

VBA 语句集

(第 1 辑)

定制模块行为

(1) Option Explicit '强制对模块内所有变量进行声明

Option Private Module '标记模块为私有，仅对同一工程中其它模块有用，在宏对话框中不显示

Option Compare Text '字符串不区分大小写

Option Base 1 '指定数组的第一个下标为 1

(2) On Error Resume Next '忽略错误继续执行 VBA 代码,避免出现错误消息

(3) On Error GoTo ErrorHandler '当错误发生时跳转到过程中的某个位置

(4) On Error GoTo 0 '恢复正常的错误提示

(5) Application.DisplayAlerts=False '在程序执行过程中使出现的警告框不显示

(6) Application.ScreenUpdating=False '关闭屏幕刷新

Application.ScreenUpdating=True '打开屏幕刷新

(7) Application.Enable.CancelKey=xlDisabled '禁用 Ctrl+Break 中止宏运行的功能

工作簿

(8) Workbooks.Add() '创建一个新的工作簿

(9) Workbooks("book1.xls").Activate '激活名为 book1 的工作簿

(10) ThisWorkbook.Save '保存工作簿

(11) ThisWorkbook.close '关闭当前工作簿

(12) ActiveWorkbook.Sheets.Count '获取活动工作簿中工作表数

(13) ActiveWorkbook.name '返回活动工作簿的名称

(14) ThisWorkbook.Name '返回当前工作簿名称

ThisWorkbook.FullName '返回当前工作簿路径和名称

(15) ActiveWindow.EnableResize=False '禁止调整活动工作簿的大小

(16) Application.Window.Arrange xlArrangeStyleTiled '将工作簿以平铺方式排列

(17) ActiveWorkbook.WindowState=xlMaximized '将当前工作簿最大化

工作表

(18) ActiveSheet.UsedRange.Rows.Count '当前工作表中已使用的行数

(19) Rows.Count '获取工作表的行数(注：考虑向前兼容性)

(20) Sheets(Sheet1).Name="Sum" '将 Sheet1 命名为 Sum

(21) ThisWorkbook.Sheets.Add Before:=Worksheets(1) '添加一个新工作表在第一工作表前

(22) `ActiveSheet.Move After:=ActiveWorkbook. _`

`Sheets(ActiveWorkbook.Sheets.Count)` '将当前工作表移至工作表的最后

(23) `Worksheets(Array("sheet1","sheet2")).Select` '同时选择工作表 1 和工作表 2

(24) `Sheets("sheet1").Delete` 或 `Sheets(1).Delete` '删除工作表 1

(25) `ActiveWorkbook.Sheets(i).Name` '获取工作表 i 的名称

(26) `ActiveWindow.DisplayGridlines=Not ActiveWindow.DisplayGridlines` '切换工作表中的网格线显示, 这种方法也可以用在其它方面进行相互切换, 即相当于开关按钮

(27) `ActiveWindow.DisplayHeadings=Not ActiveWindow.DisplayHeadings` '切换工作表中的行列边框显示

(28) `ActiveSheet.UsedRange.FormatConditions.Delete` '删除当前工作表中所有的条件格式

(29) `Cells.Hyperlinks.Delete` '取消当前工作表所有超链接

(30) `ActiveSheet.PageSetup.Orientation=xlLandscape`

或 `ActiveSheet.PageSetup.Orientation=2` '将页面设置更改为横向

(31) `ActiveSheet.PageSetup.RightFooter=ActiveWorkbook.FullName` '在页面设置的表尾中输入文件路径

`ActiveSheet.PageSetup.LeftFooter=Application.UserName` '将用户名放置在活动工作表的页脚

单元格/单元格区域

(32) `ActiveCell.CurrentRegion.Select`

或 `Range(ActiveCell.End(xlUp),ActiveCell.End(xlDown)).Select`

'选择当前活动单元格所包含的范围, 上下左右无空行

(33) `Cells.Select` '选定当前工作表的所有单元格

(34) `Range("A1").ClearContents` '清除活动工作表上单元格 A1 中的内容

`Selection.ClearContents` '清除选定区域内容

`Range("A1:D4").Clear` '彻底清除 A1 至 D4 单元格区域的内容, 包括格式

(35) `Cells.Clear` '清除工作表中所有单元格的内容

(36) `ActiveCell.Offset(1,0).Select` '活动单元格下移一行, 同理, 可下移一列

(37) `Range("A1").Offset(ColumnOffset:=1)`或 `Range("A1").Offset(,1)` '偏移一列

`Range("A1").Offset(RowOffset:=-1)`或 `Range("A1").Offset(-1)` '向上偏移一行

(38) `Range("A1").Copy Range("B1")` '复制单元格 A1, 粘贴到单元格 B1 中

`Range("A1:D8").Copy Range("F1")` '将单元格区域复制到单元格 F1 开始的区域中

`Range("A1:D8").Cut Range("F1")` '剪切单元格区域 A1 至 D8, 复制到单元格 F1 开始的区域中

`Range("A1").CurrentRegion.Copy Sheets("Sheet2").Range("A1")` '复制包含 A1 的单元格区域到工作表 2 中以 A1 起始的单元格区域中

