



为你提供 **13** 款拿来即用的 Excel 工具

完美 Excel 出品
微信公众号: [excelperfect](#)



内容提要

如果已经有人帮你做好的模板，搭好了框架，实现了某些想要的功能，为什么不拿来用呢？

本电子书《[为你提供 13 款拿来即用的 Excel 工具](#)》整理了 13 款 Excel 实用工具，为你在使用 Excel 时提供便利。



目录

| | |
|----------------------------|----|
| 1. 一个简单的 VBA 语句翻译器模型 | 1 |
| 2. VBA 通用代码库 | 9 |
| 3. Excel 函数宝典 | 15 |
| 4. 全面获取工作簿的使用信息 | 23 |
| 5. 在 Excel 中创建专业的甘特图 | 26 |
| 6. 工作时间及薪水计算器 | 29 |
| 7. Excel 日历模板 | 33 |
| 8. 一款实用的自定义功能区加载项 | 37 |
| 9. 一款定制右键菜单的加载项 | 41 |
| 10. Excel 工作任务清单模板 | 47 |
| 11. 任务矩阵模板 | 51 |
| 12. 随机分组模板 | 55 |
| 13. 每日事项清单——与日历结合 | 59 |
| 关于完美 Excel | 61 |
| 完美 Excel 微信公众号使用指南 | 62 |



温馨提示：

在完美 Excel 微信公众号中发送消息：**Excel 实用工具**，即可获得本电子书文档下载链接和密码。



如果您对本书有什么建议或者还有什么好的示例，欢迎前往完美 Excel 微信公众号沟通交流。

欢迎分享本电子书，让更多的人方便地得到所需要的知识。



本章内容 2017 年 8 月 28 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
**Excel 实用工具 1: 一个简单的 VBA 语
句翻译器模型**

1. 一个简单的 VBA 语句翻译器模型

引言: 如果能够制作一款 VBA 程序翻译器, 将代码翻译成中文, 那一定会对很多 VBA 学习者有帮助。这里试图给出一个思路并搭建一个基本的框架, 希望有兴趣的朋友能继续研究完善。

正如本章标题所表达的, 下面介绍的只是一个简单的模型, 只是搭建了一个基本的框架, 实现了最简单的 VBA 语句翻译, 如下图 1.1 所示。要实现更丰富的功能, 还需要花时间精雕细琢。



图 1.1



框架模型

用户窗体

在 VBE 编辑器中插入一个用户窗体，用于输入或粘贴 VBA 代码，并呈现译文，如下图 1.2 所示。



图 1.2

数据工作表

工作表 Sheet1 用于存放临时数据。

工作表 Sheet2 用于存放 VBA 关键词和语句，以及相应的中文。



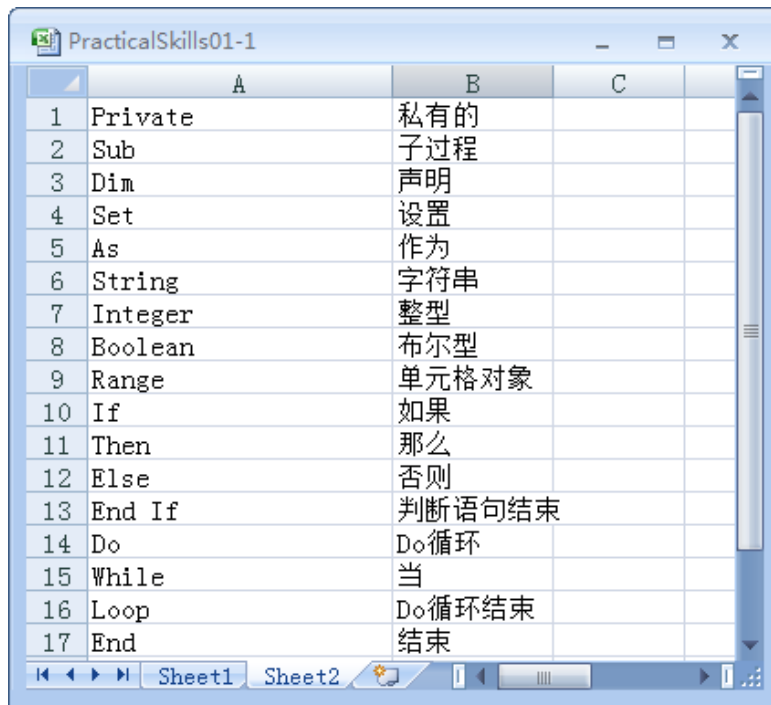


图 1.3

为方便自动扩展，给存放 VBA 关键词和语句的单元格区域定义动态名称 VBASTatementDB:

```
=Sheet2!$A$1:INDEX(Sheet2!$A:$B,COUNTA(Sheet2!$A:$A),2)
```

