

Excel VBA Range 对象等基本操作应用示例

第一大类：示例

一、赋值给某单元格

1、

```
Sub test1()  
Worksheets("Sheet1").Range("A5").Value = 22  
MsgBox "工作表 Sheet1 内单元格 A5 中的值为" _  
& Worksheets("Sheet1").Range("A5").Value  
End Sub
```

2、

```
Sub test2()  
Worksheets("Sheet1").Range("A1").Value = _  
Worksheets("Sheet1").Range("A5").Value  
MsgBox "现在 A1 单元格中的值也为" & _  
Worksheets("Sheet1").Range("A5").Value  
End Sub
```

3、

```
Sub test3()  
MsgBox "用公式填充单元格,本例为随机数公式"  
Range("A1:H8").Formula = "=Rand()  
End Sub
```

4、

```
Sub test4()  
Worksheets(1).Cells(1, 1).Value = 24  
MsgBox "现在单元格 A1 的值为 24"  
End Sub
```

5、

```
Sub test5()  
MsgBox "给单元格设置公式,求 B2 至 B5 单元格区域之和"  
ActiveSheet.Cells(2, 1).Formula = "=Sum(B1:B5)"  
End Sub
```

6、

```
Sub test6()  
MsgBox "设置单元格 C5 中的公式."  
Worksheets(1).Range("C5:C10").Cells(1, 1).Formula = "=Rand()  
End Sub
```

二、引用单元格

```
Sub Random()  
Dim myRange As Range '设置对单元格区域的引用  
Set myRange = Worksheets("Sheet1").Range("A1:D5") '对 Range 对象进行操作  
myRange.Formula = "=RAND()  
myRange.Font.Bold = True  
End Sub
```

示例说明：可以设置 Range 对象变量来引用单元格区域，然后对该变量所代表的单元格区域进行操作。

三、清除单元格

1、清除单元格中的内容(ClearContents 方法)

```
Sub testClearContents()  
MsgBox "清除指定单元格区域中的内容"  
Worksheets(1).Range("A1:H8").ClearContents  
End Sub
```

2、清除单元格中的格式(ClearFormats 方法)

```
Sub testClearFormats()  
MsgBox "清除指定单元格区域中的格式"  
Worksheets(1).Range("A1:H8").ClearFormats  
End Sub
```

3、清除单元格中的批注(ClearComments 方法)

```
Sub testClearComments()  
MsgBox "清除指定单元格区域中的批注"  
Worksheets(1).Range("A1:H8").ClearComments  
End Sub
```

4、清除单元格中的全部，包括内容、格式和批注(Clear 方法)

```
Sub testClear()  
MsgBox "彻底清除指定单元格区域"  
Worksheets(1).Range("A1:H8").Clear  
End Sub
```

四、Range 和 Cells

```
Sub test() '设置单元格区域 A1:J10 的边框线条样式  
With Worksheets(1).Range(.Cells(1, 1), _  
.Cells(10, 10)).Borders.LineStyle = xlThick  
End With  
End Sub
```

示例说明：可用 Range(cell1, cell2) 返回一个 Range 对象，其中 cell1 和 cell2 为指定起始和终止位置的 Range 对象。

五、选取单元格区域(Select 方法)

```
Sub testSelect() '选取单元格区域 A1:D5  
Worksheets("Sheet1").Range("A1:D5").Select  
End Sub
```

六、基于所选区域偏离至另一区域(Offset 属性)

1、

```
Sub testOffset()  
Worksheets("Sheet1").Activate  
Selection.Offset(3, 1).Select  
End Sub
```

示例说明：可用 Offset(row, column)(其中 row 和 column 为行偏移量和列偏移量)返回相对于另一区域在指定偏移量处的区域。如上例选定位于当前选定区域左上角单元格的向下三行且向右一列处单元格区域。

2、选取距当前单元格指定行数和列数的单元格

```
Sub ActiveCellOffice()  
MsgBox "显示距当前单元格第 3 列、第 2 行的单元格中的值"  
MsgBox ActiveCell.Offset(3, 2).Value  
End Sub
```

七、调整区域的大小(Resize 属性)

```
Sub ResizeRange()  
Dim numRows As Integer, numcolumns As Integer  
Worksheets("Sheet1").Activate  
numRows = Selection.Rows.Count  
numcolumns = Selection.Columns.Count  
Selection.Resize(numRows + 1, numcolumns + 1).Select  
End Sub
```

示例说明：本示例调整所选区域的大小，使之增加一行一列。

八、选取多个区域(Union 方法)

```
Sub testUnion()  
Dim rng1 As Range, rng2 As Range, myMultiAreaRange As Range  
Worksheets("sheet1").Activate  
Set rng1 = Range("A1:B2")  
Set rng2 = Range("C3:D4")  
Set myMultiAreaRange = Union(rng1, rng2)  
myMultiAreaRange.Select  
End Sub
```