注: `CurrentRegion` 属性等价于定位命令, 由一个矩形单元格块组成, 周围是一个或多个空行或列

- (39) `ActiveWindow.RangeSelection.Value=XX` '将值 XX 输入到所选单元格区域中
- (40) `ActiveWindow.RangeSelection.Count` '活动窗口中选择的单元格数
- (41) `Selection.Count` '当前选中区域的单元格数
- (42) `GetAddress=Replace(Hyperlinkcell.Hyperlinks(1).Address,"mailto:","")` '返回单元格中超级链接的地址并赋值
- (43) `TextColor=Range("A1").Font.ColorIndex` '检查单元格 A1 的文本颜色并返回颜色索引
`Range("A1").Interior.ColorIndex` '获取单元格 A1 背景色
- (44) `cells.count` '返回当前工作表的单元格数
- (45) `Selection.Range("E4").Select` '激活当前活动单元格下方 3 行，向右 4 列的单元格
- (46) `Cells.Item(5,"C")` '引单元格 C5
`Cells.Item(5,3)` '引单元格 C5
- (47) `Range("A1").Offset(RowOffset:=4,ColumnOffset:=5)`
或 `Range("A1").Offset(4,5)` '指定单元格 F5
- (48) `Range("B3").Resize(RowSize:=11,ColumnSize:=3)`
`Rnage("B3").Resize(11,3)` '创建 B3: D13 区域
- (49) `Range("Data").Resize(,2)` '将 Data 区域扩充 2 列
- (50) `Union(Range("Data1"),Range("Data2"))` '将 Data1 和 Data2 区域连接
- (51) `Intersect(Range("Data1"),Range("Data2"))` '返回 Data1 和 Data2 区域的交叉区域
- (52) `Range("Data").Count` '单元格区域 Data 中的单元格数
`Range("Data"). Columns.Count` '单元格区域 Data 中的列数
`Range("Data"). Rows.Count` '单元格区域 Data 中的行数
- (53) `Selection.Columns.Count` '当前选中的单元格区域中的列数
`Selection.Rows.Count` '当前选中的单元格区域中的行数
- (54) `Selection.Areas.Count` '选中的单元格区域所包含的区域数
- (55) `ActiveSheet.UsedRange.Row` '获取单元格区域中使用的第一行的行号
- (56) `Rng.Column` '获取单元格区域 Rng 左上角单元格所在列编号
- (57) `ActiveSheet.Cells.SpecialCells(xlCellTypeAllFormatConditions)` '在活动工作表中返回所有符合条件格式设置的区域
- (58) `Range("A1").AutoFilter Field:=3,VisibleDropDown:=False` '关闭由于执行自动筛选命令产生的第 3 个字段的下拉列表

名称

- (59) `Range("A1: C3").Name="computer"` '命名 A1: C3 区域为 computer
或 `Range("D1: E6").Name="Sheet1!book"` '命名局部变量，即 Sheet1 上区域 D1: E6 为 book

或 Names("computer").Name="robot" '将区域 computer 重命名为 robot

(60) Names("book").Delete '删除名称

(61) Names.Add Name:="ContentList", _

RefersTo:="=OFFSET(Sheet1!A2,0,0,COUNTA(Sheet2!\$A:\$A))" '动态命名列

(62) Names.Add Name:="Company",RefersTo:="CompanyCar" '命名字符串 CompanyCar

(63) Names.Add Name:="Total",RefersTo:=123456 '将数字 123456 命名为 Total。注意数字不能加引号，否则就是命名字符串了。

(64) Names.Add Name:="MyArray",RefersTo:=ArrayNum '将数组 ArrayNum 命名为 MyArray。

(65) Names.Add Name:="ProduceNum",RefersTo:="=\$B\$1",Visible:=False '将名称隐藏

(66) ActiveWorkbook.Names("Com").Name '返回名称字符串

公式与函数

(67) Application.WorksheetFunction.IsNumber("A1") '使用工作表函数检查 A1 单元格中的数据是否为数字

(68) Range("A:A").Find(Application.WorksheetFunction.Max(Range("A:A"))).Activate
'激活单元格区域 A 列中最大值的单元格

(69) Cells(8,8).FormulaArray="=SUM(R2C[-1]:R[-1]C[-1]*R2C:R[-1]C)" '在单元格中输入数组公式。
注意必须使用 R1C1 样式的表达式

图表

(70) ActiveSheet.ChartObjects.Count '获取当前工作表中图表的个数

(71) ActiveSheet.ChartObjects("Chart1").Select '选中当前工作表中图表 Chart1

(72) ActiveSheet.ChartObjects("Chart1").Activate

ActiveChart.ChartArea.Select '选中当前图表区域

(73) WorkSheets("Sheet1").ChartObjects("Chart2").Chart. _

ChartArea.Interior.ColorIndex=2 '更改工作表中图表的图表区的颜色

(74) Sheets("Chart2").ChartArea.Interior.ColorIndex=2 '更改图表工作表中图表区的颜色

(75) Charts.Add '添加新的图表工作表

(76) ActiveChart.SetSourceData Source:=Sheets("Sheet1").Range("A1:D5"), _

PlotBy:=xlColumns '指定图表数据源并按列排列

(77) ActiveChart.Location Where:=xlLocationAsNewSheet '新图表作为新图表工作表

(78) ActiveChart.PlotArea.Interior.ColorIndex=xlNone '将绘图区颜色变为白色

(79) WorkSheets("Sheet1").ChartObjects(1).Chart. _

Export FileName:="C: MyChart.gif",FilterName:="GIF" '将图表 1 导出到 C 盘上并命名为 MyChart.gif

窗体

(80) MsgBox "Hello!" '消息框中显示消息 Hello

(81) Ans=MsgBox("Continue?",vbYesNo) '在消息框中点击“是”按钮，则 Ans 值为 vbYes；点击“否”按钮，则 Ans 值为 vbNo。

If MsgBox("Continue?",vbYesNo)<>vbYes Then Exit Sub '返回值不为“是”，则退出

(82) Config=vbYesNo+vbQuestion+vbDefaultButton2 '使用常量的组合，赋值组 Config 变量，并设置第二个按钮为缺省按钮

(83) MsgBox “This is the first line.” & vbNewLine & “Second line.” 在消息框中强制换行，可用 vbCrLf 代替 vbNewLine。

(84) MsgBox "the average
is "&Format(Application.WorksheetFunction.Average(Selection),"#,##0.00"),vbInformation,
"selection count average" & Chr(13) '应用工作表函数返回所选区域的平均值并按指定格式显示

(85) Userform1.Show ‘显示用户窗体’

