# 域名表达式判断

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| 1 | 输入 | 输出 | 公式 |
| 2 | http://www.baidu.com/ | baidu.com | =MID(A2,SEARCH(".",A2)+1,SEARCH("/",A2,SEARCH(".",A2)+1)-SEARCH(".",A2)-1) |

If Left( cell( i , j ), 11 )= “http://www.” Then

cell( i , j )= MID(A2,SEARCH(".",A2)+1,SEARCH("/",A2,SEARCH(".",A2)+1)-SEARCH(".",A2)-1)

Else cell( i , j )= Left( cell( i , j ), LEN(cell( i , j ))-1 )

Cell(i+1,j+1)=IF(LEFT(cell(i,j),11)="http://www.",cell(i,j)= MID(cell(i,j),SEARCH(".",cell(i,j))+1,SEARCH("/",cell(i,j),SEARCH(".",cell(i,j))+1)-SEARCH(".",cell(i,j))-1), cell( i , j )= Left( cell( i , j ), LEN(cell( i , j ))-1 ))

# VBA代码

## If判断语句

### 单条件判断

判断a、b的值是否相等

Dim a%, b%

a = 2

b = 2

If a = b Then #换行后需要加End If，没有换行不需要，或者使用“ :”表示#

MsgBox” 相等 “

End If

Dim a%, b%

a = 2

b = 2

If a = b Then MsgBox” 相等 “ : a=a+1 : b=b+1

#多行可以使用“ :”表示#

Dim a%, b%

a = 2

b = 2

If a = b Then

MsgBox” 相等“

Else

MsgBox” 不相等 ”

End If

### 多条件判断

判断B1单元格分数所符合的分数等级，在B2单元格输出对应等级

If Sheet1.Cells(1, 2) >= 90 Then

Sheet1.Cells(2, 2) = "优秀"

ElseIf Sheet1.Cells(1, 2) >= 80 Then

Sheet1.Cells(2, 2) = "良好"

ElseIf Sheet1.Cells(1, 2) >= 60 Then

Sheet1.Cells(2, 2) = "中等"

Else

Sheet1.Cells(2, 2) = "差"

End If

函数表示：

Cells(2,2) = IIf(Cells(1,2)>=90,”优秀”, IIf(cells(1,2)>=80,”良好”,IIf(Cells(1,2)>=60,”中等”,”差”)))

## Select判断语句

判断i的正负

i = 1

Select case i

Case Is > 0 #Is 后接<、>、=、符号，以比较是否符合对应值，没有写，程序自动增添#

MsgBox”正数”

Case Else

MsgBox”负数”

End Select

多条件判断

Select case sheet1.[d1].value

Case “A”

Sheet1.[a3] = “A型血”

Case “B”

Sheet1.[a3] = “B型血”

Case “AB”

Sheet1.[a3] = “AB型血”

Case “O”

Sheet1.[a3] = “O型血”

Case Else

Sheet1.[a3] = “无这种血型”

End Select

# Do…Loop循环语句

**先判断后循环：**

**Do** [{**While** | **Until**} *condition*]  
[*statements*]  
[**Exit Do**]  
[*statements*]

**Loop**

**先循环后判断：**

**Do**  
[*statements*]  
[**Exit Do**]  
[*statements*]

**Loop** [{**While** | **Until**} *condition*]

a大于10时弹出对话框并退出循环

Dim a%

Dim a%

Do

a = a + 1

If a > 10 Then

MsgBox a & "大于10"

Exit Do#退出Do循环#

End If

Loop

### Do While …Loop语句——当While条件成立时执行循环

判断是否大于90，大于则在（a,3）输出“大于90分”

Dim a%

a = 2

Do While cells(a,2) <> “”#判断是否空值，当不是空值时执行循环#

If cells(a,2) >= 90 Then cells(a,3)=”大于90分”

a = a +1

Loop

### Do Until …Loop语句——执行循环，直到Until条件成立时才停止循环

判断是否大于90，大于则在（a,3）输出“大于90分”

Dim a%

a = 2

Do Until cells(a,2) = “”#判断是否空值，当不是空值时执行循环#

If cells(a,2) >= 90 Then cells(a,3)=”大于90分”

a = a +1

Loop

隔行填色

Dim a%

a = 2

Do Until Sheet1.Range(“A” & a) = “”

Sheet1.Range(“A” & a & “ : ” & “G” & a).Interior.ColorIndex = 7

a = a +2

Loop

Until在其他位置：

Dim pss$,

Do Until pss = “ABC” # Pss等于ABC时退出循环#

Pss = InputBox(“输入密码”)

Loop

#先判断Pss的值，由于一开始没有赋值，第一次判断无意义#

Dim pss$,

Do

Pss = InputBox(“输入密码”)

Loop Until pss = “ABC”

#先对Pss赋值，再判断是否相符#

Dim pss$,i！

Do

i = i + 1

If I > 3 Then exit do#赋值3次不符后退出循环#

Pss = InputBox(“输入密码”)

Loop Until pss = “ABC”

# For…Next…循环语句：以指定次数执行一组语句

**For** *counter* **=** *start* **To** *end* [**Step** *step*]  
[*statements*]  
[**Exit For**]  
[*statements*]

**Next** [*counter*]

*counter* ：计数变量

*star*：初始值

*end*：终止值

*step*：步长值

*counter*：计数变量

Exit For：常与If … Then一起使用

嵌套循环

Dim I!, J!, k$

For I = 1 To 10

For J = 1 To 10

k = “外层第”& I & “次， 内层第”& J & “次”

Next J

Next I #嵌套循环，counter使用了不同的变量名#

Dim i!, j!

For i = 1 To 16

If Cells(i, 1) = "AA" Then Exit For

Next i

For j = 1 To 16

If Cells(j, 1) = "BB" Then Exit For

Next j

MsgBox "BB的数量为" & j - i

### For Each … Next循环语句: 对某范围的单元格做循环

改变将a2:a10中的值等于Human的单元格的底色

Dim r as Range#声明变量r为单元格集合#, n!

For Each r In sheet1.range(“a2:a10”)

#声明需要循环的对象为a2:a10的每个单元格#

If r = “Human” Then r.Interior.ColorIndex = 3

Next

将当所有工作表的名称赋到C列上

Dim w as Worksheet, n as Byte#字节类型，范围为0~255#, m$

For Each w In Worksheets

#声明需要循环的对象为所有工作表#

n = n + 1

m = w.Name

Sheet1.Cells(n,3) = m

Next

### For … to… Next循环语句：含有一个内置计数器

求数列：1+2+3+…+99+100的和

Dim i!, j!

For i=1 To 100#内置计数器i，语句中i自动从1递增至100 ，即循环100次#

j = j + i

Next

Msgbox j

Dim i!, j!

For i=1 To 100 Step 2#Step表示递增的差额为2，即1+3+5+7…#

j = j + i

Next

Msgbox j

Dim i!, j!

For i=100 To 1 Step -1#Step表示递减的差额为1，即100+99+98+…#

j = j + i

Next

Msgbox j

求第二行至十六行（A, rng）和(B,rng)的乘积，输出至C

Dim rng!

For rng = 2 To 16

Sheet1.Cells(rng, 3) = Sheet1.Cells(rng, 2) \* Sheet1.Cells(rng, 1)

Next

Dim l!, r!

For l = 2 To 10

For r = 2 To 10

Cells(l, r) = Cells(l, 1) \* Cells(1, r)

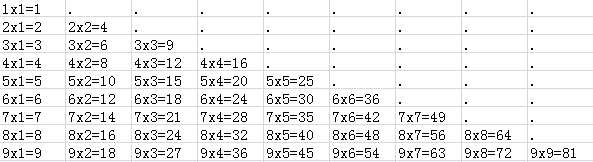
If i > 10 Then Exit For

Next r

If l > 10 Then Exit For

Next l

九九乘数表



Dim a!, b!

For a = 1 To 9

For b = 1 To 9

If b > a Then

Sheet2.Cells(a, b) = ""

Else

Sheet2.Cells(a, b) = a & "x" & b & "=" & a \* b

End If

Next

Next

# Goto line 无条件地转移到过程中指定的行

### Goto line

除非输入ID值为admin，否则输入框循环

Dim str$

Line: #冒号为必须，用于识别要跳转的行#

Str = InputBox(“input ID”)

If str < >”admin” Then goto Line #Line 非固定值#

### GoSub...Return

Sub gotoreturn()

Dim i!

For i = 2 To 10

If Sheet1.Range(“a” & i) > 1/3 Then GoSub 100#跳转至100：所在位置#

Next i

Exit Sub#等同End#

100:

Sheet1.Range(“b” & i) = “迟到”

Return#返回至GoSub的下一行#

End Sub

# On Error 错误语句

### On Error Resume Next错误发生时，控件转到紧接着发生错误的语句之后的语句

Dim i!

On Error Resume Next ‘当错误的时候继续执行下去

For i= 2 to 8

Cells(i,4) = Cells(i,3) + Cells(i, 2)

Next i

End Sub

### On Error GoTo *line*运行时错误，则控件会跳到 *line*

Sub a()

On Error GoTo 100

For i = 2 To 8

K = Sheet1.Cells(i, 2) + Sheet1.Cells(i, 3)

Next i

100:

MsgBox”错误发生在第” & i & “行”

End Sub

# With 语句：在一个单一对象或一个用户定义类型上执行一系列的语句

**With** *object*  
[*statements*]

**End With**

With Range(“a1”)

A=.Address ‘ ” . ”号不能省略

B=.Parent.Name

.Value=”1234”

End With

嵌套

With Range(“a1”)

.Value = “Who am i?”

.Parent.Name = “Hello World”

With .Font

.Size = 20

.Bold = True

End With

End With

# 函数

### 借用工作表函数

Application.WorksheetFunction.*函数*

显示A1至A10单元格中的值为AA的数量，以下代码效果相同

MsgBox Application.WorksheetFunction.CountIf（range（”a1:a10”）,”AA”）

‘表达式使用VBA格式，如COUNTIF(A1:A10,"AA")转换成CountIf（range（”a1:a10”）,”AA”）

MsgBox WorksheetFunction.CountIf（range（”a1:a10”）,”AA”）

MsgBox Application.CountIf（range（”a1:a10”）,”AA”）

### 使用VBA函数

### 自定义函数

# Like运算

*result* **=** *string* **Like** *pattern*

result 必需的；任何数值变量。

string 必需的；任何字符串表达式。

pattern 必需的；任何字符串表达式，遵循“说明”中的模式匹配约定。

通配符：

? 任何单一字符。

\* 零个或多个字符。

# 任何一个数字 (0–9)。

[charlist] charlist.中的任何单一字符。

[!charlist] 不在 charlist 中的任何单一字符。

结果为False

Sub a1()

a = “admin” Like “Admin” ‘区分大小写

End Sub

以下结果为True

Sub a2()

b = “abc” Like “a?c” ‘通配符

b2=”abc” Like “???”

