

## 16. Range单元格进阶知识

2020年4月24日 20:53

```
Sub a()
```

```
Dim i
```

```
i=2
```

```
Do while range("A"&i)<>"" and i<1048576
```

```
    i=i+1
```

```
loop
```

```
End sub
```

你有没有想过，所有所有的行全占满了，你的程序会报错？  
用佩丰老师教的For循环找到某例最后一个非空单元格也不行  
因为全占满了，就没有非空单元格了

旧版是65536

## 16.1 Rows、Columns、Address、Count

2020年4月25日 10:25

### 与Range对象位置有关的属性:

Range.Rows	单元格行号
Range.Clounms	单元格列号
Range.Address	各对角顶点的绝对引用地址
Range.Count	单元格数量

---

```
Sub a()  
    ' 选中活动工作表第3行  
    ' ActiveSheet.Rows(3).Select  
    ' 选中G3:G10区域的3行5列  
    ' Range("C3:G10").Rows("3:5").Select  
    ' 使用MsgBox显示C3:G10区域第2行的地址  
    Set r = Range("C3:G10")  
    ' MsgBox "第2行范围是: " & r.Rows(2).Address  
    MsgBox "一共有多少" & r.Rows.Count & "行"  
  
End Sub
```

## 16.2 Cells与UsedRange

2020年4月25日 10:48

1. 代表工作表中全部单元格的Range对象 【Cells】
2. 涵盖工作表中全部使用过的单元格的Range对象 【UsedRange】

```
Sub a()  
Dim i  
i=2  
Do while range("A"&i)<>"" and i < ActiveSheet.Cells.Rows.Count  
    i=i+1  
loop  
End sub
```



活动工作表，全部范围，所有行，总数

无论在哪个版本的Excel都适用

	A	B	C	D
1	1	A	B	C
2	2	A	B	C
3	3	A	B	C
4	4	A	B	C
5	5	A	B	C
6				
7	6	A	B	C
8	7	A	B	C
9				
10	8	A	B	C
11	9	A	B	C

```
Sub a()  
    Dim r1, i  
    '当前工作表所使用的区域  
    Set r1 = ActiveSheet.UsedRange  
    'Row从第几行开始+总计多少行-1就得到最后一行的位置了  
    i = r1.Row + r1.Rows.Count - 1  
    MsgBox "最后一行是" & i  
End Sub
```

	A	B	C	D
1	1	A	B	C
2	2	A	B	C
3	3	A	B	C
4	4	A	B	C
5	5	A	B	C
6				
7	6	A	B	C
8	7	A	B	C
9				
10	8	A	B	C
11	9	A	B	C
12				
13				
14				
15				

```
Sub for循环()  
Dim i, r1  
Set r1 = ActiveSheet.UsedRange  
For i = 1 To r1.Row + r1.Rows.Count - 1  
    Range("G" & i) = 520  
Next  
End Sub
```

## 16.3 【推荐】取得最后一个非空单元格

2020年4月25日 11:21

可设置的参数	参数说明
xlToLeft	End+左方向键
xlToRight	End+右方向键
xlUp	End+上方向键
xlDown	End+下方向键

**Sub 取某列中最后一个非空单元格()**

**Dim r**

**'Range("A" & Rows.Count)获取工作表第一列最后一个单元格**

**'通过End属性获得A列最后一个非空单元格**

**Set r = Range("A" & Rows.Count).End(xlUp)**

**MsgBox r.Row**

**End Sub**

**Sub 取某行中最后个非空单元格()**

**Dim r**

**Set r = Cells(1, Columns.Count).End(xlToLeft)**

**MsgBox r.Column**

**End Sub**

**Sub 利用到for循环()**

**Dim r, i**

**Set r = Range("A" & Rows.Count).End(xlUp)**

**For i = 1 To r.Row**

**'这里写For循环要执行的代码**

**Next**

**End Sub**

## 16.4 往单元格里写数据，建议配合数组

2020年4月25日 11:55

**Sub for循环()**

**Dim i**

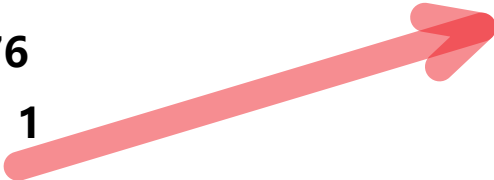
**For i = 1 To 1048576**

**Range("A" & i) = 1**

**Next**

**End Sub**

3到5分钟



**Sub 配合数组使用()**

**Dim arr(), i**

**arr = Range("A1:A1048576")**

**For i = 1 To 1048576**

**arr(i, 1) = 1**

**Next**

**Range("A1:A1048576") = arr**

**MsgBox "已经完成"**

**End Sub**

3到5秒钟



Sub 配合数组使用 ()

Dim arr(), i

arr = Range("A1:A1048576")

For i = LBound(arr, 1) To UBound(arr, 1)

    arr(i, 1) = 1

Next

Range("A1:A1048576") = arr

MsgBox "已经完成"

End Sub

## 16.5 选择性粘贴

2020年4月25日 12:24

	A	B	C	D	E	F
1						
2		1	1	1	1	1
3		1	1	1	1	1
4		1	1	1	1	1
5		1	1	1	1	1
6		1	1	1	1	1
7		1	1	1	1	1
8		1	1	1	1	1
9		1	1	1	1	1
10						

**Sub 单元格拷贝()**

**Range("B2").CurrentRegion.Copy Sheets("Sheet2").Range("A1")**

**End Sub**

**你们检查一下，列宽一致吗？**

## 16.5.1 Copy单元格保证列宽不变

2020年4月25日 12:48

### Sub 单元格拷贝()

Range("B2").CurrentRegion.Copy

With Sheets("Sheet2").Range("A1")

'使用Range对象的PasteSpecial方法选择性粘贴剪贴板中Range对象的列宽

.PasteSpecial xlPasteColumnWidths

'粘贴剪贴板中Range对象全部内容

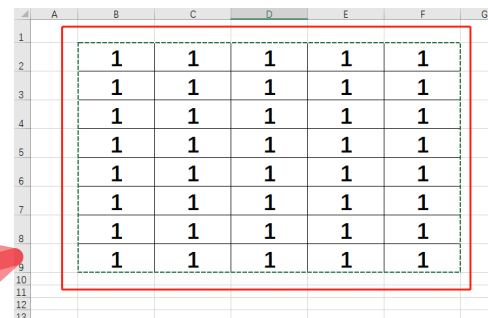
.PasteSpecial xlPasteAll

End With

'取消应用程序复制模式，如果没有关闭，如图所示

Application.CutCopyMode = False

End Sub



1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	1
7	1	1	1	1	1
8	1	1	1	1	1
9	1	1	1	1	1
10					
11					
12					
13					

---

Sub 单元格拷贝()

Range("B2").CurrentRegion.Copy

With Sheets("Sheet2").Range("A1")

.PasteSpecial xlPasteValues

仅复制数值

End With

'取消应用程序复制模式，如果没有关闭，如图所示

Application.CutCopyMode = False

End Sub



## 16.5.2 【选择性粘贴参数】PasteSpecial 方法

2020年4月25日 12:48

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

- 参数1: Paste XlPasteType 类型, 可选。指定要粘贴的区域部分
- 参数2: Operation XlPasteSpecialOperation 类型, 可选。指定粘贴操作
- 参数3: SkipBlanks 指示是否跳过空单元格, 默认值为False, 若参数值为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
- 参数4: Transpose 指示是否进行转置, 默认值为False, 若参数值为True, 则粘贴区域时转置行和列

参数1: XlPasteType 类型

常量	说明
xlPasteAll <b>【不包含列宽】</b>	全部 (默认)
xlPasteAllExceptBorders	边框除外
xlPasteColumnWidths	列宽
xlPasteComments	批注
xlPasteFormats	格式
xlPasteFormulas	公式
xlPasteFormulasAndNumberFormats	公式和数字格式
xlPasteValidation	有效性验证
xlPasteValues	数值
xlPasteValuesAndNumberFormats	值和数字

参数2: XlPasteSpecialOperation类型

常量	说明
xlPasteSpecialOperationNone	无 (默认)
xlPasteSpecialOperationAdd	加
xlPasteSpecialOperationSubtract	减
xlPasteSpecialOperationMultiply	乘
xlPasteSpecialOperationDivide	除



### 16.5.3 直接赋值，速度快【但是无法保留原数据格式】

2020年4月25日 13:37

以前我们直接赋值，需要写清楚单元格区域和目标单元格区域，先不说你有多累，如果这个区域不固定怎么办？

**Sub 直接赋值()**

**With Range("B2").CurrentRegion**

**Sheets("Sheet2").Range("A1").Resize(.Rows.Count, .Columns.Count).Value = .Value**

**End With**

**End Sub**

## 16.6 Intersect获取单元格交叉区域

2020年4月25日 12:48

### 【通用壳子】

Sub Intersect交叉区域(r1, r2)

Dim 单元格对象

'Intersect方法至少指定两个参数，最多30个，必须是Range对象

Set 单元格对象 = Excel.Application.Intersect(r1, r2)

'如果单元格对象什么都没有 (Nothing)

If 单元格对象 Is Nothing Then

MsgBox "不存在交叉区域"

Else

MsgBox "交叉区域地址为：" & 单元格对象.Address

End If

'释放单元格对象占用的内存，不写不会报错，但是数据大了以后会卡死

Set 单元格对象 = Nothing

End Sub

Sub 调用()

Call Intersect交叉区域(Range("C3:G14"), Range("E7:I20"))

End Sub

## 16.7 Union单元格多选

2020年4月25日 17:14

**Application.Union方法：**把多个Range联合在一起，作为一个新的Range对象返回

	A	B	C	D	E	F	G
1	张三丰1	张三丰2	张三丰3	张三丰4	孙兴华	张三丰6	张三丰7
2	李小龙1	孙兴华	李小龙3	李小龙4	李小龙5	李小龙6	李小龙7
3	叶问1	叶问2	叶问3	孙兴华	叶问5	叶问6	叶问7
4	黄飞鸿1	黄飞鸿2	黄飞鸿3	黄飞鸿4	黄飞鸿5	黄飞鸿6	黄飞鸿7
5	霍元甲1	霍元甲2	孙兴华	霍元甲4	霍元甲5	霍元甲6	霍元甲7
6	陈真1	陈真2	陈真3	陈真4	孙兴华	陈真6	陈真7
7	岳飞1	岳飞2	孙兴华	岳飞4	岳飞5	岳飞6	岳飞7

### 【通用壳子】

Sub 多选单元格()

'因为省去了set，所以这里要定义成单元格对象

Dim r1 As Range, r2 As Range

For Each r1 In Range("A1:G7")

If r1.Value = "孙兴华" Then

'如果r2对象变为Nothing

If r2 Is Nothing Then

'就指定r2为range对象r1

Set r2 = r1

Else

'否则r2单元格区域与r1对象多选联合后区域重新为r2赋值

Set r2 = Excel.Application.Union(r2, r1)

End If

End If

Next

If Not r2 Is Nothing Then

r2.Select

r2.Interior.Color = vbRed

End If

**Set r1 = Nothing**

**Set r2 = Nothing**

**End Sub**

**实际工作中，100张工作表，把所有工作表中的孙兴华都给我设置上颜色，你就用到了**

## 16.8 单元格边框

2020年4月25日 17:58

### Sub 添加边框()

```
Dim r1 As Range
Set r1 = Range("A1:G7")
With r1.Borders
    '边框线条样式
    .LineStyle = xlContinuous
    '边框线条粗细
    .Weight = xlThin
    '边框线条颜色
    .ColorIndex = 5
End With
'使用BorderAround方法为单元格区域添加一个加粗外框
r1.BorderAround xlContinuous, xlMedium, 5
Set r1 = Nothing
End Sub
```

### Sub 外实内虚()

```
Dim r1 As Range
Set r1 = Range("A1:G7")
With r1.Borders(xlInsideHorizontal) '内部水平
    .LineStyle = xlDot
    .Weight = xlThin
    .ColorIndex = 5
End With
With r1.Borders(xlInsideVertical) '内部垂直
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = 5
End With
r1.BorderAround xlContinuous, xlMedium, 5
Set r1 = Nothing
End Sub
```

### Sub 清除线条()

```
Columns("A:G").Borders.LineStyle = xlNone
End Sub
```

xlDiagonalDown	斜下边框
xlDiagonalUp	斜上边框
xlEdgeLeft	左边框
xlEdgeTop	顶部边框
xlEdgeBottom	底部边框
xlEdgeRight	右边框
xlInsideVertical	内部垂直
xlInsideHorizontal	内部水平

## 16.8.1 ColorIndex颜色表

2020年4月25日 18:58

Index Value	Color	Index Value	Color	Index Value	Color
1		20		39	
2		21		40	
3		22		41	
4		23		42	
5		24		43	
6		25		44	
7		26		45	
8		27		46	
9		28		47	
10		29		48	
11		30		49	
12		31		50	
13		32		51	
14		33		52	
15		34		53	
16		35		54	
17		36		55	
18		37		56	
19		38			

## 16.8.2 VBA边框的两个属性LineStyle和Weight

2020年4月25日 19:10

Borders.Weight	它代表边框的粗细
xlHairline	细线(最细的边框)
xlMedium	中等
xlThick	粗(最宽的边框)
xlThin	细

Borders.LineStyle	边框的线型
xlContinuous	实线
xlDash	虚线
xlDashDot	点划相间线
xlDashDotDot	划线后跟两个点
xlDot	点式线
xlDouble	双线
xlLineStyleNone	无线条
xlSlantDashDot	倾斜的划线



16.9 数据排序

2020年4月26日 16:33

Range对象的Sort方法对区域进行排序，其语法格式如下：  
Sort(Key1, Order1, Key2, Type, Order2, Key3, Order3, Header, OrderCustom, MatchCase, Orientation, SortMethod, DataOption1, DataOption2, DataOption3)

其中Key1、Key2、Key3是可选的，分别代表第1排序字段，第2排序字段，第3排序字段（Range的Sort方法就支持3个字段）  
Order1，Order2，Order3是可选的，指的是排序方式（升序或降序），值见下表

xlAscending	升序（默认）
xlDescending	降序

Header是可选的，指定第1行是否包含标题

xlGuess	Excel确定是否有标题，如果有，确定标题位于何处
xlNo	不包含标题(默认)
xlYes	包含标题

Type 可选，仅在对数据透视表排序时才使用本参数

xlSortType	可为以下 XlSortType 常量之一。
xlSortLabels	按标签对数据透视表排序。
xlSortValues	按值对数据透视表排序。

OrderCustom 可选。本参数是从 1 开始的整数，指定了在自定义排序顺序列表中的索引号。如果省略 OrderCustom参数，则使用常规排序。  
MatchCase 可选。如果为 True，则进行区分大小写的排序；如果为 False，则排序时不区分大小写。对数据透视表进行排序时，不能使用本参数。  
Orientation 可选，排序方向

XlSortOrientation	可为以下 XlSortOrientation 常量之一。
xlSortRows	默认值。按行排序。
xlSortColumns	按列排序。

SortMethod 可选。排序类型。对于所选择或安装的不同语言支持（例如：美国英语），以上某些常量可能不可用

XlSortMethod	可为以下 XlSortMethod 常量之一。
xlStroke	按每个字符的笔划数量排序。
xlPinYin	默认值。按字符的汉语拼音顺序排序。

DataOption1 XlSortDataOption 类型，可选。指定如何对 key 1 中的文本进行排序。对数据透视表进行排序时，不能使用本参数。

XlSortDataOption	可为以下 XlSortDataOption 常量之一。
xlSortTextAsNumbers	将文本作为数字型数据排序。
xlSortNormal	默认值。分别对数字和文本数据进行排序。

DataOption2 XlSortDataOption 类型，可选。指定如何对 key 2 中的文本进行排序。对数据透视表进行排序时，不能使用本参数。

XlSortDataOption	可为以下 XlSortDataOption 常量之一。
xlSortTextAsNumbers	将文本作为数字型数据排序。
xlSortNormal	默认值。分别对数字和文本数据进行排序。

DataOption3 XlSortDataOption 类型，可选。指定如何对 key 3 中的文本进行排序。对数据透视表进行排序时，不能使用本参数。

XlSortDataOption	可为以下 XlSortDataOption 常量之一。
xlSortTextAsNumbers	将文本作为数字型数据排序。
xlSortNormal	默认值。分别对数字和文本数据进行排序。

一、按总分排序

```
Sub 数据排序1()  
    Range("A1").Sort "总成绩", xlDescending, Header:=xlYes  
End Sub
```

二、最多支持3个字段排序

```
Sub 数据排序2()  
    Range("A1").Sort key1:="总成绩", order1:=xlDescending, _  
        key2:="数学", order2:=xlDescending, _  
        key3:="语文", order3:=xlDescending, _  
        Header:=xlYes  
End Sub
```

## 16.10 多关键字排序

2020年4月26日 16:49

Range对象的Sort方法对区域进行排序时，同时最多只能指定3个关键字，当需要按超过3个关键字对区域进行排序时，可以通过多次执行Sort方法来实现。但是，需要注意，排序时请按关键字倒序顺序，例如：希望按ABCD的关键字顺序进行排序，则按DCBA顺序执行Sort方法。

Sub 多关键字排序\_不推荐()

```
With Range("A1")
    .Sort Key1:="英语", order1:=xlDescending, Header:=xlYes
    .Sort Key1:="语文", order1:=xlDescending, Header:=xlYes
    .Sort Key1:="数学", order1:=xlDescending, Header:=xlYes
    .Sort Key1:="总成绩", order1:=xlDescending, Header:=xlYes
End With
End Sub
```

Sub 多关键字排序()

```
With Range("A1")
    .Sort Key1:="英语", order1:=xlDescending, _
        Key2:="语文", order2:=xlDescending, Header:=xlYes
    .Sort Key1:="数学", order1:=xlDescending, _
        Key2:="总成绩", order2:=xlDescending, Header:=xlYes
End With
End Sub
```