(86) Load Userform1 ‘加载一个用户窗体,但该窗体处于隐藏状态’

(87) Userform1.Hide ‘隐藏用户窗体’

(88) Unload Userform1 或 Unload Me ‘卸载用户窗体’

(89) (图像控件).Picture=LoadPicture(“图像路径”) ‘在用户窗体中显示图形

(90) UserForm1.Show 0 或 UserForm1.Show vbModeless ‘将窗体设置为无模式状态

(91) Me.Height=Int(0.88*ActiveWindow.Height) ‘窗体高度为当前活动窗口高度的 0.88

Me.Width=Int(0.88*ActiveWindow.Width) ‘窗体宽度为当前活动窗口高度的 0.88

事件

(92) Application.EnableEvents=False '禁用所有事件

Application.EnableEvents=True '启用所有事件

注：不适用于用户窗体控件触发的事件

对象

(93) Set ExcelSheet = CreateObject("Excel.Sheet") '创建一个 Excel 工作表对象

ExcelSheet.Application.Visible = True '设置 Application 对象使 Excel 可见

ExcelSheet.Application.Cells(1, 1).Value = "Data" '在表格的第一个单元中输入文本

ExcelSheet.SaveAs "C:\TEST.XLS" '将该表格保存到 C:\test.xls 目录

ExcelSheet.Application.Quit '关闭 Excel

Set ExcelSheet = Nothing '释放该对象变量

(94) ‘声明并创建一个 Excel 对象引用

```
Dim xlApp As Excel.Application  
Dim xlBook As Excel.Workbook  
Dim xlSheet As Excel.WorkSheet  
Set xlApp = CreateObject("Excel.Application")  
Set xlBook = xlApp.Workbooks.Add  
Set xlSheet = xlBook.Worksheets(1)
```

(95) '创建并传递一个 Excel.Application 对象的引用

```
Call MySub (CreateObject("Excel.Application"))
```

(96) Set d = CreateObject(**Scripting.Dictionary**) '创建一个 **Dictionary** 对象变量

(97) d.Add "a", "Athens" '为对象变量添加关键字和条目

其他

(98) Application.OnKey “^I”, “macro” '设置 Ctrl+I 键为 macro 过程的快捷键

(99) Application.CutCopyMode=False '退出剪切/复制模式

(100) Application.Volatile True '无论何时工作表中任意单元格重新计算，都会强制计算该函数

Application.Volatile False '只有在该函数的一个或多个参数发生改变时，才会重新计算该函数

By fanjy in 2006-6-4

VBA 语句集

(第 2 辑)

定制模块行为

(101) Err.Clear ‘清除程序运行过程中所有的错误

工作簿

(102) ThisWorkbook.BuiltinDocumentProperties("Last Save Time")

或 Application.Caller.Parent.Parent.BuiltinDocumentProperties("Last Save Time") ‘返回上次保存工作簿的日期和时间

(103) ThisWorkbook.BuiltinDocumentProperties("Last Print Date")

或 Application.Caller.Parent.Parent.BuiltinDocumentProperties("Last Print Date") ‘返回上次打印或预览工作簿的日期和时间

(104) Workbooks.Close ‘关闭所有打开的工作簿

(105) ActiveWorkbook.LinkSources(xlExcelLinks)(1) ‘返回当前工作簿中的第一条链接

(106) ActiveWorkbook.CodeName

ThisWorkbook.CodeName ‘返回工作簿代码的名称

(107) ActiveWorkbook.FileFormat

ThisWorkbook.FileFormat ‘返回当前工作簿文件格式代码

(108) ThisWorkbook.Path

ActiveWorkbook.Path ‘返回当前工作簿的路径(注:若工作簿未保存,则为空)

(109) ThisWorkbook.ReadOnly

ActiveWorkbook.ReadOnly ‘返回当前工作簿的读/写值(为 False)

(110) ThisWorkbook.Saved

ActiveWorkbook.Saved ‘返回工作簿的存储值(若已保存则为 False)

(111) Application.Visible = False ‘隐藏工作簿

Application.Visible = True ‘显示工作簿

注:可与用户窗体配合使用,即在打开工作簿时将工作簿隐藏,只显示用户窗体.可设置控制按钮控制工作簿可见

工作表

(112) ActiveSheet.Columns("B").Insert ‘在 A 列右侧插入列, 即插入 B 列

ActiveSheet.Columns("E").Cut

移
ActiveSheet.Columns("B").Insert ‘以上两句将 E 列数据移至 B 列, 原 B 列及以后的数据相应后

ActiveSheet.Columns("B").Cut

列
ActiveSheet.Columns("E").Insert ‘以上两句将 B 列数据移至 D 列, 原 C 列和 D 列数据相应左移一

(113) ActiveSheet.Calculate ‘计算当前工作表

(114) ThisWorkbook.Worksheets("sheet1").Visible=xlSheetHidden ‘正常隐藏工作表，同在 Excel 菜单中选择“格式——工作表——隐藏”操作一样

ThisWorkbook.Worksheets("sheet1").Visible=xlSheetVeryHidden ‘隐藏工作表，不能通过在 Excel 菜单中选择“格式——工作表——取消隐藏”来重新显示工作表

ThisWorkbook.Worksheets("sheet1").Visible=xlSheetVisible ‘显示被隐藏的工作表

(115) ThisWorkbook.Sheets(1).ProtectContents ‘检查工作表是否受到保护

(116) ThisWorkbook.Worksheets.Add Count:=2, _

Before:=ThisWorkbook.Worksheets(2)

或 ThisWorkbook.Worksheets.Add ThisWorkbook.Worksheets(2), , 2 ‘在第二个工作表之前添加两个新的工作表

(117) ThisWorkbook.Worksheets(3).Copy ‘复制一个工作表到新的工作簿

(118) ThisWorkbook.Worksheets(3).Copy ThisWorkbook.Worksheets(2) ‘复制第三个工作表到第二个工作表之前