Sub a3()

c = “excel函数” Like “\*函\*” ‘通配符

End Sub

Sub a4()

d = 556 Like “###” ‘通配符

End Sub

Sub a5()

e = “a” Like “[a-z]” ‘通配符

f = 8 Like “[!1-7]” ‘通配符

g = 8 Like “[1-4,6-9]” ‘通配符

End Sub

\*：[A-Z][A-Z]###表示前两个时任意大写字母，再接三个任意数字

库存盘点表

范例：

Dim rng As Range, rngs As Range, K%

For Each rng In [a2:a6]

For Each rngs In [b2:b4]

If rng = rngs Then

GoTo 100

End If

Next rngs

k = k + 1

Cells(k+1, “c”) = rng

100:

Next rng

自己写的：

l = 2

For j = 2 To 6

n = 0

For i = 2 To 4

If Cells(i, 2) Like Cells(j, 1) Then n = n + 1

Next

If n < 1 Then

Cells(l, 3) = Cells(j, 1)

l = l + 1

End If

Next

# 工作簿表示方法

Workbooks(“工作簿名称”)

Workbooks(索引号)

Workbooks(窗口数)

当前工作簿：thisworkbook

活动工作簿：activeworkbook

验证当前工作簿是否已打开

Dim wk As Workbook, a

For Each wk In Workbooks

a = wk.Name

If wk.Name = “ABC.xlsm” Then

wk.Activate

MsgBox “已经激活工作簿：” & wk.Name

Exit Sub

End If

Next wk

MsgBox “没有发现工作簿：ABC.xlsm”

End Sub

### 新建、保存工作簿

Dim wkb As Workbook ‘声明wkb为工作簿

Set wkb = Workbooks.Add ‘新建工作簿

wkb.SaveAs “d:\123.xlsx” ‘保存工作簿

### 打开工作簿

Dim wkb As Workbook

Set wkb = Workbooks.Open(“d:\123.xlsx”)

### 关闭工作簿

Workbooks(“123”).Close True ‘true表示保存，false表示不保存

### 复制、删除工作簿

FileCopy “d:\123.xlsx”, “d:\312.xlsx” ‘对所有文件类型都起作用

Kill “d:\312.xlsx”

检查文件是否存在

a = Dir(“d:\123.xlsx”)

If a = “” Then

MsgBox “不存在”

Else

MsgBox “存在”

End If

打开指定目录下的所有文件

Dim a$, n!, wbs As Workbook

a = Dir(“d:\\*.xlsx”)

Workbooks.Open “d:\” & a

Do

a = Dir ‘当Dir运行第二次时可以省略路径

If a <> “” Then

Workbooks.Open “d:\” & a

Else

Exit sub

End If

Loop

End Sub

# 工作表表示方法

Worksheets表示工作表

Sheets包含图表、工作表、宏表

Worksheets(“工作表名称”)

Sheets(“图表名称”)

Worksheets (索引号)

Sheets (工作表代码索引号)

将所有表名称输出至单元格

For Each shs In Sheets

k = k + 1

Cells(k, 1) = shs.Name

Next

将所有工作表输出至单元格

For Each shs In Worksheets

k = k + 1

Cells(k, 2) = shs.Name

Next

表名为ABC的表是否存在

Dim sn$

For Each sht In Sheets

sn = sht.Name

If sn = “ABC” Then

MsgBox “存在”

Exit Sub

End If

Next

MsgBox “不存在”

Dim sn$

For i = 1 To Sheets.Count

If Sheet(i).Name = “ABC” Then

MsgBox “存在”

Exit Sub

End If

Next

MsgBox “不存在”

### 增加工作表

表达式.Add(Before, After, Count, XlSheetType)

表达式:一个代表 Sheets 对象的变量。

Before：指定工作表的对象，新建的工作表将置于此工作表之前。

After：指定工作表的对象，新建的工作表将置于此工作表之后。

Count：要添加的工作表数。默认值为 1。

XlSheetType：

xlWorksheet:工作表（不指定的，默认为工作表）

xlChart :图表

xlExcel4IntlMacroSheet：Excel 4 国际宏工作表

xlExcel4MacroSheet：Excel 4宏工作表

xlDialogSheet:对话框工作表

Sheets.Add ‘新建工作表，且默认在当前活动工作表前

Sheets.Add Sheets(“abc”) ‘在abc工作表前新建

Sheets.Add , Sheets(“abc”) ‘在abc后新建

Sheets.Add after:= Sheets(“abc”) ‘在abc后新建

Sheets.Add , , 2 ‘新建2个表

Sheets.Add Count:=2 ‘新建2个表

Sheets.Add , , , xlChart ‘新建图表

批量增加12个工作表，工作表命名为1至12月

Dim i%

For i = 12 To 1 Step -1

Sheets.Add.Name = i & “月”

Next

### 删除工作表

Sheet1.Delete

批量删除12个工作表

On Error Resume Next

Application.DisplayAlerts = False

Dim i%

For i = 1 To 12

Sheets(i & “月” ).Delete

Next

Application.DisplayAlerts = True

### 移动工作表

表达式.Move(Before, After)

Sheet1.Move , Sheet3 ‘将工作表移动至Sheet3之后

Sheet1.Move after:=Sheet3

Sheet1.Move after:=Sheets(Sheets.Count) ‘将工作表移动至最后

批量增加12个工作表，工作表命名为1至12月

Dim i%, sth As Worksheet

For i = 1 To 12

Set sth = Sheets.Add

sth.Move after:=Sheets(Sheets.Count)

sth.Name = i & “月”

Next

### 复制工作表

表达式.Copy(Before, After)

Sheet1.Copy Sheets(Sheets.Count) ‘将Sheet1复制到最后一个工作表前

### 选择工作表

工作表名称.Select ‘不支持选取隐藏的工作表

Sheet1.Select

Worksheets.Select ‘快速选择所有工作表

Sheets.Select ‘快速选择所有表

Worksheets(Array(1,3,5)).Select ‘自定义选择

### 激活工作表

Worksheet.Activate ‘支持选取隐藏的工作表

### 拆分工作簿

Dim wk As Workbook, ss$, k%

Application.DisplayAlerts = False

For Each sht In Workbooks(“ABC”).Sheets

Set wk = Workbooks.Add

k = k + 1

Workbooks(1).Sheets(k).Copy Workbooks(2).Sheets(1)

ss = ThisWorkbook.Path & “\” & sht.Name & “.xlsx”

wk.SaveAs ss

wk.Close

Next

Application.DisplayAlerts = True

MsgBox “拆分工作簿完成”

# 单元格区域

### Range表示

除非指定，默认是当前活动工作表。如对象不是工作表的（如图表）会出现错误。

Range(“a1”).Select ‘选择A1单元格

Range(“a:a”).Select ‘选择A列

Range(“1:1”).Select ‘选择1行

Range(“a1:b10”).Select ‘选择相邻区域

Range(“a1”,”b10”).Select ‘选择相邻区域 等同上

Range(Range(“a1”), Range(“b10”)).Select ‘选择相邻区域 等同上

Range(“a1:b10,e1:g10”).Select ‘逗号表示选择不相邻区域

Range(“a1:b10 b1:d10”).Select ‘空格表示选择相邻区域的集合部分

Sheet1.Range(“a1”)=”44” ‘指定sheet1的A1单元格的值为44

在立即窗口显示

Debug.Print Range(“a:b”).Count ‘计算A、B列最大的行数，在立即窗口显示

Debug.Print Range(“1:1”).Count ‘计算最大的列数，在立即窗口显示

Debug.Print Application.CountA(Range(“a:a”)) ‘计算工作表已使用的行数，在立即窗口显示 计算出非空单元格数量

Debug.Print Application.CountA(Range(“1:1”)) ‘计算工作表已使用的列数，在立即窗口显示 计算出非空单元格数量

相对引用Range(“单元格区域”).Range(“单元格区域”)

Range(“b2:d4”).Range(“a1”).Select ‘选择B2：D4单元格的左上角单元格

变量表示

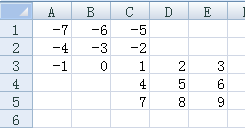
Dim a%

a = 3

Range(“a” & a).Select ‘选择A3单元格， 中间使用&连接

Range(“c3:e5”)(2).Select ‘选择C3：E5区域横向第二个单元格，即D3。2是索引值，可以是正、负、零值、小数。横向计数如下图

Range(“c3:e5”)(2.4).Select ‘选择C3：E5区域横向第二个单元格，即D3。索引值若出现小数按“银行家舍入法”（四舍六入五单双）



动态查找单元格区域

Dim i%

i = Application.CountA(Range(“c:c”)) ‘找出C列已经使用的非空单元数量，该数量为动态，可以使用变量表示，在以下单元格选取中使用变量表示

Range(“c” & i).Select ‘选择C列最后一格

Range(“a1”,”c” & i).Select ‘选择A1到C列最后一格包含的区域，使用逗号表达

Range(“a1:c” & i).Select ‘选择A1到C列最后一格包含的区域，使用冒号表达

算出大于等于2500的平均分

Dim rs%, rng%, lj&, k%

Dim rs = 1 To 60

Range(“b2:c20”)(rs).Select

rng = Range(“b2:c20”)(rs)

If rng >= 2500 Then lj = lj + rng: k = k +1 ‘将lj累加， 使用变量k计数

Next rs

MsgBox “大于等于2500的平均分为：” & Int(lj / k)

### Cells表示：只能引用一个单元格

Cells(行号,列号)

Cells(3,4).Select ‘选择D3单元格

Cells(2,”c”).Select ‘选择C2单元格

Cells.Select ‘全选

相对引用，参照前面的单元格位置

Range(“b3:f11”).Cells(2,2).Select ‘选择C4单元格

Range(“b3:f11”).Cells(6).Select ‘选择C4单元格，顺序从左至右，从上至下

Range(“b3:f11”).(6).Select ‘和上相同



和Range嵌套使用

Range(Cells(3,2),Cells(11,6)).Select ‘选择B2至F11的区域

动态查找单元格区域

Dim a%, b%

a = Application.CountA(Range(“a:a”)) ‘找出A列已经使用的非空单元数量，该数量为动态，可以使用变量表示，在以下单元格选取中使用变量表示

b = Application.CountA(Range(“1:1”)) ‘找出第1列已经使用的非空单元数量

Range(Cells(1,1),Cells(a,b)).Select ‘选择A1到右下单元格包含的区域达

### 简写单元格

[单元格地址] 注：括号中单元格地址不需要双引号

[a3].Select

[b2:c6].Select

[a3,b2:c6,b8:d12].Select

[a:a].Select

[1:1].Select

[b2:c6].Item(3).Select ‘选择B3单元格

[b2:c6].Cells(3).Select ‘同上

Range(“b2:c6”).Item(3).Select ‘同上

Range(“b2:c6”) (3).Select ‘同上

[b2:c6](3).Select ‘该表示**错误**

动态引用

Dim a%, b%

a = Application.CountA([a:a]) ‘找出A列已经使用的非空单元数量，该数量为动态，可以使用变量表示，在以下单元格选取中使用变量表示

b = Application.CountA([1:1]) ‘找出第1列已经使用的非空单元数量

Range(Range(“a1”), Range(Chr(64+b) & a)).Select ‘选择A1到右下单元格包含的区域达,**使用Chr函数，将数字函数变换至字母函数，Chr（64+1）=A，Chr（64 + 2）= B…..**。

Chr函数数字循环

For i = 1 To 65535

Cells(i, 1) = i

Cells(i,2) = Chr(i)

Next

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | Rangel | Cells | 简写方式[ ] |
| 变量支持 | 单元格、区域、行、列 | 单元格 | 单元格、区域、行、列 |
| 变量支持 | 支持 | 支持 | 不支持 |
| 书写难易 | 难 | 难 | 易 |

Range：引用单元格是区域而且有变量

Cells：引用的是单个单元格而且有变量

简写方式[ ] ：不含变量的

### Columns和Rows表示

Columns(1).Select **‘选择A列**

Columns(“b”).Select **‘选择B列**

Columns(“c:e”).Select **‘选择C、D、E列 引用多列时不能使用数字表示**

Columns.Select **‘选择所有列**

i = Columns.Count **‘计算列数**

Rows(1).Select **‘选择第1行**

Rows(“2”).Select **‘选择第2行**

Rows(“3:4”).Select **‘选择第3、4行 引用多行时需要加引号**

Rows.Select **‘选择所有行**

j =Rows.Count **‘计算行数**

动态引用：

a =Application.CountA(Columns(1)) **‘计算第1列中连续非空单元格数量**

b =Application.CountA(Rows(1)) **‘计算第1行中连续非空单元格数量**

Range(“a1”, Cells(a,b)).Select

### Range.Column 和Range. Row属性

Range.Column返回指定区域中第一块中的第一列的列号。Long 类型，只读。

Range. Row返回区域中第一个子区域的第一行的行号。Long 类型，只读。

i = Range(“a3:b9”).Range(“a5”).Row **‘相对引用，返回值为7**

j = Range(“a3:b9”).Row **‘返回值为3**

i = Range(“b3:d9”).Range(“b5”).Column **‘相对引用，返回值为3**

j = Range(“b3:d9”).Column **‘返回值为2**

将双数行的行高设置为5磅

For Each rw In Rows(“1:13”)