**推荐**

## 16.11 自定义序列排序

2020年4月26日 16:50

**Sub SortByLists()**

**Dim arr, 序号**

**arr = Range("E2:E6")**

**'通过AddCustomList方法为数组添加自定义序列**

**Excel.Application.AddCustomList arr**

**'返回数组在自定义序列中的序列号，保存在序号这个变量中**

**序号 = Application.GetCustomListNum(arr)**

**'因为OrderCustom从1开始，如果有一行表头我们就要加1**

**Range("A1").Sort Key1:="部门", Order1:=xlAscending, Header:=xlYes, OrderCustom:=序号 + 1**

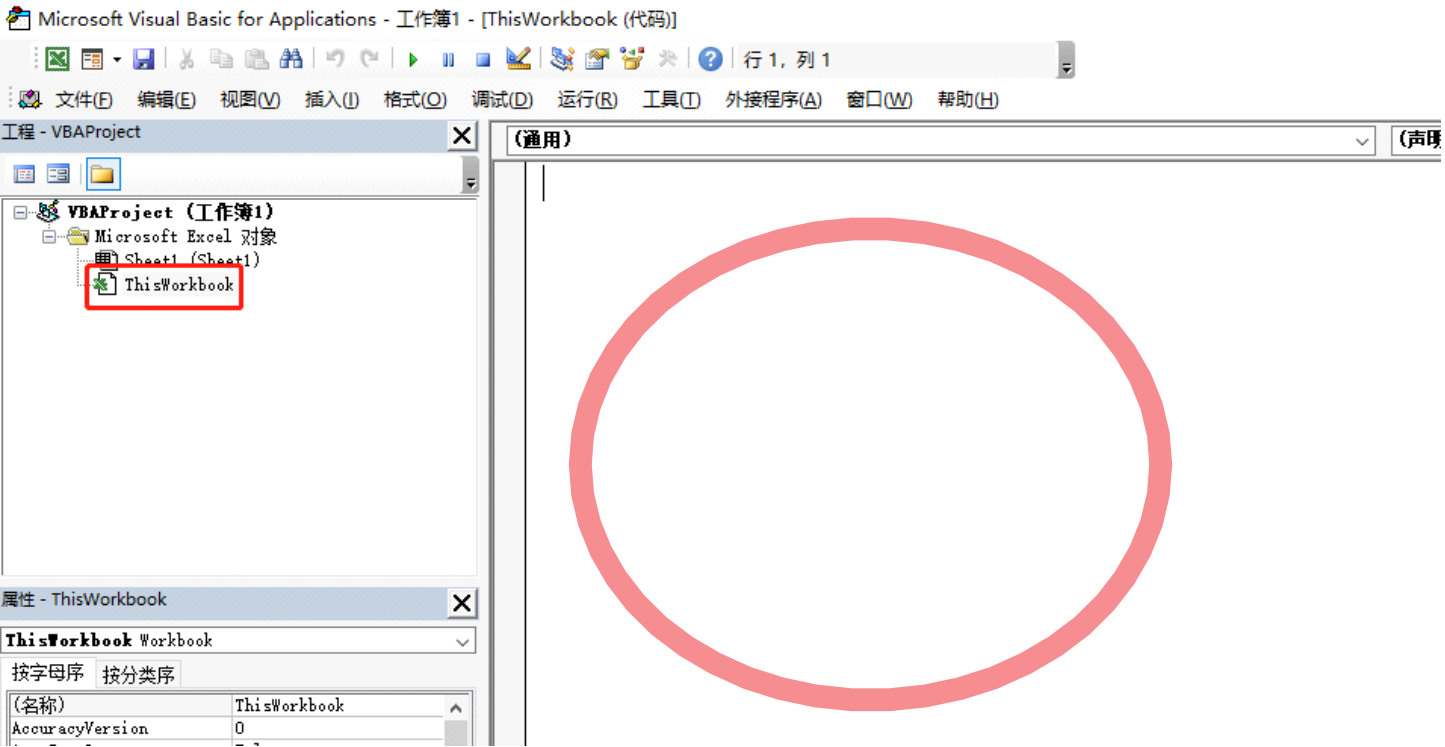
**'使用DeleteCustomList删除新添加的自定义序列**

**Application.DeleteCustomList 序号**

**End Sub**

# 17.事件编程，执行程序的自动开关

2020年4月26日 17:59



常用工作簿事件名称与对应处理过程名称示例

事件名称	触发该事件的时间	规定的处理过程名称
Activate	工作簿被激活时	Workbook_Activate
Open	工作簿被打开时	Workbook_Open
BeforeClose	关闭工作簿之前	Workbook_BeforeClose
BeforeSave	保存工作簿之前	Workbook_BeforeSave
BeforePrint	打印工作簿之前	Workbook_BeforePrint
NewSheet	在工作簿中新建工作表	Workbook_NewSheet
.....	.....	.....

# 【重要】Workbook事件

2020年4月26日 19:16

Worksheet对象的事件列表		
序号	事件名称	事件说明
1	Activate	当激活工作簿是发生
2	AddinInstall	当工作簿作为加载宏安装时发生
3	AddinUninstall	当工作表作为加载宏卸载时发生
4	AfterXmlExport	在保存或导出指定工作簿中的XML数据之后发生
5	BeforePrint	在打印指定工作簿（或其中任何内容）之前发生
6	BeforeSave	在保存工作簿前发生
7	BeforeXmlExport	在保存或导出指定工作簿中的XML数据之前发生
8	BeforeXmlImport	在刷新现有XML数据连接或新的XML数据被导入任意一个打开的工作簿前发生
9	Deactivate	在工作簿从活动状态转变为非活动状态时发生
10	NewSheet	在工作簿中新建工作表时发生
11	Open	在打开工作簿时发生
12	PivotTableCloseConnection	在数据透视表的连接关闭之后发生
13	PivotTableOpenConnection	在数据透视表的连接打开之后发生
14	SheetActivate	在激活任意工作表时发生
15	SheetBeforeDoubleClick	在双击任意工作表时（默认的双击操作之前）发生
16	SheetBeforeRightClick	在右键单击任意工作表时（默认的右键单击操作之前）发生
17	SheetCalculate	在重新计算工作表时或在图表上绘制更改的数据之后发生
18	SheetChange	当更改了任何工作表中的单元格时发生
19	SheetDeactivate	当工作表从活动工作表变为不活动工作表时发生
20	SheetFollowHyperlink	当单击工作簿中的任何超链接时发生
21	SheetPivotTableUpdate	在更新数据透视表的工作表后发生
22	SheetSelectionChange	当任意工作表上的选定区域发生更改时发生（但图表工作表上的选定区域发生改变时，不会发生此事件）
23	Sync	当“文档工作区”一部分的工作簿的本地副本与服务器上的副本进行同步时发生
24	WindowActive	在激活任意工作簿窗口时发生
25	WindowDeactivate	在任意工作簿窗口由活动窗口变为非活动窗口时发生
26	WindowResize	在调整任意工作簿窗口的大小时发生

## 【重要】Worksheet事件

2020年4月26日 19:19

Worksheet对象的事件列表	
事件名称	事件说明
Active	激活工作表时发生
BeforeDoubleClick	双击工作表之后，默认是双击动作之前发生
BeforeRightClick	右击工作表之后，默认是右击操作之前发生
Calculate	重新计算工作表之后发生
Change	工作表中的单元格发生改变是发生
Deactivate	工作表由活动工作表变为非活动工作表时发生
FollowHyerlink	单击工作表中的任意超链接时发生
PivotTableUpdate	在工作表更新数据透视表之后发生
SelectionChange	工作表中所选内容发生改变时发生

## 17.1 用事件替程序装一个自动执行的开关

2020年4月26日 19:05

将这段代码放到工作簿事件中，让工作簿打开时就提醒过生日的人名

```
Sub a()  
Dim i  
i = 2: j = 1  
Do While Range("A" & i) <> ""  
    If Month(Range("C" & i)) = Month(Date) And Day(Range("C" & i)) = Day(Date) Then  
        MsgBox "今天是" & Range("A" & i) & "的生日"  
        Sheets("生日名单").Range("A" & j) = Range("A" & i)  
        j = j + 1  
    End If  
    i = i + 1  
Loop  
End Sub
```

---

Private Sub Workbook\_Open()

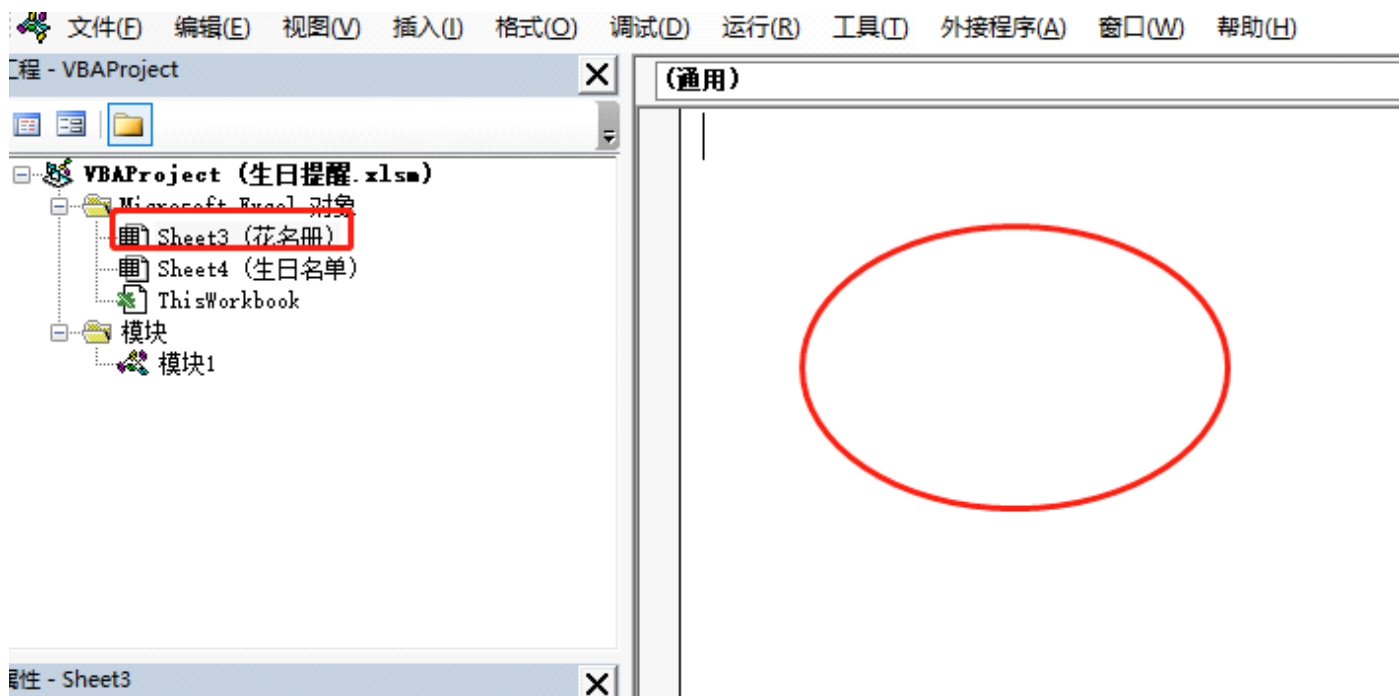


对象名称

事件名称

## 17.2 工作表事件

2020年4月26日 19:43





## 17.2.1 Worksheet\_Change (单元格的值变了去执行宏)

2020年4月26日 19:52

### 一、写一个筛选程序

Sub 筛选()

'笔记9.6和笔记9.2

Range("K1").CurrentRegion.Clear

'筛选这个区域内第2列，等于H2单元格的名称

Range("A1").CurrentRegion.AutoFilter field:=2, Criteria1:=Range("H2")

'将这个区域复制到K1单元格

Range("A1").CurrentRegion.Copy Range("K1")

'取消自动筛选

Range("A1").CurrentRegion.AutoFilter

End Sub

### 二、禁用事件，让事件过程不再自动执行

Private Sub Worksheet\_Change(ByVal Target As Range)

'关闭事件

Excel.Application.EnableEvents = False

'因为筛选本身就是修改单元格

Call 筛选

'打开事件

Excel.Application.EnableEvents = True

End Sub

### 三、只要部分单元格被更改时才执行指定的代码

Target: 代表任意单元格

1、例如只要让某列发生变化时，才执行指定代码

Private Sub Worksheet\_Change(ByVal Target As Range)

'判断单元格列号是否为8，也就是H列

**If Target.Column = 8 Then**

**Call 筛选**

**End If**

**End Sub**

**或**

**Private Sub Worksheet\_Change(ByVal Target As Range)**

**'判断单元格列号是否为8，也就是H列**

**If Target.Column <> 8 Then Exit Sub**

**Call 筛选**

**End Sub**

**2、例如只要让某单元格发生变化时，才执行指定代码**

**Private Sub Worksheet\_Change(ByVal Target As Range)**

**'笔记9.10**

**If Target.Address <> "\$H\$2" Then Exit Sub**

**Call 筛选**

**End Sub**

## 17.2.2 【进阶】用Change事件模仿收银机

2020年4月26日 19:43

```
Private Sub Worksheet_Change(ByVal Target As Range)
```

```
    '同时更改多个单元格时结束执行程序，CountLarge和count功能一样
```

```
    'CountLarge不会溢出，但是count会，xlsx单元格太多了，容易发生数据类型溢出
```

```
    If Target.CountLarge <> 1 Then Exit Sub
```

```
    '输入数据为空时退出
```

```
    If Target.Value = "" Then Exit Sub
```

```
    '输入行号是第1行时退出
```

```
    If Target.Row = 1 Then Exit Sub
```

```
    'i代表商品表的行数
```

```
    Dim i
```

```
    '当输入等于第2列时
```

```
    If Target.Column = 2 Then
```

```
        '如果出错了，就是没找到，跳转到标签a
```

```
        On Error GoTo a
```

```
        'Excel基础课函数篇06和07，match返回输入的商品名称来自参照表第几行
```

```
        i = Excel.Application.WorksheetFunction.Match(UCase(Target.Value), Range("H:H"), 0)
```

```
        '禁止事件，防止将字母改为商品名称时，再次执行程序
```

```
        Excel.Application.EnableEvents = False
```

```
        With Target
```

```
            .Value = Range("I" & i).Value
```

```
            '笔记9.4 下移行，右移列
```

```
            .Offset(0, -1).Value = Now
```

```
            .Offset(0, 1) = Range("J" & i).Value
```

```
            .Offset(0, 2) = Range("K" & i).Value
```

```
            '输入商品名称后，选中销售数量的单元格
```

```
            .Offset(0, 3).Select
```

```
        End With
```

```
        Excel.Application.EnableEvents = True
```

```
        Exit Sub
```

```
a: MsgBox "没有该商品，请联系维护人员"
```

```
    Target.Value = ""
```

Else

    If Target.Column = 5 Then

        Application.EnableEvents = False

        Target.Offset(0, 1) = Target \* Target.Offset(0, -1)

        Cells(Target.Row + 1, 2).Select

        Application.EnableEvents = True

    End If

End If

End Sub

### 17.2.3 Worksheet\_SelectionChange (选区发生变化去执行宏)

2020年4月27日 9:51

**Private Sub Worksheet\_SelectionChange(ByVal Target As Range)**

**'改颜色前需要将工作表的背景色改为透明色(xlNone)**

**Cells.Interior.Color = xlNone**

**'选中的单元格整行改颜色**

**Target.EntireRow.Interior.Color = RGB(255, 0, 0)**

**'选中的单元格整列改颜色**

**Target.EntireColumn.Interior.Color = RGB(255, 0, 0)**

**End Sub**

## 17.2.4 用批注记录单元格修改情况

2020年4月27日 10:03

'在所有过程之前用Dim语句定义的变量r1是模块级变量，应模块中所有的过程都可以使用它

Dim r1 '定义一个模块级变量，用户保存单元格的数据

'第一个事件过程，用于记录被更改前单元格中保存的数据

Private Sub Worksheet\_SelectionChange(ByVal Target As Range)

If Target.Cells.Count <> 1 Then Exit Sub '选中多个单元格时退出程序

If Target.Formula = "" Then '根据选中单元格中保存的数据，确定给变量r1赋什么值

    r1 = "空"

Else

    r1 = Target.Text

End If

End Sub

'第二个事件过程，用于批注记录单元格修改前后的信息

Private Sub Worksheet\_Change(ByVal Target As Range)

If Target.Cells.Count <> 1 Then Exit Sub

'定义变量保存单元格修改后的内容

Dim r2

'判断单元格是否被修改为空单元格

If Target.Formula = "" Then

    r2 = "空"

Else

    r2 = Target.Formula

End If

'如果单元格修改前后的内容一样则退出程序

If r1 = r2 Then Exit Sub

'定义一个批注变量

Dim r3

'定义一个变量保存批注内容

Dim r4

'将被修改单元格的批注赋给变量r3

Set r3 = Target.Comment

'如果单元格中没有批注则新建批注

If r3 Is Nothing Then Target.AddComment

'将批注的内容保存到变量r4中

r4 = Target.Comment.Text

'重新修改批注的内容=原批注内容+当前日期和时间+原内容+修改后的新内容

Target.Comment.Text Text:=r4 & Chr(10) & Format(Now(), "yyyy-mm-dd hh:mm") & "原内容:" & r1 & "修改为:" & r2

'根据批注内容自动调整批注大小

Target.Comment.Shape.TextFrame.AutoSize = True

End Sub

## 17.2.5 激活工作表

2020年4月27日 2:17

**Private Sub Worksheet\_Activate()**

**ActiveWorkbook.RefreshAll '刷新透视表**

**End Sub**

## 17.3 工作簿事件

2020年4月27日 1:05



## 17.3.1 Open事件

2020年4月27日 1:05

```
Private Sub Workbook_Open()  
    Sheets("首页").Select  
End Sub
```

## 17.3.2 BeforeClose事件：在关闭工作簿之前发生

2020年4月27日 1:12

**'Cancel是过程参数，用来确定是否响应关闭操作，值为F关闭，值为T不能关闭**

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
If MsgBox("你确定要关闭工作簿吗？ ", vbYesNo) = vbNo Then
    Cancel = True
End If
End Sub
```

**恶作剧：永远无法关闭的工作簿**

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    MsgBox "有本事你把我关闭！ "
    Cancel = True
End Sub
```

# 【重要资料】MsgBox语法和参数

2020年4月27日 1:18

## 语法

MsgBox (Prompt [,Buttons] [,Title] [,Helpfile,Context])

### MsgBox函数参数:

Prompt	必选。字符串表达式，显示在对话框中的消息。Prompt的最大长度大约为1024个字符，由所用字符的字节大小决定。如果Prompt的内容超过一行，则可以在每一行之间用回车符（Chr(13)）、换行符（Chr(10)）或是回车与换行符的组合（Chr(13) & Chr(10)，即vbCrLf）将各行分隔开来。
Buttons	可选。数值表达式，是一些数值的总和，指定所显示的按钮的数目及形式、使用的图标样式（及声音）， 缺省按钮以及消息框的强制性等。如果省略，则其缺省值为0。具体数值见“参数设置值”表格。
Title	可选。字符串表达式，在对话框标题栏中显示的内容。如果省略Title，则将应用程序标题（App.Title）放在标题栏中。
Helpfile	可选。字符串表达式，用来向对话框提供上下文相关帮助的帮助文件。如果提供了Helpfile，则也必须提供Context。
Context	可选。数值表达式，由帮助文件的作者指定给适当的帮助主题的帮助下文编号。如果提供了Context，则也必须提供Helpfile。

Buttons参数设置值

常数	值	描述
对话框中显示的按钮的类型与数目		
vbOKOnly	0	只显示“确定”按钮。（缺省）
vbOKCancel	1	显示“确定”和“取消”按钮。
vbAbortRetryIgnore	2	显示“终止”、“重试”和“忽略”按钮。
vbYesNoCancel	3	显示“是”、“否”和“取消”按钮。
vbYesNo	4	显示“是”和“否”按钮。
vbRetryCancel	5	显示“重试”和“取消”按钮。
图标的样式（根据系统设置，可能伴有声音）		
vbCritical	16	显示“错误信息”图标。
vbQuestion	32	显示“询问信息”图标。
vbExclamation	48	显示“警告消息”图标。
vbInformation	64	显示“通知消息”图标。
默认按钮		
vbDefaultButton1	0	第一个按钮是默认按钮。（缺省）
vbDefaultButton2	256	第二个按钮是默认按钮。
vbDefaultButton3	512	第三个按钮是默认按钮。
vbDefaultButton4	768	第四个按钮是默认按钮。
对话框的强制返回性		
vbApplicationModal	0	应用程序强制返回：应用程序一直被挂起，直到用户对消息框作出响应才继续工作。
vbSystemModal	4096	系统强制返回：全部应用程序都被挂起，直到用户对消息框作出响应才继续工作。
对话框特殊设置		
vbMsgBoxHelpButton	16384	将帮助按钮添加到消息框。
vbMsgBoxSetForeground	65536	指定消息框窗口作为前景窗口。
vbMsgBoxRight	524288	文本为右对齐。
vbMsgBoxRtlReading	1048576	指定文本应在在希伯来和阿拉伯语系统中的从右到左显示。