(119) ThisWorkbook.ActiveSheet.Columns.ColumnWidth = 20 ‘改变工作表的列宽为 20

ThisWorkbook.ActiveSheet.Columns.ColumnWidth = _

ThisWorkbook.ActiveSheet.StandardWidth ‘将工作表的列宽恢复为标准值

ThisWorkbook.ActiveSheet.Columns(1).ColumnWidth = 20 ‘改变工作表列 1 的宽度为 20

(120) ThisWorkbook.ActiveSheet.Rows.RowHeight = 10 ‘改变工作表的行高为 10

ThisWorkbook.ActiveSheet.Rows.RowHeight = _

ThisWorkbook.ActiveSheet.StandardHeight ‘将工作表的行高恢复为标准值

ThisWorkbook.ActiveSheet.Rows(1).RowHeight = 10 ‘改变工作表的行 1 的高度值设置为 10

(121) ThisWorkbook.Worksheets(1).Activate ‘当前工作簿中的第一个工作表被激活

(122) ThisWorkbook.Worksheets("Sheet1").Rows(1).Font.Bold = True ‘设置工作表 Sheet1 中的行 1 数据为粗体

(123) ThisWorkbook.Worksheets("Sheet1").Rows(1).Hidden = True ‘将工作表 Sheet1 中的行 1 隐藏

ActiveCell.EntireRow.Hidden = True ‘将当前工作表中活动单元格所在的行隐藏

注：同样可用于列。

(124) ActiveSheet.Range("A:A").EntireColumn.AutoFit ‘自动调整当前工作表 A 列列宽

(125) ActiveSheet.Cells.SpecialCells(xlCellTypeConstants,xlTextValues) ‘选中当前工作表中常量和文本单元格

ActiveSheet.Cells.SpecialCells(xlCellTypeConstants,xlErrors+xlTextValues) ‘选中当前工作表中常量和文本及错误值单元格

公式与函数

(126) Application.MacroOptions Macro:="SumPro",Category:=4 ‘将自定义的 SumPro 函数指定给 Excel 中的“统计函数”类别

(127) Application.MacroOptions Macro:="SumPro", _

Description:="First Sum,then Product" ‘为自定义函数 SumPro 进行了功能说明

(128) Application.WorksheetFunction.CountA(Range("A:A"))+1 ‘获取 A 列的下一个空单元格

(129) WorksheetFunction.CountA(Cell.EntireColumn) ‘返回该单元格所在列非空单元格的数量

WorksheetFunction.CountA(Cell.EntireRow) ‘返回该单元格所在行非空单元格的数量

(130) WorksheetFunction.CountA(Cells) ‘返回工作表中非空单元格数量

(131) `ActiveSheet.Range("A20:D20").Formula="=Sum(R[-19]C:R[-1]C)"` 对 A 列至 D 列前 19 个数值求和

图表

(132) `ActiveWindow.Visible=False`

或 `ActiveChart.Deselect` ‘使图表处于非活动状态

(133) `TypeName(Selection)="Chart"` ‘若选中的为图表，则该语句为真，否则为假

(134) `ActiveSheet.ChartObjects.Delete` ‘删除工作表上所有的 `ChartObject` 对象

`ActiveWorkbook.Charts.Delete` ‘删除当前工作簿中所有的图表工作表

窗体和控件

(135) `UserForms.Add(MyForm).Show` ‘添加用户窗体 `MyForm` 并显示

(136) `TextBox1.SetFocus` ‘设置文本框获取输入焦点

(137) `SpinButton1.Value=0` ‘将数值调节钮控件的值改为 0

(138) `TextBox1.Text=SpinButton1.Value` ‘将数值调节钮控件的值赋值给文本框控件

`SpinButton1.Value=Val(TextBox1.Text)` ‘将文本框控件值赋给数值调节钮控件

`CStr(SpinButton1.Value)=TextBox1.Text` ‘数值调节钮控件和文本框控件相比较

(139) `UserForm1.Controls.Count` ‘显示窗体 `UserForm1` 上的控件数目

(140) `ListBox1.AddItem "Command1"` ‘在列表框中添加 `Command1`

(141) `ListBox1.ListIndex` ‘返回列表框中条目的值，若为 -1，则表明未选中任何列表框中的条目

(142) `RefEdit1.Text` ‘返回代表单元格区域地址的文本字符串

`RefEdit1.Text=ActiveWindow.RangeSelection.Address` ‘初始化 `RefEdit` 控件显示当前所选单元格区域

`Set FirstCell=Range(RefEdit1.Text).Range("A1")` ‘设置某单元格区域左上角单元格

(143) `Application.OnTime Now + TimeValue("00:00:15"), "myProcedure"` ‘等待 15 秒后运行 `myProcedure` 过程

(144) `ActiveWindow.ScrollColumn=ScrollBarColumns.Value` ‘将滚动条控件的值赋值给 `ActiveWindow` 对象的 `ScrollColumn` 属性

`ActiveWindow.ScrollRow=ScrollBarRows.Value` ‘将滚动条控件的值赋值给 `ActiveWindow` 对象的 `ScrollRow` 属性

(145) `UserForm1.ListBox1.AddItem Sheets("Sheet1").Cells(1,1)` ‘将单元格 A1 中的数据添加到列表框中

`ListBox1.List=Product` ‘将一个名为 `Product` 数组的值添加到 `ListBox1` 中

`ListBox1.RowSource="Sheet2!SumP"` ‘使用工作表 `Sheet2` 中的 `SumP` 区域的值填充列表框

(146) `ListBox1.Selected(0)` ‘选中列表框中的第一个条目(注：当列表框允许一次选中多个条目时，必须使用 `Selected` 属性)