If rw.Row Mod 2 = 0 Then  **‘计算行号除以2余数是否为0**

Rw.RowHeight = 5

End If

### 单元格的值的表达方式：Range.Value和Range.Text 属性

Range.Value:返回或设置一个 Variant 型，它代表指定单元格的值。单一单元格可以省略Value，多个单元格不能省略。

Range.Text:返回或设置指定对象中的文本。String 型，只读。

a = [a1].Value  **‘实际是什么就返回什么**

b = [a1].Text **‘看到是什么就返回什么,与单元格格式设置有关**

c = [a1] **‘和a相同，即Value可以省略**

Range(“e1:e4”) = Range(“d1:d4”).Value **‘将e1至e4单元格的值赋到d1至d4，此时Value不能省略**

### 单元格的地址与引用：Range.Address

Range.Address：返回一个 String 值，它代表宏语言的区域引用。

Range.Address(行参数，列参数) 参数是1和0，1是绝对引用，0是相对引用，默认参数是1。

Set rng = [b2:f2]

[a9] = rng.Address(1, 1) **‘绝对引用，返回$B$2:$F$2**

[b9] = rng.Address(0, 0) **‘相对引用，返回B2:F2**

[c9] = rng.Address(1, 0) **‘混合引用，返回B$2:F$2**

[d9] = rng.Address(0, 1) **‘混合引用，返回$B2:$F2**

[e9] = rng.Address **‘使用默认值，绝对引用，返回$B$2:$F$2**

标记将空白单元格

Dim rng As Range, rn$

On Error Resume Next

For Each rng In Range("b2:b20")

If rng = "" Then rn = rn & rng.Address & ","

Next

Range(Left(rn, Len(rn) - 1)) = "空白"

### 剪切与复制：Range.Cut、Range.Copy

Range.Cut ：表达式.Cut(Destination)

Range.Copy：表达式.Copy(Destination)

#如果Destination参数，Microsoft Excel 会将区域复制到剪贴板。

Range(“a1:d8”).Cut Range(“f1”) **‘将A1至D8单元格移动至F1**

Range(“f1:i8”).Copy Range(“a1”) **‘将F1至I8单元格移动至A1**

Range(“a10:d17”) = Rangel(“a1:d8”).Value **‘使用Value方法等价Copy，但只粘贴值，不复制格式**

出差登记表记录保存

Dim rs%, crs%

rs = Application.CountA([d:d]) **‘返回D列的非空单元格数量**

crs = Application.CountA(Sheets(“出差记录表”).[a:a])+1 **‘返回“出差记录表”A列非空单元格+1**

If rs = 1 Then GoTo 100 **‘如果rs=1则执行100**

Range(“a2:d” & rs).Copy Sheets(“出差记录表”).Range(“a” & crs)

End

100:

MsgBox “没有要保存的数据”

### 插入与删除：Range.Insert和Range.Delete

Range.Insert：表达式.Insert(Shift, CopyOrigin)

# CopyOrigin 可选 Variant 复制的起点。

# Shift 可选 Variant 指定单元格的调整方式。如果省略此参数，Microsoft Excel 将根据区域的形状确定调整方式。

Range.Delete：表达式.Delete(Shift)

# Shift 可选 Variant 仅用于 Range 对象。指定如何调整单元格以替换删除的单元格。如果省略此参数，Microsoft Excel 将根据区域的形状确定调整方式。

Rows(2).Insert **‘在第二行上方插入一行**

Rows(1).Delete **‘删除第一行**

隔行插入

Dim r%

Do

r = r + 2

Rows(r).Insert

Loop Until Cells(r + 1, 1) = “”

隔行删除

Dim r, s

m = Application.CountA(Columns(1))

For r = 1 To m/2

Rows(r).Delete

Next

### 激活选取单元格（区域）：activecell、selection

a = activecell.Address **‘返回活动单元格地址**

Cells(2, 3).Activate **‘激活C2单元格**

Selection = 1 **‘将所选区域的值赋为1**

#Selection的好处在于可以自由地选取单元格#

计算所选单元格个数，将其逐个激活

For i = 1 To Selection.Count

Selection(i).Activate

Next

将所选区域内单元格值为空或“缺勤”的单元格值变为“No”

Dim i As Range, j$

For Each i In Selection

j = i.Address

If i = “” Or i =”缺勤” Then

i = “No”

End If

Next i

### 工作表所使用的区域：UsedRange

Sheet2.UsedRange.Select **‘选取Sheet2工作表已使用的区域**

#定位方式：已使用的最小单元格：最大单元格#

#即使单元格内没有赋值，但设定了格式，也被认为是已使用的区域#

#如果没有已使用单元格，默认为A1单元格#

统计大于90分的人数

For Each Rng In Sheet1.UsedRange

If IsNumberic(Rng) And Rng >= 90 Then k = k +1

**‘函数IsNumeric()：检查一个单元格是否数字，如果是返回True，如果不是数字或是日期表达式则返回False**

**‘如果所在单元格的值是数字且大于等于90，则k变量加1**

Next Rng

MsgBox “大于等于90分的人数为” & k & “人”

### 工作表当前区域：CurrentRegion

返回一个 Range 对象，该对象表示当前区域。当前区域是以空行与空列的组合为边界的区域。

[a1]. CurrentRegion.Select **‘选取A1单元格相邻的区域**

[F8]. CurrentRegion.Select **‘选取F8单元格相邻的区域**



将第二行、第三行的结果求和，计算结果在第8行显示

Rows(8).clear **‘每次都清除第8行**

a = [b2].CurrentRegion.Address **‘将b2单元格相邻的单元格地址赋值至a**

b = [b5].CurrentRegion.Address

c = [b2].CurrentRegion.Count + 1

Set d = Range(“b8”, Cells(8,c))

d.FormulaArray = “=” & a & “+” & b **‘调用数组公式**

CurrentRegion与UsedRange的异同：

#当表中只有一个从A1开始的区域，两者返回结果相同

#表达方式不同

Sheet1. UsedRange.Select **‘UsedRange指明所在的表**

[a1].CurrentRegion.Select **‘CurrentRegion指明所在单元格**

### 偏移区域Offset属性

对指定单元格区域进行偏移，它代表位于指定单元格区域的一定的偏移量位置上的区域。

Range.Offset(RowOffset, ColumnOffset)

RowOffset，可选，Variant，区域偏移的**行**数（正数、负数或 0（零））。正数表示向下偏移，负数表示向上偏移。默认值是 0。

ColumnOffset，可选，Variant，区域偏移的列数（正数、负数或 0（零））。正数表示向右偏移，负数表示向左偏移。默认值是 0。

[a1].Offset(1,2).Select **‘偏移1行，2列，选择C2单元格**

[a1].Offset(2).Select **‘偏移2行，选择A3单元格**

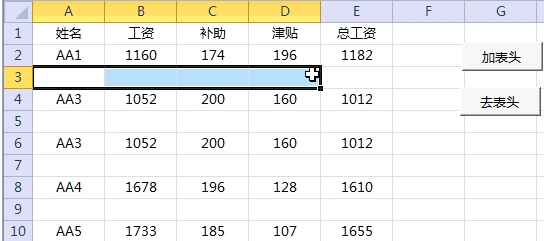
[a1].Offset(,2).Select **‘偏移2列，选择C1单元格**

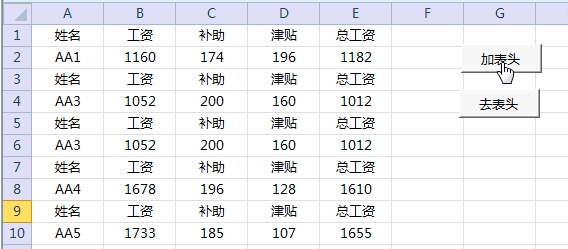
[a1:d1].Offset(1,2).Select **‘整个区域偏移1行，2列，选择C2:F2单元格**

[a1:d1].Offset(2).Select **‘整个区域偏移2行，选择A3:D3单元格**

[a1:d1].Offset(,2).Select **‘整个区域偏移2列，选择C1:F1单元格**

添减表头





添加表头

Dim i%

For i = 2 To 8 Step 2

[a1:e1].Copy [a1:e1].Offset(i)

Next i

减去表头

Dim i%

For i = 2 To 8 Step 2

[a1:e1].Offset(i).Clear

Next i

### 调整指定区域的大小：Resize

Resize(RowSize, ColumnSize)

RowSize：新区域中的行数。如果省略该参数，则该区域中的行数保持不变。

ColumnSize：新区域中的列数。如果省略该参数。则该区域中的列数保持不变。

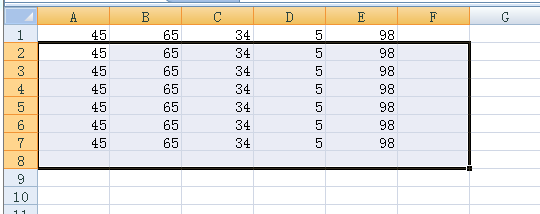
[a1].resize(1,1).select **‘选择A1单元格**

[a1].resize(2,3).select **‘选择A1:C2区域**

[a1].resize(2).select **‘选择A1:A2区域**

[a1].resize(,3).select **‘选择A1:C1区域**

调整 Sheet1 中选定区域的大小，使之增加一行和一列。



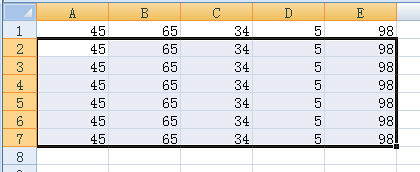
Worksheets("Sheet1").Activate

numRows = Selection.Rows.Count

numColumns = Selection.Columns.Count

Selection.Resize(numRows + 1, numColumns + 1).Select

选定该表，但不选定标题行。运行本示例之前，活动单元格必须处于该表中。

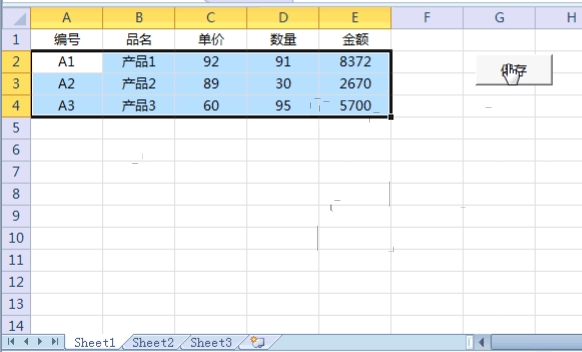
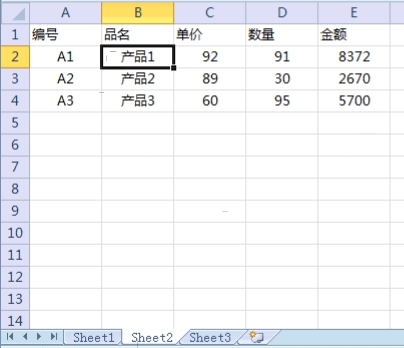


Set tbl = ActiveCell.CurrentRegion

tbl.Offset(1, 0).Resize(tbl.Rows.Count - 1, \_

tbl.Columns.Count).Select

将sheet1的数据保存至sheet2

Dim i%, j%, k%

i=[a1].CurrentRegion.Rows.Count - 1

j=[a1].CurrentRegion.Columns.Count - 1

k=Application.CountA(Sheet2.Columns(1))

[a2].Resize(i,j).Copy Sheet2.[a1].Offset(k)