代码及解析

用户窗体代码

“翻译”按钮的代码:

```
Private Sub cmdParse_Click()
    Dim strString As String
    Dim intStart As Integer, intEnd As Integer
    Dim intStrLen As Integer, intCrLf As Integer
    Dim blnLines As Boolean
    Dim rng As Range

    Sheet1.Range("A:B").ClearContents
    Set rng = Sheet1.Range("A1")
```



```

intStart = 1

strString = Trim(txtString.Text)

intStrLen = Len(strString)

intCrLf = InStr(1, strString, vbCrLf)

If intCrLf Then blnLines = True

Do While intStart > 0
    intEnd = InStr(intStart, strString, " ") - 1
    If intEnd <= 0 Then intEnd = intStrLen
    If blnLines And (intCrLf < intEnd) Then
        intEnd = intCrLf - 1
        intCrLf = InStr(intEnd + 2, strString, vbCrLf)
        If intCrLf = 0 Then blnLines = False
        rng.Value = Mid(strString, intStart, intEnd - intStart + 1)
        Set rng = rng.Offset(1, 0) '在每行后面插入一个标记
        rng.Value = "Finished"
        intStart = intEnd + 3
    Else
        rng.Value = Mid(strString, intStart, intEnd - intStart + 1)
        intStart = intEnd + 2
    End If
    Set rng = rng.Offset(1, 0)
    If intStart > intStrLen Then intStart = 0
Loop

'翻译

Translation

End Sub

```



代码解析：

- 代码主要是以空格和换行符为分隔符来分解文本框中的句子成单个的单词，并将它们依次放置到工作表 Sheet1 的列 A 中。
- Trim 函数删除文本框中字符串的前面和末尾的空格。
- Len 函数计算文本框中字符的数量。
- InStr 函数分别用来查找回车/换行符、空格在字符串中的位置，以提取单词或确定一行语句的结束。
- Mid 函数用来提取单词。

“清空”按钮的代码：

```
Private Sub cmdClear_Click()  
    Dim intCancel As Integer  
    txtString.SetFocus  
    intCancel = MsgBox("文本框中共有" & txtString.LineCount & "行文本." & vbCrLf  
& _  
        "要清空它们吗？ ", vbOKCancel)  
    If intCancel = 1 Then  
        txtString.Text = ""  
        txtChinese.Text = ""  
    End If  
End Sub
```

代码解析：

- 如果用户要清除文本框中的内容，那么单击提示框中的确定按钮后，两个文本框中的内容都被清空。

标准模块代码

模块开头，声明一个公共变量，用于确定工作表中最后一行：

```
Dim lngLastRow As Long
```

Translation 过程用于调用下面的 3 个子过程：

```
Sub Translation()
```



```
TransToCell  
JoiningAndPutTextBox  
ClearSheet  
End Sub
```

TransToCell 过程用于将 Sheet1 工作表中提取的单词“翻译”成对应的中文：

```
Sub TransToCell()  
    Dim rng As Range, rngFound As Range  
  
    lngLastRow = Sheet1.Range("A" & Rows.Count).End(xlUp).Row  
  
    For Each rng In Sheet1.Range("A1:A" & lngLastRow)  
        Set rngFound = [VBAStatementDB].Find(What:=rng, LookAt:=xlWhole)  
        If Not rngFound Is Nothing Then  
            rng.Offset(0, 1) = rngFound.Offset(0, 1)  
        Else  
            rng.Offset(0, 1) = rng  
        End If  
    Next rng  
End Sub
```

代码中使用 Find 方法在名称区域 VBAStatementDB 中查找并将找到的内容放置在对应该单元格相邻的单元格中。如果没有找到，则存放原值。

JoiningAndPutTextBox 过程用于将放置“翻译”后的单词的单元格连接，并放置到译文内容的文本框：

```
Sub JoiningAndPutTextBox()  
    Dim rng As Range  
    Dim str As String  
  
    For Each rng In Sheet1.Range("B1:B" & lngLastRow)  
        If rng.Value <> "Finished" Then
```



```
        If rng.Value = "" Then
            str = str & " "
        Else
            str = str & rng
        End If
    Else
        str = str & vbCrLf
    End If
Next rng

    UTran.txtChinese.Text = str
End Sub
```

ClearSheet 过程用于清除存放在工作表 Sheet1 中的临时数据：

```
Sub ClearSheet()
    Sheet1.Cells.Clear
End Sub
```

结语

上述给出的思路和框架都很简单，代码也不复杂，但要实现更强大的功能，翻译不同情形的 VBA 语句，还需要下大功夫。

这里抛了一块“砖头”出来，有兴趣的朋友可以继续研究。





本章内容 2017 年 9 月 5 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 实用工具 2: VBA 通用代码库](#)

