_FindByText

通过文本正则表达式获取表格对象。

函数

J_IE_Tab_FindByText(IE对象,正则表达式)

返回值

成功返回表格对象，失败返回0。

样例

```
Func Example30()
;声明一个名为$ie的局域变量，将通过浏览器窗口标题“淘宝TSP”获取浏览器对象保存这个变量中
Local $ie = J_IE_Catch("淘宝TSP", "标题")
;声明一个名为$tab的局域变量，在变量$ie对应的浏览器页面中通过表格可见文本的正则表达式定位表格，并将表格文本对象保存至变量$tab中
Local $tab = J_IE_Tab_FindByText($ie, "买家真实姓名", "文本")
;声明一个名为$tab2的局域变量，在变量$ie对应的浏览器页面中通过表格的内嵌HTML代码的正则表达式定位表格，并将表格html代码对象保存至变量$tab中
Local $tab2 = J_IE_Tab_FindByHtml($ie, "<TD class=ktile a", "html")
```

```
;弹出一个弹出框，显示$tab中对应的表格对象
J_IE_Tab_Debug($tab)
;弹出一个弹出框，显示$tab2中对应的表格对象
J_IE_Tab_Debug($tab2)
;声明一个名为$maxcol的局域变量，获取$tab变量中所对应的表格的列数，并将其保存至$maxcol中
Local $maxcol = J_IE_Tab_MaxCol($tab)
;声明一个名为$maxcount的局域变量，获取$tab变量中所对应的表格的行数，并将其保存至$maxcount中
Local $maxcount = J_IE_Tab_MaxCount($tab)
;创建一个For循环，从1循环到$maxcount
For $c = 1 To $maxcount
;在第一个循环下创建一个循环，从1循环到$maxcol
For $w = 1 To $maxcol
;声明一个名为$text的局域变量，将$tab表格中对应的$c行$w列的数据保存其中
Local $text = J_IE_Tab_Read($tab, $c, $w)
;通过系统信息提示框的方式显示出$text中的数据
J_MsgBox($text)
;下一次循环，对应For $w = 1 To $maxcol
Next
;在上述循环执行完毕之后才能执行到这里
;下一次循环，对应For $c = 1 To $maxcount
Next
EndFunc ;==>Example30
```

J_IE_Tab_MaxCol

获得IE表格一共多少列。

函数

J_IE_Tab_MaxCol(表格对象)

返回值

获得IE表格一共多少列。

样例

```
Func Example30()
;声明一个名为$ie的局域变量，将通过浏览器窗口标题“淘宝TSP”获取浏览器对象保存这个变量中
Local $ie = J_IE_Catch("淘宝TSP", "标题")
;声明一个名为$tab的局域变量，在变量$ie对应的浏览器页面中通过表格可见文本的正则表达式定位表格，并将表格文本对象保存至变量$tab中
Local $tab = J_IE_Tab_FindByText($ie, "买家真实姓名", "文本")
```



```
;声明一个名为$tab2的局域变量，在变量$ie对应的浏览器页面中通过表格的内嵌HTML代码的正则表达式定位表格，并将表格html代码对象保存至变量$tab中
Local $tab2 = J_IE_Tab_FindByHtml($ie, "<TD class=ktitle a", "html")
;弹出一个弹出框，显示$tab中对应的表格对象
J_IE_Tab_Debug($tab)
;弹出一个弹出框，显示$tab2中对应的表格对象
J_IE_Tab_Debug($tab2)
;声明一个名为$maxcol的局域变量，获取$tab变量中所对应的表格的列数，并将其保存至$maxcol中
Local $maxcol = J_IE_Tab_MaxCol($tab)
;声明一个名为$maxcount的局域变量，获取$tab变量中所对应的表格的行数，并将其保存至$maxcount中
Local $maxcount = J_IE_Tab_MaxCount($tab)
;创建一个For循环，从1循环到$maxcount
For $c = 1 To $maxcount
;在第一个循环下创建一个循环，从1循环到$maxcol
For $w = 1 To $maxcol
;声明一个名为$text的局域变量，将$tab表格中对应的$c行$w列的数据保存其中
Local $text = J_IE_Tab_Read($tab, $c, $w)
;通过系统信息提示框的方式显示出$text中的数据
J_MsgBox($text)
;下一次循环，对应For $w = 1 To $maxcol
Next
;在上述循环执行完毕之后才能执行到这里
;下一次循环，对应For $c = 1 To $maxcount
Next
EndFunc ;==>Example30
```

J_IE_Tab_MaxCount

获得IE表格一共多少行。

函数

J_IE_Tab_MaxCount(表格对象)

返回值

获得IE表格一共多少行。

样例

```
Func Example30()
;声明一个名为$ie的局域变量，将通过浏览器窗口标题“淘宝TSP”获取浏览器对象保存这个变量中
Local $ie = J_IE_Catch("淘宝TSP", "标题")
```

```
;声明一个名为$tab的局域变量，在变量$ie对应的浏览器页面中通过表格可见文本的正则表达式定位表格，并将表格文本对象保存至变量$tab中
Local $tab = J_IE_Tab_FindByText($ie, "买家真实姓名", "文本")
;声明一个名为$tab2的局域变量，在变量$ie对应的浏览器页面中通过表格的内嵌HTML代码的正则表达式定位表格，并将表格html代码对象保存至变量$tab中
Local $tab2 = J_IE_Tab_FindByHtml($ie, "<TD class=ktitle a", "html")
;弹出一个弹出框，显示$tab中对应的表格对象
J_IE_Tab_Debug($tab)
;弹出一个弹出框，显示$tab2中对应的表格对象
J_IE_Tab_Debug($tab2)
;声明一个名为$maxcol的局域变量，获取$tab变量中所对应的表格的列数，并将其保存至$maxcol中
Local $maxcol = J_IE_Tab_MaxCol($tab)
;声明一个名为$maxcount的局域变量，获取$tab变量中所对应的表格的行数，并将其保存至$maxcount中
Local $maxcount = J_IE_Tab_MaxCount($tab)
;创建一个For循环，从1循环到$maxcount
For $c = 1 To $maxcount
;在第一个循环下创建一个循环，从1循环到$maxcol
For $w = 1 To $maxcol
;声明一个名为$text的局域变量，将$tab表格中对应的$c行$w列的数据保存其中
Local $text = J_IE_Tab_Read($tab, $c, $w)
;通过系统信息提示框的方式显示出$text中的数据
J_MsgBox($text)
;下一次循环，对应For $w = 1 To $maxcol
Next
;在上述循环执行完毕之后才能执行到这里
;下一次循环，对应For $c = 1 To $maxcount
Next
EndFunc ;==>Example30
```

J_IE_Tab_Read

获得IE表格中的文本。

函数

J_IE_Tab_Read(表格对象,第几行,第几列)

返回值

成功返回表格相应的文本，失败返回空。

样例

```
Func Example30()  
;声明一个名为$ie的局域变量，将通过浏览器窗口标题“淘宝TSP”获取浏览器对象保存这个变量中  
Local $ie = J_IE_Catch("淘宝TSP", "标题")  
;声明一个名为$tab的局域变量，在变量$ie对应的浏览器页面中通过表格可见文本的正则表达式定位表格，并将表格文本对象  
保存至变量$tab中  
Local $tab = J_IE_Tab_FindByText($ie, "买家真实姓名", "文本")  
;声明一个名为$tab2的局域变量，在变量$ie对应的浏览器页面中通过表格的内嵌HTML代码的正则表达式定位表格，并将表格  
html代码对象保存至变量$tab中  
Local $tab2 = J_IE_Tab_FindByHtml($ie, "<TD class=ktitle a", "html")  
;弹出一个弹出框，显示$tab中对应的表格对象  
J_IE_Tab_Debug($tab)  
;弹出一个弹出框，显示$tab2中对应的表格对象  
J_IE_Tab_Debug($tab2)  
;声明一个名为$maxcol的局域变量，获取$tab变量中所对应的表格的列数，并将其保存至$maxcol中  
Local $maxcol = J_IE_Tab_MaxCol($tab)  
;声明一个名为$maxcount的局域变量，获取$tab变量中所对应的表格的行数，并将其保存至$maxcount中  
Local $maxcount = J_IE_Tab_MaxCount($tab)  
;创建一个For循环，从1循环到$maxcount  
For $c = 1 To $maxcount  
;在第一个循环下创建一个循环，从1循环到$maxcol  
For $w = 1 To $maxcol  
;声明一个名为$text的局域变量，将$tab表格中对应的$c行$w列的数据保存其中  
Local $text = J_IE_Tab_Read($tab, $c, $w)  
;通过系统信息提示框的方式显示出$text中的数据  
J_MsgBox($text)  
;下一次循环，对应For $w = 1 To $maxcol  
Next  
;在上述循环执行完毕之后才能执行到这里  
;下一次循环，对应For $c = 1 To $maxcount  
Next  
EndFunc ;==>Example30
```

J_IE_Text

获得IE浏览器上的全部文本。

函数

J_IE_Text(IE对象)

返回值

获得IE浏览器上的全部文本。

样例

```
Func Example25()  
;新建一个IE窗口并打开www.crm.admin.com链接,声明一个名为$ie的局域变量,装入这个新创建浏览器的对象  
Local $ie = J_IE_Create("www.taobao.com")  
;声明 $text的一个局域变量,将$ie对应的浏览器对象获取页面所有可见文本,并将其保存至此变量中  
Local $text = J_IE_Text($ie)  
;信息提示窗口显示出变量$text中保存的全部文本  
J_MsgBox($text)  
EndFunc ;==>Example25
```

J_IE_Wait

等待录制的控件刷新。

函数

J_IE_Wait(浏览器对象,MAP名称,控件名称,[超时时间:10000毫秒])

返回值

参数	说明
1	成功
0	失败

样例

```
Func Example29()  
;新建一个IE窗口并打开www.crm.admin.com链接,声明一个名为$ie的局域变量,装入这个新创建浏览器的对象  
Local $ie = J_IE_Create("www.baidu.com")  
;声明 $res1的一个局域变量, $ie对应的浏览器页面等待录制控件工具中相对应的控件在20000毫秒内出现,并将等待结果保存至变量中(成功1,失败0)  
Local $result = J_IE_Wait($ie, "淘宝网", "搜索框exp", 20000)  
;等待结束并将等待的结果通过信息提示窗口的形式显示出来  
J_MsgBox("等待结束：" & $result)  
EndFunc ;==>Example29
```

图片相关函数

J_Pic_ReadText

抓取图片上的文字，可以是url地址或者是本地路径。

函数

J_Pic_ReadText(图片路径/图片地址)

返回值

图片上的文字，如果返回值为[OCR:Error]那么证明图片识别失败。

样例

```
Func Example33()  
;声明一个名为$path1的局域变量，保存文本 "D:\1.jpg"  
Local $path1 = "D:\1.jpg"  
;声明一个名为$path2的局域变量，保存文本  
"http://img04.taobaocdn.com/bao/uploaded/i4/T1h8VkJX8XXXb1upjX.jpg"  
Local $path2 = "http://img04.taobaocdn.com/bao/uploaded/i4/T1h8VkJX8XXXb1upjX.jpg"  
;使用此功能需要电脑装有相应插件  
;声明一个名为$value1的局域变量，保存从$path1路径的图片中提取的中文文本  
Local $value1 = J_Pic_ReadText($path1, "中文")  
;声明一个名为$value2的局域变量，保存从$path2路径的图片中提取的英文文本  
Local $value2 = J_Pic_ReadText($path2, "英文")  
;以系统信息提示框的形式显示出$value1保存的数据  
J_MsgBox($value1)  
;以系统信息提示框的形式显示出$value2保存的数据  
J_MsgBox($value2)  
EndFunc ;==>Example33
```

J_Pic_WaterMark_Start

开启目标图片作为打水印的主图，必须先使用此函数。

函数

J_Pic_WaterMark_Start(原始图片地址,保存图片地址)

返回值

无。

样例

```
Func Example40()  
;截取屏幕起始坐标1,1 结束坐标为500,800 的屏幕矩形区域，并保存文件为D:\1.jpg  
J_Pic_Capture("D:\1.jpg", 1, 1, 500, 800)  
;开始图片处理，初始文件是"D:\1.jpg"，处理之后打算把文件另存为"D:\2.jpg"  
J_Pic_WaterMark_Start("D:\1.jpg", "D:\2.jpg")  
;在被处理图片的100,50 像素处打上logo水印（水印路径：D:\logo.png）  
J_Pic_WaterMark_Logo("D:\logo.png", 100, 50)  
;在被处理图片的400,400 像素处打上文本水印，文本字体为红色，字体微软雅黑，字号12号，带下划线的  
J_Pic_WaterMark_Text("taobao.com", 400, 400, 0xFFFF0000, "微软雅黑", 12, "下划线")  
;在被处理图片的450,450像素处打上文本水印，字型字号等都是默认的  
J_Pic_WaterMark_Text("copy right", 450, 450)  
;结束打水印操作，设置保存的图片清晰度为50%  
J_Pic_WaterMark_End(50)  
;等比例设置图片D:\2.jpg高度为1000，并另存为D:\3.jpg  
J_Pic_Resize("D:\2.jpg", "D:\3.jpg", 1000, "高度")  
;等比例设置图片D:\2.jpg宽度为2000，并另存为D:\4.jpg  
J_Pic_Resize("D:\2.jpg", "D:\4.jpg", 2000, "宽度")  
;设置图片D:\2.jpg宽度为700高度为1000，并另存为D:\5.jpg  
J_Pic_Resize("D:\2.jpg", "D:\5.jpg", 700, 1000)  
EndFunc ;==>Example40
```