### 整行、列：Range.EntireRow、Range.EntireColumn

Range.EntireRow：返回一个 Range 所在区域的整行（或多行）。

Range.EntireColumn：返回一个 Range 所在区域的整列（或多列）。

ActiveCell.EntireColumn.Cells(1, 1).Value = 5 **‘对包含活动单元格的列中的第一个单元格赋值。**

ActiveCell.EntireRow.Cells(1, 1).Value = 5 **‘对包含活动单元格的列中的第一个单元格赋值。**

[a1].EntireRow.Select **‘选择第1行**

[a1].EntireColumn.Select **‘选择第A列**

[a1:a4].EntireRow.Select **‘选择1至4行**

[a1:d4].EntireColumn.Select **‘选择A至D列**

删除A1：A10为空值的整行

Dim rng As Range, ads As String

For Each rng In [a1:a10]

addr = rng.Address

If rng = "" Then ad = ad & rng.Address & ","

Next

ads = Left(ad, Len(ad) - 2)

Range(ads).EntireRow.Delete

### Range.specialcells: 返回与指定类型和值匹配的所有单元格

表达式.SpecialCells(Type, Value)

Type：必选，指定类型

Value：可选

选定工作表 Sheet1 中已用区域的最后一个单元格。

Worksheets("Sheet1").Activate

ActiveSheet.Cells.SpecialCells(xlCellTypeLastCell).Activate

汇总计算所选区域批注个数

Msgbox Application.Sum(Selection.SpecialCells(-4144))

删除空行

On Error GoTo 100

Selection.SpecialCells(xlCellTypeBlanks).Select

Selection.EntireRow.Delete

Exit Sub

100:

MsgBox "没有空行"

### Range.Find:在区域中查找特定信息

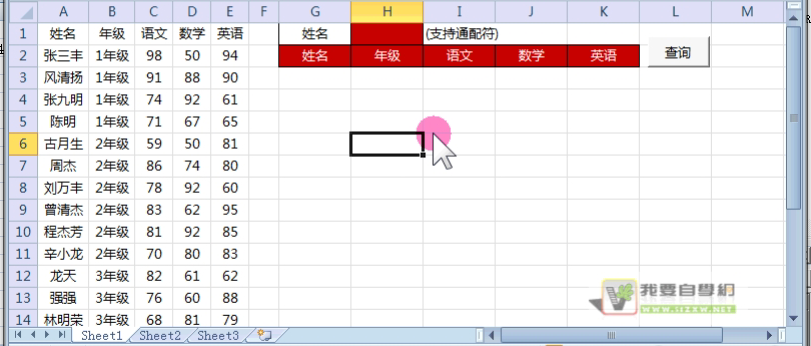
如果未发现匹配项，则返回 Nothing。Find 方法不影响选定区域或当前活动的单元格

动态选择

查找最后一格单元格，选定A1至最后一个单元格的区域

Set endrng = Cells.Find("\*", , , , , xlPrevious) **‘设置endrng为一格Range对象**

Range([a1], endrng).Select



单元格查询

ends = Columns(1).Find("\*", , , , , searchdirection:=xlPrevious).Row **‘找到A列最后一格单元格的行号**

Range("g3:1999").Clear

For Each rng In Range("a2:a" & ends)

m = m + 1

If rng Like Range("h1") Then

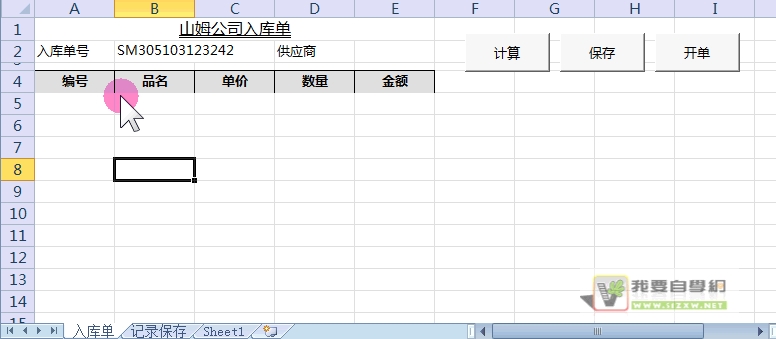
k = k + 1

Range("a" & m + 2 & ":e" & m + 1).Copy Range("g" & k + 2)

End If

Next

**实例：入库单**



计算

Set es = Columns(3).Find("\*", , xlFormulas, , , xlPrevious) **‘找到C列最后一格单元格**

For Each rng In Range([c5], es)

rng.Offset(0, 2) = rng.Offset(0, 1) \* rng

Next

开单

Set es = Cells.Find("\*", , xlFormulas, , , xlPrevious) **‘找到整个表的最后一格单元格**

a = es.Address

[b2] = "sm" & Format(Now(), "ymdhms")

Range([a5], es.Offset(4)) = ""

e2 = ""

保存

On Error GoTo 100

Dim es As Range, a%

If sheets2.[f:f].Find([b2]) = [b2] Then **‘如果表2F列中单元格和表1 B2单元格相同则弹出提示框**

MsgBox "已经保存过了"

Else

100:

Set es = Cells.Find("\*", , xlFormulas, , , xlPrevious)

a = Application.CountA(Sheet2.[a:a])

If es.Row = 4 Then MsgBox "没有填写美容": End

Range([a5], es).Copy Sheet2.Cells(a + 1, 1) **‘复制数据**

Sheet2.Cells(a + 1, "f").Resize(es.Row - 4) = [b2] **‘保存入库单号**

Sheet2.Cells(a + 1, "g").Resize(es.Row - 4) = [e2] **‘保存供应商**

Sheet2.Cells(a + 1, "h").Resize(es.Row - 4) = Now() **‘保存日期**

MsgBox "保存成功"

End If

### Range.End区域尾端的单元格

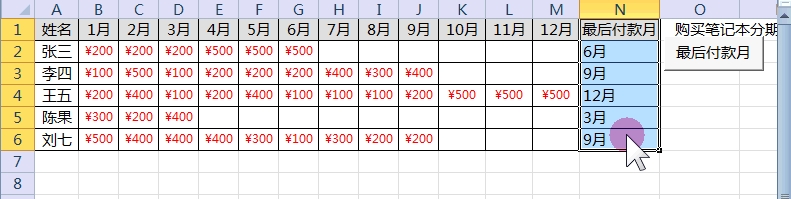
i = Cells(Rows.Count, 3).End(xlUp).Row **‘计算出第三列最底端的单元格行号**

j = Cells(Rows.Count, 3).End(xlUp).Address **‘计算出第三列最底端的单元格的地址**

Range("a1", j).Select

Range("a1", Cells(i, 3)).Select

**实例：查找分期付款最后月份**



i = Cells(Rows.Count, 1).End(xlUp).Row **'找到A列最后一行的行号**

Range("n2", Cells(i, "n")) = "" **'清空最后付款月所在单元格**

For j = 2 To i

k = Cells(j, "n").End(xlToLeft).Column **'找到最后付款月所在的列号**

Cells(j, "n") = Cells(1, k) **'将相应月份填入单元格**

Next j

### 查找最后一个常规值

方法：

a = Cells(Rows.Count, 1).End(xlUp).Row  **'end属性**

b = Columns(1).Find("\*", , , , , xlPrevious).Row **'find方法**

c =Cells.SpecialCells(xlCellTypeLastCell).Row **'specialcells方法**

d = Sheet1.UsedRange.Rows.Count **'usedrange属性**

e = [a1].Currenttegion.Rows.Count **'Currenttegion属性，非连续区域不能用**

f = WorksheetFunction.CountA([a:a])  **'工作表函数counta，非连续区域不能用**

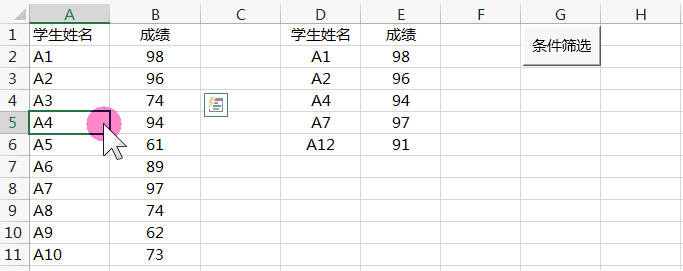
g = Application.CountIf([a:a], "<>")  **'工作表函数countif，非连续区域不能用**

### 并集单元格区域Application.Union

Range("a1:b3,c5:d8").Select **‘选择A1：B3和C5：D8的区域**

Union([a1:b3], [c5:d8]).Select **‘同上效果相同，但range引用不能超过256个字符，union无限制**

实例：筛选成绩大于90分的学生



For Each rng In Range([b2], Cells(Application.CountA([a:a]), 2))

If rng > 90 Then

k = k + 1

If k = 1 Then

Set rn = rng

Else

Set rn = Union(rn, rng) **‘使用Union将符合条件的单元格集中到rn对象**

aa = rn.Address

End If

End If

Next

For Each rngs In rn

n = n + 1

Cells(n + 1, "d") = Cells(rngs.Row, "a")

Cells(n + 1, "e") = rngs

Next

### 并集单元格区域Application.Intersect

返回一个 Range 对象，该对象表示两个或多个区域**重叠**的矩形区域。

判断已选区域是否在A1：D10区域，并返回交集区域的地址

If Intersect([a1:d10], Selection) Is Nothing Then

**‘Is Nothing用于判断对象是否存在，**

MsgBox "没有交集"

Else

MsgBox Intersect([a1:d10], Selection).Address

Intersect([a1:d10], Selection).Select

End If

隔行插入行

For i = 0 To Application.CountA(Columns(1)) \* 2 Step 2

Intersect([a1:d2].Offset(i), [a2:d3].Offset(i)).EntireRow.Insert

### 单元格格式设置——数字：Range.NumberFormatLocal

获取单元格的格式代码

For Each rng In [a1:a3]

Cells(rng.Row, 2) = rng.NumberFormatLocal

Next rng

设置单元格格式

For Each rng In [a1:a3]

rng.NumberFormatLocal = "0.00"

### 单元格格式设置——字体：Range.Font

清除格式设置

Range.ClearFormats

With [a2:a6].Font

.Name = "微软雅黑" **'字体**

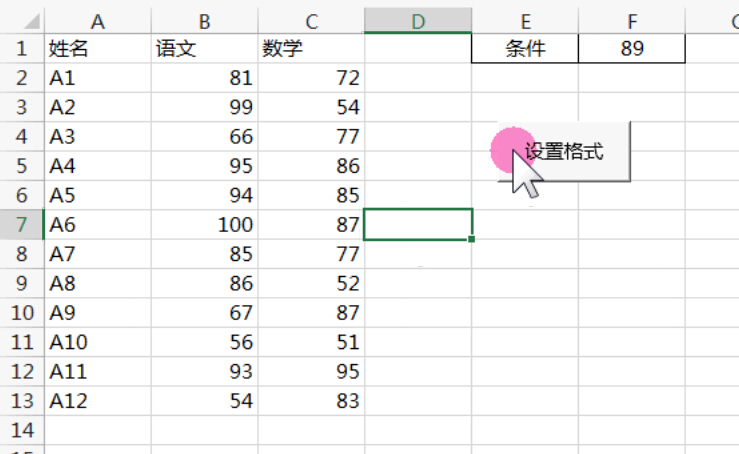
.Size = 8 **'字号**

.Bold = True **'加粗**

.Color = RGB(255, 0, 255) **'颜色**

End With

**实例：大于90分的单元格颜色设置为红色**



Set i = Cells(Rows.Count, 3).End(xlUp)

Range([b2], i).ClearFormats **‘清除格式设置**

For Each rng In Range([b2], i)

If rng.Value >= [f1] Then

With rng.Font

.Name = "华文琥珀"

.Size = 20

.Bold = True

.Color = RGB(255, 0, 0)