2.VBA 通用代码库

引言：如果能够做个 VBA 通用代码库，将常用的 VBA 代码整理在一起，想要使用时随时调出来，然后将其直接复制到需要的地方，岂不很方便。此外，这些通用代码也是 VBA 应用的精华，阅读和研究这些代码也助于学习 VBA，提高编程水平。一举两得！

下面是捣鼓了半天弄出来的一个初步的东西，还很不成熟，先放出来，让大家提提建议，在后面的日子里逐步完善。

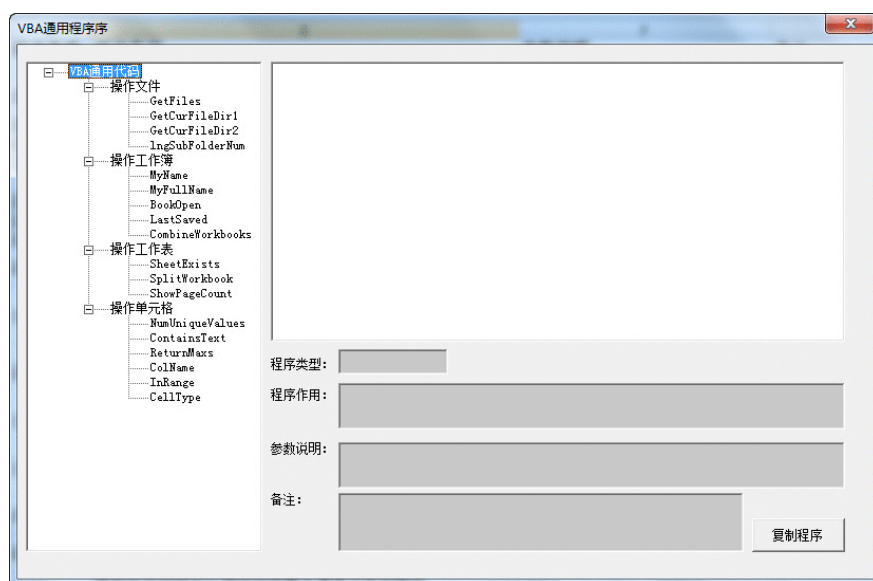


图 2.1

准备工作

“子过程清单”工作表：用来放置从 VBE 编辑器中获取的所有子过程，包括过程名、所在的模块名、以及过程在模块中的开始行号。

“数据库”工作表：用来放置程序的分类，以及每个分类下的过程名称、相应的程序类型、用途、参数说明、备注说明等信息。



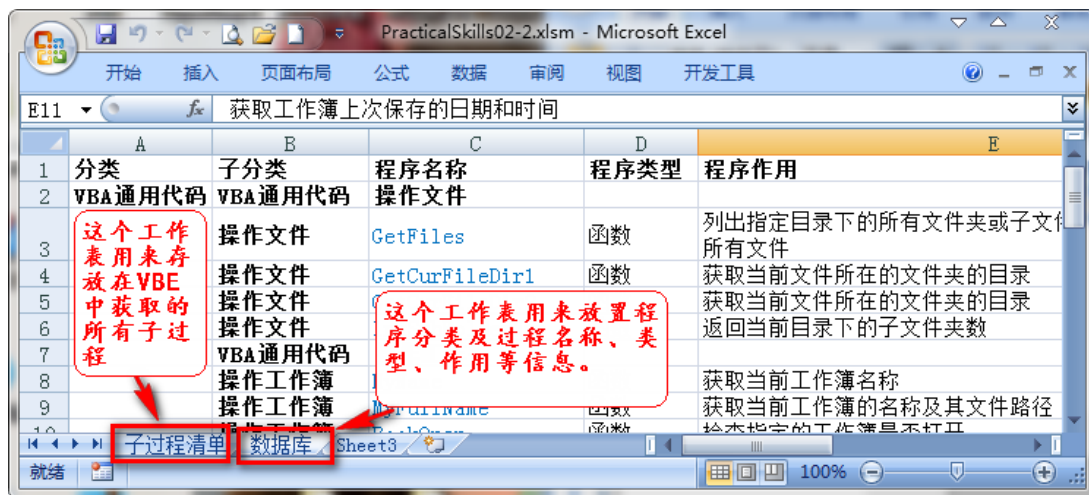


图 2.2

定义动态名称

定义动态名称 ProcList:

=子过程清单!\$A\$2:INDEX(子过程清单!\$A:\$C,COUNTA(子过程清单!\$A:\$A),3)

这样，当“子过程清单”中的数据增减时，区域会自动调整。

该名称用于“数据库”工作表中作为 VLOOKUP 函数的参数来获取相应的子过程名所开始的行号，以及其所在的模块名称。

由于我们要使用 VBA 代码操作 VBE，因此切换到 VBE 编辑器中，单击“工具——引用”，在对话框中找到“Microsoft Visual Basic for Applications Extensibility 5.3”并选中，如下图 2.3 所示。