MsgBox函数返回值

常数	值	描述
vbOK	1	单击了“确定”按钮。
vbCancel	2	单击了“取消”按钮。
vbAbort	3	单击了“终止”按钮。
vbRetry	4	单击了“重试”按钮。
vbIgnore	5	单击了“忽略”按钮。
vbYes	6	单击了“是”按钮。
vbNo	7	单击了“否”按钮。

### 17.3.3 SheetChange事件：更新任意工作表中的单元格时发生

2020年4月27日 1:28

与工作表中的Change事件不同的是，工作表事件针对 一张工作表，如果你所有工作表都要执行这个事情，就可以用工作簿事件中的SheetChange

当工作簿中任意一张工作表单元格被更改时，都自动执行该事件编写的事件过程

比如那个十字修改颜色的代码就可以写在这里

'sh代表被更改单元格所在的工作表，Target代表被更改的单元格

```
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
    MsgBox "您正在修改的是:" & Sh.Name & "工作表中的" & Target.Address & "单元格"
End Sub
```

例如：如果只想让更改名称Sheet1之外的工作表中的单元格，才执行事件过程

'sh代表被更改单元格所在的工作表，Target代表被更改的单元格

```
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
    If Sh.Name <> "Sheet1" Then
        MsgBox "您正在修改的是:" & Sh.Name & "工作表中的" & Target.Address & "单元格"
    End If
End Sub
```

### 17.3.4 BeforeSave 保存之前执行事件

2020年4月27日 1:36

```
Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)
    ThisWorkbook.SaveCopyAs "c:\" & Format(Now(), "yyyymmddhhmmss") & ".xlsx"
End Sub
```

**注：**format相当于Excel函数中的Text（转成文本函数），Now()返回现在日期时间函数，yyyymmddhhmmss指的是年月日时分秒的格式。

如果你想知道谁喜欢偷看你电脑上的文件，你可以在打开工作簿事件中，写上这行代码，当你发现具体日期和时间被人查看过文件后，去调监控吧

## 17.3.5 NewSheet创建新表时

2020年4月27日 1:48

```
Private Sub Workbook_NewSheet(ByVal Sh As Object)  
Sheets("表1").UsedRange.Copy Sh.Range("A1")  
Sh.Range("A2:F11").Value = ""  
End Sub
```

## 17.4 让文件自动保存

2020年4月27日 2:18

**Sub otime()**

**'10秒后自动运行WbSave过程**

**Application.OnTime Now() + TimeValue("00:00:10"), "WbSave"**

**End Sub**

**Sub WbSave()**

**ThisWorkbook.Save '保存本工作簿**

**Call otime '再次运行otime过程**

**End Sub**

**为了让工作簿打开就自动运行：**

**Private Sub Workbook\_Open()**

**Call otime**

**End Sub**

# 18.用户信息交互

2020年4月28日 14:09

**什么是用户信息交互？ 答：就是让用户输入内容或提示内容给用户看**



## 18.1 InputBox函数【只能返回字符型数据】

2020年4月28日 14:09

```
Sub a()  
Dim i  
i = InputBox("你叫什么名字")  
Range("A1") = i  
End Sub
```

InputBox函数共有5个参数：

**prompt**：用于设置在对话框中显示的提示信息

**Title**：用于设置对话框的标题

**Default**：是对话框中默认的输入值

**xpos**：用于设置对话框左端与屏幕左端的距离

**ypos**：是对话框的顶端与屏幕顶端的距离

```
Sub a()  
Dim i  
i = InputBox(prompt:="我是提示信息", Title:="我是标题", Default:="默认输入值", xpos:=2000, ypos:=2500)  
Range("A1") = i  
End Sub
```

**可以简写：但是位置不能变，只有prompt必选，其它都是可选，如果中间的参数省略一定要用逗号隔开**

```
Sub a()  
Dim i  
i = InputBox("我是提示信息", "我是标题", "默认输入值", 2000, 2500)  
Range("A1") = i  
End Sub
```

## 18.2 InputBox方法

2020年4月28日 14:31

用Application对象的InputBox方法可以创建与程序互动的对话框

```
Sub a()  
Dim i  
i = Excel.Application.InputBox(prompt:="我是提示信息", Title:="我是标题", Default:="默认输入值", Left:=2000, Top:=2500, Type:=1)  
Range("A1") = i  
End Sub
```

Inputbox方法Type参数的可设置项及说明:

类型	参数值
公式	0
数字	1
文本 (字符串)	2
逻辑值 (T或F)	4
单元格引用(Range对象)	8
错误值, 如#N/A	16
数值数组	64

3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15这些数值都没有

原因是留给你组合的空间, 想返回多种类型, 可以用:

Type:=1+2 当然也可以直接写成Type:=3

例如: 让Inputbox返回单元格对象

```
Sub a()  
Dim r1  
Set r1 = Excel.Application.InputBox("请输入单元格区域: ", Type:=8)  
r1.Value = 520  
End Sub
```

18.3 MsgBox语法和参数

2020年4月27日 1:18

语法

MsgBox (Prompt [,Buttons] [,Title] [,Helpfile,Context])

MsgBox函数参数:

Prompt	必选。字符串表达式，显示在对话框中的消息。Prompt的最大长度大约为1024个字符，由所用字符的字节大小决定。如果Prompt的内容超过一行，则可以在每一行之间用回车符（Chr(13)）、换行符（Chr(10)）或是回车与换行符的组合（Chr(13) & Chr(10)，即vbCrLf）将各行分隔开来。
Buttons	可选。数值表达式，是一些数值的总和，指定所显示的按钮的数目及形式、使用的图标样式（及声音）， 缺省按钮以及消息框的强制性等。如果省略，则其缺省值为0。具体数值见“参数设置值”表格。
Title	可选。字符串表达式，在对话框标题栏中显示的内容。如果省略Title，则将应用程序标题（App.Title）放在标题栏中。
Helpfile	可选。字符串表达式，用来向对话框提供上下文相关帮助的帮助文件。如果提供了Helpfile，则也必须提供Context。
Context	可选。数值表达式，由帮助文件的作者指定给适当的帮助主题的帮助下文编号。如果提供了Context，则也必须提供Helpfile。

Buttons参数设置值

常数	值	描述
对话框中显示的按钮的类型与数目		
vbOKOnly	0	只显示“确定”按钮。（缺省）
vbOKCancel	1	显示“确定”和“取消”按钮。
vbAbortRetryIgnore	2	显示“终止”、“重试”和“忽略”按钮。
vbYesNoCancel	3	显示“是”、“否”和“取消”按钮。
vbYesNo	4	显示“是”和“否”按钮。
vbRetryCancel	5	显示“重试”和“取消”按钮。
图标的样式（根据系统设置，可能伴有声音）		
vbCritical	16	显示“错误信息”图标。
vbQuestion	32	显示“询问信息”图标。
vbExclamation	48	显示“警告消息”图标。
vbInformation	64	显示“通知消息”图标。
默认按钮		
vbDefaultButton1	0	第一个按钮是默认按钮。（缺省）
vbDefaultButton2	256	第二个按钮是默认按钮。
vbDefaultButton3	512	第三个按钮是默认按钮。
vbDefaultButton4	768	第四个按钮是默认按钮。
对话框的强制返回性		
vbApplicationModal	0	应用程序强制返回：应用程序一直被挂起，直到用户对消息框作出响应才继续工作。
vbSystemModal	4096	系统强制返回：全部应用程序都被挂起，直到用户对消息框作出响应才继续工作。
对话框特殊设置		
vbMsgBoxHelpButton	16384	将帮助按钮添加到消息框。
vbMsgBoxSetForeground	65536	指定消息框窗口作为前景窗口。
vbMsgBoxRight	524288	文本为右对齐。
vbMsgBoxRtlReading	1048576	指定文本应在在希伯来和阿拉伯语系统中的从右到左显示。

6种按钮设定

对话框中显示的图表

```
Sub a()  
MsgBox "只显示【确定】按钮", Buttons:=vbOKOnly  
MsgBox "显示【确定】和【取消】按钮", Buttons:=vbOKCancel  
MsgBox "显示【中止】、【重试】和【忽略】按钮", Buttons:=vbAbortRetryIgnore  
MsgBox "显示【是】、【否】和【取消】按钮", Buttons:=vbYesNoCancel  
MsgBox "显示【是】和【否】", Buttons:=vbYesNo  
MsgBox "显示【重试】和【取消】", Buttons:=vbRetryCancel  
End Sub
```

```
Sub a()  
MsgBox "显示【错误信息】图标", Buttons:=vbCritical  
MsgBox "显示【询问信息】图标", Buttons:=vbQuestion  
MsgBox "显示【警告消息】图标", Buttons:=vbExclamation  
MsgBox "显示【通知消息】图标", Buttons:=vbInformation  
End Sub
```

```
'以上两种组合使用:  
Sub a()  
MsgBox "跟着孙兴华学习VBA你学会了吗？", Buttons:=vbYesNo + vbQuestion  
End Sub
```

```
Sub a()  
MsgBox "跟着孙兴华学习VBA你学会了吗？", Buttons:=vbYesNo + vbQuestion + vbDefaultButton1  
End Sub
```

指定对话框的类型

MsgBox函数返回值

常数	值	描述
vbOK	1	单击了“确定”按钮。
vbCancel	2	单击了“取消”按钮。
vbAbort	3	单击了“终止”按钮。
vbRetry	4	单击了“重试”按钮。
vbIgnore	5	单击了“忽略”按钮。
vbYes	6	单击了“是”按钮。
vbNo	7	单击了“否”按钮。

```
Sub a()  
  
Dim i  
i = MsgBox("你是男性吗？", Buttons:=vbYesNo + vbQuestion)  
If i = vbYes Then  
    Range("A1")= "很Man!"  
Else  
    Range("A1")= "是小姐姐！"  
End If  
End Sub
```

## 18.4 【打开文件】 FindFile方法

2020年4月28日 14:09

Sub a()

'判断是否成功打开了选择的文件

If Excel.Application.FindFile = True Then

MsgBox "你选择的文件已经打开了! "

Else

MsgBox "你选择了【取消】按钮，操作没有完成"

End If

End Sub

如果成功打开一个文件，返回T,如果取消返回F

**FindFile方法：打开在对话框中选中的文件**

## 18.5【获取路径+文件名】 GetOpenFilename方法

2020年4月28日 19:39

### GetOpenFilename方法：获得在对话框中选中的文件的文件名称（包含路径）

如果你希望程序在运行过程中，手动选择文件，再根据文件路径及名称进行其它操作，就使用GetOpenFilename方法

例如：多个工作簿合并其内容，运行中可以让用户手工选择文件进行合并。

参数：

GetOpenFilename (文件类型，优先类型(1或2)，对话框标题，按钮文字，是否支持多选)

GetOpenFilename(FileFilter, FilterIndex, Title, ButtonText, MultiSelect)

#### 一、让对话框显示**所有类型**的文件

```
Sub a()  
Dim i  
i = Excel.Application.GetOpenFilename  
If i = False Then  
    MsgBox "没有选择任何文件! "  
    Exit Sub  
Else  
    Range("A1") = i  
End If  
End Sub
```

#### 二、只在对话框中显示**指定类型**的文件

```
Sub a()  
Dim i  
i = Excel.Application.GetOpenFilename("图片文件,*.jpg")  
If i = False Then  
    MsgBox "没有选择任何文件! "  
    Exit Sub  
Else  
    Range("A1") = i  
End If  
End Sub
```

#### 三、让对话框同时显示**同一类型的多种扩展名**的文件

```
Sub a()  
Dim i  
i = Excel.Application.GetOpenFilename("Excel文件,*.xls;*.xlsx;*.xlsm")  
If i = False Then  
    MsgBox "没有选择任何文件! "  
    Exit Sub  
Else  
    Range("A1") = i  
End If  
End Sub
```

注意：后缀是分号

#### 四、让对话框选择显示**多种类型**的文件

```
Sub a()  
Dim i  
i = Excel.Application.GetOpenFilename("Excel文件,*.xls;*.xlsx;*.xlsm,Word文件,*.doc;*.docx;*.docm")  
If i = False Then
```

```

    MsgBox "没有选择任何文件! "
    Exit Sub
Else
    Range("A1") = i
End If
End Sub

```

## 五、通过FilterIndex参数设置默认显示的文件类型

```

Sub a()
Dim i
i = Excel.Application.GetOpenFilename("Excel文件,*.*xls;*.xlsx;*.xlsm,Word文件,*.*doc;*.docx;*.docm", FilterIndex:=2)
If i = False Then
    MsgBox "没有选择任何文件! "
    Exit Sub
Else
    Range("A1") = i
End If
End Sub

```

## 六、设置允许同时选择多个文件

```

Sub a()
Dim i
i = Excel.Application.GetOpenFilename("Excel文件,*.*xls;*.xlsx;*.xlsm", MultiSelect:=True)
If IsArray(i) Then
    Range("A1").Resize(UBound(i), 1) = Excel.Application.WorksheetFunction.Transpose(i)
Else
    MsgBox "已取消操作! "
End If
End Sub

```

## 七、修改对话框标题

```

Sub a()
Dim i
i = Excel.Application.GetOpenFilename("Excel文件,*.*xls;*.xlsx;*.xlsm", Title:="孙兴华")
If i = False Then
    MsgBox "没有选择任何文件! "
    Exit Sub
Else
    Range("A1") = i
End If
End Sub

```

## 18.6 【另存为】对话框 GetSaveAsFilename方法

2020年4月28日 19:47

Sub a()

Dim i

文件名 = "孙兴华"

文件类型 = "Excel文件,\*.xls;\*.xlsx;\*.xlsm,word文件,\*.doc;\*.docx;\*.docm"

标题 = "我是标题"

i = Excel.Application.GetSaveAsFilename(InitialFileName:=文件名, filefilter:=文件类型, FilterIndex:=2, Title:=标题)

Range("A1") = i

End Sub



文件类型的：索引号

## 18.7 【获取路径】FileDialog属性

2020年4月28日 20:45

只需要获取路径，不要文件名

Excel.Application.FileDialog(filedialogtype:=参数)

参数见下表

msoFileDialogFilePicker	允许选择一个文件
msoFileDialogFolderPicker	允许选择一个文件夹
msoFileDialogOpen	允许打开一个文件
msoFileDialogSaveAs	允许保存一个文件

```
Sub test1() '选择文件
Dim dig
Set dig = Application.FileDialog(msoFileDialogFilePicker) '定义对象，指定参数
With dig
.AllowMultiSelect = True '允许多选
.Filters.Add "Excel文件", "*.xls*", 1 '默认文件类型
.InitialFileName = ThisWorkbook.FullName 'd:\ '默认路径
.InitialView = msoFileDialogViewDetails '文件显示方式
.Title = "对话框测试" '对话框标题
If .Show = 0 Then 'show显示对话框，判断返回值
MsgBox "你点了取消"
Else
MsgBox dig.SelectedItems(1) '取得选择的文件名
End If
End With
End Sub
```

注：这里显示对话框用到的是show方法，而show方法是可以返回值的，返回0代表点了取消，否则返回-1。选择的文件名是保存在SelectedItems数组中，通过该数组来调用相应的文件名，选了一个可以单独调用，多个就循环调用。

```
Sub test2() '选择文件夹
Dim dig
Set dig = Application.FileDialog(msoFileDialogFolderPicker) '选择文件夹
With dig
.AllowMultiSelect = True
.InitialFileName = "D:\"
.Title = "对话框测试"
If .Show = 0 Then
MsgBox "你点了取消"
Else
MsgBox dig.SelectedItems(1)
End If
End With
End Sub
```

```
Sub test3() '打开文件
Dim dig
Set dig = Application.FileDialog(msoFileDialogOpen)
With dig
.AllowMultiSelect = True
.Filters.Add "Excel文件", "*.xls*", 1
.InitialFileName = "D:\"
.Title = "对话框测试"
```



```

If .Show = 0 Then      '判断是否点了取消
MsgBox "你点了取消"
Else
.Execute              '执行打开命令
MsgBox "文件已打开"
End If
End With
End Sub

```

注：执行打开命令需要在Show方法之后调用Execute方法，而之前的两个代码是没有Execute方法的，如果在Show之后不执行Execute那这个代码也可以用来返回文件名，和前两个代码没有区别。

```

Sub test4() '另存文件
Dim dig
Set dig = Application.FileDialog(msoFileDialogSaveAs)
With dig
.AllowMultiSelect = True
.InitialFileName = ThisWorkbook.FullName
.Title = "对话框测试"
If .Show = 0 Then      '判断是否点了取消
MsgBox "你点了取消"
Else
.Execute              '执行打开命令
MsgBox "文件已保存"
End If
End With
End Sub

```

## 18.7.1 【通用】文件选择对话框

2020年4月28日 21:14

'选择单个文件对话框

Sub SelectSingleFileDialog()

'通过对话框选择文件

With Application.FileDialog(msoFileDialogFilePicker)

.Title = "Select A File" '选择窗口的标题

.InitialFileName = "D:\TestFolder\TestFile.txt" '初次打开窗口的路径以及默认名称

.AllowMultiSelect = False '是否允许选择多个文件

.Filters.Clear '清除现有规则

.Filters.Add "Text File", "\*.txt" '增加规则

.Filters.Add "EXCEL File", "\*.xlsx; \*.xls", 1 '增加规则到第一位

.Filters.Add "All File", "\*.\*", 1 '增加规则到第一位

If .Show Then '显示文件选择对话框

.ButtonName = "Select Me"

Set ipath = .SelectedItems '获取选择项,无论是否选择一项还是多项, 返回的选项都是多项

End If

End With

If IsEmpty(ipath) Then Exit Sub '如果按取消键, 退出

ipath = ipath(1) '获取第一项选择

Debug.Print ipath '输出选择文件名

End Sub

## 18.7.2 【通用】文件夹选择对话框

2020年4月28日 21:14

'选择一个文件夹

Sub SelectFolderDialog()

'通过对话框选择文件夹

With Application.FileDialog(msoFileDialogFolderPicker)

.Title = "Select Folders"