End With

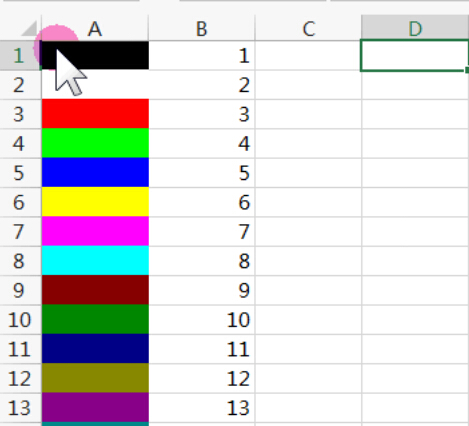
End If

Next rng

### 单元格格式设置——填充：Interior

代表一个对象的内部

获取颜色值



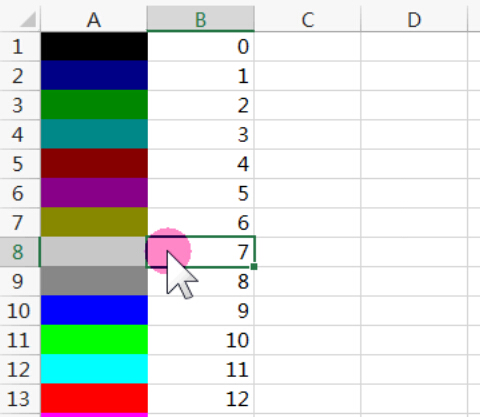
For i =1 to 56

Cells(i, 1).interior.colorindex=i

Cells(i, 2)=i

Next i

早期颜色值



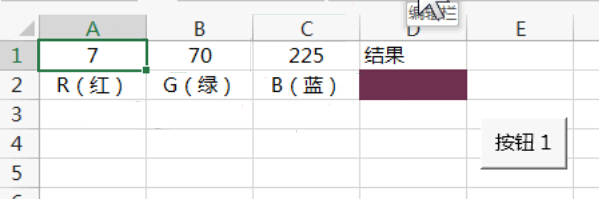
For i = 0 To 15

Cells(i + 1, 1).Interior.Color = QBColor(i)

Cells(i + 1, 2) = i

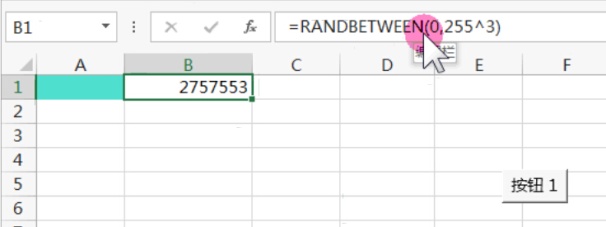
Next i

三原色颜色值



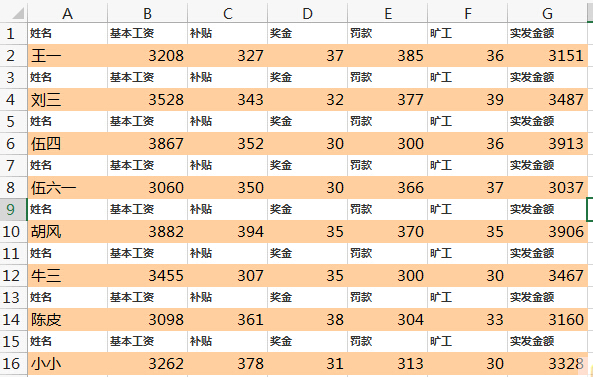
Cells(2, 4).Interior.Color = RGB([a1], [b1], [c1])

直接颜色值



Cells(1, 1).Interior.Color = [b1] **'在B1单元格使用函数RANDBETWEEN(0,255^3)获取0至255的三次方之间随机数，按F9刷新**

**实例：格式化工资条**



Dim i%

i = Cells(Rows.Count, 1).End(xlUp).Row

For j = 1 To i

If j Mod 2 Then **'函数Mod 2返回除2后的余数，如果为1则执行Then，如果为0则执行Else**

With Cells(j, 1).EntireRow.Range("a1:g1").Font **'Cells(j,1)所在整行中的A1：G1区域的字体**

.Bold = True

.Size = 8

.ColorIndex = 56

End With

Else

With Cells(j, 1).EntireRow.Range("a1:g1").Interior **'Cells(j,1)所在整行中的A2：G2区域的字体，这里使用了相对引用，虽然代码指的是A1：G1，但实际上相对引用了A2：G2**

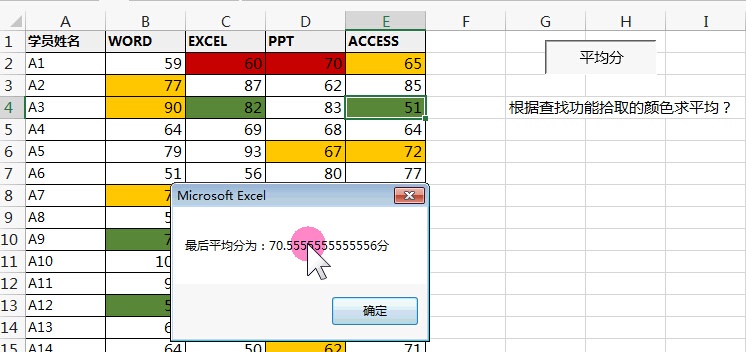
.ColorIndex = 40

End With

End If

Next j

**实例：查找相同颜色单元格**



On Error GoTo 100

Dim erng As Range, rng As Range, i As Long

i = Application.FindFormat.Interior.Color **‘取得颜色值，该颜色值使用Ctrl+F的格式选取确定**

Set erng = Cells(Rows.Count, "e").End(xlUp)

For Each rng In Range([b2], erng)

If rng.Interior.Color = i Then k = k + rng.Value: n = n + 1

Next

MsgBox "最后平均分为：" & k / n & "分"

End

100:

MsgBox "查找功能没有拾取到颜色"

### 粘贴：Expression.Paste(Destination, Link)

将“剪贴板”中的内容粘贴到工作表上

复制的区域没有公式

Range("b1:b6").Copy Range("c9")

Range("b1:b6").Copy

Sheet1.Paste Range("c9") **'效果和以上等价**

复制的区域没有公式

Range("c1:c6").Copy

Sheet1.Paste **'如果不指定destination参数，则使用该方法之前必须选择目标单元格**

Range("c1:c6").Copy

Sheet1.Paste , True **'以链接的形式粘贴**

Application.CutCopyMode = True **'将复制/移动的高亮光标去除**

### 选择性粘贴：Range.PasteSpecial

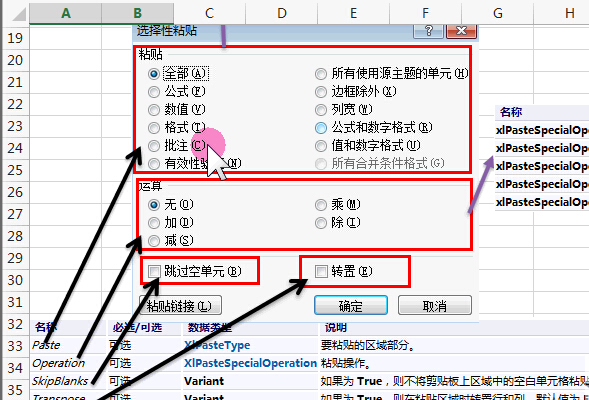
PasteSpecial(*Paste, Operation, SkipBlanks, Transpose*)

*Paste*：要粘贴的区域部分。

*Operation：*粘贴操作，计算。

*SkipBlanks ：*如果为 True，则不将剪贴板上区域中的空白单元格粘贴到目标区域中。默认值为 False。

*Transpose：*如果为 True，则在粘贴区域时转置行和列。默认值为 False。



粘贴值和数字格式

Range("c2:c10").Copy

Range("d2").PasteSpecial 12

Application.CutCopyMode = True **'将复制/移动的高亮光标去除**

粘贴运算

Range("b2:b9").Copy

Range("d2").PasteSpecial , 2 **'复制的数据与目标单元格中的值相加**

Range("c2:c9").Copy

Range("d2").PasteSpecial , 2

Application.CutCopyMode = True

粘贴转置

Range("a2:b9").Copy

Range("a11").PasteSpecial 12, , , True

Application.CutCopyMode = True

跳过空单元格

Range("b2:b9").Copy

Range("a15").PasteSpecial , , True

### （解除）合并单元格：Range.Merge、Range.MergeArea、Range.UnMerge

Range.Merge：由指定的 Range 对象创建合并单元格。

* Range.MergeArea：返回一个 Range 对象，该对象代表包含指定单元格的合并区域。如果指定的单元格不在合并区域内，则该属性返回指定的单元格。

Range.UnMerge：将合并区域分解为独立的单元格。

Range.MergeCells：如果区域包含合并单元格，则为 True。

Selection.Merge **‘对选择的区域进行合并**



Dim er%, rng%, rg As Range

Application.DisplayAlerts = False **‘关掉弹出警告窗口**

er = Application.CountA([a:a])

For rng = er To 2 Step -1

Set rg = Range("a" & rng)

If rg = rg.Offset(-1) Then rg.Offset(-1).Resize(2).Merge

Next

Application.DisplayAlerts = True **‘恢复弹出警告窗口**

判断A3单元格是否合并的单元格，然后解除合并

With Range("a3")

If .MergeCells Then

.MergeArea.UnMerge

Else

MsgBox "not merged"

End If

End With

解除单元格合并



Dim b%, rng As Range

For Each rng In Selection

ad = rng.Address

b = rng.MergeArea.Count

rng.UnMerge

rng.Resize(b) = rng

next

### 单元格批注：Comment

Range. AddComment.Text ”ABC” 添加内容为ABC的批注

Range.Comments.Visible = True / False 显示/隐藏批注

Worksheet.Comments(index) 可返回 Comments 集合中指定的单条批注（其中 index 为批注号）

Range.ClearComments 清除批注

Range.Comment.Shape.Fill.UserPicture ThisWorkbook.Path & "\pic.png" 修改批注框背景

先判断A1是否有批注，如果没有就添加内容为“123”的批注

With [a1]

If .Comment Is Nothing Then

.AddComment.Text "123"

.Comment.Visible = True

End If

End With

先判断所选区域是否有批注，如果有就删除批注

For Each rng In Selection

If Not rng.Comment Is Nothing Then

rng.ClearComments

End If

Next

将A2：A20区域大于90分的单元格添加内容为优秀的批注

For Each rng In Range("a2:a20")

rng.ClearComments

If rng >= 90 Then rng.AddComment.Text "优秀"

Next

修改批注框

Range("a2").AddComment

[a2].Comment.Shape.Height = 50 **'设置批注高度**

[a2].Comment.Shape.Width = 40 **'设置批注宽度**

[a2].Comment.Shape.Fill.UserPicture ThisWorkbook.Path & "\pic.png" **'将批注背景变更为指定图片**

批量增加批注框背景

On Error Resume Next

For Each rng In Selection

paths = ThisWorkbook.Path & "\pic" & rng.Value & ".png"

rng.ClearComments

rng.AddComment

rng.Comment.Shape.Height = 50 **'设置批注高度**

rng.Comment.Shape.Width = 40 **'设置批注宽度**

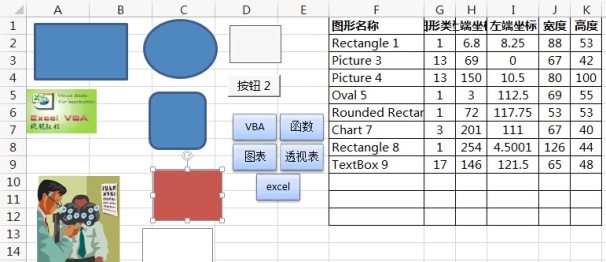
rng.Comment.Shape.Fill.UserPicture paths