J_Pic_WaterMark_Text

在已经打开的图片中打文字水印。

函数

J_Pic_WaterMark_Text(文本, X坐标,Y坐标,[颜色:0xff000000],[字体:宋体], [字符大小:12], [模式:正常/粗体/斜体/下划线/删除线])

返回值

无。

样例

```
Func Example40()  
;截取屏幕起始坐标1,1 结束坐标为500,800 的屏幕矩形区域，并保存文件为D:\1.jpg  
J_Pic_Capture("D:\1.jpg", 1, 1, 500, 800)  
;开始图片处理，初始文件是"D:\1.jpg"，处理之后打算把文件另存为"D:\2.jpg"  
J_Pic_WaterMark_Start("D:\1.jpg", "D:\2.jpg")  
;在被处理图片的100,50 像素处打上logo水印（水印路径：D:\logo.png）  
J_Pic_WaterMark_Logo("D:\logo.png", 100, 50)  
;在被处理图片的400,400 像素处打上文本水印，文本字体为红色，字体微软雅黑，字号12号，带下划线的  
J_Pic_WaterMark_Text("taobao.com", 400, 400, 0xFFFF0000, "微软雅黑", 12, "下划线")  
;在被处理图片的450,450像素处打上文本水印，字型字号等都是默认的  
J_Pic_WaterMark_Text("copy right", 450, 450)  
;结束打水印操作，设置保存的图片清晰度为50%  
J_Pic_WaterMark_End(50)  
;等比例设置图片D:\2.jpg高度为1000，并另存为D:\3.jpg  
J_Pic_Resize("D:\2.jpg", "D:\3.jpg", 1000, "高度")  
;等比例设置图片D:\2.jpg宽度为2000，并另存为D:\4.jpg  
J_Pic_Resize("D:\2.jpg", "D:\4.jpg", 2000, "宽度")  
;设置图片D:\2.jpg宽度为700高度为1000，并另存为D:\5.jpg  
J_Pic_Resize("D:\2.jpg", "D:\5.jpg", 700, 1000)  
EndFunc ;==>Example40
```

文本字符串相关

J_EXP

通过正则表达式获得指定的文本。

函数

J_EXP(总字符串,正则表达式,[位置:1])

返回值

当位置等于0的时候返回值为1或者0，代表是否符合此正则表达式。

样例

```
Func Example35()  
;声明一个名为$string的局域变量，保存文本 "hello1234hi4567good7890"  
Local $string = "hello1234hi4567good7890"  
;方法1,优点，速度快，一般用于批量提取  
;声明一个名为$open的局域变量，保存从文本$string中提取满足正则表达式的数据集合  
Local $open = J_EXP_Open($string, "\d+")  
;声明一个名为$max的局域变量，保存$open数据集合中的数据个数  
Local $max = J_EXP_MaxCount($open, "\d+")  
;创建一个循环，从1开始遍历这个集合  
For $i = 1 To $max Step 1  
;声明一个名为$sub_string的局域变量，保存逐个取出集合open中的数据  
Local $sub_string = J_EXP($open, "\d+", $i)  
;通过系统提示框的形式显示出$sub_string里装载的数据  
J_MsgBox($sub_string, $i)  
Next  
;方法2，优点，简单，一般用于单条提取  
;声明一个名为$sub_string2的局域变量，保存文本$string中满足正则表达式\d+的第6个数据  
Local $sub_string2 = J_EXP($string, "\d+", 6)  
J_MsgBox($sub_string2)  
;用于判断是否满足正则表达式,第三个参数（位置）必须为0  
Local $result = J_EXP($string, "\w+\d+", 0)  
If $result = 1 Then  
J_MsgBox("满足正则表达式\w+\d+")  
Else  
J_MsgBox("不满足正则表达式\w+\d+")  
EndIf  
EndFunc ;==>Example35
```

J_EXP_MaxCount

返回满足此正则表达式匹配的个数。

函数

J_EXP_MaxCount(总字符串,正则表达式)

返回值

返回满足此正则表达式匹配的个数。

样例

```
Func Example35()  
;声明一个名为$string的局域变量，保存文本 "hello1234hi4567good7890"  
Local $string = "hello1234hi4567good7890"  
;方法1,优点，速度快，一般用于批量提取  
;声明一个名为$open的局域变量，保存从文本$string中提取满足正则表达式的数据集合  
Local $open = J_EXP_Open($string, "\d+")  
;声明一个名为$max的局域变量，保存$open数据集合中的数据个数  
Local $max = J_EXP_MaxCount($open, "\d+")  
;创建一个循环，从1开始遍历这个集合  
For $i = 1 To $max Step 1  
;声明一个名为$sub_string的局域变量，保存逐个取出集合open中的数据  
Local $sub_string = J_EXP($open, "\d+", $i)  
;通过系统提示框的形式显示出$sub_string里装载的数据  
J_MsgBox($sub_string, $i)  
Next  
;方法2，优点，简单，一般用于单条提取  
;声明一个名为$sub_string2的局域变量，保存文本$string中满足正则表达式\d+的第6个数据  
Local $sub_string2 = J_EXP($string, "\d+", 6)  
J_MsgBox($sub_string2)  
;用于判断是否满足正则表达式,第三个参数（位置）必须为0  
Local $result = J_EXP($string, "\w+\d+", 0)  
If $result = 1 Then  
J_MsgBox("满足正则表达式\w+\d+")  
Else  
J_MsgBox("不满足正则表达式\w+\d+")  
EndIf  
EndFunc ;==>Example35
```

J_Screct

用于加密一段文本。

函数

J_Scret(文本,[模式:加密])

返回值

加密/解密后的文本。

样例

```
Func Example41()  
Local $string = "hello123456"  
Local $scret1 = J_Scret($string, "加密")  
J_MsgBox($scret1)  
Local $scret2 = J_Scret($scret1, "解密")  
J_MsgBox($scret2)  
EndFunc ;=>Example41
```

J_String

对字符串操作函数的汇总。

函数

J_String(总字符串，操作行为，参数值)

操作行为

是否包含长度、取左子串、取右子串、取子串、转小写、转大写、仅数字、仅字母、仅小写、仅大写和拆分。

样例

```
Func StringDemo()  
;1、判断字符串中是否有包含某个子串,如果包含，返回子串包含的位置，否则返回0  
local $str1 = "我是一个字符串"  
if J_String($str1,'是否包含','字符') > 0 then  
J_MsgBox('包含子串')
```

```
endif

;2、取字符串长度
local $len = J_String($str1,'长度')
J_MsgBox("长度:"&$len)

;3、从左边取子串:从左边第一个字符开始，参数值为要取的字符串长度
local $subStr1 = J_String($str1,'取左子串','2')
J_MsgBox($subStr1);我是

;4、从右边取子串：从最后一个字符开始，参数值为要取的字符串长度
local $subStr2 = J_String($str1,'取右子串','3')
J_MsgBox($subStr2);

;5、取子串:参数值需要有提供2个数据，以英文逗号割开：开始位置,截取子串长度
;将其中的 “一个” 提取出来,从第3个字符开始取，取2个字符
local $subStr3 = J_String($str1,'取子串','3,2')
J_MsgBox($subStr3)

;6、转化
local $str2 = "Hello"
;转成小写
local $subStr4 = J_String($str2,'转小写','')
J_MsgBox($subStr4);hello
;转成大写
local $subStr5 = J_String($str2,'转大写','')
J_MsgBox($subStr5);HELLO

;7、判断
;是否纯数字
local $str3="1234"
if J_String($str3,'仅数字','') = 1 then
J_MsgBox("字符串是纯数字")
endif

;是否纯平字母a到z,A到Z
local $str4="abcADK"
if J_String($str4,'仅字母') = 1 then
J_MsgBox("字符串是纯字母")
endif

;字母串中是否仅包括：小写字母 a到z
local $str6='abcd'
if J_String($str6,'仅小写') = 1 then
J_MsgBox("字符串中全是小写字母")
endif

;字母串中是否仅包括：大写字母 A到Z
local $str6='CDFEF'
if J_String($str6,'仅大写') = 1 then
J_MsgBox("字符串中全是大写字母")
endif

;8、拆分：返回值是数组，数组第一个元素值为子串的数量
local $str7 = "123,456,788"
local $array = J_String($str7,'拆分',';')
```

```
;拆分成几个串了  
local $len = $array[0]  
;其它元素  
for $i = 1 to UBound($array) -1  
J_LOG('元素 : '&$array[$i])  
next  
EndFunc ;==>
```

J_StringReplace

将一个字符串中的一段文本替换成其他文本。

函数

J_StringReplace(总字符串,目标字符串,替换成的字符串)

返回值

换后的字符串。

样例

```
Func Example34()  
Local $text = "1122333"  
;将文本$text中的文本22替换成本8888  
Local $new_text = J_StringReplace($text, "22", "8888")  
J_MsgBox($new_text)  
EndFunc ;==>Example34
```

视图功能函数

J_LOG

在码栈执行日志列表中打印出调试信息。

函数

J_LOG (日志信息)

返回值

参数	说明
1	打印成功

样例

```
Func Example18()  
;在当前“我的应用”中日志列表中追加并显示一条日志"你好，世界"  
J_LOG("你好，世界")  
EndFunc ;==>Example18
```

J_Panel_Read

获得码栈参数列表中的某一行值或指定某个参数名的值。

函数

J_Panel_Read (第几个/参数名)

返回值

返回相应参数值。

样例

```
Func Example20()
```

```
;声明 $result1的一个局域变量，并将当前“我的应用”中参数列表中第2个的数据保存至整个变量中
Local $result = J_Panel_Read(2)

;将当前“我的应用”参数列表中参数名“旺旺发送内容”的数据保存到$result2变量中
local $result2 = J_Panel_Read("旺旺发送内容")
EndFunc ;==>Example20
```

J_Panel_Write

设置参数列表第几个或指定参数名的参数值。

函数

J_Panel_Write (第几个/参数名,新值)

返回值

参数	说明
1	设置成功

样例

```
Func Example19()
;在当前“我的应用”中参数列表中第2个参数设置文本"hello",此功能一般用于用户交互，可使用J_Panel_Read读取
J_Panel_Write(2, "hello")

;在当前“我的应用”中参数列表中参数“姓名”设置文本"叶子",此功能一般用于用户交互，可使用
J_Panel_Write("姓名","叶子")
EndFunc ;==>Example19
```

J_Record

用于累加记录用户成功执行的自动化数据，每执行成功一次调用一次这个函数即可。

函数

J_Record()

返回值

无

样例

```
Func Example42()  
;获取当前“我的应用”中视图数据的行数  
Local $max = J_View_MaxCount()  
;创建一个循环用来变遍历每一行数据  
For $i = 1 To $max  
;if语句的特殊用法，当if语句中只有一个执行的表达式，那么if语句可以和判断之后的内容写在一行  
If J_View_GetRowChecked($i) = 0 Then ContinueLoop  
;用来记录用，注意，提交的发布的工具必须要使用此功能  
;调用一次J_Record那么代表此工具执行了一条数据，此功能非常重要！！！！  
J_Record()  
Next  
EndFunc ;==>Example42
```

J_View_Clear

清空浏览器视图数据。

函数

J_View_Clear()

返回值

参数	说明
1	清空成功

样例

```
func test()  
J_View_clear()  
endfunc; ==>test
```

J_View_CreateRow

在数据列表中创建一条新数据。

函数

J_View_CreateRow (新值)

返回值

返回值为新创建数据的行数。

样例

```
Func Example12()  
  
;声明 $index的一个局域变量，在当前“我的应用”中数据列表中最后追加一条数据并在这条数据的第1列设置文本“你好，世界”，并将新追加的数据的行号保存在这个变量中  
Local $index = J_View_CreateRow("你好，世界")  
  
EndFunc ;==>Example12
```

J_View_DeleteRow

删除一条数据列表中的数据。

函数

J_View_DeleteRow (行号)

返回值

参数	说明
1	删除成功

样例

```
Func Example()  
  
;从视图中删除一行数据,参数是行数,从1开始  
Local $index = J_View_DeleteRow(1);  
  
EndFunc ;==>Example
```

J_View_GetRowChecked

获取数据列表中某行数据是否为勾选状态。

函数

J_View_GetRowChecked (行号)