If .Show Then

    ipath = .SelectedItems

End If

End With

If IsEmpty(ipath) Then Exit Sub     '如果按取消键，退出

Debug.Print ipath(1)             '输出文件夹路径

End Sub

## 18.8 课堂练习

2020年4月28日 21:31

**打开一个或多个文件并写入数据【多选】**

**Sub a()**

**Dim i, w1, arr**

**arr = Excel.Application.GetOpenFilename("Excel文件,\*.xls\*", MultiSelect:=True)**

**If IsArray(arr) Then**

**For i = LBound(arr) To UBound(arr)**

**Set w1 = Workbooks.Open(arr(i))**

**w1.Sheets(1).Range("A1") = 520**

**w1.Save**

**w1.Close**

**Next**

**Else**

**MsgBox "已取消操作! "**

**End If**

**End Sub**

## 18.8.1 多个Excel文件合并

2020年4月28日 23:15

**Sub a()**

**Dim i, w1, arr**

**Set w2 = ActiveWorkbook**

**Set s2 = ActiveSheet**

**arr = Excel.Application.GetOpenFilename("Excel文件,\*.xls\*", MultiSelect:=True)**

**If IsArray(arr) Then**

**For i = LBound(arr) To UBound(arr)**

**Set w1 = Workbooks.Open(arr(i))**

**For Each s1 In w1.Sheets**

**s1.Copy after:=w2.Sheets(w2.Sheets.Count)**

**w2.Sheets(w2.Sheets.Count).Name = Split(w1.Name, ".")(0) & s1.Name**

**Next**

**w1.Close**

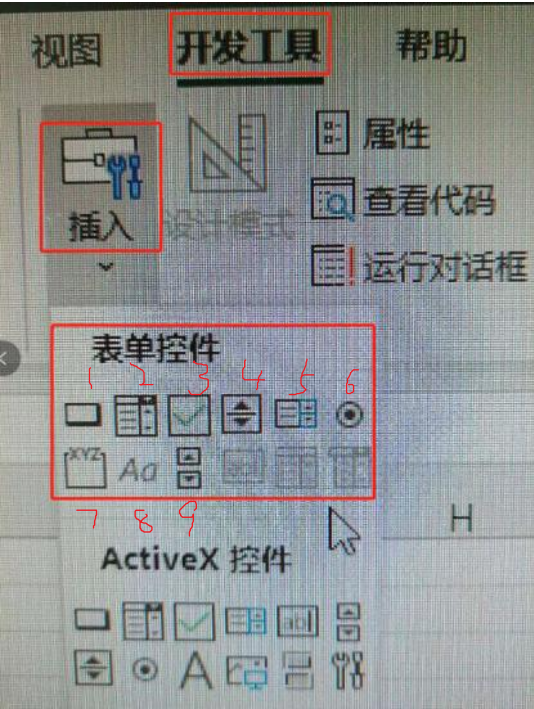
**Next**

**End If**

**End Sub**

19.图形界面设计

2020年4月26日 18:59



序号	控件名称	控件说明
1	按钮	用于执行宏命令
2	组合框	提供可选择的多个选项，用户可以选择其中的一个项目
3	复选框	用于选择的控件，可以多项选择
4	数值调节钮	通过单击控件的箭头来选择数值
5	列表框	显示多个选项的列表，用户可以从选择一个选项
6	选项按钮	用于选择的控件，通常几个选项按钮用组合框组合在一起使用，在一组中只能同时选择一个选项按钮。
7	分组框	用于组合其它多个控件
8	标签	用于输入和显示静态文本
9	滚动条	包括水平滚动条和垂直滚动条

## 19.1 在工作表中使用

2020年4月28日 12:29

**表单控件与ActiveX控件区别：**

**表单控件：只能用来指定宏，只能用在工作表中**

**ActiveX控件：有很多属性和事件,不仅可以用在工作表中，也可以使用在窗体中**

# 19.1.1 表单控件

2020年4月28日 12:29

	A	B	C	D	E	F	G	H	I	J
1	姓名								序号	6
2	丁智敏									
3	李平平			张伊						
4	王刚			丁智敏						
5	卢海军			李平平						
6	王松			王刚						
7	张伊			卢海军						
				王松						
				张伊						



## 19.1.2 ActiveX控件

2020年4月28日 12:29

Private Sub xb1\_Click()

'如果控件xb1已选中则执行if与End if之间的代码

If xb1.Value = True Then

Range("D1") = "男"

xb2.Value = False

End If

End Sub

Private Sub xb2\_Click()

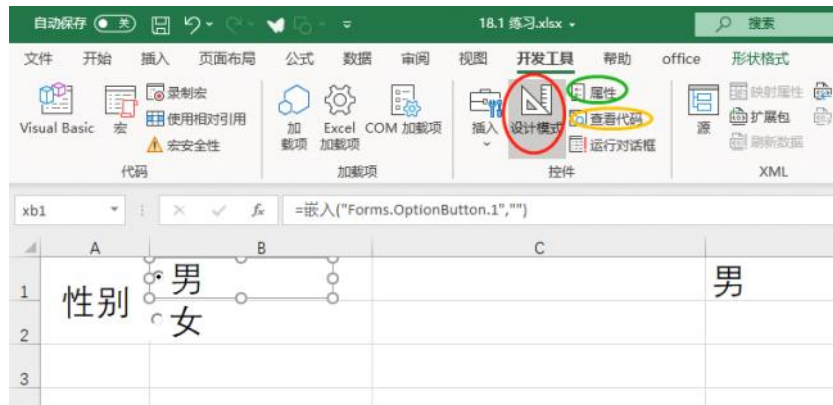
If xb2.Value = True Then

Range("D1") = "女"

xb1.Value = False

End If

End Sub



# 19.2 窗体对象设计交互界面

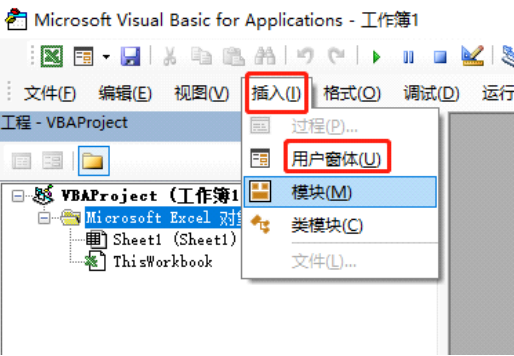
2020年4月29日 16:23

在VBE中有两种方式可以插入窗体

## 1、通过菜单命令插入窗体

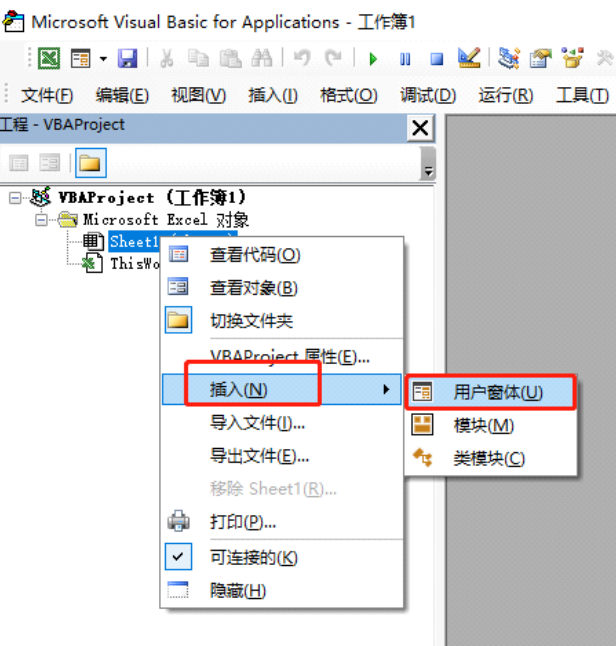
VBE-【插入】-【用户窗体】

无论哪个窗口找不到了，到视图下拉菜单里去找



## 2、利用右键菜单插入窗体

在插入模块那里，选择用户窗体



## 19.2.1 修改窗体外观

2020年4月29日 16:33

### 一、修改大小：用鼠标直接拖拽

### 二、常用窗体属性

属性	说明	默认值
名称	指定对象引用时的名称	
BackColor	控件控件的背景颜色	&H8000000F&
Caption	指定窗体标题栏上显示的文本	UserForm
Enabled	指定窗体或控件能否相应由用户引发的事件	True
Font	指定窗体中字体的样式。	“宋体”、“常规”、“小五”
ForeColor	指定窗体的前景色，如控件上字体的颜色。	&H80000012&
Height	指定窗体的高度，以磅为单位。	无
MouseIcon	指定鼠标指针在窗体中的图标，类型为.ico 的图标文件。	None
MousePointer	指定当用户把鼠标放到特定对象上时，所显示鼠标指针的类型。当值为 99 值，由 MouseIcon 属性指定的图标有效。	0-fmMousePointerDefault
Picture	指定窗体的背景图片	None
PictureAlignment	指定由 Picture 指定的背景图片在窗体中的位置	2-fmPictureAlignmentCenter
PictureSizeMode	指定在窗体中显示图片的方式	0-fmPictureSizeModeClip
PictureTiling	允许在窗体或页面中平铺图片。	False
ScrollBars	指定窗体是否有垂直或水平滚动条	0-fmScrollBarsNone
ShowModal	指定窗体在显示时是模式的还是无模式的。	True
StartPosition	返回或设置一个值，用来指定用户窗体在第一次出现时的位置	1-所有者中心
Width	指定窗体的宽度，以磅为单位。	

## 19.3 添加和设置控件（1）

2020年4月29日 17:09

### 1、【VBE】-【视图】-【工具箱】

### 2、添加控件

**切记：每个控件都要自己命名，方便记忆**

学生成绩管理系统

### 成绩录入窗口

姓名	<input type="text"/>
性别	<input type="text"/>
成绩	<input type="text"/>

### 19.3.1 【用户输入】TextBox文本框

2020年4月29日 17:18

**就是：让用户输入内容**

**常用属性：**

Font	字体、颜色、字号
TextAlign	对齐方式
Enabled	可用性
Visible	可见性
Value	返回值
TabIndex	按下Tab键时的切换顺序（1. 2. 3. 4……代表TAB键时先跳到哪个控件上）

### 19.3.2 【选择菜单】ComboBox复合文本框

2020年4月29日 17:47

就是：让用户选择

常用属性：

Font	字体、颜色、字号
TextAlign	对齐方式

让复选框下拉菜单中有内容：

对着窗体点右键-查看代码

Private Sub UserForm\_Initialize()

txt性别.List = Array("男", "女")

End Sub

常用方法：

AddItem	增加一个下拉项目
RemoveItem	移除一个项目
Clear	全部清除

### 19.3.3 【显示内容】 Lable标签

2020年4月29日 17:25

**Lable标签**：用于窗体上显示一段文本的控件，用于无法手动修改其显示内容。  
**就是：只让用户看，不让用户改**

**常用属性：**

Capt ion	显示在标签中的文字
Font	字体、颜色、字号
TextAl ign	对齐方式

## 19.3.4 【命令按钮】CommandButton

2020年4月29日 17:30

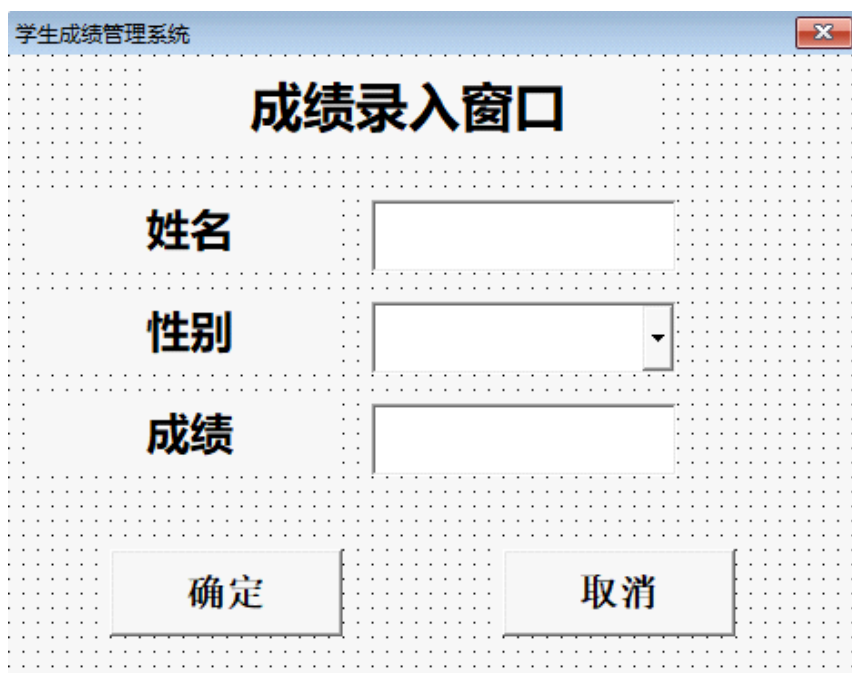
常用属性：

Caption	显示在标签中的文字
Font	字体、颜色、字号
Caption	重命名
Enabled	可用性
Visible	可见性
Height	按钮高度
Width	按钮宽度
Top	按钮离上边线的距离，对齐时使用
Left	按钮离左边线的距离



## 19.3.5 实战

2020年4月29日 17:59



按下确定按钮时，在工作表里录入文本框中的内容，所以我们要对确定按钮写事件

如果，你需要的是对文本框内容修改后触发的事件，那么才对文本框写事件

例如：

按下“确定”按钮（单击时）修改单元格内容

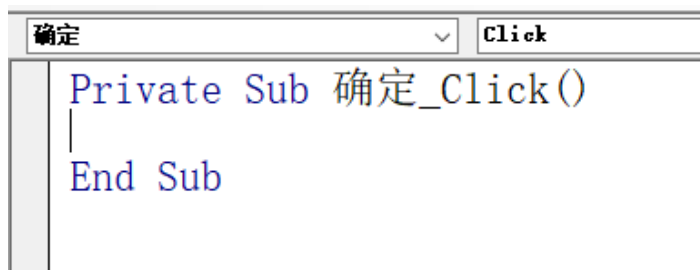
当窗体被单击时，弹出一个消息框

当“成绩”标签被双击时，将成绩显示在文本框中

当“文本框”被修改时，弹出消息框提示警告

当窗体被右键单击时，显示另一个窗体

双击确定按钮，在Click事件中写代码



```
Private Sub 确定_Click()  
End Sub
```

写代码：逐行输入用UsedRange.Row找到最大行号，给最大行号+1

```
Private Sub 确定_Click()  
Dim r1 As Range, i As Long  
Set r1 = Worksheets(1).UsedRange  
'i代表当前表格最后一行数据的下一行  
'起始行+使用区域一共多少行  
i = r1.Row + r1.Rows.Count  
Range("A" & i) = txt姓名.Value  
Range("B" & i) = txt性别.Value  
Range("C" & i) = txt成绩.Value  
'将窗体中输入的数据清除，等待下次输入  
txt姓名.Value = ""  
txt性别.Value = ""  
txt成绩.Value = ""  
End Sub
```

## 19.3.6 窗体的显示位置

2020年4月29日 19:41

默认情况下，显示一个窗体时，Excel会将其显示在Excel窗口的中心位置，但可以通过设置属性来定义其显示的位置。

在模块里面写：

**Sub 显示窗口()**

**With UserForm1**

'属性为0时，可以设置窗体初次显示时的位置由用户定义

**.StartPosition = 0**

'设置窗体顶端离屏幕窗口顶端的距离

**.Top = 100**

'设置窗体左端离屏幕窗口左端的距离

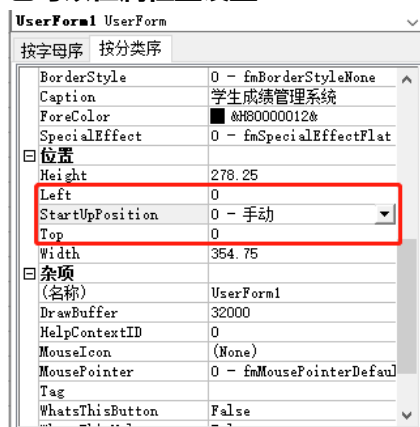
**.Left = 200**

**.Show**

**End With**

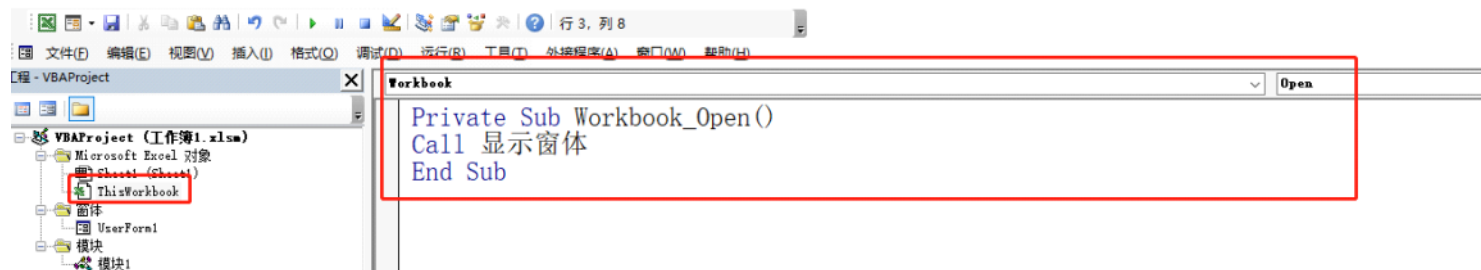
**End Sub**

也可以在属性里设置



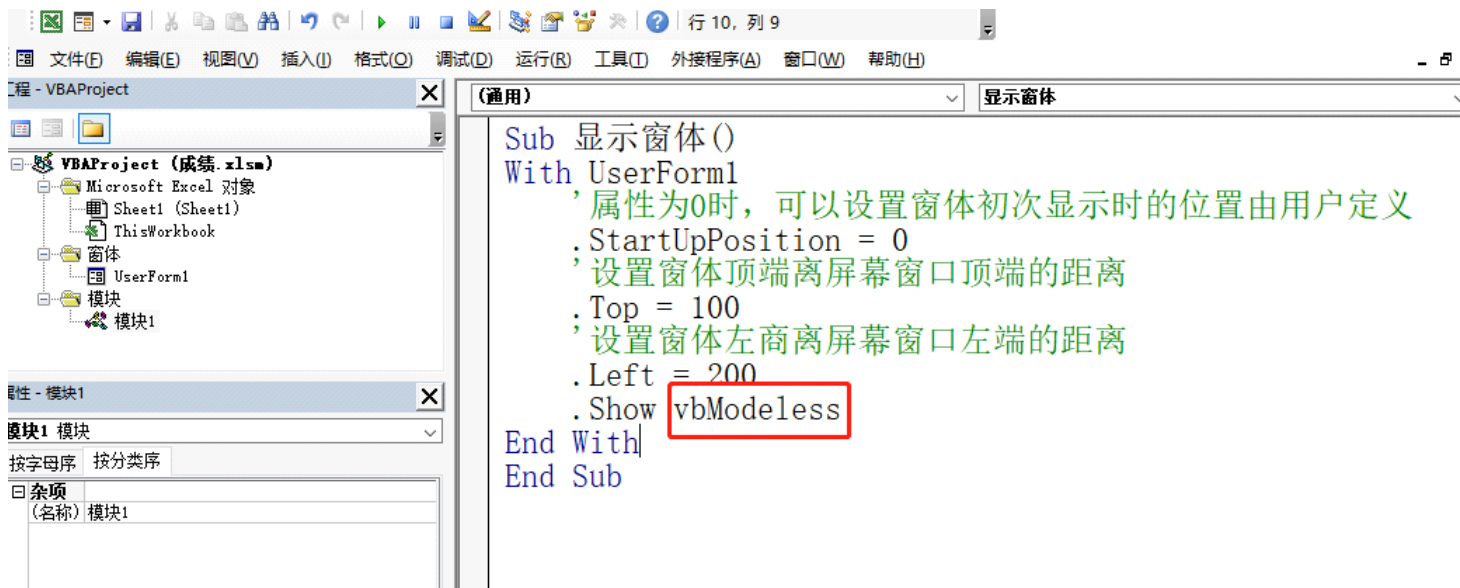
## 19.3.7 文件打开时就自动显示窗体

2020年4月29日 19:48



## 19.3.8 操作窗体之外的内容

2020年4月29日 19:56



## 19.3.9 【显示/隐藏】窗体

2020年4月29日

19:32

**如果是密码登录窗口，我们输入用户名和密码后，需要让窗体消失**

```
Private Sub 确定_Click()  
Dim r1 As Range, i As Long  
Set r1 = Worksheets(1).UsedRange  
' i代表当前表格最后一行数据的下一行  
' 起始行+使用区域一共多少行  
i = r1.Row + r1.Rows.Count  
Range("A" & i) = txt姓名.Value  
Range("B" & i) = txt性别.Value  
Range("C" & i) = txt成绩.Value  
' 将窗体中输入的数据清除，等待下次输入  
txt姓名.Value = ""  
txt性别.Value = ""  
txt成绩.Value = ""  
UserForm1.Hide  
End Sub
```