Next

# 图形Shapes

每个 Shape 对象都代表绘图层中的一个对象，如自选图形、任意多边形、OLE 对象或图片

读取工作表中图形的信息



Dim ob As Shape

n = Sheet3.Shapes.Count

For Each ob In Sheet3.Shapes

k = k + 1

ob.Select

Cells(k + 1, "f") = ob.Name **'图形名称**

Cells(k + 1, "g") = ob.Type **'图形类型**

Cells(k + 1, "h") = ob.Top **'顶端坐标**

Cells(k + 1, "i") = ob.Left **'左端坐标**

Cells(k + 1, "j") = ob.Width **'宽度**

Cells(k + 1, "k") = ob.Height **'高度**

Next

### 插入图片Shapes.AddPicture

Shapes.AddPicture(Filename, LinkToFile, SaveWithDocument, Left, Top, Width, Height) 添加图片，7个参数是必须的

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **必选/可选** | **数据类型** | **描述** |
| *Filename* | 必选 | **String** | 要在其中创建 OLE 对象的文件。 |
| *LinkToFile* | 必选 | [**MsoTriState**](ms-help://MS.EXCEL.DEV.12.2052/EXCEL.DEV/content/HV10073109.htm) | 要链接至的文件。 |
| *SaveWithDocument* | 必选 | [**MsoTriState**](ms-help://MS.EXCEL.DEV.12.2052/EXCEL.DEV/content/HV10073109.htm) | 将图片与文档一起保存。 |
| *Left* | 必选 | **Single** | 图片左上角相对于文档左上角的位置（以磅为单位）。 |
| *Top* | 必选 | **Single** | 图片左上角相对于文档顶部的位置（以磅为单位）。 |
| *Width* | 必选 | **Single** | 图片的宽度（以磅为单位）。 |
| *Height* | 必选 | **Single** | 图片的高度（以磅为单位）。 |

图形插入

Sheet3.Shapes.AddPicture ThisWorkbook.Path & "\pic1.png", \_

True, True, 100, 100, 70, 70

删除所有图形

For Each shp In Sheet3.Shapes

shp.Delete

Next

批量插入图片

For Each shap In Sheet3.Shapes **‘清除以往的图形**

If shap.Type <> 8 Then shap.Delete **‘参数8表示控件类型，即清除非控件的图形**

Next shap

For Each rng In Range([a2], Cells(Application.CountA(Columns(1)), 1))

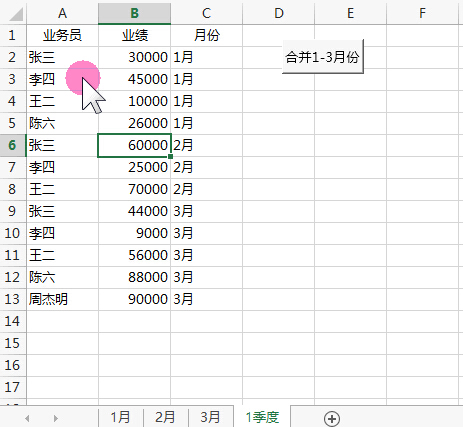
i = ThisWorkbook.Path & "\pic" & rng & ".png"

Set rngs = Cells(rng.Row, 2)

Sheet3.Shapes.AddPicture i, True, True, rngs.Left, rngs.Top, rngs.Width, rngs.Height

Next

# 综合实例：多表合并



Dim i%, rs%, rss%, st As Worksheet, zst As Worksheet

Set zst = Sheets("1季度") **‘定义zst为工作表 “1季度”**

For i = 1 To 3

Set st = Sheets(i & "月") **‘定义st为工作表 1至3月**

rs = st.UsedRange.Rows.Count **‘计算1-3月每个表的最后一行**

rss = zst.UsedRange.Rows.Count + 1 **‘计算”1季度”表的最后一行**

st.Range("a2:b" & rs).Copy zst.Cells(rss, 1) **‘复制1-3月表的数据至1季度**

zst.Cells(rss, 3).Resize(rs - 1) = i & "月" **‘记录月份数**

Next

# 综合实例：多表拆分



For f = 1 To 3

Worksheets.Add.Name = f & "月" **‘新建工作表，以月份命名**

For Each rng In Sheets("1季度").Range("a2:a20")

If rng.Value = f & "月" Then **‘判断a列值是否等于当前月份**

n = "a" & rng.Row & ":d" & rng.Row **‘构造被复制的源表区域**

y = y + 1 **‘新表行数累计**

If y = 1 Then

Sheets("1季度").Range("a1:d1").Copy Sheets(f & "月").Cells(y, 1) **‘复制表头**

End If

Sheets("1季度").Range(n).Copy Sheets(f & "月").Cells(y + 1, 1)

End If

Next

y = 0 **‘分表行数累计归零**

Next

# 事件程序

工作表事件



改变选择时弹出对话框

Private Sub Worksheet\_SelectionChange(ByVal Target As Range)

MsgBox "123"

End Sub

改变选择时弹出对话框，显示选取的单元格的地址（使用括号内参数）

Private Sub Worksheet\_SelectionChange(ByVal Target As Range)

MsgBox Target.Address

End Sub

# 数组

### 定义数组

Dim arr1(3) **‘一维数组arr1 从0开始到3，共4个**

Dim arr2(1 to 3) **‘一维数组arr2 从1开始到3，共3个**

Dim arr3(1 to 3,1 to 2) **‘定义二维数组，相当于3行2列的表格**

Dim arr4(3,2) **‘定义二维数组，相当于4行3列的表格**

**‘一行是一维数组，一列是二维数组**

arr1= [{“a”,”b”,”c”,”d”}]  **‘赋值一维数组arr1,相当于对一行赋值，使用“，”间隔**

arr2 =[{1;2;3;4}] **‘赋值二维数组arr2,相当于对一列赋值，使用“；”间隔**

arr3 = Application.Transpose([{1;2;3;4}]) **‘赋值一维数组arr3，效果与arr1相同，“；”定义一列数组，然后用Application.Transpose函数转置**

arr4= Application.Transpose([{1,2,3,4}])  **‘‘赋值二维数组arr3，效果与arr2相同，“，”定义一行数组，然后用Application.Transpose函数转置**

arr5 = [{"a",1;"b",2;"c",3}]  **‘赋值二维数组arr3，相当于对3行2列赋值**

### Array数组函数

arr1 = Array(1, 2, 3, 4)  **'使用函数Array赋值给一维数组arr1**

arr2 = Array(Array("a", "b"), Array(1, 2, 3)) **'使用函数Array赋值给二维数组arr2，相当于对2行3列赋值**

arr3 = Array(Array("a", "b"), Array(1, 2, 3), Array(2, 3, 4, 5)) **'使用函数Array赋值给二维数组arr3，相当于对4行3列赋值**

### 赋值

向指定位置赋值

Dim arr(1 To 9) **'声明数组**

arr(3) = "A" **'将3的位置赋值为A**

arr(5) = "B" **'将5的位置赋值为B**

arr(7) = "c" **'将7的位置赋值为c**

### 一维数组转化

一列数组使用一次application.transpose；

一行数组使用两次application.transpose；

多行多列无法转化。

arr = [a1:a4]  **‘将A1:A4的值写入数组，当写入多行时，结果是二维数组**

arr = Application.Transpose([a1:a4]) **‘使用application.transpose转换为一维数组**

arr1 = [a1:d1] **‘将A1:D1的值写入数组，该数组是二维数组，相当于1行4列的表格**

arr2 = application.transpose(application.transpose([a1:d1])) **‘将arr1转置为一维数组，使用两侧application.transpose**

使用循环命令写入

#将A1：A4的单元格的值写入数组

Dim arr(1 To 4)

For Each Rng In [a1:a4]

n = n + 1

arr(n) = Rng

Next

向数组写入常量

Dim arr() **‘定义一个动态数组**

arr = Array("A", "B", 1, 2, 5)

### 读取值

arr = [a2:a5] **‘赋值**

MsgBox arr(2, 1) **‘使用消息框显示值**

MsgBox arr(4, 1)

使用循环命令

arr = [a2:a9]

[b1] = arr(2, 1)

For i = 1 To 8

Cells(i, 3) = arr(i, 1)

Next

将A2:A5的值赋到D1:D4区域

arr = [a2:a5]

Range("d1:d" & 4) = arr

转置，注意尺寸

arr = [a2:a5]

arr1 = Application.Transpose(arr)  **‘转置后为一维数组**

[a7:d7] = arr1 **‘将A2：A5的值复制到A7:D7**

[a8:c8] = arr1  **‘将A2：A4的值复制到A8:C8**

[a9:e9] = arr1 **‘将A2：A5的值复制到A9:D9，由于E9超过尺寸，所以显示“#N/A”的错误值**

[b1:b4] = arr1  **‘如果没有转置，只显示一维数组的第一组的值**

[c1:c4] = Application.Transpose(arr1) **‘使用转置将A2:A5的值复制到C1:C4**

### 上下标函数UBoud，LBound

UBoud 返回数组上界

UBound(arrayname[, dimension])

LBoud 返回数组下界

Dim arr(4 To 8, 1 To 3, 3 To 9)

MsgBox UBound(arr, 1) **'可简写为UBound(arr) 返回值是8**

MsgBox UBound(arr, 2) **'返回值是3**

MsgBox UBound(arr, 3) **'返回值是9**

MsgBox LBound(arr, 1) **'可简写为LBound(arr) 返回值是4**

实例：数据汇总

在数组中求和、平均、最大、最小、极大、极小值

arr = [E2:E19]

MsgBox WorksheetFunction.Sum(arr)

MsgBox WorksheetFunction.Average(arr)

MsgBox WorksheetFunction.Max(arr)

MsgBox WorksheetFunction.Min(arr)

MsgBox WorksheetFunction.Small(arr, 2) **‘求第二个最小值**

MsgBox WorksheetFunction.larger(arr, 2) **‘求第二个最大值**

实例：将期末大于等于80的进行



Dim arr1(1 To 99)

arr = [b2:c9]

For Each a In arr

If a >= 80 Then

n = n + 1

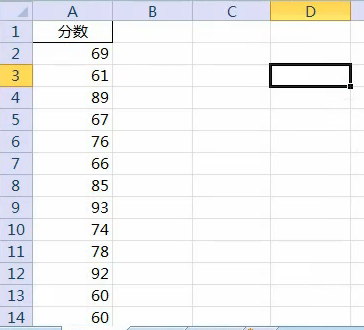
arr1(n) = a **‘使用arr1储存满足条件的值**

End If

Next

MsgBox WorksheetFunction.Average(arr1)

实例：将大于等于分数80的汇总到C列



方法一：非数组方法

t = Timer

Set Rng = Cells(Rows.Count, 1).End(xlUp)

For Each a In Range([a1], Rng)

If a >= 80 Then n = n + 1: Cells(n, 3) = a

Next

MsgBox Format(Timer - t, "0.0000")

方法二：非数组方法

t = Timer

Dim arr1(1 To 5000, 1 To 1)

Set Rng = Cells(Rows.Count, 1).End(xlUp)

arr = Range("a1", Rng) **‘将区域数据写入arr**

For Each a In arr

If a >= 80 Then

k = k + 1

arr1(k, 1) = a ‘将符合条件的数据写入数组arr1

End If

Next

[d1].Resize(k) = arr1 ‘arr1的数据写入D列

MsgBox Format(Timer - t, "0.0000")

### 实例：提取不重复值



Dim arr1(1 To 5000, 1 To 1)

Set lastcell = Cells(Rows.Count, 2).End(xlUp)

arr = Range("b2", lastcell) '将区域数据写入arr

For i = 1 To lastcell.Row - 1

For j = 1 To UBound(arr1, 1)

If arr(i, 1) = arr1(j, 1) Then

GoTo 111 '如符合条件则跳出循环

End If

Next j

k = k + 1 '计数器，计算不重复值数量

arr1(k, 1) = arr(i, 1) '将不重复的值赋给arr1

111:

Next i

[e2].Resize(k) = arr1 ‘arr1的数据写入e列

### 实例：提取不重复值并求和



Dim arr1(1 To 10, 1 To 2)

Set lastcell = Cells(Rows.Count, ”c”).End(xlUp)

arr = Range("b2", lastcell) '将区域数据写入arr

For i = 1 To lastcell.Row - 1

For j = 1 To UBound(arr1, 1)