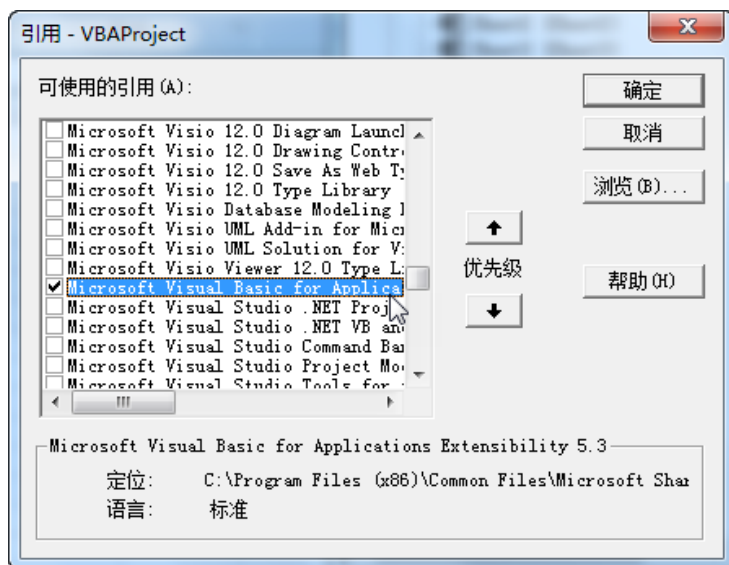


图 2.3



然后，在“Excel 选项”中，打开“信任中心”，并选中“信任对 VBA 工程对象模型的访问”，如下图 2.4 所示。

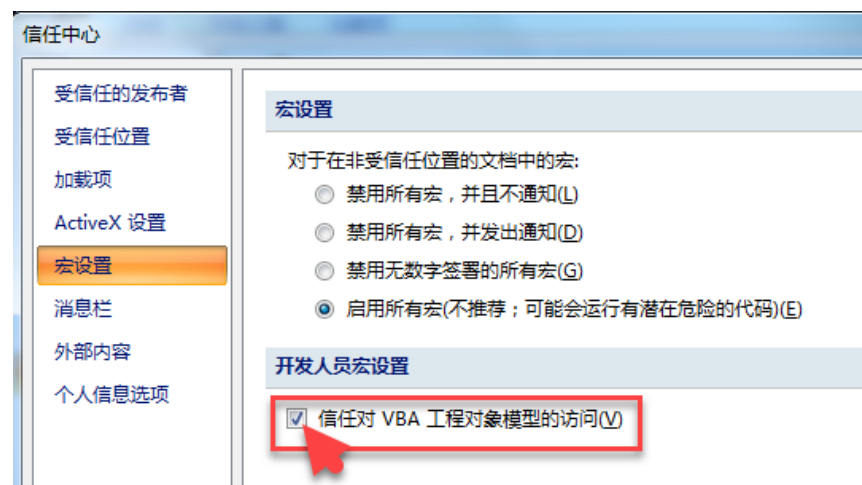


图 2.4

用户窗体设计

在 VBE 中，插入一个用户窗体，并添加相应的控件，如下图 2.5 所示。



图 2.5

编写代码

ThisWorkbook 模块

在 ThisWorkbook 模块中，输入下面的代码：



```

Private Sub Workbook_Open()
    ShowProc
End Sub

Sub ShowProc()
    Dim VBP As VBIDE.VBProject
    Dim VBC As VBCComponent
    Dim CM As CodeModule
    Dim lngStartLine As Long
    Dim strProcName As String
    Dim lngRow As Long

    Set VBP = ActiveWorkbook.VBProject
    lngRow = 1

    With Sheet1
        .Cells.ClearContents
        .Range("A1:C1") = Array("过程名", "模块名", "起始行")

        For Each VBC In VBP.VBComponents
            Set CM = VBC.CodeModule
            lngStartLine = CM.CountOfDeclarationLines + 1
            Do Until lngStartLine >= CM.CountOfLines
                lngRow = lngRow + 1
                .Cells(lngRow, 2) = VBC.Name
                .Cells(lngRow, 1) = CM.ProcOfLine(lngStartLine, vbext_pk_Proc)
                .Cells(lngRow, 3) = lngStartLine
                lngStartLine = lngStartLine + CM.ProcCountLines - 1
                .Cells(lngRow, 1) = CM.ProcOfLine(lngStartLine, vbext_pk_Proc), vbext_pk_Proc)
            Loop
        Next VBC
    End With
End Sub