**UserForm1.Show : 显示窗体**

**UserForm1.Hide : 隐藏窗体**

## 19.3.10 【关闭】窗体

2020年4月29日 19:58

关闭窗体方法：Unload 窗体名称

关闭代码所在的窗体：Unload Me

二者区别：

使用【Unload 窗体名称】可以关闭任意的窗体

使用【Unload Me】只能关闭代码所在的窗体

如果是要关闭代码所在的窗体，使用【Unload Me】关闭会更安全。

因为：如果用【Unload 窗体名称】依赖于窗体名称，如果窗体名称改变了，你就要回来改代码。如果用【Unload Me】无论你的窗体名称怎么变，都不用关心代码问题。

那么【关闭】和【隐藏】有什么区别呢？

**关闭：**不但会从屏幕上删除，还会将其从内存中卸载，当窗体从内存中卸载后，窗体及窗体中的控件都将还原成最初的值，代码将不能操作或访问窗体及其中的控件，也不能再访问保存在窗体中的变量。

**隐藏：**只是从屏幕上消失，等候你再次使用。

就好比一个你输入了很多文字的Word文档，你没保存到硬盘上，它就保存在内存中，你关闭了，就没有了，你最小化，就叫隐藏。

双击【取消】按钮

```
Private Sub 取消_Click()
```

```
Unload Me
```

```
End Sub
```

19.3.11 【禁止关闭窗口】利用Queryclose事件

2020年4月29日 20:12

**警告：**在禁用关闭按钮的用户窗体中，一定要在窗体中设置【退出】按钮，否则用户无法关闭此窗体，只能在Windows任务管理器中结束进程。

需求：让自带的【关闭】按钮失效

窗体-右键-查看代码

```
Private Sub UserForm_Initialize()  
txt性别.List = Array("男", "女")  
End Sub  
  
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)  
If CloseMode = 0 Then Cancel = True  
End Sub  
  
Private Sub 取消_Click()  
Unload Me  
End Sub  
  
Private Sub 确定_Click()  
Dim rl As Range, i As Long  
Set rl = Worksheets(1).UsedRange  
' i代表当前表格最后一行数据的下一行  
' 起始行+使用区域一共多少行  
i = rl.Row + rl.Rows.Count  
Range("A" & i) = txt姓名.Value  
Range("B" & i) = txt性别.Value  
Range("C" & i) = txt成绩.Value
```

Cancel参数确定是否关闭窗口，值为True不关闭，反之关闭。

CloseMode参数是我们关闭窗体的方式，值为0或1

值	说明
0	在窗体中单击【关闭】按钮关闭窗口
1	通过Unload语句关闭窗口

当然这不是唯一方法，我会在《跟着孙兴华学习Excel VBA》第三季或第四季，给大家讲API函数，利用API函数也可以禁用关闭窗口按钮



## 19.3.12 给控件设置快捷键

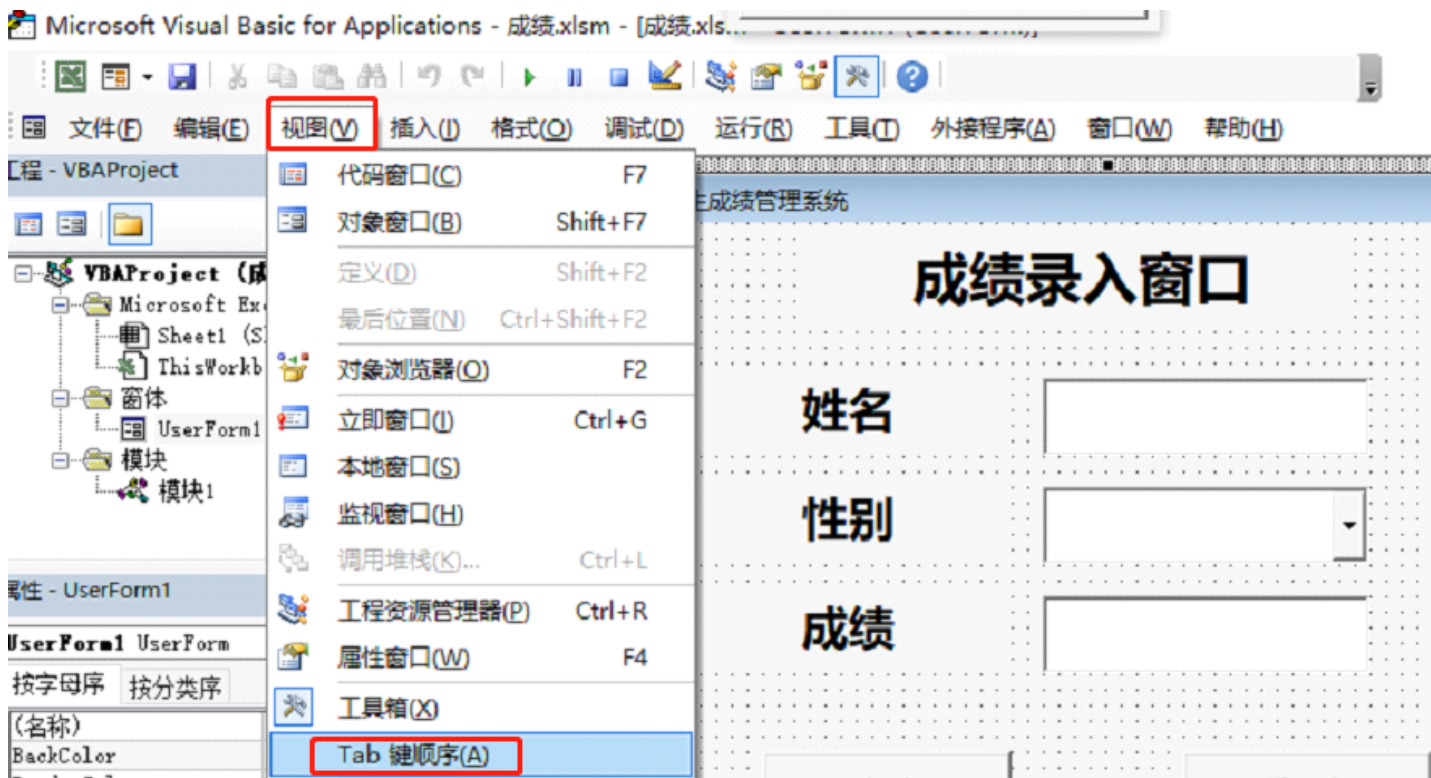
2020年4月29日 20:34

确定 CommandButton	
按字母序	按分类序
(名称)	确定
Accelerator	N
AutoSize	False
BackColor	□ &H8000000F&
BackStyle	1 - fmBackStyleOpaque
Cancel	False
Caption	确定
ControlTipText	
Default	False
Enabled	True
Font	宋体
ForeColor	■ &H80000012&
Height	36
HelpContextID	0
Left	42

以后这个按钮的快捷键就是【Alt+N】

## 19.3.13 更改控件的Tab顺序

2020年4月29日 20:41



### 19.3.14 在表格上双击某一行数据自动弹出窗体

2020年4月29日 20:50

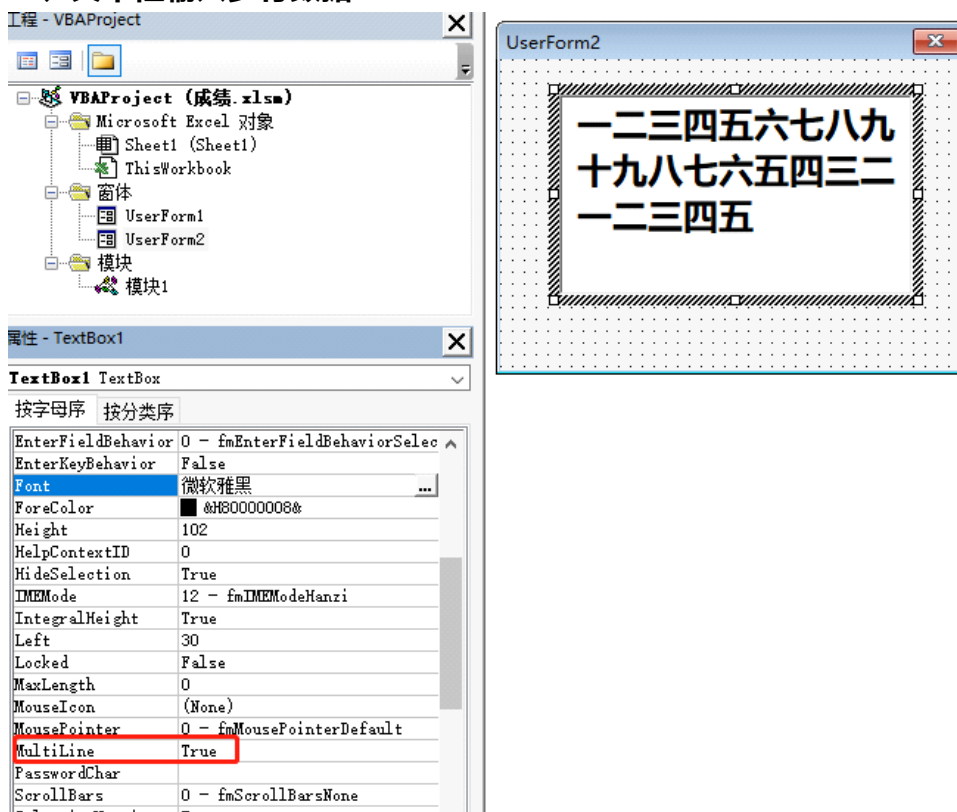
分析：当在工作表中双击单元格时，自动弹出窗体，并且显示这一行的内容



## 19.3.15 【输入多行文本】文本框

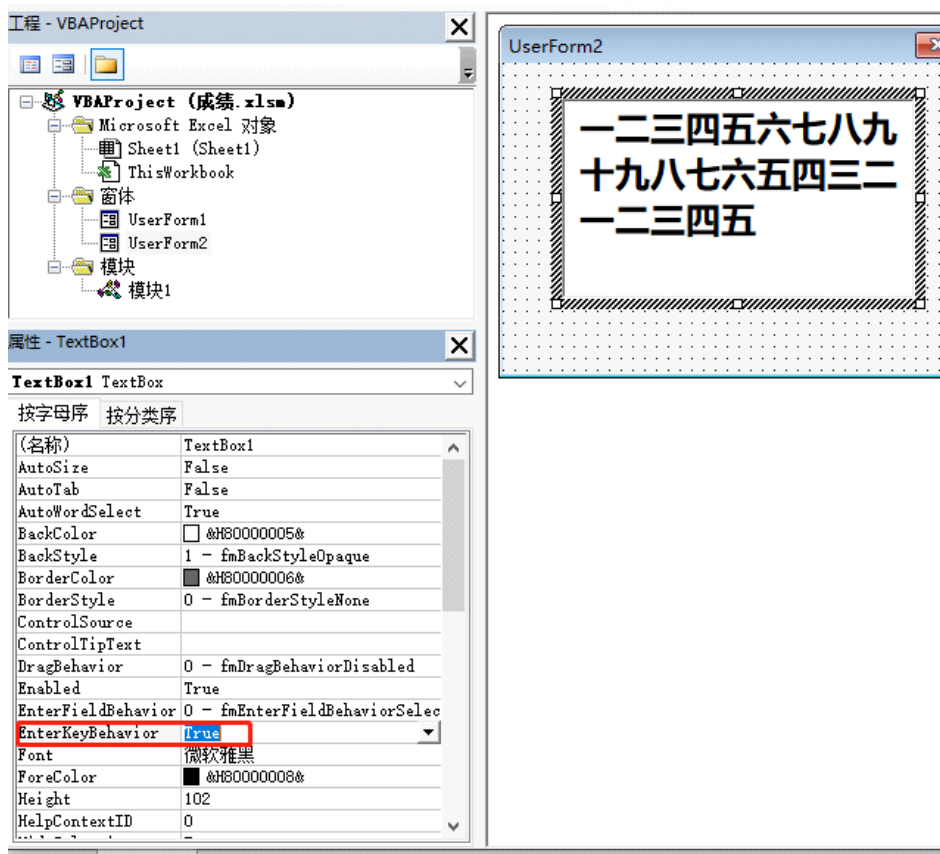
2020年4月29日 21:04

### 一、文本框输入多行数据



但是，默认情况下，文本框中直接按下【回车】后跳出文本框，只有按下Ctrl+回车才能在文本框中换行

如果在这个文本框中想按回车键换行：





## 19.3.15 补充【多行文本的相关参数】

2020年4月29日 21:16

Multiline	True支持换行
EnterKeyBehavior	True支持回车键换行
WordWrap	True代表自动换行
ScrollBars	是否显示滚动条

## 19.4 添加和设置控件（2）

2020年4月29日 21:18

## 19.4.1 【选择题】ListBox列表框

2020年4月29日 21:19

对着窗体点右键-查看代码

```
Private Sub UserForm_Initialize()
```

```
    孙兴华.List = Array("张三", "李四", "王五")
```

```
End Sub
```

一、向列表框中添加项目：

列表.AddItem "新的项目"

二、从列表框中删除项目：

列表.RemoveItem num

num(从0开始计数)

三、被用户选中的项目序号

列表.listindex

从0开始计数，用户未做任何选择，则为-1



```
Private Sub UserForm_Initialize()
```

```
    列表.List = Array("张三", "李四", "王五")
```

```
End Sub
```

```
Private Sub 删除_Click()
```

```
If 列表.ListIndex <> -1 Then
```

```
    列表.RemoveItem 列表.ListIndex
```

```
End If
```

```
End Sub
```

```
Private Sub 添加_Click()
```

```
If 文本.Value <> "" Then
```

```
    列表.AddItem 文本.Value
```

```
    文本.Value = ""
```

```
End If
```

```
End Sub
```

```
Private Sub 显示人名_Click()
```

**避免用户没有选择就点了按钮**

```
If Not IsNull(列表.Value) Then
```

```
    MsgBox 列表.Value
```



**End If**  
**End Sub**

## 19.4.1 【补充】常用属性和方法

2020年4月30日 0:51

<b>Enabled</b>	<b>可用性</b>
<b>Value</b>	<b>返回值</b>
<b>Visible</b>	<b>可见性</b>
<b>List</b>	<b>数据源列表</b>
<b>ColumnCount</b>	<b>列</b>
<b>AddItem</b>	<b>增加一个项目</b>
<b>RemoveItem</b>	<b>移动一个项目</b>
<b>Clear</b>	<b>全部清除</b>
<b>Listindex</b>	<b>选中的序号（从0开始）</b>

## 19.4.1 课堂练习

2020年4月30日 1:03

```
Private Sub UserForm_Initialize()  
列表.List = Range("A2:A7").Value  
End Sub
```

```
Private Sub 显示_Click()  
If 列表.ListIndex <> -1 Then  
    '因为数据从第2行起,而ListIndex从0开始  
    i = 列表.ListIndex + 2  
    数学.Value = Range("B" & i)  
    语文.Value = Range("C" & i)  
    英语.Value = Range("D" & i)  
End If  
End Sub
```

姓名	数学	语文	英语
丁智敏	70	68	96
李平平			
王刚			
张伊			
王松			
卢海军			

去掉显示成绩的按钮，在列表框上选择就显示，这是一个列表框发生选择变化的事件

```
Private Sub UserForm_Initialize()  
列表.List = Range("A2:A7").Value  
End Sub  
  
Private Sub 列表_Change()  
If 列表.ListIndex <> -1 Then  
    '因为数据从第2行起,而ListIndex从0开始  
    i = 列表.ListIndex + 2  
    数学.Value = Range("B" & i)  
    语文.Value = Range("C" & i)  
    英语.Value = Range("D" & i)  
End If  
End Sub
```

## 19.4.2 选项按钮和复选框

2020年4月30日 12:42

他们的Value被选中时为True,否则为False



```
Private Sub 确定_Click()
```

```
Dim 变量1, 变量2
```

```
'因为返回值都是T或F
```

```
If 男.Value Then
```

```
    变量1 = 男.Caption
```

```
Elseif 女.Value Then
```

```
    变量1 = 女.Caption
```

```
End If
```

```
Range("A2") = 变量1
```

```
'注意复选框不要用Elseif因为是多选
```

```
If 协议.Value Then 变量2 = 变量2 & 协议.Caption
```

```
If 体验.Value Then 变量2 = 变量2 & 体验.Caption
```

```
Range("B2") = 变量2
```

```
End Sub
```

总结:

- 1、选项按钮与复选框的Value都是返回T或False, 被选中返回T
- 2、如果想让值等于按钮或选框名字的时候, 例如: 按钮.Caption
- 3、复选框不建议用Elseif, 因为是多选
- 4、默认选项按钮是单选, 复选框是多选
- 5、利用GroupName 组名 可以分组

## 19.4.3 【框架】对按钮分组

2020年4月30日

13:08

性别

☒ 男

☐ 女

党员

☐ 是

☒ 否

## 19.4.4 可用性Ebable和可见性Visible

2020年4月30日 13:14

**可用性Ebable：** 值为F,按钮为灰色不可用

**可见性Visible：** 值为F,按钮被隐藏

启用.Enable=True

禁用.Enable=False

显示.Visible=True

隐藏.Visible=False

## 19.4.5 【旋转按钮】SpinButton

2020年4月30日 14:12

除了可见性，可用性 和 标题之外还多了两个属性

MIN最小值 和 MAX最大值

它的事件中没有单击，只有改变

## 19.4.6 常用窗体事件

2020年4月30日 20:30

UserForm.Activate 激活

UserForm.QueryClose 退出

```
Private Sub UserForm_Activate() '当这个用户窗体被激活时  
Excel.Application.Visible = False '把Excel窗体隐藏  
End Sub
```

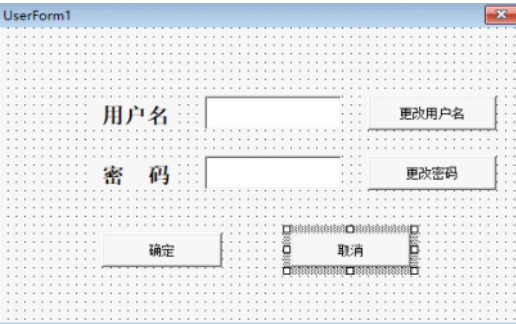
---

```
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer) '用户窗体退出  
Excel.Application.Quit '退出EXCEL  
End Sub
```