If arr(i, 1) = arr1(j, 1) Then

arr1(j, 2) = arr(i, 2) + arr1(j, 2) '将姓名的工资进行累加

GoTo 111

End If

Next j

k = k + 1

arr1(k, 1) = arr(i, 1) '将不重复的值赋给arr1

arr1(k, 2) = arr(i, 2)

111:

Next i

[e2].Resize(k, 2) = arr1

### 动态数组与重新声明

ReDim [Preserve] varname(subscripts) [As type]

Preserve：使用此关键字可以保持数组中原来的数据。

Varname：变量的名称

Type：变量的数据类型

Dim arr()

i=9

arr = [{1,2,3}]

Redim Preserve arr(1 To 5) ‘保留数组原来的值，同时将数组上界扩展到5

Redim Preserve arr(1 To i) ‘保留数组原来的值，同时将数组上界扩展到i

Redim Preserve arr(1 To 2) ‘将数组上界减为2，同时保留相应的值

实例：提取大于等于80分的单元格

Dim arr(), arr1() ‘事前无法确定数组边界，先使用动态数组定义，后使用redim重新声明数组边界

rn = cells(Rows.Count, 1).End(xlUp).Address

arr1 = Range("a1", rn)

m = WorksheetFunction.CountIf(Range("a1", rn), ">=80") ‘调用工作表函数计算出大于等于80的个数

ReDim arr(1 To m + 1) ‘重新声明数组

For Each ar In arr1

If ar >= 80 Then

n = n + 1

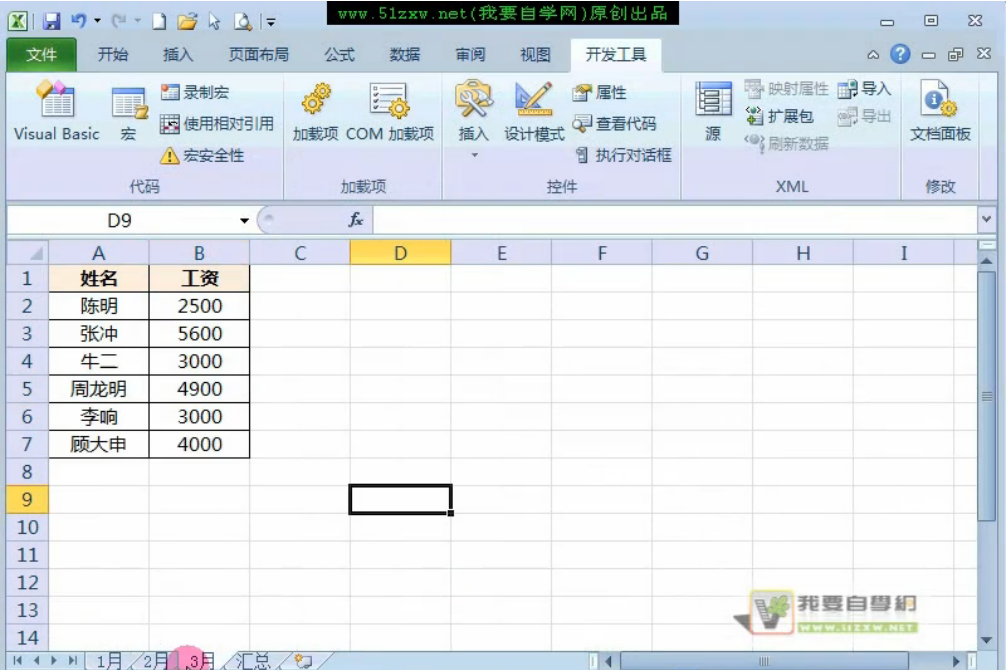
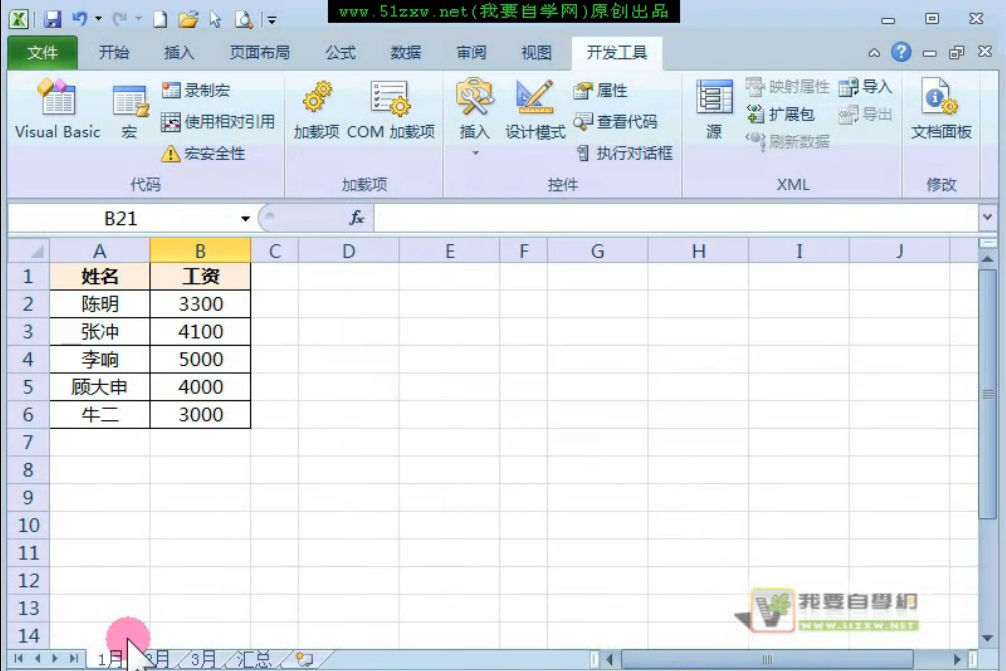
arr(n) = ar

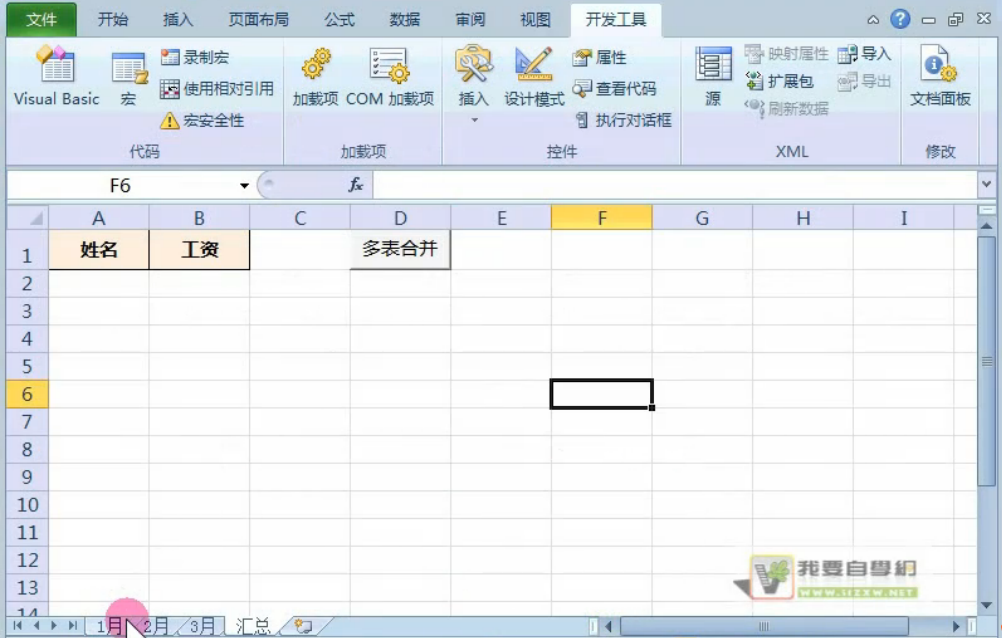
End If

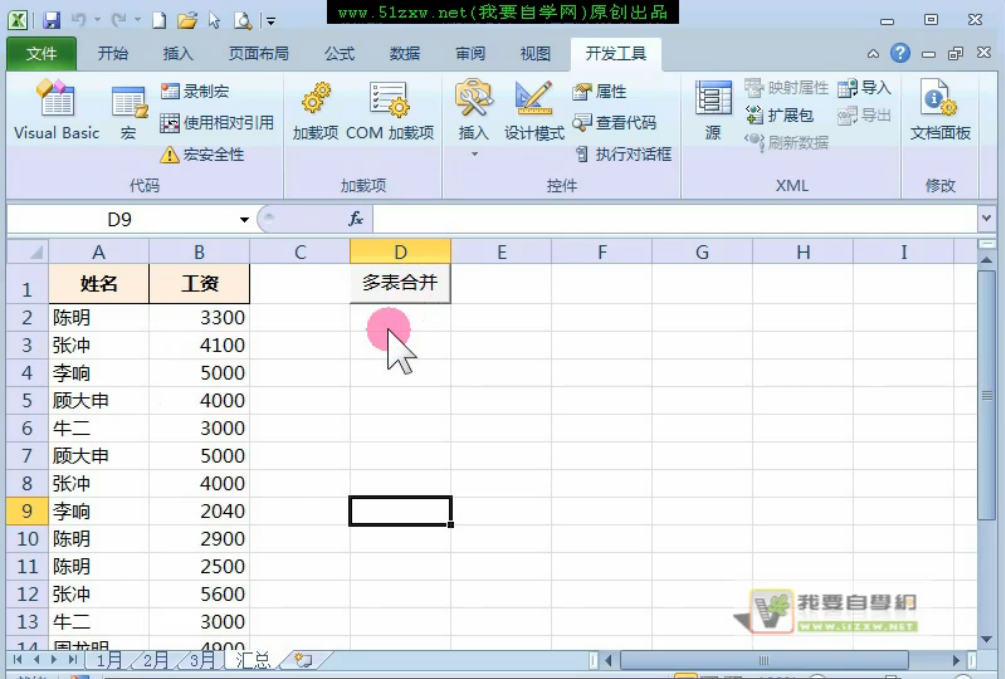
Next

[c1].Resize(UBound(arr)) = Application.Transpose(arr)

实例：多表合并







Dim arr()

For Each sh In Sheets

If sh.Name <> "汇总" Then ‘只对名称不是“汇总”的工作表进行循环

arr1 = sh.Range("a2:b" & sh.UsedRange.Rows.Count) ‘将各表的数据写入arr1

act = act + UBound(arr1) ‘累加各表的行数

ReDim Preserve arr(1 To 2, 1 To act) ‘重新声明arr，保留原来数据，重新定义数组上界，变量值act只能写在最后

For j = 1 To UBound(arr1)

n = n + 1

arr(1, n) = arr1(j, 1)

arr(2, n) = arr1(j, 2)

Next

End If

Next

Sheets("汇总").[a2].Resize(n, 2) = Application.Transpose(arr1)

### Split函数

将字符串拆分成一维数组

Dim i$

i = "a-b-c-d-e-f"

arr = Split(i, "-") ‘以“-”为分隔符对i进行拆分

[a1].Resize(1, UBound(arr)) = arr

实例：数据互换



arr = [a1].CurrentRegion

For Each a In arr

arr1 = Split(a, "-") ‘对每个单元格进行拆分

n = n + 1

Cells(n, "c") = arr1(1) & "-" & arr1(0)

Next

### Join函数

将一个一维数组生成一个字符串，（可以使用Application.Transpose(Application.Transpose(arr))转换成一维数组）。

Dim i$

arr = [{1,2,3,4}]