返回值

参数	说明
True	在码栈中True=1
False	在码栈中False=0

样例

```
Func Example15()
```

```
;声明 $result的一个局域变量，获取当前“我的应用”中数据列表中第20行数据的勾选状态，并将结果保存在这个变量中（勾选状态值为1，未勾选状态值为0）
```

```
Local $result = J_View_GetRowChecked(20)
```

```
EndFunc ;==>Example15
```

J_View_MaxCount

查看数据列表中一共有多少条数据，一般情况下用于循环。

函数

J_View_MaxCount ()

返回值

返回码栈视图中一共多少行（数字）。

样例

```
Func Example17()
```

```
;声明 $result1的一个局域变量，并将当前“我的应用”中数据列表中全部数据的条数保存在这个变量中  
Local $max = J_View_MaxCount()
```

```
;意图把视图中的数据从第一条开始全部遍历一遍
```

```
;从1循环到$max中保存的总数据条数，步长为1，即1,2,3,4...
```

```
For $i = 1 To $max Step 1
```

```
;信息提示窗，显示变量 $i的数据
```

```
J_MsgBox($i)
```

```
;下一次循环
```

```
Next
```

```
EndFunc ;==>Example17
```

J_View_Read

获取数据列表中的数据。

函数

J_View_Read (行号,列号/列名称)

返回值

返回指定列表中的数据。

样例

```
Func Example11()

;声明 $result的一个局域变量，读取当前“我的应用”中数据列表的第1行第1列单元格中的数据，并将读取的数据保存在这个变量中
Local $result = J_View_Read(1, 1)

;读取当前“我的应用”中数据列表的第1行列名为“部门”单元格中的数据，并将读取的数据保存在这个变量中
local $result2 = J_View_Read(1, "部门")

EndFunc ;==>Example11
```

J_View_SetRowChecked

设置码栈数据列表中的某行勾选状态为勾选。

函数

J_View_SetRowChecked (行号,[勾选状态:是/否])

返回值

参数	说明
1	设置成功
0	设置失败

样例

```
Func Example16()

;声明 $result1的一个局域变量，设置当前 “我的应用” 中数据列表中第5行数据的勾选状态为勾选状态，并将结果保存在这个变量中（结果一定是1）
Local $result1 = J_View_SetRowChecked(5, "是")

;声明 $result1的一个局域变量，设置当前 “我的应用” 中数据列表中第6行数据的勾选状态为未勾选状态，并将结果保存在这个变量中（结果一定是1）
Local $result2 = J_View_SetRowChecked(6, "否")

EndFunc ;==>Example16
```

J_View_Show

让数据列表中的某行数据显示出来。

函数

J_View_Show (行号)

返回值

参数	说明
1	返回成功

样例

```
Func Example()

;声明 $index的一个局域变量，将当前 “我的应用” 中数据列表中第20行数据置为可见状态，并将结果保存在这个变量中（结果一定是1）
Local $index = J_View_Show(20)

EndFunc ;==>Example
```

J_View_Write

设置数据列表中的数据。

函数

J_View_Write (行号,列号/列名称,设置值)

返回值

参数	说明
1	设置成功
0	设置失败

样例

```
Func Example10()  
  
;声明 $result的一个局域变量，将当前“我的应用”中数据列表的第1行第1列单元格设置文本“hello”，并将插入数据的结果  
保存在这个变量中  
Local $result = J_View_Write(1, 1, "Hello")  
  
;将当前“我的应用”中数据列表的第1行列名为“部门”单元格设置文本“hello”，并将插入数据的结果保存在这个变量中  
Local $result2 = J_View_Write(1, "部门", "Hello")  
  
EndFunc ;==>Example10
```

Word相关

J_Word_Close

关闭WordDoc文档。

函数

J_Word_Close(Word对象)

返回值

成功返回1，失败返回0。

样例

```
Func Example39()  
;前台创建一个word文档，可见  
Local $doc1 = J_Word_Create("是")  
;前台打开D:\1.docx word文档，可见  
Local $doc2 = J_Word_Open("D:\1.docx", "是")  
;光标向下移动4行  
J_Word_CursorMove($doc2, 4, "下")  
;光标向右移动3个字符  
J_Word_CursorMove($doc2, 3, "右")  
;插入文本 你好啊~，字号12，宋体，颜色位置是3，加粗  
J_Word_Write($doc2, "你好啊~", 12, "宋体", 3, "是")  
;从当前位置向后查找第二个"你好啊2"文本，并把光标放在你好啊2前面  
J_Word_SeatchText($doc2, "你好啊2", "当前", 2, "前")  
;插入文本，“在这里”，字号12，字体微软雅黑，颜色位置1，不加粗  
J_Word_Write($doc2, "在这里", 12, "微软雅黑", 1, "否")  
;保存doc2  
J_Word_Save($doc2)  
;读取doc2全部文本内容  
Local $allstring = J_Word_Read($doc2)  
;把读取的内容写入到doc1里面  
J_Word_Write($doc1, $allstring)  
;将doc1文档保存为D:\newdoc.docx  
J_Word_Save($doc1, "D:\newdoc.docx")  
;关闭文档1  
J_Word_Close($doc1)  
;关闭文档2  
J_Word_Close($doc2)  
;结束word操作  
J_Word_Quit()  
;将D:\newdoc.docx转换成D:\1.pdf，注意此函数非word操作，而是文件格式的转变  
J_WordToPDF("D:\newdoc.docx", "D:\1.pdf")  
EndFunc ;==>Example39
```

J_Word_Create

创建一个新的Word文档，并返回word对象。

函数

J_Word_Create([是否可见:否])

返回值

Word对象。

样例

```
Func Example39()  
;前台创建一个word文档，可见  
Local $doc1 = J_Word_Create("是")  
;前台打开D:\1.docx word文档，可见  
Local $doc2 = J_Word_Open("D:\1.docx", "是")  
;光标向下移动4行  
J_Word_CursorMove($doc2, 4, "下")  
;光标向右移动3个字符  
J_Word_CursorMove($doc2, 3, "右")  
;插入文本 你好啊~，字号12，宋体，颜色位置是3，加粗  
J_Word_Write($doc2, "你好啊~", 12, "宋体", 3, "是")  
;从当前位置向后查找第二个"你好啊2"文本，并把光标放在你好啊2前面  
J_Word_SeatchText($doc2, "你好啊2", "当前", 2, "前")  
;插入文本，“在这里”，字号12，字体微软雅黑，颜色位置1，不加粗  
J_Word_Write($doc2, "在这里", 12, "微软雅黑", 1, "否")  
;保存doc2  
J_Word_Save($doc2)  
;读取doc2全部文本内容  
Local $allstring = J_Word_Read($doc2)  
;把读取的内容写入到doc1里面  
J_Word_Write($doc1, $allstring)  
;将doc1文档保存为D:\newdoc.docx  
J_Word_Save($doc1, "D:\newdoc.docx")  
;关闭文档1  
J_Word_Close($doc1)  
;关闭文档2  
J_Word_Close($doc2)  
;结束word操作  
J_Word_Quit()  
;将D:\newdoc.docx转换成D:\1.pdf，注意此函数非word操作，而是文件格式的转变  
J_WordToPDF("D:\newdoc.docx", "D:\1.pdf")  
EndFunc ;==>Example39
```

J_Word_CursorMove

移动Word文档中的光标，移动方向:上/下/左/右。

函数

J_Word_CursorMove(Word对象, 移动步数,[移动方向:右])

返回值

成功返回1，失败返回0。

样例

```
Func Example39()  
;前台创建一个word文档，可见  
Local $doc1 = J_Word_Create("是")  
;前台打开D:\1.docx word文档，可见  
Local $doc2 = J_Word_Open("D:\1.docx", "是")  
;光标向下移动4行  
J_Word_CursorMove($doc2, 4, "下")  
;光标向右移动3个字符  
J_Word_CursorMove($doc2, 3, "右")  
;插入文本 你好啊~，字号12，宋体，颜色位置是3，加粗  
J_Word_Write($doc2, "你好啊~", 12, "宋体", 3, "是")  
;从当前位置向后查找第二个"你好啊2"文本，并把光标放在你好啊2前面  
J_Word_SeatchText($doc2, "你好啊2", "当前", 2, "前")  
;插入文本，“在这里”，字号12，字体微软雅黑，颜色位置1，不加粗  
J_Word_Write($doc2, "在这里", 12, "微软雅黑", 1, "否")  
;保存doc2  
J_Word_Save($doc2)  
;读取doc2全部文本内容  
Local $allstring = J_Word_Read($doc2)  
;把读取的内容写入到doc1里面  
J_Word_Write($doc1, $allstring)  
;将doc1文档保存为D:\newdoc.docx  
J_Word_Save($doc1, "D:\newdoc.docx")  
;关闭文档1  
J_Word_Close($doc1)  
;关闭文档2  
J_Word_Close($doc2)  
;结束word操作  
J_Word_Quit()  
;将D:\newdoc.docx转换成D:\1.pdf，注意此函数非word操作，而是文件格式的转变  
J_WordToPDF("D:\newdoc.docx", "D:\1.pdf")  
EndFunc ;==>Example39
```


J_Word_Open

打开指定的word文档，并返回word对象。

函数

J_Word_Open(文件路径, [是否可见:否])

返回值

Word对象。

样例

```
Func Example39()
;前台创建一个word文档，可见
Local $doc1 = J_Word_Create("是")
;前台打开D:\1.docx word文档，可见
Local $doc2 = J_Word_Open("D:\1.docx", "是")
;光标向下移动4行
J_Word_CursorMove($doc2, 4, "下")
;光标向右移动3个字符
J_Word_CursorMove($doc2, 3, "右")
;插入文本 你好啊~，字号12，宋体，颜色位置是3，加粗
J_Word_Write($doc2, "你好啊~", 12, "宋体", 3, "是")
;从当前位置向后查找第二个"你好啊2"文本，并把光标放在你好啊2前面
J_Word_SeatchText($doc2, "你好啊2", "当前", 2, "前")
;插入文本，“在这里”，字号12，字体微软雅黑，颜色位置1，不加粗
J_Word_Write($doc2, "在这里", 12, "微软雅黑", 1, "否")
;保存doc2
J_Word_Save($doc2)
;读取doc2全部文本内容
Local $allstring = J_Word_Read($doc2)
;把读取的内容写入到doc1里面
J_Word_Write($doc1, $allstring)
;将doc1文档保存为D:\newdoc.docx
J_Word_Save($doc1, "D:\newdoc.docx")
;关闭文档1
J_Word_Close($doc1)
;关闭文档2
J_Word_Close($doc2)
;结束word操作
J_Word_Quit()
```

```
;将D:\newdoc.docx转换成D:\1.pdf，注意此函数非word操作，而是文件格式的转变
J_WordToPDF("D:\newdoc.docx", "D:\1.pdf")
EndFunc ;=>Example39
```

J_Word_Quit

结束Word文档操作。

注意：在使用Word函数后，必须调用这个函数。

函数

J_Word_Quit()

返回值

返回1。

样例

```
Func Example39()
;前台创建一个word文档，可见
Local $doc1 = J_Word_Create("是")
;前台打开D:\1.docx word文档，可见
Local $doc2 = J_Word_Open("D:\1.docx", "是")
;光标向下移动4行
J_Word_CursorMove($doc2, 4, "下")
;光标向右移动3个字符
J_Word_CursorMove($doc2, 3, "右")
;插入文本 你好啊~，字号12，宋体，颜色位置是3，加粗
J_Word_Write($doc2, "你好啊~", 12, "宋体", 3, "是")
;从当前位置向后查找第二个"你好啊2"文本，并把光标放在你好啊2前面
J_Word_SeatchText($doc2, "你好啊2", "当前", 2, "前")
;插入文本，“在这里”，字号12，字体微软雅黑，颜色位置1，不加粗
J_Word_Write($doc2, "在这里", 12, "微软雅黑", 1, "否")
;保存doc2
J_Word_Save($doc2)
;读取doc2全部文本内容
Local $allstring = J_Word_Read($doc2)
;把读取的内容写入到doc1里面
J_Word_Write($doc1, $allstring)
;将doc1文档保存为D:\newdoc.docx
```

```
J_Word_Save($doc1, "D:\newdoc.docx")
;关闭文档1
J_Word_Close($doc1)
;关闭文档2
J_Word_Close($doc2)
;结束word操作
J_Word_Quit()
;将D:\newdoc.docx转换成D:\1.pdf，注意此函数非word操作，而是文件格式的转变
J_WordToPDF("D:\newdoc.docx", "D:\1.pdf")
EndFunc ;=>Example39
```

J_Word_Read

读取并返回Word文件中的内容。

函数

J_Word_Read(Word对象/文件路径)