19.4.7 【案例1】登录系统

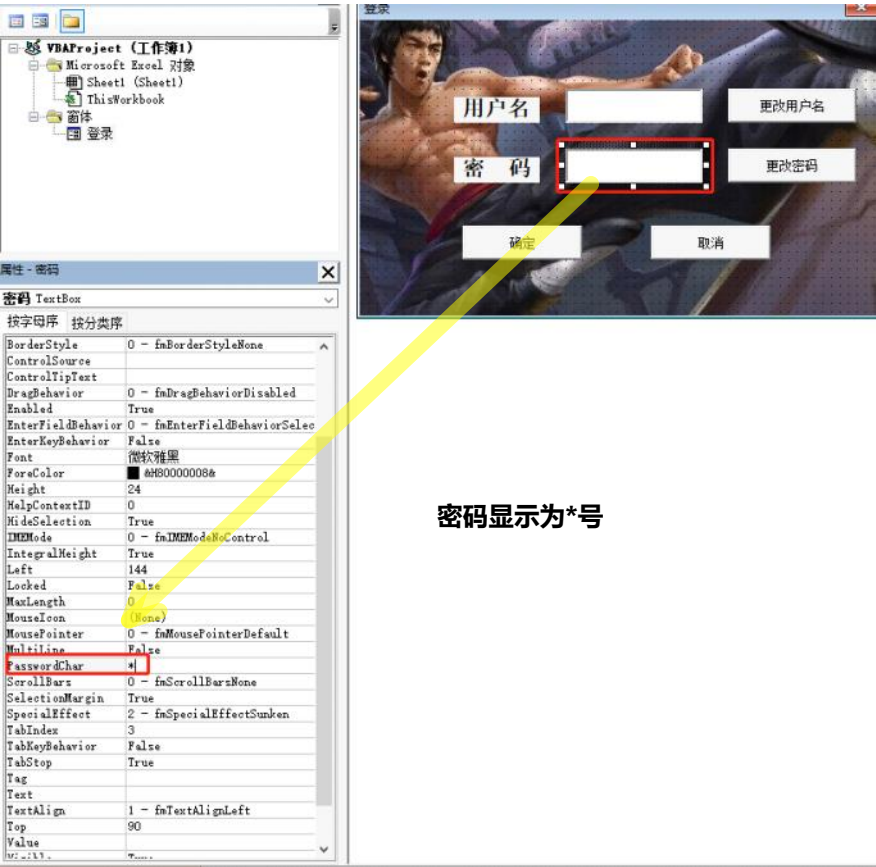
2020年4月30日 14:52



(名称)	登录
BackColor	0 - fmBackColorDefault
BorderColor	0 - fmBorderStyleNone
BorderStyle	0 - fmBorderStyleNone
Caption	登录
Cycle	0 - fmCycleAllForms
DrawBuffer	32000
Enabled	True
Font	宋体
ForeColor	0 - fmForeColorDefault
Height	228.75
HelpContextID	0
KeepScrollBarsVisible	3 - fmScrollBarsBoth
Left	0
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Picture	(Bitmap)
PictureAlignment	2 - fmPictureAlignmentCent
PictureSizeMode	0 - fmPictureSizeModeClip
PictureTiling	False
RightToLeft	False
ScrollBars	0 - fmScrollBarsNone
ScrollHeight	0
ScrollLeft	0
ScrollTop	0
ScrollWidth	0
ShowModal	True
SpecialEffect	0 - fmSpecialEffectFlat
StartUpPosition	1 - 所有者中心
Tag	
Top	0
WhatsThisButton	False

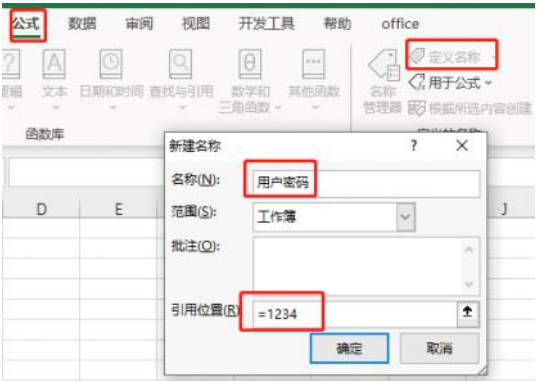
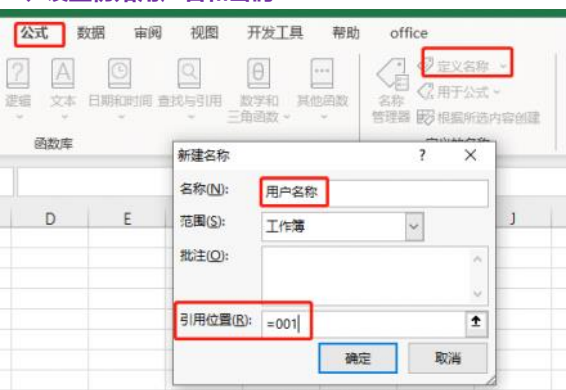


一、设计界面



密码显示为\*号

二、设置初始用户名和密码



### 三、定义公共变量

```
Public j, k
```

### 四、为【确定】按钮添加代码

```
Private Sub 确定_Click()
```

```
'动态变量程序结束后释放，静态变量workbook结束后释放
```

```
Static i
```

```
j = Names("用户名称").RefersTo
```

```
k = Names("用户密码").RefersTo
```

```
'Excel基础篇宏表函数
```

```
If 用户名.Value = Evaluate(j) And CStr(密码.Value) = Evaluate(k) Then
```

```
Unload Me
```

```
Excel.Application.Visible = True
```

```
Else
```

```
'用变量i记录密码或用户名输入错误的次数
```

```
i = i + 1
```

```
If i = 3 Then
```

```
MsgBox "对不起！您的用户名或密码错误！你无权打开工作簿"
```

```
ThisWorkbook.Close
```

```
Else
```

```
MsgBox "输入错误，您还有" & (3 - i) & "次输入机会", vbExclamation, "提示"
```

```
用户名.Value = ""
```

```
密码.Value = ""
```

```
End If
```

```
End If
```

```
End Sub
```

### 五、为【更改用户名】按钮添加代码

```
Private Sub 更改用户名_Click()
```

```
Dim 原用户名, 新用户名1, 新用户名2
```

```
j = Names("用户名称").RefersTo
```

```
原用户名 = InputBox("请输入原用户名：", "提示")
```

```
'判断用户名输入是否正确
```

```
If 原用户名 <> Evaluate(j) Then
```

```
MsgBox "原用户名输入有误，不能修改", vbCritical, "错误"
```

```
Exit Sub
```

```
End If
```

```
新用户名1 = InputBox("请输入新用户名：", "提示")
```

```
'判断新用户名是否为空
```

```
If 新用户名1 = "" Then
```

```
MsgBox "新用户名不能为空，修改不成功", vbCritical, "错误"
```

```
Exit Sub
```

```
End If
```

```
新用户名2 = InputBox("请再次输入新用户名：", "提示")
```

```
'判断两次输入是否相同
```

```
If 新用户名1 = 新用户名2 Then
```

```
Names("用户名称").RefersTo = "=" & Chr(34) & 新用户名1 & Chr(34)
```

```
MsgBox Names("用户名称").RefersTo
```

```
ThisWorkbook.Save
```

```
MsgBox "用户名修改完成，下次登录请使用新用户名！", vbInformation, "提示"
```

```
Else
```

```
MsgBox "两次输入用户名不一致，修改没有完成！", vbCritical, "错误"
```

```
End If
End Sub
```

#### 六、为【更改密码】按钮添加代码

```
Private Sub 更改密码_Click()
Dim 原密码
k = Names("用户密码").RefersTo
原密码 = InputBox("请输入原密码: ", "提示")
'判断原密码是否正确
If Ctr(原密码) <> Evaluate(k) Then
    MsgBox "原密码输入错误, 不能修改! ", vbCritical, "错误"
    Exit Sub
End If
新密码1 = InputBox("请输入新密码:", "提示")
'判断新密码是否为空
If 新密码1 = "" Then
    MsgBox "新密码不能为空, 修改没有完成! ", vbCritical, "错误"
    Exit Sub
End If
新密码2 = InputBox("请再次输入新密码: ", "提示")
'判断两次输入的亲密码是否相同
If 新密码1 = 新密码2 Then
    k = "=" & 新密码1
    ThisWorkbook.Save
    MsgBox "密码修改完成, 下次登录请使用新密码! ", vbInformation, "提示"
Else
    MsgBox "两次输入的密码不一致, 修改没有完成! ", vbCritical, "错误"
End If
End Sub
```

#### 七、禁用关闭按钮

```
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
If CloseMode = 0 Then Cancel = True
End Sub
```

#### 八、为【退出】按钮添加代码

```
Private Sub 退出_Click()
Unload Me
ThisWorkbook.Close
End Sub
```

#### 九、设置工作簿打开时显示登录窗体

```
Private Sub Workbook_Open()
Excel.Application.Visible = False
登录.Show
End Sub
```

拓展：为了不让别人看到名称管理器中保存的用户名和密码：

```
Sub 隐藏密码()

Names("用户名称").Visible = False
Names("用户密码").Visible = False

End Sub
```

## 19.4.8 【案例2】考试系统

2020年4月30日 14:53

### 控件

```
Private Sub AA_Click()  
Range("G" & 题号.Caption + 1) = "A"  
End Sub
```

```
Private Sub BB_Click()  
Range("G" & 题号.Caption + 1) = "B"  
End Sub
```

```
Private Sub CC_Click()  
Range("G" & 题号.Caption + 1) = "C"  
End Sub
```

```
Private Sub DD_Click()  
Range("G" & 题号.Caption + 1) = "D"  
End Sub
```

```
Private Sub UserForm_Activate()  
'隐藏Excel  
Excel.Application.Visible = False  
End Sub
```

```
Private Sub 第一题_Click()  
Call 下一步(1)  
选择.Value = 1  
End Sub
```

```
Private Sub 结束考试_Click()  
Call 答案  
Unload Me  
End Sub
```

```
Private Sub 开始考试_Click()  
Call 确定最大和最小  
Call 下一步(选择.Value)  
End Sub
```

```
Private Sub 选择_Change()  
Call 确定最大和最小
```

### 模块

```
Sub 答案()  
Dim i, r, arr1, arr2  
i = 2  
Set r = Range("F" & Rows.Count).End(xlUp)  
j = r.Row  
arr1 = Range("F2:F" & j)  
arr2 = Range("G2:G" & j)  
Do While Range("F" & i) <> ""  
    If arr1(i - 1, 1) = arr2(i - 1, 1) Then  
        a = a + 1  
    End If  
    i = i + 1  
Loop  
MsgBox "你一共做对了" & a & "道题"  
End Sub
```

```
Sub 确定最大和最小()  
Dim j  
'确定最大最小值  
j = Sheet1.UsedRange.Rows.Count - 1  
考试.选择.Min = 1  
考试.选择.Max = j  
End Sub
```

```
Sub 下一步(i)  
Call 确定最大和最小  
With 考试  
    '初始显示  
    .题号.Enabled = True  
    .题目.Enabled = True  
    .AA.Enabled = True  
    .BB.Enabled = True  
    .CC.Enabled = True  
    .DD.Enabled = True  
    .a.Enabled = True  
    .b.Enabled = True  
    .c.Enabled = True  
    .d.Enabled = True
```

```
Private Sub 选择_Change()
```

```
Call 确定最大和最小
```

```
Call 下一步(选择.Value)
```

```
End Sub
```

```
Private Sub 最后一题_Click()
```

```
Dim k
```

```
k = Sheet1.UsedRange.Rows.Count - 1
```

```
Call 下一步(k)
```

```
选择.Value = k
```

```
End Sub
```

```
Private Sub Workbook_Open()
```

```
考试.Show
```

```
End Sub
```

```
.c.Enabled = True
```

```
.d.Enabled = True
```

```
.题号.Caption = Sheet1.UsedRange.Row
```

```
.题目.Caption = Range("A2")
```

```
.开始考试.Enabled = False
```

```
.结束考试.Enabled = True
```

```
.第一题.Enabled = True
```

```
.最后一题.Enabled = True
```

```
.选择.Enabled = True
```

'每次清空4个选项

```
.AA.Value = False
```

```
.BB.Value = False
```

```
.CC.Value = False
```

```
.DD.Value = False
```

'写入

```
.题号.Caption = i
```

```
.题目.Caption = Sheet1.Range("A" & i + 1)
```

```
.a.Caption = Sheet1.Range("B" & i + 1)
```

```
.b.Caption = Sheet1.Range("C" & i + 1)
```

```
.c.Caption = Sheet1.Range("D" & i + 1)
```

```
.d.Caption = Sheet1.Range("E" & i + 1)
```

'隐藏C和D选项

```
If .c.Caption = "" Then
```

```
.CC.Visible = False
```

```
.DD.Visible = False
```

```
Else
```

```
.CC.Visible = True
```

```
.DD.Visible = True
```

```
End If
```

'返回之前的答案

```
If Range("G" & i + 1) = "A" Then
```

```
.AA.Value = True
```

```
Elseif Range("G" & i + 1) = "B" Then
```

```
.BB.Value = True
```

```
Elseif Range("G" & i + 1) = "C" Then
```

```
.CC.Value = True
```

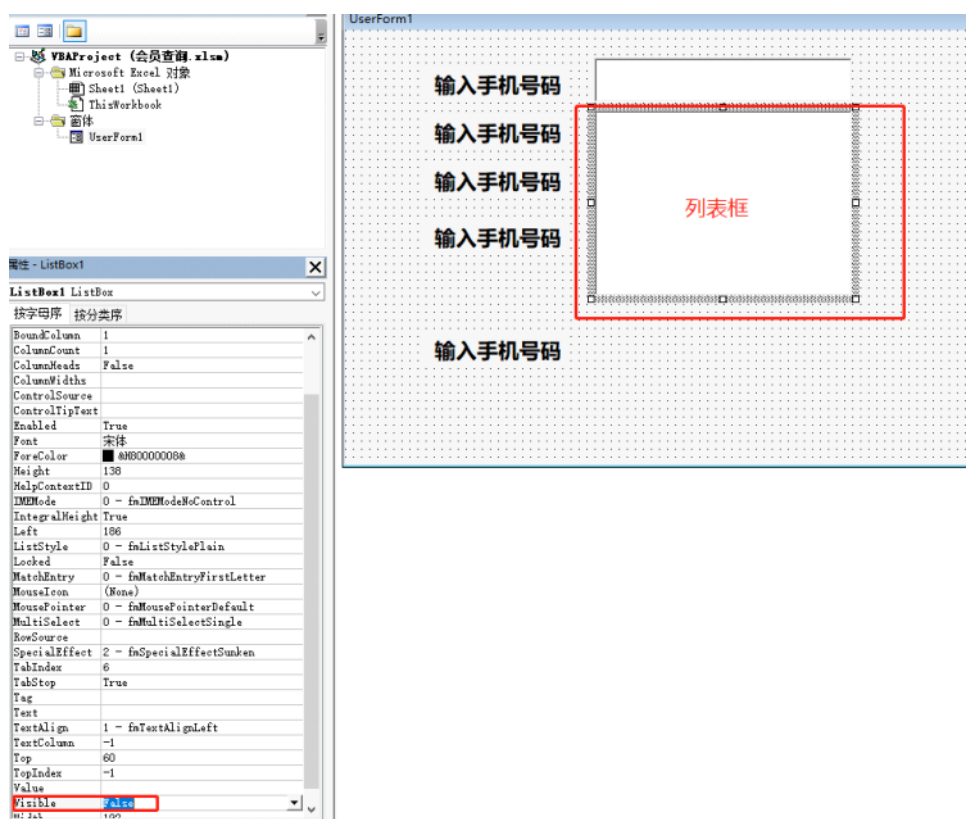
```
Elseif Range("G" & i + 1) = "D" Then
```

```
.DD.Value = ture
```

**End If**  
**End With**  
**End Sub**

## 19.4.9 【案例3】会员信息查询系统

2020年4月30日 20:31



### '第一步

```
Private Sub 手机号_Change()
```

```
Dim arr, r, j, i
```

```
Set r = Range("I" & Rows.Count).End(xlUp)
```

```
j = r.Row
```

```
If Len(手机号.Value) >= 4 Then
```

```
'Me就是我的意思。如果你的代码写在worksheets之下,
```

```
'则me代表这个工作表, 如果代码写在workbook里面则me代表thisworkbook,
```

```
'如果代码写在自定义窗体下, 则me代表这个窗体。
```

```
Me.列表.Clear
```

```
arr = Sheet1.Range("I2:I" & j)
```

```
For i = LBound(arr) To UBound(arr)
```

```
    If InStr(arr(i, 1), Me.手机号.Value) > 0 Then
```

```
        Me.列表.AddItem arr(i, 1)
```

```
    End If
```

```
Next
If Me.列表.ListCount > 0 Then
    Me.列表.Visible = True
Else
    Me.列表.Clear
    Me.列表.Visible = False
End If
End If
End Sub
```

### '第二步

```
Private Sub 列表_Click()
Me.手机号 = Me.列表.Value
Me.列表.Visible = False
End Sub
```

### '第三步

```
Private Sub 查询_Click()
Dim arr, r, j, k
Set r = Range("I" & Rows.Count).End(xlUp)
j = r.Row
arr = Sheet1.Range("I2:I" & j)
For i = LBound(arr) To UBound(arr)
    If arr(i, 1) = Int(手机号.Value) Then
        k = i + 1
    End If
Next
会员编码.Caption = Range("A" & k)
办理日期.Caption = Range("B" & k)
姓名.Caption = Range("C" & k)
性别.Caption = Range("D" & k)
```



```
出生日期.Caption = Range("E" & k)
省.Caption = Range("F" & k)
市.Caption = Range("G" & k)
地址.Caption = Range("H" & k)
End Sub
```

## 19.5 添加和设置控件（3）

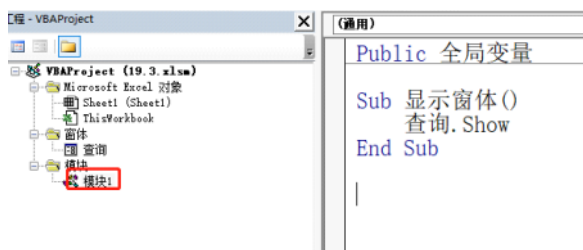
2020年4月29日 21:18

## 19.5.1 使用全局变量互通消息

2020年5月1日 23:17

```
Worksheet BeforeDoubleClick
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
Dim i
'得到用户当前点击的单元格所在的行号
i = Target.Row
'将该行各单元格的信息分别显示到窗体上
查询.姓名.Value = Range("A" & i)
查询.数学.Value = Range("B" & i)
查询.语文.Value = Range("C" & i)
查询.英语.Value = Range("D" & i)
'显示窗体
查询.Show vbModeless
End Sub
```

在单元格上双击成绩，就显示在窗体上，但是现在我需要的是点修改，可以修改单元格内容



m1

	数学	
	语文	
	英语	
<input type="button" value="修改成绩"/>		

工作表事件把内容放到模块，窗体再从模块中把内容拿走，模块只是负责保管

全局变量（公共变量）：

特点：可以被所有模块中的各个过程调用，读取，因此可以实现不同模块之间的数据交流。

用法：

- (1) 不能声明在过程或函数内部
- (2) 使用Public,表示其为公共变量

```
Worksheet BeforeDoubleClick
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
Dim i
'得到用户当前点击的单元格所在的行号
i = Target.Row
' Target和i都只存在于本子过程中，其它模块（包括窗体）无法读取其数值
' 我们把i存到了全局变量，之后VBA中所有工程都可以访问
模块1.全局变量 = Target.Row

' 将该行各单元格的信息分别显示到窗体上
查询.姓名.Value = Range("A" & i)
查询.数学.Value = Range("B" & i)
查询.语文.Value = Range("C" & i)
查询.英语.Value = Range("D" & i)
'显示窗体
查询.Show vbModeless
End Sub
```