示例说明：可用 Union(range1, range2, ...) 返回多块区域，即该区域由两个或多个连续的单元格区域所组成。如上例创建由单元格区域 A1:B2 和 C3:D4 组合定义的对象，然后选定该定义区域。

九、激活已选区域中的单元格

```
Sub ActivateRange()  
MsgBox "选取单元格区域 B2:D6 并将 C4 选中"  
ActiveSheet.Range("B3:D6").Select  
Range("C5").Activate  
End Sub
```

十、选取指定条件的单元格(SpecialCells 方法)

```
Sub SelectSpecialCells()  
MsgBox "选择当前工作表中所有公式单元格"  
ActiveSheet.Cells.SpecialCells(xlCellTypeFormulas).Select  
End Sub
```

十一、选取矩形区域(CurrentRegion 属性)

'选取包含当前单元格的矩形区域
'该区域周边为空白行和空白列

```
Sub SelectCurrentRegion()  
MsgBox "选取包含当前单元格的矩形区域"  
ActiveCell.CurrentRegion.Select  
End Sub
```

十二、选取当前工作表中已用单元格(UsedRange 属性)

'选取当前工作表中已使用的单元格区域

```
Sub SelectUsedRange()  
MsgBox "选取当前工作表中已使用的单元格区域" _  
& vbCrLf & "并显示其地址"  
ActiveSheet.UsedRange.Select  
MsgBox ActiveSheet.UsedRange.Address  
End Sub
```

十三、选取最边缘单元格(End 属性)

'选取最下方的单元格

```
Sub SelectEndCell()  
MsgBox "选取当前单元格区域内最下方的单元格"  
ActiveCell.End(xlDown).Select  
End Sub
```

示例说明：可以改变参数 xlDown 以选取最左边、最右边、最上方的单元格。

十四、设置当前单元格的前一个单元格和后一个单元格的值

```
Sub SetCellValue()  
MsgBox "将当前单元格中前面的单元格值设为""我前面的单元格"" & vbCrLf _  
& "后面的单元格值设为""我后面的单元格""  
ActiveCell.Previous.Value = "我前面的单元格"  
ActiveCell.Next.Value = "我后面的单元格"  
End Sub
```

十五、确认所选单元格区域中是否有公式(HasFormula 属性)

```
Sub IfHasFormula()  
If Selection.HasFormula = True Then  
MsgBox "所选单元格中都有公式"  
Else  
MsgBox "所选单元格中，部分单元格没有公式"  
End If  
End Sub
```

十六、公式单元格操作

1、获取与运算结果单元格有直接关系的单元格

```
Sub CalRelationCell()  
MsgBox "选取与当前单元格的计算结果相关的单元格"  
ActiveCell.DirectPrecedents.Select  
End Sub
```

2、追踪公式单元格

```
Sub Call()  
MsgBox "选取计算结果单元格相关的所有单元格"  
ActiveCell.Precedents.Select  
End Sub  
Sub TrackCell()  
MsgBox "追踪运算结果单元格"  
ActiveCell.ShowPrecedents
```

```
End Sub
Sub DelTrack()
MsgBox "删除追踪线"
ActiveCell.ShowPrecedents Remove:=True
End Sub
```

十七、复制单元格(Copy 方法)

```
Sub CopyRange()
MsgBox "在单元格 B7 中写入公式后,将 B7 的内容复制到 C7:D7 内"
Range("B7").Formula = "=Sum(B3:B6)"
Range("B7").Copy Destination:=Range("C7:D7")
End Sub
```

十八、获取单元格行列值(Row 属性和 Column 属性)

```
Sub RangePosition()
MsgBox "显示所选单元格区域的行列值"
MsgBox "第 " & Selection.Row & "行 " & Selection.Column & "列"
End Sub
```

十九、获取单元格区域的单元格数及行列数(Rows 属性、Columns 属性和 Count 属性)

```
Sub GetRowColumnNum()
MsgBox "显示所选取单元格区域的单元格数、行数和列数"
MsgBox "单元格区域中的单元格数为:" & Selection.Count
MsgBox "单元格区域中的行数为:" & Selection.Rows.Count
MsgBox "单元格区域中的列数为:" & Selection.Columns.Count
End Sub
```