(147) `ListBox1.RemoveItem ListBox1.ListIndex` ‘移除列表框中选中的条目

对象

Application 对象

- (148) Application.UserName ‘返回应用程序的用户名
- (149) Application.Caller ‘返回代表调用函数的单元格
- (150) Application.Caller.Parent.Parent ‘返回调用函数的工作簿名称
- (151) Application.StatusBar=”请等待…….” ‘将文本写到状态栏
Application.StatusBar=”请等待…….” & Percent & “% Completed” ‘更新状态栏文本，以变量 Percent 代表完成的百分比
Application.StatusBar=False ‘将状态栏重新设置成正常状态
- (152) Application.Goto Reference:=Range(“A1:D4”) ‘指定单元格区域 A1 至 D4，等同于选择“编辑——定位”，指定单元格区域为 A1 至 D4，不会出现“定位”对话框
- (153) Application.Dialogs(xlDialogFormulaGoto).Show ‘显示“定位”对话框，但定位条件按钮无效
- (154) Application.Dialogs(xlDialogSelectSpecial).Show ‘显示“定位条件”对话框
- (155) Application.Dialogs(xlDialogFormatNumber).show ‘显示“单元格格式”中的“数字”选项卡
Application.Dialogs(xlDialogAlignment).show ‘显示“单元格格式”中的“对齐”选项卡
Application.Dialogs(xlDialogFontProperties).show ‘显示“单元格格式”中的“字体”选项卡
Application.Dialogs(xlDialogBorder).show ‘显示“单元格格式”中的“边框”选项卡
Application.Dialogs(xlDialogPatterns).show ‘显示“单元格格式”中的“图案”选项卡
Application.Dialogs(xlDialogCellProtection).show ‘显示“单元格格式”中的“保护”选项卡
注：无法一次显示带选项卡的“单元格格式”对话框，只能一次显示一个选项卡。
- (156) Application.Dialogs(xlDialogFormulaGoto).show Range(“b2”), True ‘显示“引用位置”的默认单元格区域并显示引用使其出现在窗口左上角(注：内置对话框参数的使用)
- (157) Application.CommandBars(1).Controls(2).Controls(16).Execute ‘执行“定位”对话框，相当于选择菜单“编辑——定位”命令
- (158) Application.Transpose(Array(“Sun”, “Mon”, “Tur”, “Wed”, “Thu”, “Fri”, “Sat”)) ‘返回一个垂直的数组
- (159) Application.Version ‘返回使用的 Excel 版本号
- (160) Application.Cursor = xlNorthwestArrow ‘设置光标形状为北西向箭头
Application.Cursor = xlIBeam ‘设置光标形状为 I 字形
Application.Cursor = xlWait ‘设置光标形状为沙漏(等待)形
Application.Cursor = xlDefault ‘恢复光标的默认设置
- (161) Application.WindowState ‘返回窗口当前的状态
Application.WindowState = xlMinimized ‘窗口最小化
Application.WindowState = xlMaximized ‘窗口最大化
Application.WindowState = xlNormal ‘窗口正常状态
- (162) Application.UsableHeight ‘获取当前窗口的高度
Application.UsableWidth ‘获取当前窗口的宽度
- (163) Application.ActiveCell.Address ‘返回活动单元格的地址(注:返回的是绝对地址)
- (164) Application.ActivePrinter ‘返回当前打印机的名称
- (165) Application.ActiveSheet.Name ‘返回活动工作表的名称
- (166) Application.ActiveWindow.Caption ‘返回活动窗口的标题
- (167) Application.ActiveWorkbook.Name ‘返回活动工作簿的名称
- (168) Application.Selection.Address ‘返回所选区域的地址
- (169) Application.ThisWorkbook.Name ‘返回当前工作簿的名称
- (170) Application.CalculationVersion ‘返回 Excel 计算引擎版本(右边四位数字)及 Excel 版本(左边两位数字)

- (171) Application.MemoryFree ‘以字节为单位返回 Excel 允许使用的内存数(不包括已经使用的内存)
- (172) Application.MemoryUsed ‘以字节为单位返回 Excel 当前使用的内存数
- (173) Application.MemoryTotal ‘以字节为单位返回 Excel 可以使用的内存数(包括已使用的内存,是 MemoryFree 和 MemoryUsed 的总和)
- (174) Application.OperatingSystem ‘返回所使用的操作系统的名称和版本
- (175) Application.OrganizationName ‘返回 Excel 产品登记使用的组织机构的名称
- (176) Application.FindFormat ‘查找的格式种类
Application.ReplaceFormat ‘替换查找到的内容的格式种类
ActiveSheet.Cells.Replace What:=” “, _
Replacement:=” “, SearchFormat:=True, ReplaceFormat:=True ‘替换查找到的格式
- (177) Application.Interactive=False ‘忽略键盘或鼠标的输入
- (178) Application.Evaluate("Rate") ‘若在工作表中定义了常量 0.06 的名称为"Rate",则本语句将返回值 0.06
- (179) Application.OnUndo "Undo Option","Undo Procedure" ‘选择 UndoOption 后, 将执行 Undo Procedure 过程

Range 对象

- (180) Range(A1:A10).Value=Application.WorksheetFunction.Transpose(MyArray) ‘将一个含有 10 个元素的数组转置成垂直方向的工作表单元格区域(A1 至 A10)

注: 因为当把一维数组的内容传递给某个单元格区域时, 该单元格区域中的单元格必须是水平方向的, 即含有多列的一行。若必须使用垂直方向的单元格区域, 则必须先将数组进行转置, 成为垂直的。

- (181) Range("A65536").End(xlUp).Row+1 ‘返回 A 列最后一行的下一行
- (182) rng.Range("A1") ‘返回区域左上角的单元格
- (183) cell.Parent.Parent.Worksheets ‘访问当前单元格所在的工作簿
- (184) Selection.Font.Bold=Not Selection.Font.Bold ‘切换所选单元格是否加粗
- (185) ActiveSheet.Range("A:B").Sort Key1:=Columns("B"), Key2:=Columns("A"), _
Header:=xlYes ‘两个关键字排序, 相邻两列, B 列为主关键字, A 列为次关键字, 升序排列
- (186) cell.Range("A1").NumberFormat ‘显示单元格或单元格区域中的第一个单元格的数字格式
- (187) cell.Range("A1").HasFormula ‘检查单元格或单元格区域中的第一个单元格是否含有公式
或 cell.HasFormula ‘工作表中单元格是否含有公式
- (188) Cell.EntireColumn ‘单元格所在的整列
Cell.EntireRow ‘单元格所在的整行