返回值

成功返回word文件中的内容，失败返回0。

样例

```
Func Example39()
;前台创建一个word文档，可见
Local $doc1 = J_Word_Create("是")
;前台打开D:\1.docx word文档，可见
Local $doc2 = J_Word_Open("D:\1.docx", "是")
;光标向下移动4行
J_Word_CursorMove($doc2, 4, "下")
;光标向右移动3个字符
J_Word_CursorMove($doc2, 3, "右")
;插入文本 你好啊~，字号12，宋体，颜色位置是3，加粗
J_Word_Write($doc2, "你好啊~", 12, "宋体", 3, "是")
;从当前位置向后查找第二个"你好啊2"文本，并把光标放在你好啊2前面
J_Word_SeatchText($doc2, "你好啊2", "当前", 2, "前")
;插入文本，“在这里”，字号12，字体微软雅黑，颜色位置1，不加粗
J_Word_Write($doc2, "在这里", 12, "微软雅黑", 1, "否")
;保存doc2
J_Word_Save($doc2)
;读取doc2全部文本内容
```

```
Local $allstring = J_Word_Read($doc2)
;把读取的内容写入到doc1里面
J_Word_Write($doc1, $allstring)
;将doc1文档保存为D:\newdoc.docx
J_Word_Save($doc1, "D:\newdoc.docx")
;关闭文档1
J_Word_Close($doc1)
;关闭文档2
J_Word_Close($doc2)
;结束word操作
J_Word_Quit()
;将D:\newdoc.docx转换成D:\1.pdf, 注意此函数非word操作, 而是文件格式的转变
J_WordToPDF("D:\newdoc.docx", "D:\1.pdf")
EndFunc ;==>Example39
```

J_Word_Save

保存Word文档，当输入另存名称的时候，对此word文档进行另存操作。

函数

J_Word_Save(Word对象,[文件另存名称])

返回值

成功返回1，失败返回0。

样例

```
Func Example39()
;前台创建一个word文档, 可见
Local $doc1 = J_Word_Create("是")
;前台打开D:\1.docx word文档, 可见
Local $doc2 = J_Word_Open("D:\1.docx", "是")
;光标向下移动4行
J_Word_CursorMove($doc2, 4, "下")
;光标向右移动3个字符
J_Word_CursorMove($doc2, 3, "右")
;插入文本 你好啊~, 字号12, 宋体, 颜色位置是3, 加粗
J_Word_Write($doc2, "你好啊~", 12, "宋体", 3, "是")
;从当前位置向后查找第二个"你好啊2"文本, 并把光标放在你好啊2前面
J_Word_SeatchText($doc2, "你好啊2", "当前", 2, "前")
;插入文本, "在这里", 字号12, 字体微软雅黑, 颜色位置1, 不加粗
```

```
J_Word_Write($doc2, "在这里", 12, "微软雅黑", 1, "否")
;保存doc2
J_Word_Save($doc2)
;读取doc2全部文本内容
Local $allstring = J_Word_Read($doc2)
;把读取的内容写入到doc1里面
J_Word_Write($doc1, $allstring)
;将doc1文档保存为D:\newdoc.docx
J_Word_Save($doc1, "D:\newdoc.docx")
;关闭文档1
J_Word_Close($doc1)
;关闭文档2
J_Word_Close($doc2)
;结束word操作
J_Word_Quit()
;将D:\newdoc.docx转换成D:\1.pdf，注意此函数非word操作，而是文件格式的转变
J_WordToPDF("D:\newdoc.docx", "D:\1.pdf")
EndFunc ;==>Example39
```

J_Word_SeatchText

搜索word文档中的关键词并把光标移动至关键词。

起始位置：开始/当前是否从文档头开始搜索，还是从光标位置进行搜索，默认开始。

第几个符合：搜索词出现的个数，默认第一次出现就匹配。

光标位于关键词位置：后/前 光标处于关键词前还是关键词后。

函数

J_Word_SeatchText(Word对象, 搜索的文本, [起始位置:开始], [第几个符合:1], [光标位于关键词位置:后])

返回值

成功返回1 失败返回0。

样例

```
Func Example39()
```

```
;前台创建一个word文档，可见
Local $doc1 = J_Word_Create("是")
;前台打开D:\1.docx word文档，可见
Local $doc2 = J_Word_Open("D:\1.docx", "是")
;光标向下移动4行
J_Word_CursorMove($doc2, 4, "下")
;光标向右移动3个字符
J_Word_CursorMove($doc2, 3, "右")
;插入文本 你好啊~，字号12，宋体，颜色位置是3，加粗
J_Word_Write($doc2, "你好啊~", 12, "宋体", 3, "是")
;从当前位置向后查找第二个"你好啊2"文本，并把光标放在你好啊2前面
J_Word_SeatchText($doc2, "你好啊2", "当前", 2, "前")
;插入文本，“在这里”，字号12，字体微软雅黑，颜色位置1，不加粗
J_Word_Write($doc2, "在这里", 12, "微软雅黑", 1, "否")
;保存doc2
J_Word_Save($doc2)
;读取doc2全部文本内容
Local $allstring = J_Word_Read($doc2)
;把读取的内容写入到doc1里面
J_Word_Write($doc1, $allstring)
;将doc1文档保存为D:\newdoc.docx
J_Word_Save($doc1, "D:\newdoc.docx")
;关闭文档1
J_Word_Close($doc1)
;关闭文档2
J_Word_Close($doc2)
;结束word操作
J_Word_Quit()
;将D:\newdoc.docx转换成D:\1.pdf，注意此函数非word操作，而是文件格式的转变
J_WordToPDF("D:\newdoc.docx", "D:\1.pdf")
EndFunc ;==>Example39
```

J_Word_Write

在word文档中输入文本。

函数

J_Word_Write(Word对象, 写入的文本, [字体大小:8], [字体:微软雅黑],[颜色位置:1], [是否加粗:否])

返回值

成功返回1，失败返回0。

样例

```
Func Example39()  
;前台创建一个word文档，可见  
Local $doc1 = J_Word_Create("是")  
;前台打开D:\1.docx word文档，可见  
Local $doc2 = J_Word_Open("D:\1.docx", "是")  
;光标向下移动4行  
J_Word_CursorMove($doc2, 4, "下")  
;光标向右移动3个字符  
J_Word_CursorMove($doc2, 3, "右")  
;插入文本 你好啊~，字号12，宋体，颜色位置是3，加粗  
J_Word_Write($doc2, "你好啊~", 12, "宋体", 3, "是")  
;从当前位置向后查找第二个"你好啊2"文本，并把光标放在你好啊2前面  
J_Word_SeatchText($doc2, "你好啊2", "当前", 2, "前")  
;插入文本，“在这里”，字号12，字体微软雅黑，颜色位置1，不加粗  
J_Word_Write($doc2, "在这里", 12, "微软雅黑", 1, "否")  
;保存doc2  
J_Word_Save($doc2)  
;读取doc2全部文本内容  
Local $allstring = J_Word_Read($doc2)  
;把读取的内容写入到doc1里面  
J_Word_Write($doc1, $allstring)  
;将doc1文档保存为D:\newdoc.docx  
J_Word_Save($doc1, "D:\newdoc.docx")  
;关闭文档1  
J_Word_Close($doc1)  
;关闭文档2  
J_Word_Close($doc2)  
;结束word操作  
J_Word_Quit()  
;将D:\newdoc.docx转换成D:\1.pdf，注意此函数非word操作，而是文件格式的转变  
J_WordToPDF("D:\newdoc.docx", "D:\1.pdf")  
EndFunc ;==>Example39
```

其他

J_dialog

js弹出窗口进行操作。

函数

J_dialog([“动作名称”],[按钮名称],[窗口标题])

动作名称：点击/获取文本。

按钮名称：确定(OK)/取消(Cancle)

返回值

0：表示返回失败。

1：点击操作，表示成功。

对获取文本操作：返回值为相应的文本信息。

样例

```
Func Example49()  
local $text = J_dialog("获取文本")  
;获取值为：请用j_dialog(点击)来关闭我  
  
j_dialog("点击");点确定按钮  
EndFunc ;==>Example49
```

J_Download

从URL地址下文件至本机，如路径为空，则返回html代码。

函数

J_Download(URL地址,保存本机的路径,[编码:1])

编码

默认1,仅在保存本机路径为空的时候生效，编码为0的时候返回二进制编码。

返回值

无。

样例

```
Func Example36()  
Local $url = "https://img.alicdn.com/bao/uploaded/i2/TB1nuG0GXXXXa2aXXXXXXXXXX_!!0-  
item_pic.jpg_130x130.jpg"  
;将url地址的图片下载并保存至本地D:\1.jpg  
J_Download($url, "D:\1.jpg")  
EndFunc ;=>Example36
```

J_Email

模拟邮箱发邮件。

函数

J_Email(smtp服务器,邮件发送人,邮件发送者地址,邮件发送给谁,邮件标题,邮件正文,附件地址,CC,BCC,用户名,密码,[发送端口=465],[ssl=993])

返回值

成功返回1，失败返回0。

样例

```
Func Example44()  
;声明一个局域变量，保存邮件发送的结果  
Local $res = J_Email("smtp.taobao.com", "镖头", "biaotou@taobao.com", "gulie@taobao.com", "邮件通知测试", "邮件  
正文支持html代码", "D:1/jpg;D:2.jpg", "", "", "username", "password")  
If $res = 1 Then  
J_MsgBox("发送成功")  
Else  
J_MsgBox("发送失败")  
EndIf
```

```
EndIf  
EndFunc ;==>Example44
```

J_GetErrorMsg

获取异常信息，用于展示执行过程中的异常或失败的原因。。

函数

无。

返回值

无。

样例

```
Func Example49()  
  
local $result= J_dialog("点击")  
;有两种方式，处理失败的情况  
local $errorMsg  
;方式一：  
if @error then  
$errorMsg = J_GetErrorMsg()  
endif  
  
;方式二：  
if $result = 0 then  
$errorMsg = J_GetErrorMsg()  
EndIf  
  
EndFunc ;==>Example49
```

J_IE_FastModel

IE浏览器模式切换：

1:快捷模式。

0:普通模式。

默认为普通模式。

函数

J_IE_FastModel([模式:0])

返回值

无。

样例

```
Func Example45()  
;E浏览器模式切换,1:快捷模式 , 0:普通模式,默认为普通模式,注意快捷模式不显示图片,如果影响用户使用需要将参数设置为0再  
执行一次  
J_IE_FastModel(1)  
Local $ie = J_IE_Create("www.taobao.com")  
Sleep(2000)  
J_IE_Close($ie)  
J_IE_FastModel(0)  
Local $ie2 = J_IE_Create("www.taobao.com")  
Sleep(2000)  
J_IE_Close($ie2)  
EndFunc ;=>Example45
```

J_INPUT

弹出一个输入框接收用户输入的信息。

函数

J_INPUT(标题,内容,[类型：文本/密码],[宽度:默认200],[高度:默认125])

返回值

用户输入的内容。

样例

```
Func Example45()  
;声明一个局域变量，保存输入框返回的数据  
Local $string = J_INPUT("请输入您的公司名称", "在这里输入您公司的名称", "文本", 200, 125)  
EndFunc ;==>Example45
```

Func J_Logistics

根据物流单号获取物流详情。

函数

Func J_Logistics(“物流编号”，“物流服务商代码，可以不传”)