```

打开工作簿时，代码自动获取 VBE 中所有的过程名、模块名、以及每个过程在模块中的起始行，并放置在“子过程清单”工作表中。

用户窗体代码

下面的代码在启动用户窗体时，自动使用“数据库”工作表中的数据填充 TreeView1 控件。

```

Private Sub UserForm_Initialize()
    treeViewPopulate
End Sub

' 填充TreeView控件
Sub treeViewPopulate()
    Dim i As Long
    Dim j As Long
    Dim xNode As Node
    Dim NodeKey As String
    Dim strWksName As String
    Dim objWks As Worksheet

    strWksName = "数据库"
    Set objWks = ActiveWorkbook.Sheets(strWksName)
    j = objWks.Range("A" & Rows.Count).End(xlUp).Row
    With Me.TreeView1
        ' 第一层次的节点
        For i = 2 To j
            Set xNode = .Nodes.Add
            NodeKey = objWks.Range("A" & i).text
            With xNode
                .Key = NodeKey
                .text = NodeKey
                .Expanded = False
            End With
        Next i
        j = objWks.Range("C" & Rows.Count).End(xlUp).Row
        ' 其他节点
        For i = 2 To j
            Set xNode = .Nodes.Add(objWks.Range("B" & i).text, tvwChild)
            NodeKey = objWks.Range("C" & i).text
            With xNode
                .Key = NodeKey
                .text = NodeKey
            End With
        Next i
    End With
End Sub

```



下面的代码在单击树状目录中的子过程名时，调用子过程在右侧的文本框中显示过程代码及相应的信息。

```
Private Sub TreeView1_NodeClick(ByVal Node As MSComctlLib.Node)
    Dim strProcName As String
    strProcName = Node.text
    FindAndPass strProcName
End Sub
```

下面的代码在用户单击“复制代码”按钮后，将文本框中的代码复制到剪贴板。

```
Private Sub cmdCopy_Click()
    With txtProc
        .SelStart = 0
        .SelLength = .TextLength
        .Copy
    End With
    ShowInfo
End Sub

Sub ShowInfo()
    MsgBox "复制代码成功!" & vbNewLine &
        "可以将代码粘贴到需要的地方使用。"
End Sub
```

其他代码

下面的代码用于获取在树状目录中单击的子过程的代码及相应的信息，并放置在相应的文本框中。

```
Sub FindAndPass(strProc As String)
    Dim wks As Worksheet
    Dim rng As Range

    On Error Resume Next
    Set wks = Worksheets("数据库")
    If strProc = "" Then MsgBox "请在左侧选择过程名"
    Set rng = wks.Range("C:C").Find(What:=strProc)

    If Not rng Is Nothing Then
        UProc.txtProc.text = CopySelectedProc(rng.Offset(0, 6), vbext_pk_Proc)
        UProc.txtType.text = rng.Offset(0, 1)
        UProc.txtUse.text = rng.Offset(0, 2)
        UProc.txtPara.text = rng.Offset(0, 3)
        UProc.txtComm.text = rng.Offset(0, 4)
    Else
        UProc.txtProc.text = ""
        UProc.txtType.text = ""
        UProc.txtUse.text = ""
        UProc.txtPara.text = ""
        UProc.txtComm.text = ""
    End If

    Set wks = Nothing
    Set rng = Nothing
End Sub
```



```

Function CopySelectedProc(StartLine As Long, ProcType As Long) As String
    Dim lngStartLine As Long, lngStartCol As Long
    Dim lngEndLine As Long, lngEndCol As Long
    Dim lngProcType As Long, strProcName As String
    Dim lngProcStart As Long, lngProcEnd As Long

    lngStartLine = StartLine
    lngProcType = ProcType

    With Application.VBE.ActiveCodePane

        With .CodeModule
            If lngStartLine <= .CountOfDeclarationLines Then
                lngProcStart = 1
                lngProcEnd = .CountOfDeclarationLines
            Else
                strProcName = .ProcOfLine(lngStartLine, lngProcType)
                lngProcStart = .ProcStartLine(strProcName, lngProcType)
                lngProcEnd = lngProcStart + .ProcCountLines(strProcName, lngProcType)
            End If
            CopySelectedProc = .Lines(lngProcStart, .ProcCountLines(strProcName, lngProcType))
        End With
    End With
End Function

```

结语

这个应用程序需要完善的地方还很多，所以具体的代码语句没有详细讲解。接下来有时间，要开展下列工作：

- 完善程序代码，使程序更健壮。
- 收集整理通用程序代码，使程序库更丰富。

在此其基础上：

- 美化界面并扩展功能。



本章内容 2017 年 9 月 18 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 实用工具 3: Excel 函数宝典](#)

3. Excel 函数宝典

引言：学好并熟练运用 Excel 函数将能发挥出 Excel 无尽的威力。然而，Excel 有几百个函数，要想将它们全都熟练掌握是一件非常困难的事。并且，函数之间的相互配合，变化多端，有着多种多样的使用情形。如果我们随时能够查阅到各个函数详细的语法和使用说明，并且有丰富的示例作为参考，那一定非常有助于我们解决问题，也有利于我们函数运用水平的精进。