- (189) rng.Name.Name ‘显示 rng 区域的名称

- (190) rng.Address ‘返回 rng 区域的地址

- (191) cell.Range("A1").Formula ‘返回包含在 rng 区域中左上角单元格中的公式。

注: 若在一个由多个单元格组成的范围内使用 Formula 属性, 会得到错误; 若单元格中没有公式, 会得到一个字符串, 在公式栏中显示该单元格的值。

- (192) Range("D5:D10").Cells(1,1) ‘返回单元格区域 D5:D10 中左上角单元格

- (193) ActiveCell.Row ‘活动单元格所在的行数

ActiveCell.Column ‘活动单元格所在的列数

- (194) Range("A1:B1").HorizontalAlignment = xlLeft ‘当前工作表中的单元格区域数据设置为左对齐

- (195) ActiveSheet.Range("A2:A10").NumberFormat="#,##0" ‘设置单元格区域 A2 至 A10 中数值格式

(196) rng.Replace “,”0” ‘用 0 替换单元格区域中的空单元格

Collection 与 object

(197) Dim colMySheets As New Collection

Public colMySheets As New Collection ‘声明新的集合变量

(198) Set MyRange=Range(“A1:A5”) ‘创建一个名为 MyRange 的对象变量

(199) <object>.Add Cell.Value CStr(Cell.Value) ‘向集合中添加惟一的条目(即将重复的条目忽略)

Windows API

(200) Declare Function GetWindowsDirectoryA Lib “kernel32” _

(ByVal lpBuffer As String,ByVal nSize As Long) As Long ‘API 函数声明。返回安装 Windows 所在的目录名称，调用该函数后，安装 Windows 的目录名称将在第一个参数 lpBuffer 中，该目录名称的字符串长度包含在第二个参数 nSize 中

(By fanjy in 2006-6-24)

VBA 语句集

(第3辑)

前面已经推出了两辑VBA 语句集，共有200 句VBA 常用代码及代码功能的简要解释。根据前阶段在学习VBA 过程中总结归纳的成果，特汇编了VBA 语句集第3 辑，供大家在学习VBA编程时参考。其实，您可以在VBE编辑器中将这语句进行测试，以体验其作用或效果。

VBA语句集的特点是，一句VBA代码，后面配有代码功能简要的说明或解释。每辑100句，尽可能收录所有在程序中所要用到的代码。

(201) Set objExcel = CreateObject("Excel.Application")objExcel.Workbooks.Add '创建Excel 工作簿

(202) Application.ActivateMicrosoftApp xlMicrosoftWord '开启Word应用程序

(203) Application.TemplatesPath '获取工作簿模板的位置

(204) Application.Calculation = xlCalculationManual '设置工作簿手动计算

Application.Calculation = xlCalculationAutomatic '工作簿自动计算

(205) Worksheets(1).EnableCalculation = False '不对第一张工作表自动进行重算

(206) Application.CalculateFull '重新计算所有打开的工作簿中的数据

(207) Application.RecentFiles.Maximum = 5 '将最近使用的文档列表数设为5

(208) Application.RecentFiles(4).Open '打开最近打开的文档中的第4个文档

(209) Application.OnTime DateSerial(2006,6,6)+TimeValue("16:16:16"),"BaoPo" '在2006年6月6日的16:16:16开始运行BaoPo过程

(210) Application.Speech.Speak ("Hello" & Application.UserName) '播放声音，并使用用户的姓名问候用户

(211) MsgBox Application.PathSeparator '获取"\ "号

(212) MsgBox Application.International(xlCountrySetting) '返回应用程序当前所在国家的设置信息

(213) Application.AutoCorrect.AddReplacement "葛洲坝","三峡" '自动将在工作表中进行输入的"葛洲坝"更正为"三峡"

(214) Beep '让计算机发出声音

(215) Err.Number '返回错误代码

(216) MsgBox IMEStatus '获取输入法状态

(217) Date = #6/6/2006#Time = #6:16:16 AM# '将系统时间更改为2006年6月6日上午6时16分16秒

(218) Application.RollZoom = Not Application.RollZoom '切换是否能利用鼠标中间的滑轮放大/缩小工作表

(219) Application.ShowWindowsInTaskba = True '显示任务栏中的窗口,即各工作簿占用各自的窗口

(220) Application.DisplayScrollBars = True '显示窗口上的滚动条

(221) Application.DisplayFormulaBar = Not Application.DisplayFormulaBar '切换是否显示编辑栏

(222) Application.Dialogs(xlDialogPrint).Show '显示打印内容对话框

(223) Application.MoveAfterReturnDirection = xlToRight '设置按Enter键后单元格的移动方向向右

(224) Application.FindFile '显示打开对话框

(225) ThisWorkbook.FollowHyperlink <http://fanjy.blog.excelhome.net> '打开超链接文档

(226) ActiveWorkbook.ChangeFileAccess Mode:=xlReadOnly '将当前工作簿设置为只读

(227) ActiveWorkbook.AddToFavorites '将当前工作簿添加到收藏夹文件夹中

(228) ActiveSheet.CheckSpelling '在当前工作表中执行"拼写检查"

(229) ActiveSheet.Protect userinterfaceonly:=True '保护当前工作表

(230) ActiveSheet.PageSetup.LeftHeader = ThisWorkbook.FullName '在当前工作表的左侧页眉处打印出工作簿的完整路径和文件名