返回值

物流详情的寄送对象。

样例

```
Func Example50()  
;作用1:根据物流单号获取物流详情，返回物流详情的json格式  
  
Local $sel = J_Logistics("310017000205")  
  
;作用2:根据物流单号和服务商代码获取物流详情，返回物流详情的json格式，这比仅仅根据物流单号获取物流详情更准确  
  
Local $sel = J_Logistics("310017000205","HTKY")  
  
EndFunc ;==>Example50  
<item>
```

```
<cpCode>ZTO</cpCode>
<fullName>中通快递</fullName>
</item>
<item>
<cpCode>YTO</cpCode>
<fullName>圆通速递</fullName>
</item>
<item>
<cpCode>STO</cpCode>
<fullName>申通快递</fullName>
</item>
<item>
<cpCode>YUNDA</cpCode>
<fullName>韵达快递</fullName>
</item>
<item>
<cpCode>HTKY</cpCode>
<fullName>百世汇通</fullName>
</item>
<item>
<cpCode>SF</cpCode>
<fullName>顺丰速运</fullName>
</item>
<item>
<cpCode>TTKDEX</cpCode>
<fullName>天天快递</fullName>
</item>
<item>
<cpCode>FAST</cpCode>
<fullName>快捷快递</fullName>
</item>
<item>
<cpCode>QFKD</cpCode>
<fullName>全峰快递</fullName>
</item>
<item>
<cpCode>GTO</cpCode>
<fullName>国通快递</fullName>
</item>
<item>
<cpCode>ZJS</cpCode>
<fullName>宅急送</fullName>
</item>
<item>
<cpCode>UC</cpCode>
<fullName>优速物流</fullName>
</item>
<item>
<cpCode>DBKD</cpCode>
<fullName>德邦快递</fullName>
</item>
<item>
<cpCode>EMS</cpCode>
<fullName>EMS</fullName>
</item>
<item>
```

```
<cpCode>EMS</cpCode>
<fullName>中速快递</fullName>
</item>
<item>
<cpCode>CNPOSTGJ</cpCode>
<fullName>邮政国际包裹</fullName>
</item>
<item>
<cpCode>POSTBBZ</cpCode>
<fullName>邮政标准快递</fullName>
</item>
<item>
<cpCode>POSTB</cpCode>
<fullName>邮政快递包裹</fullName>
</item>
<item>
<cpCode>DHLDE</cpCode>
<fullName>DHL德国</fullName>
</item>
<item>
<cpCode>KERRYEAS</cpCode>
<fullName>嘉里大通</fullName>
</item>
<item>
<cpCode>TMS56</cpCode>
<fullName>加运美</fullName>
</item>
<item>
<cpCode>POSTSE</cpCode>
<fullName>瑞典邮政</fullName>
</item>
<item>
<cpCode>TNT</cpCode>
<fullName>TNT</fullName>
</item>
<item>
<cpCode>UPS</cpCode>
<fullName>UPS</fullName>
</item>
<item>
<cpCode>USPS</cpCode>
<fullName>USPS</fullName>
</item>
<item>
<cpCode>DHLCN</cpCode>
<fullName>DHL中国</fullName>
</item>
<item>
<cpCode>ZTKY</cpCode>
<fullName>中铁物流</fullName>
</item>
<item>
<cpCode>ZTKY</cpCode>
<fullName>飞豹快递</fullName>
</item>
<item>
```

```
<cpCode>DHL</cpCode>
<fullName>DHL全球</fullName>
</item>
<item>
<cpCode>FEDEX</cpCode>
<fullName>FedEx国际</fullName>
</item>
<item>
<cpCode>FEDEXUS</cpCode>
<fullName>FedEx美国</fullName>
</item>
<item>
<cpCode>FEDEXCN</cpCode>
<fullName>FedEx中国</fullName>
</item>
<item>
<cpCode>POSTB</cpCode>
<fullName>包裹信件</fullName>
</item>
<item>
<cpCode>POSTTH</cpCode>
<fullName>泰国邮政</fullName>
</item>
<item>
<cpCode>XFWL</cpCode>
<fullName>信丰物流</fullName>
</item>
<item>
<cpCode>EWINSHINE</cpCode>
<fullName>万象物流</fullName>
</item>
<item>
<cpCode>RFD</cpCode>
<fullName>如风达</fullName>
</item>
<item>
<cpCode>ANE56</cpCode>
<fullName>安能物流</fullName>
</item>
<item>
<cpCode>HOAU</cpCode>
<fullName>天地华宇</fullName>
</item>
<item>
<cpCode>CNEX</cpCode>
<fullName>佳吉快运</fullName>
</item>
<item>
<cpCode>SURE</cpCode>
]]>
<fullName>速尔快递</fullName>
</item>
<item>
<cpCode>UAPEX</cpCode>
<fullName>全一快递</fullName>
</item>
```

```
<item>
<cpCode>QRT</cpCode>
<fullName>全日通</fullName>
</item>
<item>
<cpCode>ZENY</cpCode>
]]>
<fullName>增益速递</fullName>
</item>
<item>
<cpCode>GZFY</cpCode>
]]>
<fullName>凡宇快递</fullName>
</item>
<item>
<cpCode>XBWL</cpCode>
]]>
<fullName>新邦物流</fullName>
</item>
<item>
<cpCode>CAE</cpCode>
<fullName>民航快递</fullName>
</item>
<item>
<cpCode>CRE</cpCode>
<fullName>中铁快运</fullName>
</item>
<item>
<cpCode>FEC</cpCode>
<fullName>银捷速递</fullName>
</item>
<item>
<cpCode>FJSFWLJTYXGS</cpCode>
<fullName>盛丰物流</fullName>
</item>
<item>
<cpCode>HLWL</cpCode>
<fullName>恒路物流</fullName>
</item>
<item>
<cpCode>JDKD</cpCode>
<fullName>京东快递</fullName>
</item>
<item>
<cpCode>JIAYI</cpCode>
<fullName>佳怡物流</fullName>
</item>
<item>
<cpCode>KYE</cpCode>
<fullName>跨越速运</fullName>
</item>
<item>
<cpCode>LTS</cpCode>
<fullName>联昊通</fullName>
</item>
<item>
```



```
<cpCode>SZSA56</cpCode>
<fullName>圣安物流</fullName>
</item>
<item>
<cpCode>SHENGHUI</cpCode>
<fullName>盛辉物流</fullName>
</item>
<item>
<cpCode>SZKKE</cpCode>
<fullName>京广速递</fullName>
</item>
<item>
<cpCode>YCGWL</cpCode>
<fullName>远成物流</fullName>
</item>
<item>
<cpCode>YTZG</cpCode>
<fullName>运通中港快递</fullName>
</item>
<item>
<cpCode>POSTBR</cpCode>
<fullName>巴西邮政</fullName>
</item>
<item>
<cpCode>POSTCH</cpCode>
<fullName>瑞士邮政</fullName>
</item>
<item>
<cpCode>BLSYZ</cpCode>
<fullName>比利时邮政</fullName>
</item>
<item>
<cpCode>RBYZEMS</cpCode>
<fullName>日本邮政</fullName>
</item>
<item>
<cpCode>EPOST</cpCode>
<fullName>韩国邮政</fullName>
</item>
<item>
<cpCode>EYB</cpCode>
<fullName>EMS经济快递</fullName>
</item>
<item>
<cpCode>BESTQJT</cpCode>
<fullName>百世物流</fullName>
</item>
<item>
<cpCode>HUANGMAJIA</cpCode>
<fullName>黄马甲配送</fullName>
</item>
<item>
<cpCode>CHENGBANG</cpCode>
<fullName>晟邦物流</fullName>
</item>
<item>
```

```
<cpCode>GZLT</cpCode>
<fullName>飞远配送</fullName>
</item>
<item>
<cpCode>HZABC</cpCode>
<fullName>飞远(爱彼西)配送</fullName>
</item>
<item>
<cpCode>SNWL</cpCode>
<fullName>苏宁快递</fullName>
</item>
<item>
<cpCode>ZMKM</cpCode>
<fullName>芝麻开门</fullName>
</item>
<item>
<cpCode>JDYWL</cpCode>
<fullName>筋斗云物流</fullName>
</item>
<item>
<cpCode>BJCS</cpCode>
<fullName>城市一百物流</fullName>
</item>
<item>
<cpCode>LB</cpCode>
<fullName>龙邦速递</fullName>
</item>
<item>
<cpCode>YCT</cpCode>
<fullName>黑猫宅急便</fullName>
</item>
<item>
<cpCode>ROYALMAIL</cpCode>
<fullName>英国皇家邮政</fullName>
</item>
<item>
<cpCode>JKYZ</cpCode>
<fullName>捷克邮政</fullName>
</item>
<item>
<cpCode>BLYZ</cpCode>
<fullName>波兰邮政</fullName>
</item>
<item>
<cpCode>FGYZ</cpCode>
<fullName>法国邮政</fullName>
</item>
<item>
<cpCode>POSTTR</cpCode>
<fullName>土耳其邮政</fullName>
</item>
<item>
<cpCode>POSTCL</cpCode>
<fullName>智利邮政</fullName>
</item>
<item>
```

```
<cpCode>POSTBY</cpCode>
<fullName>白俄罗斯邮政</fullName>
</item>
<item>
<cpCode>POSTES</cpCode>
<fullName>西班牙邮政</fullName>
</item>
<item>
<cpCode>POSTUA</cpCode>
<fullName>乌克兰邮政</fullName>
</item>
<item>
<cpCode>POSTNO</cpCode>
<fullName>挪威邮政</fullName>
</item>
<item>
<cpCode>POSTZA</cpCode>
<fullName>南非邮政</fullName>
</item>
<item>
<cpCode>POSTPT</cpCode>
<fullName>葡萄牙邮政</fullName>
</item>
<item>
<cpCode>POSTSA</cpCode>
<fullName>沙特邮政</fullName>
</item>
<item>
<cpCode>POSTIN</cpCode>
<fullName>印度邮政</fullName>
</item>
<item>
<cpCode>POSTBG</cpCode>
<fullName>保加利亚邮政</fullName>
</item>
<item>
<cpCode>POSTAE</cpCode>
<fullName>阿联酋邮政</fullName>
</item>
<item>
<cpCode>POSTAU</cpCode>
<fullName>澳大利亚邮政</fullName>
</item>
<item>
<cpCode>POSTPK</cpCode>
<fullName>巴基斯坦邮政</fullName>
</item>
<item>
<cpCode>POSTMT</cpCode>
<fullName>马耳他邮政</fullName>
</item>
<item>
<cpCode>POSTLB</cpCode>
<fullName>黎巴嫩邮政</fullName>
</item>
<item>
```

```
<cpCode>POSTMD</cpCode>
<fullName>摩尔多瓦邮政</fullName>
</item>
<item>
<cpCode>POSTSRB</cpCode>
<fullName>塞尔维亚邮政</fullName>
</item>
<item>
<cpCode>POSTHR</cpCode>
<fullName>克罗地亚邮政</fullName>
</item>
<item>
<cpCode>POSTAM</cpCode>
<fullName>亚美尼亚邮政</fullName>
</item>
<item>
<cpCode>POSTMK</cpCode>
<fullName>马其顿邮政</fullName>
</item>
<item>
<cpCode>ONTRAC</cpCode>
<fullName>OnTrac</fullName>
</item>
<item>
<cpCode>CITYLINK</cpCode>
<fullName>City-Link</fullName>
</item>
<item>
<cpCode>COE</cpCode>
<fullName>COE</fullName>
</item>
<item>
<cpCode>DTW</cpCode>
<fullName>大田物流</fullName>
</item>
<item>
<cpCode>EES</cpCode>
<fullName>百福东方</fullName>
</item>
<item>
<cpCode>MBEX</cpCode>
<fullName>民邦快递</fullName>
</item>
<item>
<cpCode>HQKY</cpCode>
<fullName>华企快运</fullName>
</item>
<item>
<cpCode>YFEXPRESS</cpCode>
<fullName>越丰物流</fullName>
</item>
<item>
<cpCode>AIR</cpCode>
<fullName>亚风速递</fullName>
</item>
<item>
```

```
<cpCode>MANCOWL</cpCode>
<fullName>万家物流</fullName>
</item>
<item>
<cpCode>SZML56</cpCode>
<fullName>明亮物流</fullName>
</item>
<item>
<cpCode>CXCOD</cpCode>
<fullName>传喜物流</fullName>
</item>
<item>
<cpCode>4PX</cpCode>
<fullName>递四方</fullName>
</item>
<item>
<cpCode>POSTFI</cpCode>
<fullName>芬兰邮政</fullName>
</item>
<item>
<cpCode>POSTAR</cpCode>
<fullName>阿根廷邮政</fullName>
</item>
<item>
<cpCode>POSTSK</cpCode>
<fullName>斯洛伐克邮政</fullName>
</item>
<item>
<cpCode>SERPOST</cpCode>
<fullName>秘鲁邮政</fullName>
</item>
<item>
<cpCode>POSTIT</cpCode>
<fullName>意大利邮政</fullName>
</item>
<item>
<cpCode>POSTSI</cpCode>
<fullName>斯洛文尼亚邮政</fullName>
</item>
<item>
<cpCode>POSTHU</cpCode>
<fullName>匈牙利邮政</fullName>
</item>
<item>
<cpCode>POSTMU</cpCode>
<fullName>毛里求斯邮政</fullName>
</item>
<item>
<cpCode>POSTAT</cpCode>
<fullName>奥地利邮政</fullName>
</item>
<item>
<cpCode>POSTAL</cpCode>
<fullName>阿尔巴尼亚邮政</fullName>
</item>
<item>
```

```
<cpCode>POSTEE</cpCode>
<fullName>爱沙尼亚邮政</fullName>
</item>
<item>
<cpCode>ARAMEX</cpCode>
<fullName>Aramex</fullName>
</item>
<item>
<cpCode>SANTAI</cpCode>
<fullName>三态速递</fullName>
</item>
<item>
<cpCode>OCS</cpCode>
<fullName>OCS</fullName>
</item>
<item>
<cpCode>MYAAE</cpCode>
<fullName>AAE全球专递</fullName>
</item>
<item>
<cpCode>POSTLV</cpCode>
<fullName>拉脱维亚邮政</fullName>
</item>
<item>
<cpCode>YWWL</cpCode>
<fullName>燕文物流</fullName>
</item>
<item>
<cpCode>ASENDIA</cpCode>
<fullName>Asendia USA</fullName>
</item>
<item>
<cpCode>RUSTON</cpCode>
<fullName>俄速通</fullName>
</item>
<item>
<cpCode>XLOBO</cpCode>
<fullName>贝海国际速递</fullName>
</item>
<item>
<cpCode>SPSR</cpCode>
<fullName>中俄快递</fullName>
</item>
<item>
<cpCode>POSTCO</cpCode>
<fullName>哥伦比亚邮政</fullName>
</item>
<item>
<cpCode>SAD</cpCode>
<fullName>赛澳递</fullName>
</item>
<item>
<cpCode>SUJIAWL</cpCode>
<fullName>穗佳物流</fullName>
</item>
```