最后一步，给修改成绩按钮加事件

```
修改成绩
Private Sub 修改成绩_Click()
Dim i
i = 模块1.全局变量
'将各个文本框中的内容输出到第i行单元格中
Range("B" & i) = 查询.数学.Value
Range("C" & i) = 查询.语文.Value
Range("D" & i) = 查询.英语.Value
Unload Me
End Sub
```

## 19.5.2 文本框Text与Value属性

2020年5月1日 23:17

```
修改成绩 Click
Private Sub 修改成绩_Click()
Dim i
i = 模块1.全局变量
'将各个文本框中的内容输出到第i行单元格中
Range("B" & i) = 数学.Value
Range("C" & i) = 语文.Value
Range("D" & i) = 英语.Value
Range("E" & i) = Int(数学.Value) + Int(语文.Value) + Int(英语.Value)
Unload Me
End Sub
```

**Text与Value返回都是字符串，用+号代表连接，所以必需转换数据类型**

## 19.5.3 附加控件

2020年5月1日 23:17

### 【工具箱】-右键-【附加控件】

## 19.5.4 ActiveX控件同样可以用于工作表

2020年5月1日 23:17

**右键-属性** 可以修改属性

**双击** 可以写代码，不过代码是写在工作表事件中的

**思考一下：灯的开关，有必要装两个吗？一个开，一个关  
是不是按一下开，再按一下关？**

**如果是普通的按钮，你需要两个，如果是ActiveX你就可以设置事件，按一下执行开的工作并且按钮标题变成“开”，再按一下，执行关的操作，按钮标题变成“关”**

## 19.5.5 显示图片

2020年4月30日 13:22

属性:

**Picture:**要显示的图片对象。

**Picturesize:**设定图片显示模式，尺寸，拉伸，自适应

图片.Picture = **LoadPicture**("C:\Users\孙艺航\Desktop\美元.jpg")

## 19.5.6 课程练习

2020年5月1日 23:17

A	B	C	D	E
商品名称	销售金额		计价单位	RMB
西红柿	2823.60		汇率	5.7
土豆	2739.20			
黄瓜	3677.20			
茄子	1700.00			
西瓜	2610.00			
苹果	5755.20			
香蕉	5409.60			

**Sub 转美元()**

**Dim i, j**

**If Range("E1") = "RMB" Then**

**j = Range("E2")**

**i = 2**

**Do While Range("A" & i) <> ""**

**Range("B" & i) = Range("B" & i) / j**

**i = i + 1**

**Loop**

**Range("E1") = "USD"**

**End If**

**End Sub**

**Sub 转人民币()**

**Dim i, j**

**If Range("E1") = "USD" Then**

**j = Range("E2")**

**i = 2**

**Do While Range("A" & i) <> ""**

**Range("B" & i) = Range("B" & i) \* j**

**i = i + 1**

**Loop**

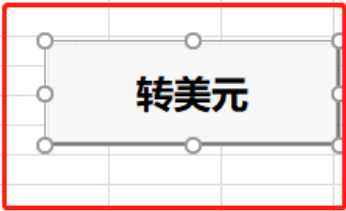
**Range("E1") = "RMB"**



End If  
End Sub

	A	B	C	D	E	F	G	H	I
1	商品名称	销售金额		计价单位	RMB				
2	西红柿	2823.60		汇率	5.7		转美元		
3	土豆	2739.20					转人民币		
4	黄瓜	3677.20							
5	茄子	1700.00							
6	西瓜	2610.00							
7	苹果	5755.20							
8	香蕉	5409.60							
9									
10									
11									
12									
13									
14									

右键属性可以修改



双击后写代码

```
Private Sub 按钮_Click()  
    If Range("E1") = "RMB" Then  
        模块1.转美元  
        按钮.Caption = "转美元"  
    ElseIf Range("E1") = "USD" Then  
        模块1.转人民币  
        按钮.Caption = "转人民币"  
    End If  
End Sub
```

	A	B	C	D	E	F	G	H
1	商品名称	销售金额		计价单位	USD			
2	西红柿	495.37		汇率	5.7		转美元	
3	土豆	480.56					转人民币	
4	黄瓜	645.12						
5	茄子	298.25						
6	西瓜	457.89						
7	苹果	1009.68						
8	香蕉	949.05						
9								
10								
11								
12								



按钮	Click
<pre>Private Sub 按钮_Click()     If Range("E1") = "RMB" Then         模块1.转美元         按钮.Caption = "转美元"         图片.Picture = LoadPicture("C:\Users\孙艺航\Desktop\美元.jpg")     ElseIf Range("E1") = "USD" Then         模块1.转人民币         按钮.Caption = "转人民币"         图片.Picture = LoadPicture("C:\Users\孙艺航\Desktop\人民币.jpg")     End If End Sub</pre>	

## 20. 正则表达式

2020年5月1日 23:17

1. 正则表达式中的每个字符，如果**不是元字符**，就会被**视做文字**。

孙兴华 13901234567 男 2020/05/02 张三 13811118888 李四 13801234567 男 2020/05/02 张三 13811118888 孙兴华 13901234567 男 2020/05/02 张三 13811118888 李四 13801234567



孙兴华 13901234XXXX 男 2020/05/02 张三 13811118XXXX 李四 13801234XXXX 男 2020/05/02 张三 13811118XXXX 孙兴华 13901234XXXX 男 2020/05/02 张三 13811118XXXX 李四 13801234XXXX



## 20.1 元字符与特殊字符

2020年5月1日 23:17

元字符	描述
✓ .	句号匹配任意单个字符除了换行符
✓ []	字符种类，匹配方括号内的任意字符， <b>中括号内每个字符是或(or)的关系</b>
✓ [^]	否定的字符种类，匹配除了方括号里的任意字符
✓ *	匹配0次或无限次，重复在*号之前的字符
✓ +	匹配1次或无限次，重复在+号之前的字符
✓ ?	匹配0次或1次，重复在?号之前的字符
✓ {n}	正好出现n次
✓ {n,m}	匹配num个大括号之前的字符，出现n到m次 ( $n \leq \text{num} \leq m$ )
✓ (xyz)	字符集又称做 <b>组</b> ，匹配与xyz完全相等的字符串， <b>每个字符是且(and)的关系</b>
✓	或运算符，匹配符号前或后的字符
✓ \	转义字符，用于匹配一些保留字符 [], ( ), { }, . , * , + , ? , ^ , \$ , \ ,
✓ ^	从字符串开始位置开始匹配
✓ \$	从字符串末端开始匹配

### 反斜杠后面跟普通字符实现特殊功能

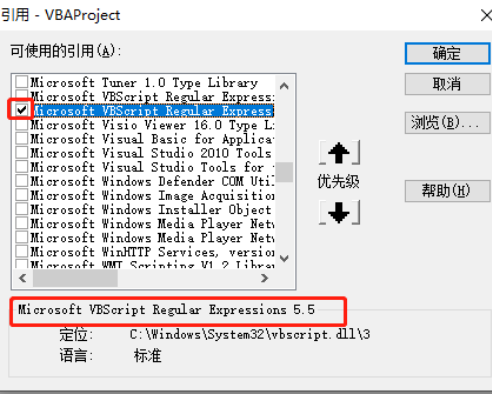
特殊字符	描述
✓ \d	匹配数字，相当于[0-9]
✓ \D	不匹配数字，相当于[^0-9]
✓ \s	匹配空白字符(包括空格、换行符、制表符等)，相当于 [\t\n]

		<code>\r\f\v]</code>
<input checked="" type="checkbox"/>	<code>\S</code>	与 <code>\s</code> 相反，相当于 <code>[^\t\n\r\f\v]</code>
<input checked="" type="checkbox"/>	<code>\w</code>	匹配中文，下划线，数字，英文，相当于 <code>[a-zA-z0-9_]</code>
<input checked="" type="checkbox"/>	<code>\W</code>	与 <code>\w</code> 相反，匹配特殊字符，如 <code>\$、&amp;、空格、\n、\t</code> 等

## 20.2 在VBA中使用正则表达式

2020年5月1日 23:17

**正则表达式 (Regular Expression)**，常被用来检验、替换符合某个模式(规则)的字符串，在代码中常缩写为`regexp`。  
**VBA中并没有直接提供正文表达式对象，需要借助VBScript的正则对象。如果使用前期绑定，那么就需要在VBE中引用如下扩展库。**



20.2.1 后期绑定，实现查找

2020年5月1日 23:17

也可以使用后期绑定的方式创建正则对象，代码如下。  
Set objRegExp = CreateObject("vbscript.regexp")

正则对象有四个属性和三个方法。

属性	说明	方法	说明
Global	Boolean值，指明单次匹配或者全部匹配	Execute	将正则模式应用于字符串，并返回Matches集合
IgnoreCase	Boolean值，指明匹配模式是否大小写敏感	Replace	替换正则模式匹配的文本
Pattern	指定用于搜索的正则模式	Test	对字符串执行正则匹配，返回Boolean值说明匹配是否成功
Multiline	指定的搜索字符串分布在多行		

待处理的文本如下，现需要提取其中的姓名汉字和电话数字。

Name:孙兴华,Phone:13901234567

A	B	C	D
字符串	姓名	电话	姓名电话
Name:孙兴华,Phone:13901234567	孙兴华	13901234567	孙兴华13901234567

```
Sub RegExpDemoSyntax()
    Dim 正则, 结果集合, 结果
    字符串 = Range("A2").Value
    '给对象指定正则表达式对象
    'CreateObject函数用于创建各种外部对象，对象的完整名称就是参数
    Set 正则 = CreateObject("vbscript.regexp")
    'Pattern后面写正则表达式
    正则.Pattern = "Name:(.*?),Phone:(\d+)"
    'Global值为True返回所有符合要求的结果，反之只返回第一个符合要求的结果
    正则.Global = True
    'Execute(字符串)
    Set 结果集合 = 正则.Execute(字符串)
    If 结果集合.Count > 0 Then
        i = 2
        For Each 结果 In 结果集合
            Range("B" & i) = 结果.submatches(0)
            Range("C" & i) = 结果.submatches(1)
            Range("D" & i) = 正则.Replace(字符串, "$1$2")
            i = i + 1
        Next
    End If
    Set 正则 = Nothing
    set 结果集合 = Nothing
End Sub
```

## 20.2.2 替换

2020年5月1日 23:17

匹配模式	含义
[^]	匹配除了叹号之外的一个字符，用于剔除已经指定工作表的单元格引用
[A-Z]{1,3}	1个至3个大小字母，匹配列标A到XFD，这里不是严格的匹配，例如ZZZ也可以被匹配，但是这不是合法的单元格引用
\d{1,6}	1位至6位数字，匹配行号，同上也不是严格的匹配
([A-Z]{1,3}\d{1,6})	圆括号代表提取匹配的字符组，也就是单元格引用

	A	B	C
1	销售表	A3=A8+A9	销售表!A3=销售表!A8+销售表!A9
2	销售表	B3=B4+B7	销售表!B3=销售表!B4+销售表!B7
3	销售表	C4=A2+B5	销售表!C4=销售表!A2+销售表!B5
4	产品表	J3=B8+C2	产品表!J3=产品表!B8+产品表!C2
5	产品表	B3=B4+B7	产品表!B3=产品表!B4+产品表!B7
6	店铺表	B3=B4+B7	店铺表!B3=店铺表!B4+店铺表!B7
7	店铺表	B3=B4+B7	店铺表!B3=店铺表!B4+店铺表!B7
8			

### Sub RegExpDemoReplace()

Dim 正则, i

'使用后期绑定正则对象

Set 正则 = CreateObject("vbscript.regexp")

'正则表达式

正则.Pattern = "[^!]\*([A-Z]{1,3}\d{1,6})"

正则.Global = True

i = 1

Do While Range("A" & i) <> ""

'在公式前添加一个空格，确保第一个单元格引用可以被正则匹配成功

公式 = " " & Range("B" & i)

表名 = Range("A" & i) & "!"

Set 结果集合 = 正则.Execute(公式)

'成功匹配的对象（字符组）数目

If 结果集合.Count > 0 Then

For j = 1 To 结果集合.Count

单元格 = 结果集合(j - 1).submatches(0)

公式 = Replace(公式, 单元格, 表名 & 单元格)

Next



```

Range("C" & i) = Trim(公式)
End If
i = i + 1
Loop
Set 正则 = Nothing
Set 结果集合 = Nothing
End Sub

```

```

或
Sub a()
Dim 正则, 字符串, 表名, 结果集合, j, 子串, i
Set 正则 = CreateObject("vbscript.regexp")
正则.Pattern = "[A-Z]{1,3}[0-9]{1,7}"
正则.Global = True
i = 1
Do While Range("A" & i) <> ""
    字符串 = Range("B" & i)
    表名 = Range("A" & i) & "!"
    Set 结果集合 = 正则.Execute(字符串)
    If 结果集合.Count > 0 Then
        For j = 1 To 结果集合.Count
            子串 = 结果集合(j - 1).submatches(0)
            字符串 = Replace(字符串, 子串, 表名 & 子串)
        Next
        Range("C" & i) = Trim(字符串)
    End If
    i = i + 1
Loop
Set 正则 = Nothing
Set 结果集合 = Nothing
End Sub

```

## 20.2.3 特殊计算

2020年5月3日 1:24

	A	B	C
1	序号	包装明细	包装个数
2	1	50*1	
3	2	50*6+100*3	
4	3	100*3+115*1	
5	4	50*1+100*2+60*1+70*3	
6	5	600*3	
7	6	400*2+300*1+400*2	
8			

匹配模式	含义
\d+	至少1位数字，匹配包装规格
\*	匹配乘号，由于在正则表达式中星号具有特殊含义，因此需要使用反斜杠进行转义

Sub RegExpDemo()

Dim 正则

'使用后期绑定创建正则对象

Set 正则 = CreateObject("vbscript.regexp")

'指定正则匹配字符串

正则.Pattern = "\d+\\*"

'设置为全局搜索模式

正则.Global = True

For Each 单元格 In Range([B2], Cells(Rows.Count, "B").End(xlUp))

新字符串 = 正则.Replace(Trim(单元格.Value), "")

单元格.Offset(0, 1).Value = Application.Evaluate(新字符串)

**Next**

**Set 正则 = Nothing**

**End Sub**

或

```
Sub a()  
Dim 正则, i, 字符串, 表名, 结果集合, j  
Set 正则 = CreateObject("vbscript.regexp")  
正则.Pattern = "\d+\*"   
正则.Global = True  
i = 2  
Do While Range("A" & i) <> ""  
    字符串 = Range("B" & i).Value  
    Set 结果集合 = 正则.Execute(字符串)  
    If 结果集合.Count > 0 Then  
        For j = 1 To 结果集合.Count  
            子串 = 结果集合(j - 1)  
            字符串 = Replace(字符串, 子串, "")  
        Next  
        Range("C" & i) = Application.Evaluate(字符串)  
    End If  
    i = i + 1  
Loop  
Set 正则 = Nothing  
Set 结果集合 = Nothing  
  
End Sub
```

## 20.2.4 提取日期和金额

2020年5月2日 14:48

匹配模式	含义
(\d{4}-\d{2}-\d{2}) \d{4}.\d{2}.\d{2})	用于匹配两种日期格式，竖线代表“或者”，最外侧的圆括号代表提取匹配的字符组
.*?	匹配日期和金额之间的字符
(([A-Z]{3})*\d+[\d.,]*元)	匹配金额部分，圆括号含义与上面相同
([A-Z]{3})*	匹配三个字母的货币简称，也可以没有
\d+[\d.,]*元	一个或者多个数字、小数点、千分位

	A	B	C
1	数据	日期	金额
2	于2020-05-04收到520元	2020/5/4	520元
3	孙兴华2020-05-03上交USD88.88元	2020/5/3	USD88.88元
4	2020.03.15补发3,000元	2020.03.15	3,000元
5	记录2020-02-15转帐3,999,000元	2020/2/15	3,999,000元
6			

Sub a()

Dim 正则

Set 正则 = CreateObject("vbscript.regexp")

正则.Pattern = "(\d{4}-\d{2}-\d{2}|\d{4}.\d{2}.\d{2}).

\*?(([A-Z]{3})\*\d+[\d.,]\*元)"

正则.Global = True

i = 2

Do While Range("A" & i) <> ""

字符串 = Range("A" & i).Value

Set 结果集合 = 正则.Execute(字符串)

If 结果集合.Count > 0 Then

如果你不理解结果集合(0)你就用For each

Range("B" & i) = 结果集合(0).submatches(0)

Range("C" & i) = 结果集合(0).submatches(1)

End If

i = i + 1

**Loop**

**Set 正则 = Nothing**

**Set 结果集合 = Nothing**

**End Sub**

## 20.2.5 提取中文

2020年5月2日 23:22

正则有两种写法:

[^一-龠]

[^\u4e00-\u9fa5]

龠頁

```
Sub a()
```

```
Dim 正则
```

```
Set 正则 = CreateObject("vbscript.regexp")
```

```
正则.Pattern = "[^一-龠]"
```

```
正则.Global = True
```

```
For Each 单元格 In Range([A2], Cells(Rows.Count,  
1).End(xlUp))
```

```
    字符串 = Trim(单元格.Value)
```

```
    单元格.Offset(0, 1).Value = 正则.Replace(字符串, "")
```

```
Next
```

```
Set 正则 = Nothing
```

```
Set 结果集合 = Nothing
```

```
End Sub
```

## 20.2.6 重复字符分组

2020年5月2日 23:42

匹配模式	含义
<code>([a-z])</code>	匹配单个小写英文字符，并提取为第一组
<code>?!</code>	零宽度否定顺序环视，不消耗字符，只用于环视判断
<code>\1</code>	第一组匹配字符（一个或者多个）
<code>\$</code>	行的结束标识

```
Sub a()
```

```
Dim 正则, arr
```

```
Set 正则 = CreateObject("vbscript.regexp")
```

```
正则.Pattern = "([a-z])(?!\\1|$)"
```

```
正则.Global = True
```

```
For Each 单元格 In Range([A1], Cells(Rows.Count, "A").End(xlUp))
```

```
    字符串 = 单元格.Value
```

```
    正则替换 = 正则.Replace(字符串, "$1 ")
```

```
    arr = Split(正则替换)
```

```
    Cells(单元格.Row, 2).Resize(1, UBound(arr) + 1).Value = arr
```

```
Next
```

```
Set 正则 = Nothing
```

```
End Sub
```

或

旧方法

```
Sub RegExpDemo()
```

```
    Dim strTxt As String
```

```
    Dim objRegExp As Object, objMatch As Object
```

```
    Dim objMH As Object, c As Range
```

```
    Set objRegExp = CreateObject("vbscript.regexp")
```

```
    objRegExp.Global = True
```

```
objRegex.Pattern = "[a-z]\1*"
Range("B:AA").ClearContents
For Each c In Range([A1], Cells(Rows.Count, "A").End(xlUp))
    strTxt = c.Value
    Set objMatch = objRegex.Execute(strTxt)
    If objMatch.Count > 0 Then
        col = 2
        For Each objMH In objMatch
            Cells(c.Row, col) = objMH.Value
            col = col + 1
        Next
    End If
Next
Set objMH = Nothing
Set objMatch = Nothing
Set objRegex = Nothing
End Sub
```

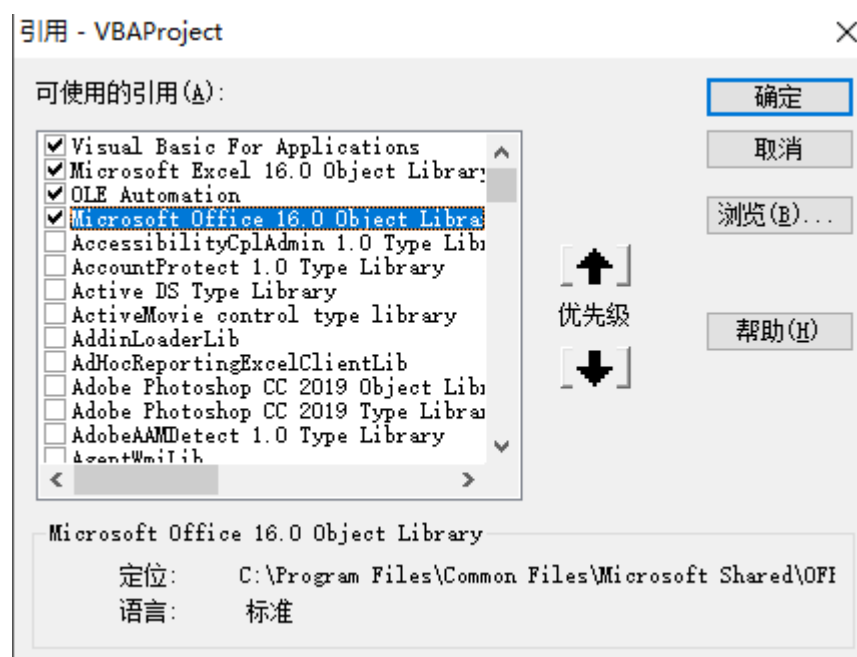


## 20.2.7 乾坤大挪移

2020年5月2日 23:42

# 21. Excel里面用SQL

2020年5月1日 23:28



#经验分享##制作过程##学习心得#



《跟着孙兴华学习MySQL》关系型数据库教程  
[初级篇完结]

此课程仅献给年轻的自己，我还是从前那个少年，没有  
一丝丝改变！我不是老师，我是自学+分享，同时也...