- (231) Worksheets("Sheet1").Range("A1:G37").Locked = False Worksheets("Sheet1").Protect
'解除对工作表Sheet1中A1:G37区域单元格的锁定
'以便当该工作表受保护时也可对这些单元格进行修改
- (232) Worksheets("Sheet1").PrintPreview '显示工作表sheet1的打印预览窗口
- (233) ActiveSheet.PrintPreview Enablechanges:=False '禁用显示在Excel 的“打印预览”窗口中的“设置”和“页边距”按钮
- (234) ActiveSheet.PageSetup.PrintGridlines = True '在打印预览中显示网格线
ActiveSheet.PageSetup.PrintHeadings = True '在打印预览中显示行列编号
- (235) ActiveSheet.ShowDataForm '开启数据记录单
- (236) Worksheets("Sheet1").Columns("A").Replace _
What:="SIN", Replacement:="COS", _
SearchOrder:=xlByColumns, MatchCase:=True '将工作表sheet1中A列的SIN替换为COS
- (237) Rows(2).Delete '删除当前工作表中的第2行
Columns(2).Delete '删除当前工作表中的第2列
- (238) ActiveWindow.SelectedSheets.VPageBreaks.Add before:=ActiveCell '在当前单元格左侧插入一条垂直分页符
ActiveWindow.SelectedSheets.HPageBreaks.Add before:=ActiveCell '在当前单元格上方插入一条垂直分页符
- (239) ActiveWindow.ScrollRow = 14 '将当前工作表窗口滚动到第14行
ActiveWindow.ScrollColumn = 13 '将当前工作表窗口滚动到第13列
- (240) ActiveWindow.Close '关闭当前窗口
- (241) ActiveWindow.Panes.Count '获取当前窗口中的窗格数
- (242) Worksheets("sheet1").Range("A1:D2").CreateNames Top:=True '将A2至D2的单元格名称设定为A1到D1单元格的内容
- (243) Application.AddCustomList listarray:=Range("A1:A8") '自定义当前工作表中单元格A1至A8中的内容为自动填充序列
- (244) Worksheets("sheet1").Range("A1:B2").CopyPicture xlScreen, xlBitmap '将单元格A1至B2的内容复制成屏幕快照
- (245) Selection.Hyperlinks.Delete '删除所选区域的所有链接
Columns(1).Hyperlinks.Delete '删除第1列中所有的链接
Rows(1).Hyperlinks.Delete '删除第1行中所有的链接
Range("A1:Z30").Hyperlinks.Delete '删除指定范围所有的链接
- (246) ActiveCell.Hyperlinks.Add Anchor:=ActiveCell, _
Address:="C:\Windows\System32\Calc.exe", ScreenTip:=" 按下我， 就会开启Windows 计算器",
TextToDisplay:="Windows 计算器" '在活动单元格中设置开启Windows计算器链接
- (247) ActiveCell.Value = Shell("C:\Windows\System32\Calc.exe", vbNormalFocus) '开启Windows计算器
- (248) ActiveSheet.Rows(1).AutoFilter '打开自动筛选。若再运行一次，则关闭自动筛选
- (249) Selection.AutoFilter '开启/关闭所选区域的自动筛选
- (250) ActiveSheet.ShowAllData '关闭自动筛选
- (251) ActiveSheet.AutoFilterMode '检查自动筛选是否开启，若开启则该语句返回True
- (252) ActiveSheet.Columns("A").ColumnDifferences(Comparison:=ActiveSheet. _
Range("A2")).Delete '在A列中找出与单元格A2内容不同的单元格并删除
- (253) ActiveSheet.Range("A6").ClearNotes '删除单元格A6中的批注，包括声音批注和文字批注

- (254) `ActiveSheet.Range("B8").ClearComments` '删除单元格B8中的批注文字
- (255) `ActiveSheet.Range("A1:D10").ClearFormats` '清除单元格区域A1至D10中的格式
- (256) `ActiveSheet.Range("B2:D2").BorderAround ColorIndex:=5, _
Weight:=xlMedium, LineStyle:=xlDouble` '将单元格B2至D2区域设置为蓝色双线
- (257) `Range("A1:B2").Item(2, 3)`或`Range("A1:B2")(2, 3)` '引用单元格C2的数据
`Range("A1:B2")(3)` '引用单元格A2
- (258) `ActiveSheet.Cells(1, 1).Font.Bold = TRUE` '设置字体加粗
`ActiveSheet.Cells(1, 1).Font.Size = 24` '设置字体大小为24磅
`ActiveSheet.Cells(1, 1).Font.ColorIndex = 3` '设置字体颜色为红色
`ActiveSheet.Cells(1, 1).Font.Italic = TRUE` '设置字体为斜体
`ActiveSheet.Cells(1, 1).Font.Name = "Times New Roman"` '设置字体类型
`ActiveSheet.Cells(1, 1).Interior.ColorIndex = 3` '将单元格的背景色设置为红色
- (259) `ActiveSheet.Range("C2:E6").AutoFormat Format:=xlRangeAutoFormatColor3` '将当前工作表中单元格区域C2至E6格式自动调整为彩色3格式
- (260) `Cells.SpecialCells(xlCellTypeLastCell)` '选中当前工作表中的最后一个单元格
- (261) `ActiveCell.CurrentArray.Select` '选定包含活动单元格的整个数组单元格区域.假定该单元格在数据单元格区域中
- (262) `ActiveCell.NumberFormatLocal = "0.000; [红色] 0.000"` '将当前单元格数字格式设置为带3位小数, 若为负数则显示为红色
- (263) `IsEmpty (ActiveCell.Value)` '判断活动单元格中是否有值
- (264) `ActiveCell.Value = LTrim(ActiveCell.Value)` '删除字符串前面的空白字符
- (265) `Len(ActiveCell.Value)` '获取活动单元格中字符串的个数
- (266) `ActiveCell.Value = UCase(ActiveCell.Value)` '将当前单元格中的字符转换成大写
- (267) `ActiveCell.Value = StrConv(ActiveCell.Value, vbLowerCase)` '将活动单元格中的字符串转换成小写
- (268) `ActiveSheet.Range("C1").AddComment` '在当前工作表的单元格C1中添加批注
- (269) `Weekday(Date)` '获取今天的星期,以数值表示,1-7分别对应星期日至星期六
- (270) `ActiveSheet.Range("A1").AutoFill Range(Cells(1, 1), Cells(10, 1))` '将单元格A1的数值填充到单元格A1至A10区域中
- (271) `DatePart("y", Date)` '获取今天在全年中的天数
- (272) `ActiveCell.Value = DateAdd("yyyy", 2, Date)` '获取两年后的今天的日期
- (273) `MsgBox WeekdayName(Weekday(Date))` '获取今天的星期数
- (274) `ActiveCell.Value = Year(Date)` '在当前单元格中输入今年的年份数
`ActiveCell.Value = Month(Date)` '在当前单元格中输入今天所在的月份数
`ActiveCell.Value = Day(Date)` '在当前单元格中输入今天的日期数
- (275) `ActiveCell.Value = MonthName(1)` '在当前单元格中显示月份的名称,本句为显示"一月"
- (276) `ActiveCell.Value = Hour(Time)` '在当前单元格中显示现在时间的小时数
`ActiveCell.Value = Minute(Time)` '在当前单元格中显示现在时间的分钟数
`ActiveCell.Value = Second(Time)` '在当前单元格中显示现在时间的秒数
- (277) `ActiveSheet.Shapes(1).Delete` '删除当前工作表中的第一个形状
- (278) `ActiveSheet.Shapes.Count` '获取当前工作表中形状的数量
- (279) `ActiveSheet.Shapes(1).TextEffect.ToggleVerticalText` '改变当前工作表中第一个艺术字的方向
- (280) `ActiveSheet.Shapes(1).TextEffect.FontItalic = True` '将当前工作表中第一个艺术字的字体设置为斜体