今天给出的这个工具——Excel 函数宝典，就是想要将 Excel 函数集中在一起，不仅有详细的语法和使用说明，还有丰富的参考示例，便于学习，便于查阅参考。

由于时间有限，整理全部的 Excel 函数语法以及应用示例并不是三两天就能完成的。因此，这里先给出工具的架构，以及功能演示，听听大家的意见，我再进行完善。

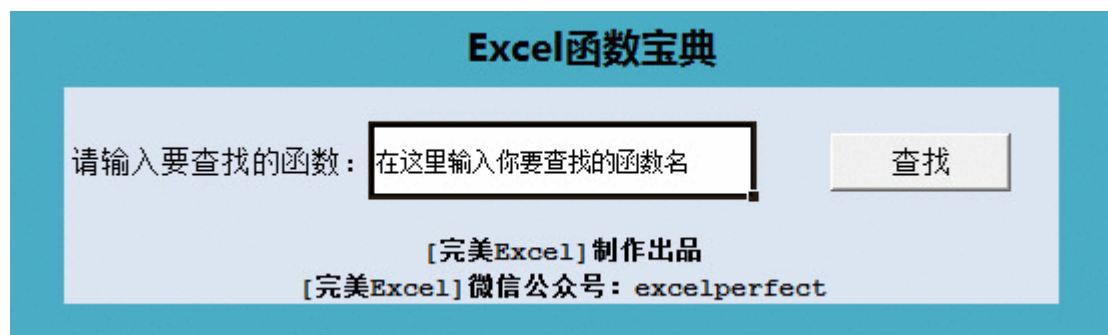


图 3.1

“Excel 函数宝典”的功能

“Excel 函数宝典”实际上就是 Excel 函数库，力图汇集所有的 Excel 函数，甚至有有用的自定义函数。主要功能有：

- 查找想要的函数并给出该函数的详细语法和使用说明。
 - ✧ 输入函数名后查找
 - ✧ 在分类框中选择想要的函数后查找



- 方便获取每个函数的应用示例，作为参考。

“Excel 函数宝典” 的使用

可以在工作表界面中输入想要查找的函数，也可以在弹出的用户窗体中输入想要查找的函数，单击“查找”按钮查找到该函数的详细语法和使用说明。

找到函数后，单击“详细示例”按钮，调出包含关于该函数详细信息以及应用示例的工作簿文档。

下面的视频演示了如何使用“Excel 函数宝典”。（视频详见[完美 Excel 微信公众号](#)）

“Excel 函数宝典”的设计

工作表设计

下图 3.2 所示为工作表主界面，在命名为“SearchFuncName”的单元格中输入函数名，单击右侧的“查找”按钮，即可调出相关函数信息。

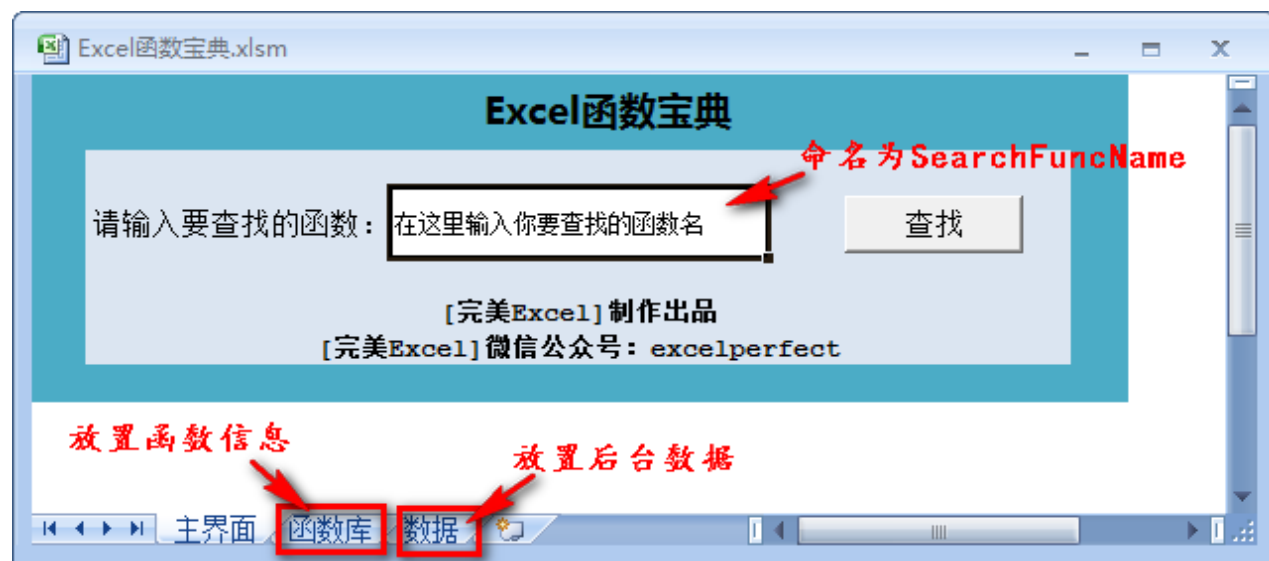


图 3.2

“函数库”工作表存放全部函数的基本信息，包括函数名、类别、版本、作用、语法、参数说明。包含着函数名的区域：

名称：FuncName = OFFSET(函数库!\$A\$3,0,0,COUNTA(函数库!\$A:\$A)-2,1)