991 37

<https://www.bilibili.com/video/BV1Dk4y1d71j>

所有讲过的SQL语句直接复制到Excel VBA中就能直接使用

## 21.1 创建数据库连接【Access数据库】

2020年5月3日 2:26

```
Sub CreateQueryExecute()
```

```
    Dim cnADO As Object
```

```
    Dim rsADO As Object
```

```
    Dim strPath As String
```

```
    Dim strSQL As String
```

```
    Dim i As Integer
```

```
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
```

```
    '创建一个连接对象，命名为cnADO
```

```
    Set cnADO = CreateObject("ADODB.Connection")
```

```
    On Error GoTo ErrMsg
```

```
    '使用cnADO将本VBA程序与目标数据库连接起来
```

```
    With cnADO
```

```
        .Provider = "Microsoft.ACE.OLEDB.12.0"
```

```
        .ConnectionTimeout = 100
```

```
        .Open strPath
```

```
    End With
```

```
    '将SQL语句写在strSQL变量中
```

```
    strSQL = "SELECT 姓名,性别 FROM 员工档案 WHERE 部门='办公室'"
```

```
    '使用Connection对象的Execute属性运行SQL语句
```

```
    Set rsADO = cnADO.Execute(strSQL)
```

```
    Cells.ClearContents
```

```
    '利用Recordset对象Fields集合取得所有字段名，写入第1行
```

```
    For i = 0 To rsADO.Fields.Count - 1
```

```
        Cells(1, i + 1) = rsADO.Fields(i).Name
```

```
    Next i
```

将SQL语句查到的内容复制到以A2单元格为左上角的区域内

```
Range("A2").CopyFromRecordset rsADO
```

```
rsADO.Close
```

```
cnADO.Close
```

```
Set rsADO = Nothing
```

```
Set cnADO = Nothing
```

```
Exit Sub
```

```
ErrMsg:
```

```
MsgBox Err.Description, , "错误报告"
```

```
End Sub
```

## 21.2 连接Excel文件

2020年5月3日 2:34

### Sub Query()

```
Dim Conn As Object, Rst As Object
Dim strConn As String, strSQL As String
Dim i As Integer, PathStr As String
Set Conn = CreateObject("ADODB.Connection")
Set Rst = CreateObject("ADODB.Recordset")
PathStr = "C:\Users\孙艺航\Desktop\销售表.xlsx" '设置工作簿的完整路径和名称
t = ActiveSheet.Name
Select Case Application.Version * 1 '设置连接字符串,根据版本创建连接
Case Is <= 11
    strConn = "Provider=Microsoft.Jet.Oledb.4.0;Extended Properties=excel 8.0;Data source=" & PathStr
Case Is >= 12
    strConn = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & PathStr & ";Extended
Properties="""Excel 12.0;HDR=YES""";""""
End Select
'设置SQL查询语句
strSQL = "select * from [销售表$]"
Conn.Open strConn '打开数据库链接
Set Rst = Conn.Execute(strSQL) '执行查询,并将结果输出到记录集对象
'ThisWorkbook.Sheets.Add after:=Worksheets("数据源") '添加一个新工作表在第一工作后
For Each ws In Worksheets
    If ws.Name = "提数结果" Then ws.Delete
Next
Sheets.Add.Name = "提数结果"
With Sheets("提数结果")
    .Cells.Clear
    .Tab.Color = 255
    For i = 0 To Rst.Fields.Count - 1 '填写标题
        .Cells(1, i + 1) = Rst.Fields(i).Name
    Next i
    .Range("A2").CopyFromRecordset Rst
    .Cells.EntireColumn.AutoFit '自动调整列宽
End With
Rst.Close '关闭数据库连接
Conn.Close
Set Conn = Nothing
Set Rst = Nothing
End Sub
```

## 22.VBA字典

2020年5月2日 0:56

字典是VBA编程中的一个对象，具有操作简单、运行高效的特点，常用于数据的条件查询、聚合汇总等，如果说数组是VBA处理数据的基础结构，那么字典就可以被称为核心。**数组用于数据清洗整理，字典用于数据汇总分析**

姓名	分数
丁智敏	65
李平平	54
王松	84
卢海军	48
张伊	34
王刚	74

数组就是循环，从最小下标到最大小标循环，找到为止

字典：就是用Key接取到值

关键字1	内容1
关键字2	内容2
关键字3	内容3
关键字4	内容4
关键字5	内容5
关键字6	内容6
关键字7	内容7

字典是由两列数据构成的特殊表，第1列是关键字(Key)，第2列是值 (Item)

注意：

**字典里面不能有重复的key值，Key可以是字符也可以是数值**

**由于对象也可以做关键字，所以如果用单元格的值做关键字时，必须写上.Value属性**

## 22.1 字典的创建与常用方法和属性

2020年5月3日 4:50

### 一、创建字典(内部连接)

**Set 字典 = CreateObject("Scripting.Dictionary")**

### 二、常用方法

#### 1、Add方法

#### 2、Key属性与Item属性

#### 3、Remove方法和Exists方法

## 22.1.1 Add方法

2020年5月3日 5:18

Sub Add方法()

Dim 字典

Set 字典 = CreateObject("Scripting.Dictionary")

字典.Add "孙兴华", "20岁"


MsgBox 字典("孙兴华")

Set 字典 = Nothing

End Sub

语法:

object.Add Key,Item



字典("孙兴华") = "18岁"

如果孙兴华已经被赋值20岁，那么再赋值只会更新Item



## 22.1.2 Exists方法 【判断关键字是否存在】

2020年5月3日 5:18

**【判断关键字是否存在】** 如果存在返回T，不存在返回F  
语法：

**字典.Exists(关键字)**

**Sub BuildDic()**

**Dim 字典**

**Set 字典 = CreateObject("Scripting.Dictionary")**

**字典.Add "孙兴华", "20岁"**

**If Not 字典.Exists("孙兴华") Then**

**字典("孙兴华") = "18岁"**

**MsgBox 字典("孙兴华")**

**Else**

**MsgBox "K值已经存在"**

**End If**

**Set 字典 = Nothing**

**End Sub**

## 22.1.3 将一组数据批量装入字典

2020年5月3日 5:40

**Sub 数据批量存入字典()**

**Dim 字典, arr, i**

**Set 字典 = CreateObject("Scripting.Dictionary")**

**arr = Range("A1").CurrentRegion**

**For i = 2 To UBound(arr)**

**字典(arr(i, 1)) = arr(i, 2)**

**Next**

**End Sub**

## 22.1.4 将数据从字典中提取

2020年5月3日 5:18

**Sub 从字典中读取数据()**

**Dim 字典, arr, brr**

**Set 字典 = CreateObject("Scripting.Dictionary")**

**arr = Range("A1").CurrentRegion**

**For i = 2 To UBound(arr)**

**'对字典里的每个key赋值**

**字典(arr(i, 1)) = arr(i, 2)**

**Next**

**'查询区域**

**brr = Range("D1:E" & Cells(Rows.Count, "D").End(xlUp).Row)**

**For i = 2 To UBound(brr)**

**'如果字典存在Key值**

**If 字典.Exists(brr(i, 1)) Then**

**'获取人名对应的条目，分数那列等于人名那列的item**

**brr(i, 2) = 字典(brr(i, 1))**

**Else**

**brr(i, 2) = "查无此人"**

**End If**

**Next**

**Range("D1:E" & Cells(Rows.Count, "D").End(xlUp).Row)  
= brr**

**Set 字典 = Nothing**

**End Sub**

## 22.1.5 将关键字或值转成一维数组

2020年5月3日 5:18

获取字典所有Key: 字典.Keys

获取字典所有Item: 字典.Items

生成并返回1维数组

Sub 去重复()

Dim 字典, arr, i

Set 字典 = CreateObject("Scripting.Dictionary")

arr = Range("a1").CurrentRegion

For i = 1 To UBound(arr)

If Not 字典.Exists(arr(i, 1)) Then

'姓名为关键字, 值为空文本

字典(arr(i, 1)) = ""

End If

Next

Range("d:d").ClearContents

'Count统计字典关键字数量

Range("d1").Resize(字典.Count, 1) = Application.Transpose(字典.Keys)

Set 字典 = Nothing

End Sub

拓展: 将不重复的姓名和分数都提取

Sub 去重复()

Dim 字典, arr, i

Set 字典 = CreateObject("Scripting.Dictionary")

arr = Range("a1").CurrentRegion

For i = 1 To UBound(arr)

If Not 字典.Exists(arr(i, 1)) Then

字典(arr(i, 1)) = arr(i, 2)

End If

Next

Range("d:d").ClearContents

Range("d1").Resize(字典.Count, 2) = Application.Transpose(Array(字典.Keys, 字典.Items))

Set 字典 = Nothing

End Sub

	A	B	C	D	E
1	姓名	分数		姓名	
2	丁智敏	65		丁智敏	
3	李平平	54		李平平	
4	王松	84		王松	
5	丁智敏	65		卢海军	
6	卢海军	48		张伊	
7	李平平	54		王刚	
8	张伊	34			
9	王刚	74			
10					

	A	B	C	D	E
1	姓名	分数		姓名	分数
2	丁智敏	65		丁智敏	65
3	李平平	54		李平平	54
4	王松	84		王松	84
5	丁智敏	65		卢海军	48
6	卢海军	48		张伊	34
7	李平平	54		王刚	74
8	张伊	34			
9	王刚	74			
10					

## 22.1.6 移除字典中的元素

2020年5月3日 15:35

语法:

**字典.Remove 关键字**

将字典一次清空:

**字典.RemoveAll**

	A	B	C	D	E	F	G	H
1	序号	数字1	数字2	数字3	数字4	数字5	数字6	不重复的开奖号
2	1	2	4	4	1	2	6	2,4,1,6
3	2	1	4	3	6	6	5	1,4,3,6,5
4	3	2	2	3	4	6	6	2,3,4,6
5	4	6	3	2	5	4	2	6,3,2,5,4
6	5	2	5	5	1	2	5	2,5,1
7	6	3	4	6	1	5	4	3,4,6,1,5
8	7	5	6	6	4	5	3	5,6,4,3
9								

**Sub 全部移除字典中的元素()**

Dim d, arr, i

Set 字典 = CreateObject("Scripting.Dictionary")

arr = Range("a1").CurrentRegion

For i = 2 To UBound(arr) '遍历每行数据

字典.RemoveAll '移除字典中所有的元素

For j = 2 To UBound(arr, 2) - 1 '遍历列

If Not 字典.Exists(arr(i, j)) Then

字典(arr(i, j)) = "" '将不重复的号码存入字典

End If

Next

arr(i, UBound(arr, 2)) = VBA.Join(字典.Keys, ",") '合并为一个字符串

Next

Range("a1").CurrentRegion = arr

Set 字典 = Nothing

**End Sub**

## 22.1.7 遍历字典中的元素

2020年5月3日 16:01

为了筛选出符合条件的数据，我们需要遍历数组一样，遍历字典中的每个元素  
通常需要先获取字典的Keys集合，再遍历每个Key去筛选字典中符合条件的数据

```
Sub 遍历字典元素_索引法()  
    Dim 字典, arr, aKey, aRes  
    Set 字典 = CreateObject("Scripting.Dictionary")  
    arr = Range("a1").CurrentRegion  
    For i = 2 To UBound(arr)  
        '将数组中的人名存入字典  
        字典(arr(i, 1)) = 字典(arr(i, 1)) + 1  
    Next  
    '返回字典中所有的Key，结果是一个下标为0的一维数组，命名为akey  
    aKey = 字典.Keys  
    '重新定义动态数组，字典个数是行，列数是列，一列放人名，一列放次数  
    ReDim aRes(1 To 字典.Count, 1 To 2) '结果数组  
    For i = 0 To UBound(aKey)  
        '用索引方式遍历数组akey，查每一个key在字典中item是否大于1  
        If 字典(aKey(i)) > 1 Then '次数大于1次  
            k = k + 1  
            '如果大于1，将key与item分别存入数组  
            aRes(k, 1) = aKey(i)  
            aRes(k, 2) = 字典(aKey(i))  
        End If  
    Next  
    Range("c:c").ClearContents  
    Range("c1") = "重复1次以上的人名"  
    Range("c2").Resize(k, 2) = aRes  
    Set 字典 = Nothing  
End Sub
```

作为赋值语句，它首先运算的是等号右侧的表达式：字典(arr(i,1))+1。

在代码运行这里的字典(arr(i,1))的时候，我们还没有将arr(i,1)的关键字存入字典，

所以正常理解，这句代码应该返回程序错误，但事实并没有。

字典(关键字)语句的运算规则是：

如果字典中存在指定关键字，则返回对应的Item，否则会将该关键字存入字典，同时将其对应的Item设置为Nothing。

因此，在第一次运行这句代码时.....

字典(arr(i, 1)) = 字典(arr(i, 1)) + 1

等同于▼

字典(arr(i, 1)) = Nothing + 1

等同于▼

字典(arr(i, 1)) = 1

也就是在字典中存入一个关键字arr(i,1)，对应的条目为1。

当出现第2次出现同名的关键字时...

字典(arr(i, 1)) = 字典(arr(i, 1)) + 1

等同于▼

字典(arr(i, 1)) = 1 + 1

等同于▼

字典(arr(i, 1)) = 2

也就是在字典将关键字arr(i,1)对应的条目更新为2。

.....以此实现了相同值出现次数在字典中不断累加的效果。

## 22.1.8 前期绑定与后期绑定的区别

2020年5月3日 16:01

有些属性前期绑定是支持的，但后期绑定并不能使用。这些属性，其实指的就是Items和Keys。

在前期绑定的情况下，我们可以使用以下语句读取Keys集合的指定索引元素。

读取字典Keys的第2个元素，并赋值为变量strKey.. ▼

```
strKey = 字典.Keys(1)
```

但后期绑定并不支持运行该语句，必须先将Key或Item转换为数组才能够索引遍历。

读取字典Keys的第2个元素，并赋值为变量strKey.. ▼

```
arr = 字典.Keys'关键字集合赋值数组arr
```

```
strKey = arr(1)
```



## 22.1.9 让字典不区分大小写

2020年5月3日 16:41

字典是默认是区分字母大小写的。如果不需要区分字母大小写，有两种解决方法.....

一种是设置字典的CompareMode属性为TextCompar，示例代码如下▼

Sub 不区分字母大小写()

Dim 字典

Set 字典 = CreateObject("Scripting.Dictionary")

字典.CompareMode = TextCompare

字典("a") = 1

MsgBox 字典("A")

End Sub

一种是将所有的字母统一转换为大写（UCase）或小写(LCase)，这种方式明显修改了原值，因此通常不建议使用。示例代码如下▼

Sub 不区分字母大小写2()

Dim 字典

Set 字典 = CreateObject("Scripting.Dictionary")

字典(LCase("a")) = 1 'LCase将字母统一转换为小写

MsgBox 字典(LCase("A"))

End Sub

## 22.1.10 案例

2020年5月3日 16:45

	A	B	C	D	E
1	姓名	倒霉次数		姓名	倒霉次数
2	丁智敏	628		丁智敏	1723
3	李平平	751		李平平	2237
4	丁智敏	524		卢海军	1146
5	卢海军	615		王松	881
6	王松	881		张伊	675
7	李平平	612		王刚	633
8	丁智敏	571			
9	卢海军	531			
10	李平平	874			
11	张伊	675			
12	王刚	633			
13					

### Sub 案例1()

Dim 字典, arr1, arr2, i

Set 字典 = CreateObject("Scripting.Dictionary")

i = 2

Do While Range("A" & i) <> ""

Key = Range("A" & i)

If 字典.Exists(Key) Then

'item是key对应的值

字典.Item(Key) = 字典.Item(Key) + Range("B" & i).Value

Else

'将key和值添加到字典中

字典.Add Key, Range("B" & i).Value

End If

i = i + 1

Loop

字典条数 = 字典.Count

If 字典条数 > 0 Then

arr1 = 字典.keys()

arr2 = 字典.items()

Range(Cells(2, 4), Cells(字典条数 + 1, 4)) = Excel.Application.Transpose(arr1)

Range(Cells(2, 5), Cells(字典条数 + 1, 5)) = Excel.Application.Transpose(arr2)

End If

**End Sub**

## 23. 第三季预告

2020年5月3日 17:33

第三季主要内容：

Chart对象与Shape对象：使用VBA制作各种图表和实现对图片的操作

FSO对象：操作文件系统

ExcelVBA实现对互联网数据的抓取

使用VBA操作其它office程序及类模块的详细讲解

数组与字典的高级应用

**第三季开课时间**，由网友决定，只要Excel VBA **第二季**的视频播放量超过**3W**立即开课，如果您觉得本课程对您有帮助，请分享视频到您的朋友圈，谢谢

- 我只看播放量，点赞和投币并不能代表视频受不受欢迎。