二十、设置单元格中的文本格式

1、对齐文本

```
Sub HorizontalAlign()
MsgBox "将所选单元格区域中的文本左右对齐方式设为居中"
Selection.HorizontalAlignment = xlHAlignCenter
End Sub

Sub VerticalAlign()
MsgBox "将所选单元格区域中的文本上下对齐方式设为居中"
Selection.RowHeight = 36
Selection.VerticalAlignment = xlVAlignCenter
End Sub
```

2、缩排文本(InsertIndent 方法)

```
Sub Indent()
MsgBox "将所选单元格区域中的文本缩排值加 1"
Selection.InsertIndent 1
MsgBox "将缩排值恢复"
Selection.InsertIndent -1
End Sub
```

3、设置文本方向(Orientation 属性)

```
Sub ChangeOrientation()
MsgBox "将所选单元格中的文本顺时针旋转 45 度"
```

```
Selection.Orientation = 45
MsgBox "将文本由横向改为纵向"
Selection.Orientation = xlVertical
MsgBox "将文本方向恢复原值"
Selection.Orientation = xlHorizontal
End Sub
```

4、自动换行(WrapText 属性)

```
Sub ChangeRow()
Dim i
MsgBox "将所选单元格设置为自动换行"
i = Selection.WrapText
Selection.WrapText = True
MsgBox "恢复原状"
Selection.WrapText = i
End Sub
```

5、将比单元格列宽长的文本缩小到能容纳列宽大小(ShrinkToFit 属性)

```
Sub AutoFit()
Dim i
MsgBox "将长于列宽的文本缩到与列宽相同"
i = Selection.ShrinkToFit
Selection.ShrinkToFit = True
MsgBox "恢复原状"
Selection.ShrinkToFit = i
End Sub
```

二十一、设置条件格式(FormatConditions 属性)

```
Sub FormatConditions()
MsgBox "在所选单元格区域中将单元格值小于 10 的单元格中的文本变为红色"
Selection.FormatConditions.Add Type:=xlCellValue, _
Operator:=xlLessEqual, Formula1:="<10"
Selection.FormatConditions(1).Font.ColorIndex = 3
MsgBox "恢复原状"
Selection.FormatConditions(1).Font.ColorIndex = xlAutomatic
End Sub
```

二十二、插入批注(AddComment 方法)

```
Sub EnterComment()
MsgBox "在当前单元格中输入批注"
ActiveCell.AddComment ("Hello")
ActiveCell.Comment.Visible = True
End Sub
```

二十三、隐藏/显示单元格批注

```
Sub CellComment()
MsgBox "切换当前单元格批注的显示和隐藏状态"
ActiveCell.Comment.Visible = Not (ActiveCell.Comment.Visible)
End Sub
```

二十四、改变所选单元格的顏色

```
Sub ChangeColor()  
Dim iro As Integer  
MsgBox "将所选单元格的顏色改为紅色"  
iro = Selection.Interior.ColorIndex  
Selection.Interior.ColorIndex = 3  
MsgBox "将所选单元格的顏色改为藍色"  
Selection.Interior.Color = RGB(0, 0, 255)  
MsgBox "恢复原状"  
Selection.Interior.ColorIndex = iro  
End Sub
```

二十五、改变单元格的图案

```
Sub ChangePattern()  
Dim p, pc, i  
MsgBox "依 Pattern 常數值的顺序改变所选单元格的图案"  
p = Selection.Interior.Pattern  
pc = Selection.Interior.PatternColorIndex  
For i = 9 To 16  
With Selection.Interior  
.Pattern = i  
.PatternColor = RGB(255, 0, 0)  
End With  
MsgBox "常数值 " & i  
Next i  
MsgBox "恢复原状"  
Selection.Interior.Pattern = p  
Selection.Interior.PatternColorIndex = pc  
End Sub
```

二十六、合并单元格

```
Sub MergeCells()  
MsgBox "合并单元格 A2:C2,并将文本设为居中对齐"  
Range("A2:C2").Select  
With Selection  
.MergeCells = True  
.HorizontalAlignment = xlCenter  
End With  
End Sub
```

二十七、限制单元格移动的范围

```
Sub ScrollArea1()  
MsgBox "将单元格的移动范围限制在单元格区域 B2:D6 中"  
ActiveSheet.ScrollArea = "B2:D6"  
End Sub  
Sub ScrollArea2()  
MsgBox "解除移动范围限制"
```

```
ActiveSheet.ScrollArea = ""
```

```
End Sub
```

二十八、获取单元格的位置(Address 属性)

```
Sub GetAddress()
```

```
MsgBox "显示所选单元格区域的地址"
```

```
MsgBox "绝对地址:" & Selection.Address
```

```
MsgBox "行的绝对地址:" & Selection.Address(RowAbsolute:=False)
```

```
MsgBox "列的绝对地址:" & Selection.Address(ColumnAbsolute:=False)
```

```
MsgBox "以 R1C1 形式显示:" & Selection.Address(ReferenceStyle:=xlR1C1)
```

```
MsgBox "相对地址:" & Selection.Address(False, False)
```

```
End Sub
```