J_MsgBox

显示一个消息窗口。

按钮个数默认为1个，可以修改成2。

函数

J_MsgBox([“窗口文本”],[“标题”],[是否顶层:是],[按钮个数:1])

返回值

单击**确定**，返回值为1。

单击**取消**，返回值为2

样例

```
Func Example49()  
;作用1:显示一个消息窗口。按钮个数默认为1个，可以修改成2，当点击确定之后返回值为1,点击取消返回值为2  
J_MsgBox("用来信息提示")  
J_MsgBox("带标题的提示", "标题")  
J_MsgBox("带标题的底层的提示", "标题", "否")  
Local $sel = J_MsgBox("确认或者取消", "标题", "是", 2)  
If $sel = 0 Then  
J_MsgBox("你啥按钮也没点")  
EndIf  
If $sel = 1 Then  
J_MsgBox("你点了确定按钮")  
EndIf  
If $sel = 2 Then  
J_MsgBox("你点了取消按钮")  
EndIf  
EndFunc ;==>Example49
```

J_Rest_Get

rest接口get方法。

函数

J_Rest_Get(\$url,\$header=' ' ,浏览器对象=' ')

若指定浏览器对象，调用rest接口时会加上该浏览器的session。

其中\$header,键值对，多个值以换行符连接。

返回值

成功返回调用结果，失败返回0。

样例

```
func test()
local $url =
"http://api.daily.taobao.net/router/rest?sign=45E8A4332BEC46289386F25954DB2706&timestamp=2016-10-28+19%3A34%3A49&v=2.0&app_key=4272&method=taobao.juwuba.user.getrsa&partner_id=top-apitools&format=json"
local $result = J_Rest_Get($url)
if $result=0 then
local $msg = J_GetErrorMsg()
j_msgbox($msg)
endif
j_log($result)
endfunc ;==>test

func test1()
local $ie = J_IE_Catch("码栈", "标题", "chrome")
local $url = "http://codestore.daily.taobao.net/services/cs/task/sdks/744/depend"
local $result = J_Rest_Get($url, "", $ie)
if $result = 0 then
local $msg = J_GetErrorMsg()
j_msgbox($msg)
endif
j_log($result)
EndFunc ;==>test1
```

J_Rest_Post

rest接口post方法。

函数

J_Rest_Post(\$url,\$header=' ',\$data=' ',浏览器对象=' ')

若指定浏览器对象，调用rest接口时会加上该浏览器的session。

其中\$header/Data为键值对，多个值以换行符连接。

返回值

成功返回调用结果，失败返回0。

样例

```
func test()
;调用登录接口，获取session
local $url = "http://codestore.daily.taobao.net/services/cs/user/login/buclogin"
local $data = "userName=XXXX" & @CRLF
$data &= "password=YYYY" & @CRLF
$data &= "ip=127.0.0.1" & @CRLF
$data &= "publicKey=XX" & @CRLF
$data &= "versionNumber=XXX" & @CRLF
local $result = J_Rest_Post($url, "", $data)
if $result = 0 then
local $msg = J_GetErrorMsg()
j_msgbox($msg)
endif

;调用请求接口
local $url2 = "http://codestore.daily.taobao.net/services/cs/task/sdks/744/depend"
local $result2 = J_Rest_Get($url2, "", "")
j_log($result2)
EndFunc ;=>test3
```

J_Say

说出文本，模式:ch/en，速度:1-10,音量:0-100。

函数

J_Say(内容,[模式:ch],[速度:1],[音量:100])

返回值

无。

样例

```
Func Example37()  
;使用此功能需要电脑装有语音插件  
J_Say("你好，世界")  
EndFunc ;==>Example37
```

J_Select

提供用户备选选项选择框,多个备选值用“ | ”分隔，能否修改，是否支持用户键盘输入，默认不支持0,1支持。

函数

J_Select(标题,提示信息,默认值,备选值,[能否修改:默认:0/1],[X坐标:中间],[Y坐标:中间])

返回值

无。

样例

```
Func Example47()  
Local $sel = J_Select("码栈", "请选择您的部门", "聚划算", "淘宝网|聚划算|1688|菜鸟|蚂蚁金服")  
J_MsgBox("您选择了部门:" & $sel)  
EndFunc ;==>Example47
```

J_ShortMsg

根据阿里大鱼的模板发送短信。

函数

J_ShortMsg(“阿里大鱼应用key”，“阿里大鱼应用密钥”，“阿里大鱼的短信模板id”，“手机号”，“阿里大鱼的短信签名”，“短信模板中的参数，如果没有可以为空”)

返回值

是否发送成功。

0表示失败。

1表示成功。

使用方法

阿里大鱼申请阿里大鱼的账户，地址为阿里大鱼账号。

按照阿里云的指引开通云通信服务。

创建阿里云的AccessKey 和Access Secret，地址为AK创建地址

设置短信模板，模板需要申请，审核通过后就可以使用，地址为审核地址。

单击[详情](#)，可以查看模板详情。

设置短信签名和短信模板一样，申请地址为申请地址。

样例

```
Func Example51()
```

```
;作用1:发送短信 $msgParams 是模板中的变量，用JSON 格式的方式传入
```

```
Local $msgParams = '{"code":"122456","product":"阿里码栈"}';  
J_ShortMsg("23330602","765xxxxxxxxxxxxxx40","SMS_5012690","13312341234",,"活动验证" $msgParams )  
  
EndFunc ;==>Example51
```

J_Windows

对窗口一系列操作,模式:激活/获取标题/获取文本/设置标题/隐藏/显示/关闭。

启用/禁用/最小化/最大化/透明/顶层/底层/闪烁/检测/等待出现/等待消失。

当模式为透明的时候,需要设置参数0-255,数字越小透明度越高。

当模式为设置标题的时候,参数表示新的标题名称。

函数

J_Windows(IE对象/窗口标题,[模式:激活],[参数])

返回值

获取标题：返回标题；

获取文本：返回页面全部可见文本；

设置标题/隐藏/显示/关闭/禁用/启用/最小化/最大化/透明/顶层/底层：成功1，失败0。

样例

```
Func Example48()  
J_Windows("淘宝网", "激活")  
Local $title = J_Windows("淘宝网", "获取标题")  
Local $text = J_Windows("淘宝网", "获取文本")  
J_Windows("淘宝网", "设置标题", "我爱淘宝网")
```

```
J_Windows("淘宝网", "隐藏")
J_Windows("淘宝网", "显示")
J_Windows("淘宝网", "关闭")
J_Windows("淘宝网", "禁用")
J_Windows("淘宝网", "启用")
J_Windows("淘宝网", "最小化")
J_Windows("淘宝网", "最大化")
;透明度0-255越大越不透明
J_Windows("淘宝网", "透明", 100)
J_Windows("淘宝网", "顶层")
J_Windows("淘宝网", "底层")
J_Windows("淘宝网", "闪烁")
Local $res = J_Windows("淘宝网", "检测")
If $res = 1 Then
J_MsgBox("检测到了~")
Else
J_MsgBox("没检测到~")
EndIf

Local $res = J_Windows("淘宝网", "等待出现", 3)
If $res = 1 Then
J_MsgBox("3秒内出现了~")
Else
J_MsgBox("3秒内没出现~")
EndIf

Local $res = J_Windows("淘宝网", "等待消失", 8)
If $res = 1 Then
J_MsgBox("8秒内消失了~")
Else
J_MsgBox("8秒内没消失~")
EndIf
EndFunc ;==>Example48
```

可视化开发

浏览器

后退网页

该模块用于将浏览器退回到上一次打开的网页。

在**后退网页**页面，选择之前打开的网页，默认选取最近一次打开的网页，如下图所示。



查看表格格式

查看指定表格对象的内容格式，仅用于测试使用。

选择之前查找到的表格，默认已经选取最近一次获取的表格。

设置复选框

用于设置网页中的复选框。

捕获网页元素：具体操作请参考

设置复选框：选择对复选框进行勾选还是取消勾选的操作。

设置网页元素位置：只有在捕获到多个网页元素时使用才有效，如果录制的结果只有一项，默认设置为1即可。

如果录制一个列表的时候会录制到多个列表项，只获取第二个列表项，那么网页元素位置就设置为2。

选择已打开网页：选择之前打开的网页，默认已经选取最近一次打开的网页

选择下拉列表

用于选择网页中的下拉列表

捕获网页元素：具体操作请参考

选择匹配方式，包含以下选项：

完全匹配: 选择与匹配内容完全一致的匹配项。

模糊匹配: 选择包含匹配内容的匹配项。

正则表达式: 选择与匹配内容中正则表达式匹配的匹配项。

匹配内容：填写需要用于匹配的文本内容或正则表达式。

网页元素位置：只有在捕获到多个网页元素时使用才有效，如果录制的结果只有一项，默认设置为1即可。

如果录制一个列表的时候会录制到多个列表项，只获取第二个列表项，那么网页元素位置就设置为2。

选择已打开网页，选择之前打开的网页，默认已经选取最近一次打开的网页。

点击网页元素

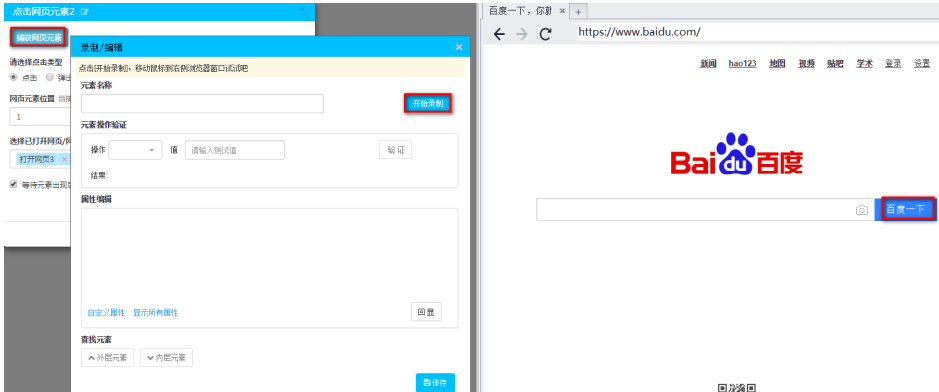
模拟鼠标点击的操作，可以用于点击按钮、下拉框、弹出框等操作。

捕获网页元素。

在**浏览器窗口**打开拥有按钮、下拉框或者弹出框等可以点击按钮的网页。

在**点击网页元素**页面，单击**捕获网页元素**。

系统显示**录制/编辑**页面，如下图所示。



单击**开始录制**，将鼠标移动至浏览器中的输入框，并单击，录制完成。

设置录制元素的控件属性。

元素名称：用于区分控件，每次都会自动生成一个随机的名字，可视化开发用户可以忽略此选项。

操作验证：用于验证本次录制操作是否成功，与应用编写无关，仅作为测试使用，如模拟输入操作，单击**验证**，会在输入框中输入“你好,码栈”。

录制结果：用于拷贝录制操作生成的代码，可视化开发用户可以忽略。

属性编辑：这里是一个属性列表，用来定位元素，有个别元素录制后无法精准定位，可以通过添加、修改此属性来自定义定位方式。

回显：单击**回显**，会在浏览器中高亮已捕获的元素，与操作验证一样是用来测试本次录制操作是否成功。

单击**保存**，返回**点击网页元素**页面。

选择**点击类型**。

点击: 用于点击网页元素。

弹出框点击: 用于点击网页中的弹出框(提示框)。

设置网页元素位置：只有在捕获到多个网页元素时使用才有效，如果录制的结果只有一项，默认设置为1即可。

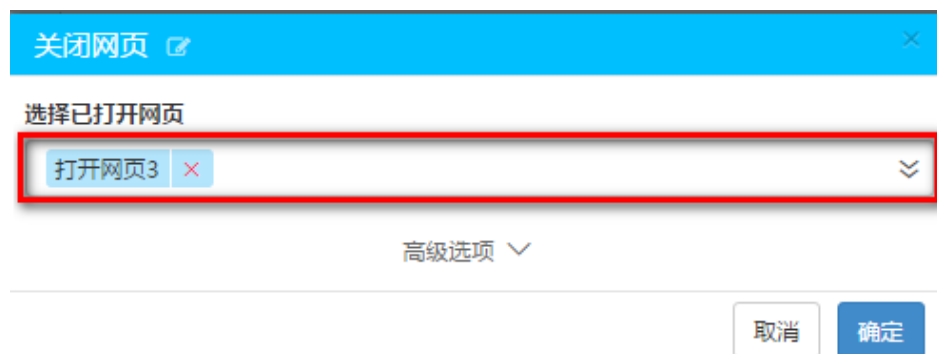
如果录制一个列表的时候会录制到多个列表项，只获取第二个列表项，那么网页元素位置就设置为2。