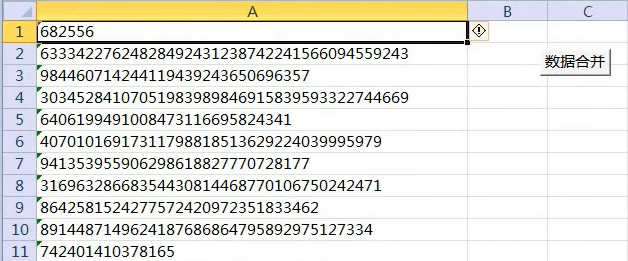
j = Join(arr) ‘生成的字符串是“1 2 3 4”，默认间隔符是空格符

i = Join(arr, ":") ‘生成的字符串是“1:2:3:4”，使用“:”分隔

k= Join(arr, "") ‘生成的字符串是“1234”

实例：数据合并





i = Sheet3.Cells(Rows.Count, 1).End(xlUp).Row

With Sheet3

For j = 1 To i

Set k = Range(.Cells(j, 1), .Cells(j, Columns.Count).End(xlToLeft))

arr = Application.Transpose(Application.Transpose(k))

Sheet4.Cells(j, 1) = "'" & Join(arr, "") ‘添加“’”，令单元格以字符格式显示，否则以科学记数格式显示。

Next j

End With

### Filter函数

对一个一维数组进行字符串筛选，返回一个符合指定条件的数组，下标从0开始

arr = [{"abc","bb","c","ba","dd","nba"}]

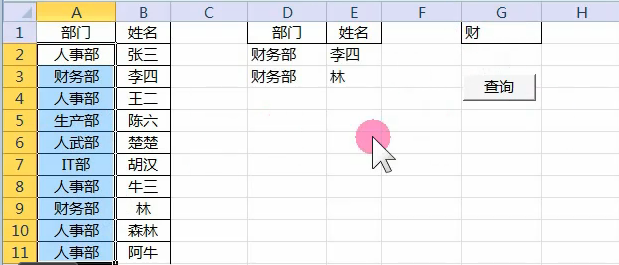
a = Filter(arr, "b", True) ‘筛选出包含“b”的值

a = Filter(arr, "b", 1) ‘效果同上

b = Filter(arr, "b", 0) ‘筛选出不包含“b”的值

b = Filter(arr, "b", 0) ‘效果同上

### 实例：数据查询（包含指定值）



[d2:f999].Clear '清楚单元格

i = Cells(Rows.Count, 1).End(xlUp).Row

Range("c2:c" & i).FormulaArray = "=a2:a" & i & " & ""\_"" & b2:b" & i

'将a列和b列合拼成一个字符串,代码中出现"，2次输入"

arr = Range("c2:c1" & i)

Range("c2:c1" & i).Clear

a = Filter(Application.Transpose(arr), [g1], True) '将arr转换成一维数组，然后进行筛选

For Each b In a

n = n + 1

c = Split(b, "\_") '将筛选结果进行拆分，随后赋值

Cells(n + 1, "d") = c(0)

Cells(n + 1, "e") = c(1)

Next

### 实例：查询系统（工作表函数+VBA数组）

Index函数

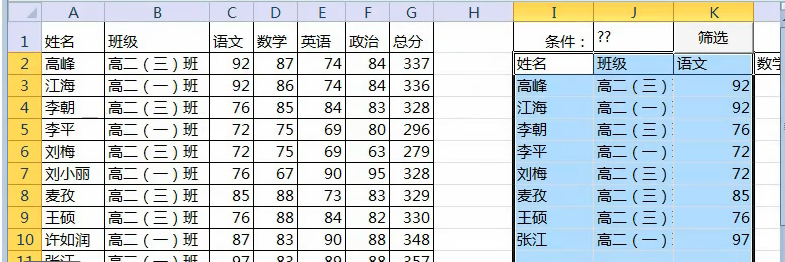
Index(区域/数组, row\_num, column\_num)

如果将 row\_num 或 column\_num 设置为 0（零），INDEX 将分别返回对整列或整行的引用。

arr = [a2:g13]

arr1 = WorksheetFunction.Index(arr, 0, 2) ‘提取数组arr第2列

arr2 = WorksheetFunction.Index(arr, 3, 0) ‘提取数组arr第3行



[i3:o999].Clear

arr = Range("a2", Cells(Rows.Count, "g").End(xlUp))

For i = 1 To UBound(arr)

If arri(i, 1) Like [j1] Then

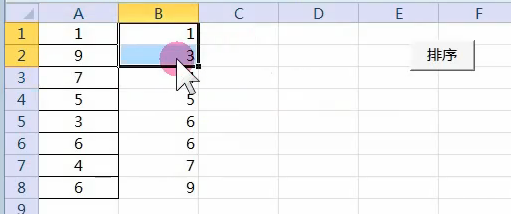
n = n + 1

Cells(n + 2, "i").Resize(1, 7) = WorksheetFunction.Index(arr, i, 0)

End If

Next

实例：排序



arr = Selection

For i = 1 To UBound(arr)

For j = i + 1 To UBound(arr)

If arr(i, 1) > arr(j, 1) Then

k = arr(i, 1)

arr(i, 1) = arr(j, 1)

arr(j, 1) = k

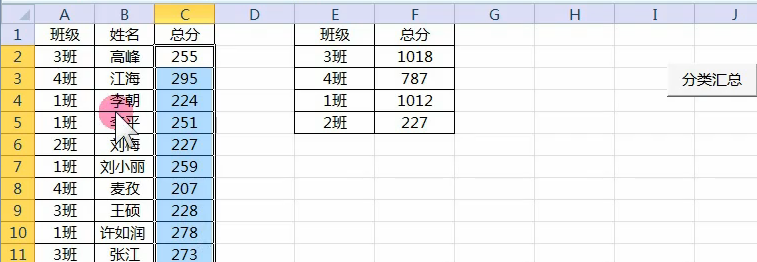
End If

Next

Next

[b1].Resize(UBound(arr)) = arr

实例：分类汇总



Dim arr1() ‘先声明一个动态数组

arr = [a2:c13]

For i = 1 To UBound(arr)

ReDim Preserve arr1(1 To 2, 1 To n + 1) ‘当下面if判断成立时，再增加数组第二维数量

For j = 1 To UBound(arr1, 2)

If arr1(1, j) = arr(i, 1) Then

arr1(2, j) = arr1(2, j) + arr(i, 3) ‘当下面if判断成立，累计总分

GoTo 100

End If

Next

n = n + 1

arr1(1, n) = arr(i, 1)

arr1(2, n) = arr(i, 3)

100:

Next

[e2].Resize(n, 2) = Application.Transpose(arr1)

# 字典

### 创建字典

前期绑定字典对象：工具 – 引用 – 添加Microsoft Scripting Runtime(Scrrun.dll文件)

后期绑定对象：

Set d = CreateObject("scripting.dictionary")

### ADD添加关键字项目

Object.add key,item

Object：必选项，Dictionary对象的名称

Key：必选项，与item相关联的key，必须是唯一值，不能与现存的key重复，否则发生错误。

Item：必选项

Set dd = CreateObject("scripting.dictionary") ‘生成一个字典对象

dd.Add "ABC", "123" ‘新增一个Key为“ABC”，Item为123的值，在本地窗口中不能看到Item的值。

dd.Add "abc", "一二三"

### Keys获取Dic中所有Key

Object.Keys

返回的是一个一维数组。

Dim d As New Dictionary

d.Add "ABC", "123"

d.Add "abc", "一二三"

方法1：前期绑定Dic对象时，后期绑定无效

i = d.Keys() ‘将d的所有Key值以数组形式赋值给i,返回的i值是一个数组

ii = d.Keys(0) ‘将d的第一个Key值赋值给ii，ii=“ABC”，维度从0开始计算，因此此处参数为0

iii = d.Keys(4) ‘由于不存在第5嘅Key值，会显示下标越界错误

方法2：函数方式

j = Application.Index(d.Keys, 2) ‘i = “abc”，维度从1开始，此处参数为2

方法3：变量方式

k = d.Keys

l = k(1) ‘i = “abc”

### Items获取所有Item

返回一个一维数组

Dim d As New Dictionary

d.Add "ABC", "123"

d.Add "abc", "一二三"

方法1：前期绑定Dic对象时，后期绑定无效

r = d.Items(1) ‘r="一二三"

方法2：函数方式

s = Application.Index(d.Items, 2)

方法3：Key值方式

t = d("abc")

tt= d("abc1") ‘由于没有abc1的key，返回的是空值

方法4：变量方式

w = d.Items

v = w(1)

### Exists方法

判断Dic中是否存在指定Key值，存在返回True，否则为False

Dim d As New Dictionary

d.Add "ABC", "123"

d.Add "abc", "一二三"

d.Add "aaa", "1314"

a = d.Exists("abc") 'a=True

aa = d.Exists("abc1") 'a=False

### Remove/ RemoveAll方法

Remove：从一个Dic中移除一个Key和Item

Dim d As New Dictionary

d.Add "ABC", "123"

d.Add "abc", "一二三"

d.Add "aaa", "1314"

d.remove(“abc”)

d.remove(“bbb”) ‘由于bbb不存在，会显示错误

If d.Exists("aaa") Then ‘先用exist判断，后remove

d.Remove ("aaa")

End If

RemoveAll：清楚一个dic中的所有关键字

d.RemoveAll

### 字典属性

CompareMode属性：对Dic中是否对Key进行大小写比较，默认值是0，区分大小写。1是区分大小写。需要在Dic中还没有数据时设置，否则报错。

Dim d As New Dictionary

i = d.CompareMode ‘返回默认值0，区分大小写

d.CompareMode = 1 ‘修改参数为1，不区分大小写

d.Add "ABC", "123"

d.Add "abc", "一二三" ‘由于不区分大小写，abc等同ABC，和已存在的Key冲突，显示报错

d.Add "aaa", "1314"

Count属性：计算当前Dic的非空值Item数量

Dim d As New Dictionary

d.Add "ABC", "123"

d.Add "abc", "一二三"

d.Add "aaa", "1314"

d.Add "aa", "" ‘添加的item为空值

j = d.Count ‘返回值为4

Key属性：修改一个Key值

Dim d As New Dictionary

d.Add "ABC", "123"

d.Add "abc", "一二三"

d.Add "aaa", "1314"

d.Key("ABC") = "CBA" ‘将已存在的“ABC”Key值修改为“CBA”

Item属性：修改一个Item值

Dim d As New Dictionary

d.Add "ABC", "123"

d.Add "abc", "一二三"

d.Add "aaa", "1314"

d.Item("ABC") = "321" ‘修改“ABC”Key的Item值为321

t = d("ABC") ‘读取“ABC”的新item值，t=321

d("ABC") = "123" ‘简写，修改“ABC”Key的Item值为123

t = d("ABC") ‘读取“ABC”的新item值，t=123

### 向字典写入数据

将姓名和计件写入字典，其中姓名为Key，计件为Item



Dim d As New Dictionary

On Error Resume Next ‘当遇到错误时程序继续运行，特别是当遇到重复Key值时

arr = Range("a2:b" & Cells(Rows.Count, 2).End(xlUp).Row)

For i = 1 To UBound(arr)

d.Add arr(i, 1), arr(i, 2)

Next

i = d.Keys ‘输入

j = d.Items

## End

With ActiveSheet.QueryTables.Add(Connection:= \_

"URL;http://t.qq.com/yesense?mode=0&id=340957093903587&pi=4&time=1384376204", \_

Destination:=Range("$F$4"))

.Name = "yesense"

.FieldNames = True

.RowNumbers = False

.FillAdjacentFormulas = False

.PreserveFormatting = False

.RefreshOnFileOpen = False

.BackgroundQuery = True

.RefreshStyle = xlInsertDeleteCells

.SavePassword = False

.SaveData = True

.AdjustColumnWidth = True

.RefreshPeriod = 0

.WebSelectionType = xlEntirePage

.WebFormatting = xlWebFormattingRTF 'RTF

.WebPreFormattedTextToColumns = True

.WebConsecutiveDelimitersAsOne = True

.WebSingleBlockTextImport = False

.WebDisableDateRecognition = False

.WebDisableRedirections = False

.Refresh BackgroundQuery:=False

End With

End Sub