二十九、删除单元格区域>Delete 方法)

```
Sub DeleteRange()
```

```
MsgBox "删除单元格区域 C2:D6 后,右侧的单元格向左移动"
```

```
ActiveSheet.Range("C2:D6").Delete (xlShiftToLeft)
```

```
End Sub
```

第二大类：小结

下面对 Range 对象的一些常用属性和方法进行简单的小结。

1、Activate 与 Select

试验下面的过程：

```
Sub SelectAndActivate()
```

```
Range("B3:E10").Select
```

```
Range("C5").Activate
```

```
End Sub
```

Selection 指单元格区域 B3：E10，而 ActiveCell 则是单元格 C5；

ActiveCell 代表单个的单元格，而 Selection 则可以代表单个单元格，也可以代表单元格区域。

2、Range 属性

可以使用 Application 对象的 Range 属性引用 Range 对象，如

Application.Range("B2") ‘代表当前工作表中的单元格 B2

若引用当前工作表中的单元格，也可以忽略前面的 Application 对象。

Range("A1:D10") ‘代表当前工作表中的单元格区域 A1:D10

Range("A1:A10,C1:C10,E1:E10") ‘代表当前工作表中非连续的三个区域组成的单元格区域

Range 属性也接受指向单元格区域对角的两个参数，如：

Range("A1","D10") ‘代表单元格区域 A1：D10

当然，Range 属性也接受单元格区域名称，如：

Range("Data") ‘代表名为 Data 的数据区域

Range 属性的参数可以是对象也可以是字符串，如：

```
Range("A1",Range("LastCell"))
```

3、单元格引用的快捷方式

可以在引用区域两侧加上方括号来快速引用单元格区域，如：

```
[B2]
```

```
[A1:D10]
```

```
[A1:A10,C1:C10,E1:E10]
```


[Data]

但其引用的是绝对区域。

4、Cells 属性

可以使用 Cells 属性来引用 Range 对象。如：

ActiveSheet.Cells

Application.Cells ‘引用当前工作表中的所有单元格

Cell(2,2)

Cell(2,”B”) ‘引用单元格 B2

Range(Cells(1,1),Cells(10,5)) ‘引用单元格区域 A1:E10

若想在一个单元格区域中循环时，使用 Cells 属性是很方便的。

也可以使用 Cells 属性进行相对引用，如：

Range(“D10:G20”).Cells(2,3) ‘表示引用单元格区域 D10:G20 中第 2 行第 3 列的单元格，即单元格 F11

也可使用语句：Range(“D10”).Cells(2,3)达到同样的引用效果。

5、Offset 属性 Offset 属性基于当前单元格按所给参数进行偏移，与 Cells 属性不同的是，它基于 0 即基准单元格为 0，如：

Range(“A10”).Cells(1,1)和 Range(“A10”).Offset(0,0)都表示单元格 A10

当想引用于基准单元格区域同样大小的单元格区域时，则 Offset 属性是有用的。

6、Resize 属性

可使用 Resize 属性获取相对于原单元格区域左上角单元格指定大小的区域。

7、SpecialCells 方法

SpecialCells 方法对应于“定位条件”对话框，如图 05-02 所示：

图 05-02: “定位条件”对话框

8、CurrentRegion 属性

使用 CurrentRegion 属性可以选取当前单元格所在区域，即周围是空行和空列所围成的矩形区域，等价于“Ctrl+Shift+*”快捷键。

9、End 属性

End 属性所代表的操作等价于“Ctrl+方向箭”的操作，使用常量 xlUp、xlDown、xlToLeft 和 xlToRight 分别代表上、下、左、右箭。

10、Columns 属性和 Rows 属性

Columns 属性和 Rows 属性分别返回单元格区域中的所有列和所有行。

11、Areas 集合

在多个非连续的单元格区域中使用 Columns 属性和 Rows 属性时，只是返回第一个区域的行或列，如：Range(“A1:B5,C6:D10,E11:F15”).Rows.Count 将返回 5。

此时应使用 Areas 集合来返回区域中每个块的地址，如：

For Each Rng In Range(“A1:B5,C6:D10,E11:F15”).Areas

MsgBox Rng.Address

Next Rng

12、Union 方法和 Intersect 方法

当想从两个或多个单元格区域中生成一个单元格区域时，使用 Union 方法；当找到两个或多个单元格区域共同拥有的单元格区域时，使用 Intersect 方法。

End Sub