选择已打开网页点击网页元素，选择之前打开的网页，默认选取最近一次打开的网页。

关闭网页

该模块用于关闭已经打开的网页。

在**关闭网页**页面，从下拉列表中选择已打开的网页，默认选取最近打开的网页，如下图所示。



获取网页信息

该模块用于获取已打开的整个网页的详细信息，如网页的文本内容、源代码、网址(URL)等。

在**获取网页信息**页面，进行一下操作：

选择已打开网页：选择之前打开的网页，默认选取最近一次打开的网页。

提取类型：选择相需要获取的信息类型。

文本: 网页的文本内容，不包括图片、标签等信息。

源代码：网页的源代码。

URL：网页的网址。

查找网页中表格

根据指定的条件，获取网页中的表格对象。

选择查找方式，包括以下选项：

根据位置索引: 指定想要获取的是第几个表格

根据表格内容: 根据提供的内容模糊匹配网页中的表格

根据表格源代码: 根据提供的源代码匹配网页中的表格

查找条件：填写位置索引、表格内容、或源代码。

选择已打开网页：选择之前打开的网页，默认已经选取最近一次打开的网页。

填写输入框内容

模拟键盘输入的操作，可以用于输入框、密码框、富文本编辑区。

在填写输入框内容页面，进行以下配置：

捕获网页元素，具体操作请参见

输入内容，填写需要输入到文本框中的内容。

设置网页元素位置：只有在捕获到多个网页元素时使用才有效，如果录制的结果只有一项，默认设置为1即可。

如果录制一个列表的时候会录制到多个列表项，只获取第二个列表项，那么网页元素位置就设置为2。

选择已打开网页，选择之前打开的网页，默认已经选取最近一次打开的网页。

鼠标移动到网页元素

模拟鼠标移动的操作，网页中的有些元素需要将鼠标移动到某个地方才能显示出来，可以通过此操作来完成。

捕获网页元素：具体操作请参见

设置网页元素位置：只有在捕获到多个网页元素时使用才有效，如果录制的结果只有一项，默认设置为1即可。

如果录制一个列表的时候会录制到多个列表项，只获取第二个列表项，那么网页元素位置就设置为2。

选择已打开网页：选择之前打开的网页，默认已经选取最近一次打开的网页。

转到新网址

该模块用于将已打开的网页定位到新的网址。

在**转到新网址**页面，进行如下配置：

选择已打开网页：选择之前打开的网页，默认选取最近一次打开的网页。

新地址：输入要打开的新的网址。

获取网页元素内容

获取网页元素的文本、值、源代码、个数、勾选状态等操作，与获取网页信息不同的是获取的是元素内容，获取网页信息获取的是整个网页的内容。

捕获网页元素：具体操作请参考

选择提取类型，包含以下选项：

文本: 获取元素的文本内容，可用于提取文本类型元素的内容。

值: 获取元素的值，可用于提取输入框、密码框、下拉框等元素的内容。

源代码: 获取元素的源代码。

个数: 获取录制到的元素个数，只有在录制到多个元素的时候可用，一般在处理循环列表数据时使用。

勾选状态: 获取元素的勾选状态，可用于复选框、单选框。

设置网页元素位置：只有在捕获到多个网页元素时使用才有效，如果录制的结果只有一项，默认设置为1即可。

如果录制一个列表的时候会录制到多个列表项，只获取第二个列表项，那么网页元素位置就设置为2。

选择已打开网页：选择之前打开的网页，默认已经选取最近一次打开的网页。

打开网页

该模块用于操作内部浏览器打开一个新的网页，或者获取一个已经打开的网页。

打开新网页

在**打开网页**页面，进行如下选择：

请选择操作类型：配置为**打开新网页**。

打开网址：在输入框中输入需要打开网页的地址，如天猫网址。

根据标题捕获已打开网页

在**打开网页**页面，进行如下选择：

请选择操作类型：配置为**捕获已打开网页**。

捕获方式：配置为**根据标题匹配**。

匹配内容：输入要捕捉的标题，如天猫，便于获取标题中包含天猫的网页。

根据网址捕获已打开网页

在**打开网页**页面，进行如下选择：

请选择操作类型：配置为**捕获已打开网页**。

捕获方式：配置为**根据网址匹配**。

匹配内容：输入要捕捉的网址，如tmall，这样就能获取网址中包含tmall的网页。

拖拽网页元素

网页中的某些元素需要执行拖拽操作，如有些网站的验证功能就是通过用户拖动图标来完成的。

捕获网页元素：具体操作请参考

横向拖动偏移量：元素在横坐标位置拖动的距离，正数是从左向右拖动，负数是从右向左拖动。

竖向拖动偏移量：元素在纵坐标位置拖动的距离，正数是从上向下拖动，负数是从下向上拖动。

选择已打开网页：选择之前打开的网页，默认已经选取最近一次打开的网页。

读取网页中表格

读取指定表格对象中的内容。

选择读取类型，包含以下选项：

总行数: 读取表格的总行数。

总列数: 读取表格的总列数。

单元格内容: 读取指定行号、列号的单元格内容。

表格：选择之前查找到的表格，默认已经选取最近一次获取的表格。

上传文件

用于操作网页中的上传控件。

捕获网页元素：具体操作请参考

请选择文件：单击**选择文件**，选取一个用于上传的本地文件，也可以直接在后面的输入框中输入文件路径。

设置网页元素位置：只有在捕获到多个网页元素时使用才有效，如果录制的结果只有一项，默认设置为1即可。

如果录制一个列表的时候会录制到多个列表项，只获取第二个列表项，那么网页元素位置就设置为2。

选择已打开网页：选择之前打开的网页，默认已经选取最近一次打开的网页

等待网页加载完毕

该模块表示等待该网页完全加载完成后，再执行后续操作。

在**选择已打开网页**下选择之前打开的网页，默认选取最近一次打开的网页。

等待网页元素加载完成

等待网页加载完成，一般由于某些延迟加载的网页元素，该操作会暂停应用的执行，只有在网页元素成功加载或超时后，才会执行后续的操作。

捕获网页元素：具体操作请参考

超时时间：，最大的等待时间，单位为秒

选择已打开网页：选择之前打开的网页，默认已经选取最近一次打开的网页

参数面板

读取参数面板

获得参数列表中的某一参数的值。

参数位置/参数名称：提供需要读取的参数名称，或者参数所在列表的行数。

设置参数面板

设置参数列表中的某一参数的值。

参数位置/参数名称：提供需要设置的参数名称，或者参数所在列表的行数。

参数值：提供需要设置的参数内容。

流程

条件判断

根据条件判断是否成功，执行不同的命令。

条件判断页面，包含以下参数配置：

判断方式：满足任意条件时成功是指当判断条件为多条时，只要有一条判断成功就代表成功；满足全部条件时成功是指当判断条件为多条时，必须全部判断成功才代表成功。

判断条件：这里分三部分，最左侧是需要判断的动态内容，如性别、年龄；中间是判断方式，如等于、大于；右侧是要比较的内容。

示例如下图所示：

条件判断 ✕

判断方式

☒ 满足任意条件时成功

☐ 满足全部条件时成功

判断条件

性别 ✕

≡

等于 ▼

男

≡

✕

年龄 ✕

≡

大于 ▼

20

≡

✕

✚ 添加新条件

取消

确定

上图的设置代表：如果性别是男或者年龄大于20，则执行第一部分命令，否则，执行第二部分命令。

退出循环

用于退出整个循环或者忽略本次循环并执行下一次循环，只在循环执行内部有效。

继续下一次循环：表示忽略本次循环并执行下一次循环。

如果条件判断成功则忽略后续操作，直接执行下一次循环，不会再执行打印日志的命令。

结束循环：表示退出整个循环。如果条件判断成功则退出整个循环，不会再执行打印日志，也不会再执行下一次循环。

循环执行

重复执行一组命令，可以按次数执行、循环一个列表或者按条件循环。

按次数循环：通过指定执行次数，重复执行一组命令。

循环次数表示重复执行的次数。

104

列表循环：指定一个列表，依次取出列表中的内容，循环次数为列表中内容的个数。

列表是命令生成的一个列表（如正则表达式取出的多个匹配项），也可以是一个文本内容，使用英文逗号分隔，依次执行。

按条件循环：该循环类型可以实现根据判断是否成功来重复执行n次命令。

判断方式：满足任意条件时成功是指当判断条件为多条时，只要有一条判断成功就代表成功；满足全部条件时成功是指当判断条件为多条时，必须全部判断成功才代表成功。

判断条件：这里分三部分，最左侧是数据视图的列名，如性别、身高；中间是判断方式，如等于、不等于；右侧是要比较的值。

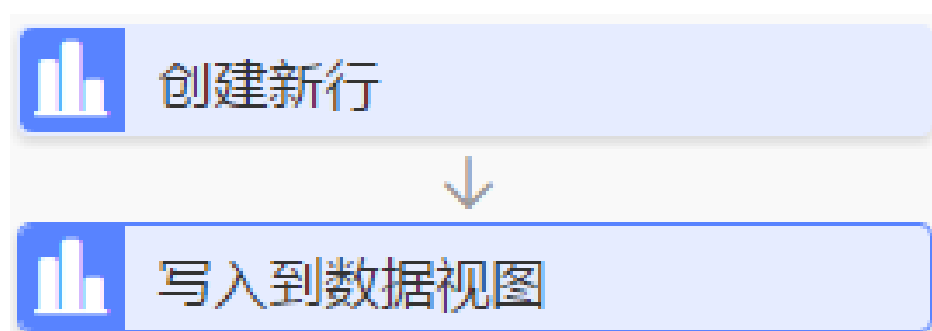
数据表格

创建新行

在数据视图中创建一个新行，并返回新行的行号。

设置值：默认是新行的第一列的单元格内容，可以为空。

在创建新行后通常是要设置单元格内容的，可以在创建新行后再拖入一个写入到数据视图来完成该操作，如下图所示：



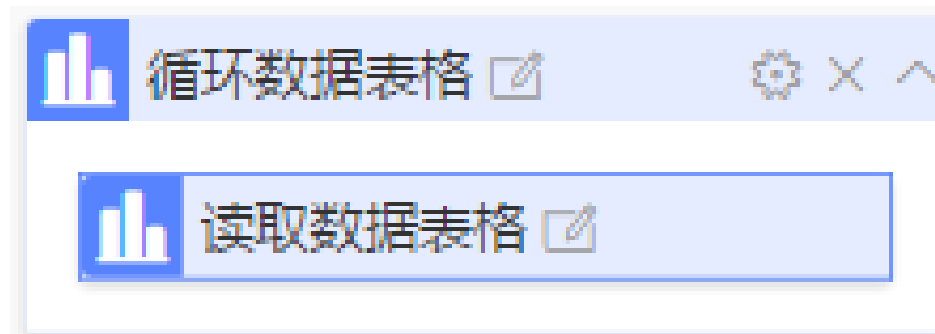
循环数据视图

用于依次遍历数据视图中的所有行，提供了3种循环方式。

循环整个数据视图

从数据视图的第一行开始遍历，直到最后一行，每遍历一行都会返回当前行的行号，再通过行号就可以方便的读取、设置单元格内容。

在循环数据内部可以再插入一个读取数据视图或者写入到数据视图的操作，如下图所示：



读取数据视图中的行号设置为**循环数据视图**，列名中填写需要读取的列名或者列号，如性别。

通过上面的示例就可以读取数据视图中每一行性别单元格中的值。

循环指定区域

该循环类型通过设置起始行号和结束行号来控制读取的范围及读取顺序，通过循环步长来控制每次读取递增的行号。

起始行号：设置遍历的起始行号，设置为1就是从数据视图中的第一行开始遍历，如果设置为-1则表示最后一行。

结束行号：设置遍历的结束行号，设置为-1就会一直遍历到数据视图的最后一行。

循环步长：设置每次遍历后递增的行号，如：起始行号为1，循环步长为1，就会依次读取1、2、3、4...行；如果循环步长为2就会依次读取1，3，5，7...