(281) `ActiveSheet.Shapes.AddTextEffect(msoTextEffect21, "三峡", _
"Arial Black", 22#, msoFalse, msoFalse, 66#, 80).Select` '在当前工作表中创建一个名为"三峡"的艺术字
并对其进行格式设置和选中

(282) `ActiveSheet.Shapes.AddLine(BeginX:=10, BeginY:=10, EndX:=250, _
EndY:=100).Select` '在当前工作表中以(10,10)为起点(250,100)为终点画一条直线并选中

(283) `ActiveSheet.Shapes.AddShape(Type:=msoShapeRightTriangle, _
Left:=70, Top:=40, Width:=130, Height:=72).Select` '在当前工作表中画一个左上角在(70,40),宽为130高
为72的三角形并选中

(284) `ActiveSheet.Shapes.AddShape(Type:=msoShapeRectangle, _
Left:=70, Top:=40, Width:=130, Height:=72).Select` '在当前工作表中画一个以点(70,40)为起点,宽130高
72的矩形并选中

(285) `ActiveSheet.Shapes.AddShape(Type:=msoShapeOval, _
Left:=70, Top:=40, Width:=130, Height:=72).Select` '在当前工作表中画一个左上角在(70,40),宽为130高
为72的椭圆

(286) `ActiveSheet.Shapes(1).Line.ForeColor.RGB = RGB(0, 0, 255)` '将当前工作表中第一个形状的线条
颜色变为蓝色

(287) `ActiveSheet.Shapes(2).Fill.ForeColor.RGB = RGB(255, 0, 0)` '将当前工作表中第2个形状的前景色
设置为红色

(288) `ActiveSheet.Shapes(1).Rotation = 20` '将当前工作表中的第1个形状旋转20度

(289) `Selection.ShapeRange.Flip msoFlipHorizontal` '将当前选中的形状水平翻转
`Selection.ShapeRange.Flip msoFlipVertical` '将当前选中的形状垂直翻转

(290) `Selection.ShapeRange.ThreeD.SetThreeDFormat msoThreeD1` '将所选取的形状设置为第1种立
体样式

(291) `ActiveSheet.Shapes(1).ThreeD.Depth = 20` '将当前工作表中第一个立体形状的深度设置为20

(292) `ActiveSheet.Shapes(1).ThreeD.ExtrusionColor.RGB = RGB(0, 0, 255)` '将当前工作表中第1个立
体形状的进深部分的颜色设为蓝色

(293) `ActiveSheet.Shapes(1).ThreeD.RotationX = 60` '将当前工作表中的第1个立体形状沿X轴旋转60度
`ActiveSheet.Shapes(1).ThreeD.RotationY = 60` '将当前工作表中的第1个立体形状沿Y轴旋转60度

(294) `Selection.ShapeRange.ThreeD.Visible = msoFalse` '将所选择的立体形状转换为平面形状

(295) `Selection.ShapeRange.ConnectorFormat.BeginDisconnect` '在形状中让指定的连接符起点脱离原
来所连接的形状

(296) `ActiveSheet.Shapes(1).PickUp` '复制当前工作表中形状1的格式

(297) `ActiveSheet.Shapes.AddTextbox(msoTextOrientationHorizontal, 260, 160, 180, 30).
TextFrame.Characters.Text = "fanjy.blog.excelhome.net"` '在工作簿中新建一个文本框并输入内容

(298) `ActiveSheet.Shapes.AddLabel(msoTextOrientationHorizontal, 20, 80, 100, 200).
TextFrame.Characters.Text = "fanjy.blog.excelhome.net"` '在当前工作表中建立一个水平文本框并输入
内容

(299) `ActiveSheet.Shapes.AddPicture "d:\sx.jpg", True, True, 60, 20, 400, 300` '在当前工作表中插入一
张d盘中名为sx 的图片

(300) `ActiveChart.ApplyCustomType xl3DArea` '将当前图表类型改为三维面积图

By fanjy in 2006-10-7