倒序循环：可以通过设置起始行号为-1，结束行号为1来实现倒序遍历。

按条件循环

该循环类型可以实现根据单元格内容是否满足条件来遍历数据视图

选择判断方式，包含以下选项：

满足任意条件时成功是指当判断条件为多条时，只要有一条判断成功就代表成功。

满足全部条件时成功是指当判断条件为多条时，必须全部判断成功才代表成功。

判断条件：这里分三部分，最左侧是数据视图的列名，如性别、身高；中间是判断方式，如等于、不等于；右侧是要比较的值。

示例：

循环数据视图

请选择循环类型

☐ 循环整个数据视图

☐ 循环指定区域

☒ 按条件循环

判断方式

☒ 满足任意条件时成功

☐ 满足全部条件时成功

判断条件

性别

⌵

等于

▼

男

⌵

🗑

身高

⌵

等于

▼

175

⌵

🗑

+ 添加新条件

取消

确定

上图的设置代表：只遍历性别是男或者身高等于175的行。

删除行

在数据视图中删除一行。

行号：指定要删除行的行号。

107

获取视图行数

返回数据视图的总行数，一般情况下用于循环，无配置项。

写入到数据视图

通过指定行号、列名、内容来设置指定单元格的内容，可以一次设置同一行的多个单元格。

示例：

写入到数据视图

行号

循环数据视图

设置项

性别	设置为	女	删除
身高	设置为	165	删除

+ 添加新设置项

取消

确定

上图的设置代表：设置当前遍历行的**性别为女，身高为165**。

读取数据视图

通过指定行号和列名来读取指定单元格的内容。

行号：可以指定为数字或者循环数据视图。如：设置为1，代表读取第1行或者设置为循环数据视图，代表读取当前遍历的行。

列名：设置为数据视图中某一列的列名称。

示例：

读取数据视图

×

行号

循环数据视图

×

⌵

列号/列名

性别

⌵

取消

确定

上图的设置代表：读取当前遍历行的**性别**单元格。

显示行数据

让数据视图中的某行数据显示出来。

行号：指定要显示的行号。

设置行勾选状态

设置数据视图中的某行勾选状态为勾选。

行号：指定要获取的行号。

是否勾选：指定要设置的状态。

编程

打印日志

可以将指定的内容打印到日志列表中，该功能在调试应用时常用。

界面参数**打印内容**表示指定需要打印的内容。

定时执行

用于指定时间点执行一次命令，或者每隔一段时间执行一次命令。

每天指定时间点执行一次：指定应用执行的时间。

每隔一段时间执行一次：执行应用执行时间间隔。

检查是否为对象

打开已有网页，检查一个变量是否是对象类型。

数组操作

用于处理数组，如创建新数组、获取数组内容、合并数组内容等操作。

数组操作类型，包括以下选择：

创建空数组：创建一个空的数组，用于后续操作。

获取数组长度：获取指定数组的内容个数。

获取数组内容：

要操作的数组: 一个数组，可以是正则表达式-匹配多项的返回结果，也可以是字符串拆分返回的结果。

数组索引: 要获取的内容在数组中的位置，从0开始计数。

获取数组内容的索引：是获取数组内容的反操作，该操作是根据内容获取索引的功能。

添加数组内容：向指定数组追加一个内容。

删除数组内容：删除数组中指定位置的内容。

拼接数组：将两个数组拼接为一个新数组。

合并字符串：将数组中的内容按照指定的分隔符合并成一个文本。

数学运算

执行一次数学运算，并返回结果。

操作类型: 目前支持 加、减、乘、除、平方、开方。

弹出提示窗口

弹出一个消息对话框，该命令会暂停脚本执行，直到在对话框中点击取消或确定。

包含以下参数：

消息内容：指定对话框中展示的消息内容。

打印内容：指定需要打印的内容。

退出程序

用于退出应用，当需要主动退出应用时可以使用此功能，无配置项。

延时执行

使应用暂停一段时间，一般用于等待网页加载。

暂停时间：指定应用需要暂停的时间，单位为毫秒，1秒=1000毫秒。

正则表达式

用于从文本中提取一段/多段文本，或者判断文本是否满足某一种格式，也可以将文本中的一段文本替换成其他内容。

正则表达式匹配类型，分为以下四种：

查询第一个匹配项：用于查找文本中的第一个满足条件的内容，并返回。

示例如下图，表示从内容中提取出1309875748：

正则表达式

正则表达式

\d+

要匹配的内容

订单编号 : 1309875748

请选择匹配类型

☒ 查询第一个匹配项

☐ 查询所有匹配项

☐ 是否匹配

☐ 替换匹配项

是否区分大小写

☒ 区分大小写

取消

确定

查询所有匹配项：用于查找文本中所有满足条件的内容，并返回。

示例如下图，表示从内容中提取出一个包含1309875748、4570975675、890987512三项内容的列表。

正则表达式

正则表达式

\d+

要匹配的内容

订单编号 : 1309875748 | 订单编号 : 4570975675 | 订单编号 : 890987512

请选择匹配类型

☐ 查询第一个匹配项

☒ 查询所有匹配项

☐ 是否匹配

☐ 替换匹配项

是否区分大小写

☒ 区分大小写

取消

确定

查询是否匹配：用于查找文本是否满足指定格式，成功返回1，失败返回0。

示例如下图，表示内容中包含数字，所以返回1。

正则表达式 ✕

正则表达式

\d+

要匹配的内容

订单编号 : 1309875748

请选择匹配类型

☐ 查询第一个匹配项 ☐ 查询所有匹配项 ☒ 是否匹配 ☐ 替换匹配项

是否区分大小写

☒ 区分大小写

取消 确定

替换匹配项：用于替换文本中的一段内容。

示例如下图，表示将会将内容中的数字替换为已过期，返回结果为等待编号：已过期。

正则表达式 ✕

正则表达式

\d+

要匹配的内容

订单编号 : 1309875748

请选择匹配类型

☐ 查询第一个匹配项 ☐ 查询所有匹配项 ☐ 是否匹配 ☒ 替换匹配项

是否区分大小写

☒ 区分大小写

要替换的内容

已过期

取消 确定

执行记录

用于记录业务执行次数，无配置项。

自定义脚本

用于执行一段自定义的码栈脚本。

字符串操作

用于处理文本，如拼接、拆分、转换大小写等操作。

字符串操作类型，包括以下选项：

拼接：将多个文本拼接成一个文本。

示例如下图，将返回你好码栈用户。

字符串操作 ✕

请选择操作类型

拼接

要操作的字符串

你好

拼接的字符串

码栈

用户

+ 添加

取消 确定

拆分：将一段文本按照指定字符拆分，返回一个数组。

示例如下图，表示分隔，获取一个包含张三、李四、王五三项的数组。

字符串操作 ✕

请选择操作类型

拆分

要操作的字符串

张三,李四,王五

分隔符

,

取消 确定

替换：将文本中的指定内容替换成另一个内容。

示例如下图，表示你好，张三替换成你好，李四。

字符串操作

请选择操作类型

替换

要操作的字符串

你好，张三

要替换的字符串

张三

替换后的字符串

李四

取消 确定

移除前后空格：将指定文本的前后空格移除。

获取长度:获取指定文本的字符个数，如你好，码栈!将返回6，空格和符号也会当做字符处理。

获取子字符串:提取文本内指定区域的子内容。

示例如下图，表示将得到欢迎使用码栈的结果。

字符串操作

请选择操作类型

获取子字符串

要操作的字符串

欢迎使用码栈平台

开始位置 0

字符个数 6

取消 确定

开始位置: 以0开始的索引，例如：如果从第一个字符开始就是0，如果是第二个字符开始就是1。

字符个数: 要截取的字符个数。

是否包含：判断一段文本中是否包含另一段文本，成功返回1，失败返回0。

是否全部为字母：判断一段文本中是否所有的字符都是英文字母，成功返回1，失败返回0。

是否全部为数字：判断一段文本中是否所有的字符都是数字，成功返回1，失败返回0。

是否全部为小写：判断一段文本中是否所有的字符都是英文小写字母，成功返回1，失败返回0。

是否全部为大写：判断一段文本中是否所有的字符都是英文大写字母，成功返回1，失败返回0。

转数字：将一个文本类型的数字转换为数字类型，如：从参数面板中读取的数字都是文本类型的，可以通过此方法得到数字类型。

转小写：将一个文本中所有的英文字母都转换为小写。

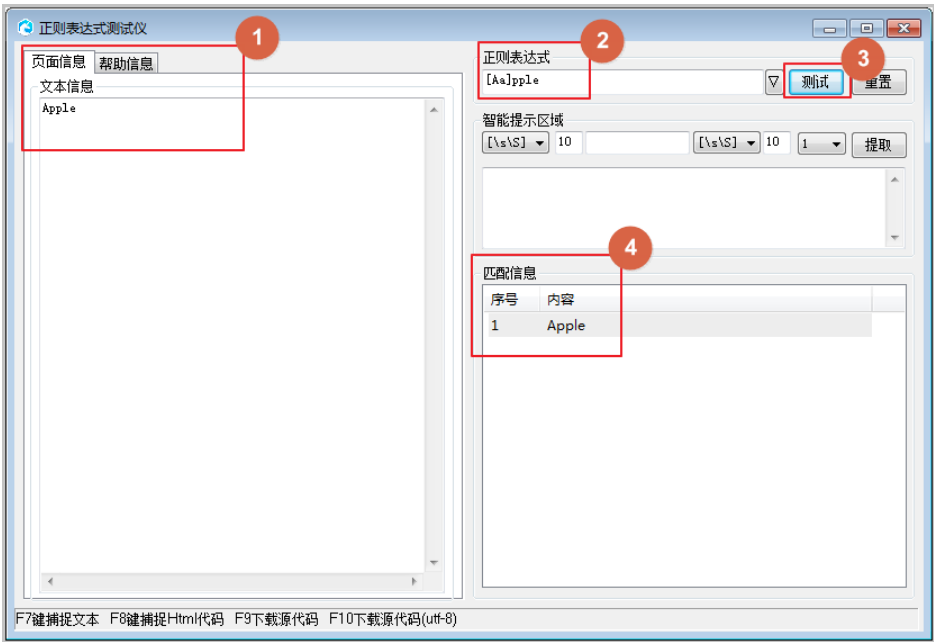
转大写：将一个文本中所有的英文字母都转换为大写。

正则表达式

正则表达式可以认为是数据筛选器，就是从一段文本中找到你想要的信息。

例如，要在一段文本中搜索匹配手机号，如13312345678，如何从中匹配到手机号码？

码栈提供了**正则表达式测试仪**帮助检测正则表达式是否正确，在编码开发模式下，您可以通过**正则表达式**按钮，打开测试仪进行验证。



正则表达式的组成

一个正则表达式有3部分组成：

- 匹配内容：筛选内容可以是具体的字符例如“手机号”，也可以是匹配符，例如\ d表示数字；
- 匹配数量：筛选数量就是你希望筛选内容出现的几次，默认1次；
- 分组提取：分组提取就是从符合筛选内容中提取具体的值，分组是可选的。

例如：通过正则表达式获得指定的文本，当位置=0的时候返回值为1或者0代表是否符合此正则表达式。

```
Local $url="www.taobao.com"
Local $middle=_EXP($url,"w+",1)
J_MsgBox($middle)
```

匹配符

匹配符号	匹配内容	样例
\d	匹配数字	- 正则表达式：\d1\d3 - 匹配值样例：0123
\D	匹配非数字	- 正则表达式： ：\DSKMNC你好啊 +S - 匹配值样例： ：QSKMNC你好啊 +S

\w	匹配数字、字母以及下滑线	- 正则表达式 ：\w\w\wd_d\d\d\d5 - 匹配值样例 ：Abcd_12345
\W	匹配非数字、字母以及下划线	- 正则表达式：你好 +\W\W=大家好 - 匹配值样例：你好+我好=大家好
\s	匹配空格、换行符等空白文本	- 正则表达式：\s\s\s - 匹配值：3个空格
\S	匹配非空格、换行符	- 正则表达式：你好 +\S\S大家好12_ab - 匹配值样例：你好+我好=大家好12_ab
.	匹配除换行符之外所有的数据	- 正则表达式：N..e - 匹配值样例：Nice
\.	匹配符号.，在正则表达式中，所有“有含义的”英文字符如果想要匹配，需要加上反斜杠	- 正则表达式：\.\. - 匹配值样例：.,
[]	匹配中括号中的数据，放在括号中表示非的意思	- 正则表达式 ：[Aa]pple或者 [^\d]pple - 匹配值样例：Apple
()	用于抓取目标数据	- 正则表达式：退款金额：(\d+.\d\d) - 匹配值样例：退款金额：100.00

匹配数量

匹配符号	匹配内容	样例
{n}	匹配指定次数，n为具体数字	- 正则表达式：\d{11} - 匹配值样例 ：17091313333

{n,m}	匹配指定n到m范围内的次数，n、m为具体数字或空值	- 样例1 正则表达式 ：\d{1,5}匹配值样例 ：匹配1~5个数字 - 样例2 正则表达式 ：\w{3,}匹配值样例 ：匹配至少3个数字
*	尽可能多的匹配，允许为空	-
+	尽可能多的匹配，至少要有一个满足	- 正则表达式：\s+ - 匹配值样例：匹配至少1个空格、换行符等空白文本
?	有可能出现，也有可能不出现	- 样例1 正则表达式 ：App?G?le匹配值样例 例：Apple - 样例2 正则表达式 ：.*?匹配值样例：尽可能少的匹配，可以为空 - 样例3 正则表达式 ：.+?匹配值样例：尽可能少的匹配，不能为空