“数据”工作表存放程序后台需要的数据，包含着一些命名区域。



函数类别名: Cate

引用位置: =OFFSET(数据!\$A\$2,0,0,COUNTA(数据!\$A:\$A)-1,1)

各分类的名称所代表的区域:

名称: 查找与引用

引用位置: =OFFSET(数据!\$D4,0,0,COUNTA(数据!\$D:\$D)-1,1)

名称: 文本

引用位置: =OFFSET(数据!\$E\$2,0,0,COUNTA(数据!\$E:\$E)-1,1)

.....

用户窗体设计

在 VBE 中, 插入用户窗体, 放置相应的控件, 设置控件格式并命名, 如图 3.3 所示。

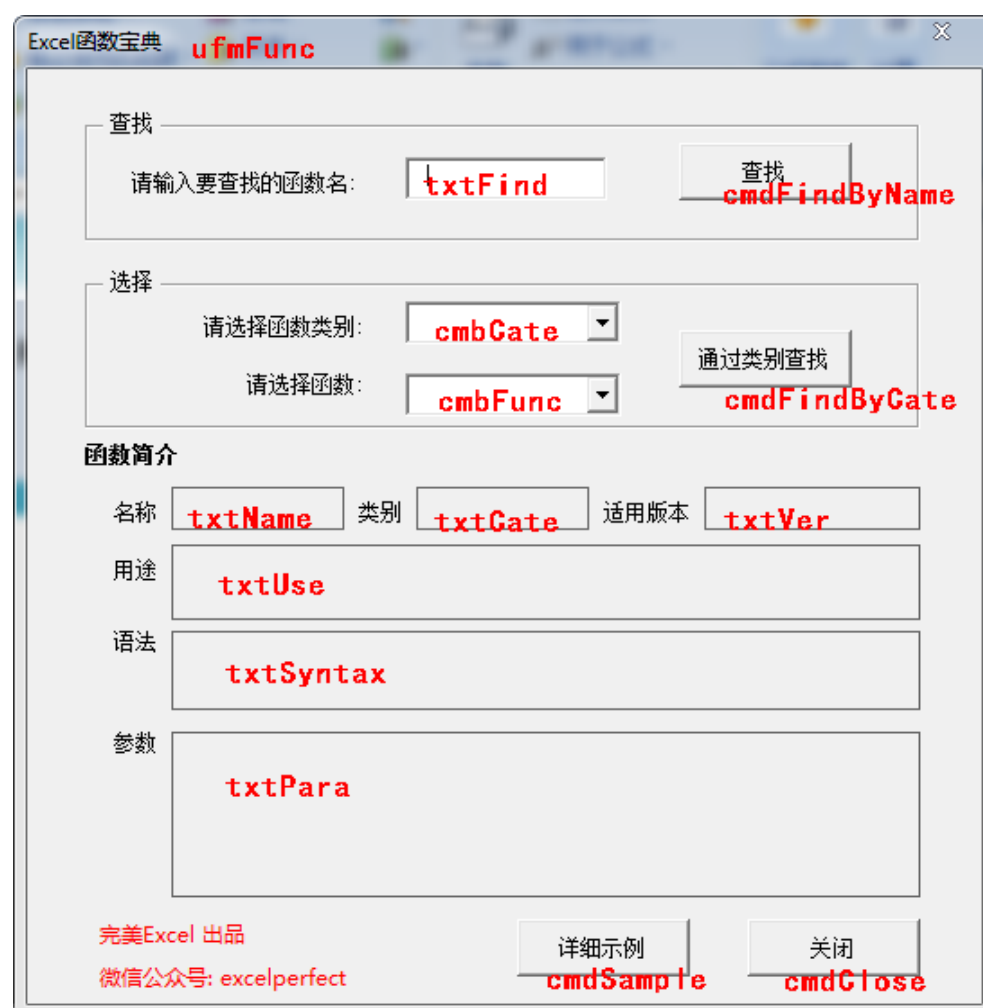


图 3.3



函数工作簿

函数工作簿包括函数的详细信息及应用示例，如图 3.4 所示。

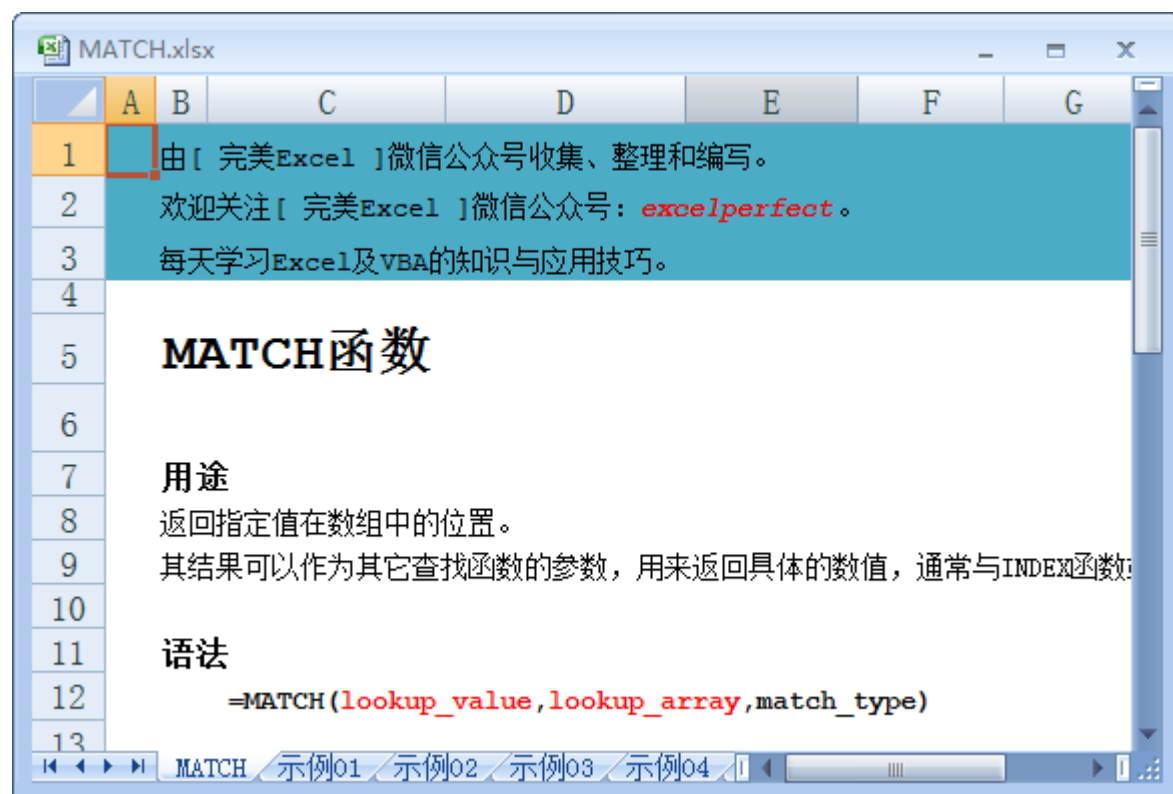


图 3.4

编写代码

工作表代码

在“主界面”工作表的 Activate 事件中，输入下面的代码：

```
Private Sub Worksheet_Activate()  
    [SearchFuncName] = "在这里输入你要查找的函数名"  
    [SearchFuncName].Select  
End Sub
```

每次激活该工作表时，代码在命名为“SearchFuncName”的单元格中输入提示信息，并选择该单元格。

```
Sub wsFindButtonClick()  
    ufmFunc.Show  
End Sub
```



单击工作表中的“查找”按钮时，调出用户窗体。

用户窗体代码

用户窗体初始化代码：

```
Private Sub UserForm_Initialize()  
    '在类别组合框中输入类别项  
    cmbCate.List = Application.Transpose([Cate])  
    '查找名称 SearchFuncName 单元格中函数的信息  
    FindData (Trim([SearchFuncName]))  
End Sub
```

“函数类别” 组合框代码：

```
Private Sub cmbCate_Change()  
    Select Case cmbCate.Text  
        Case "查找与引用"  
            cmbFunc.List = Application.Transpose([查找与引用])  
        Case "文本"  
            cmbFunc.List = Application.Transpose([文本])  
        Case "信息"  
            cmbFunc.List = Application.Transpose([信息])  
        ,  
        Case "财务"  
            cmbFunc.List = Application.Transpose([财务])  
        ,  
        Case "日期与时间"  
            cmbFunc.List = Application.Transpose([日期与时间])  
        ,  
        Case "数学与三角"  
            cmbFunc.List = Application.Transpose([数学与三角])  
        ,  
        Case "统计"  
            cmbFunc.List = Application.Transpose([统计])  
        ,  
        Case "数据库"  
            cmbFunc.List = Application.Transpose([数据库])  
        ,  
        Case "逻辑"  
            cmbFunc.List = Application.Transpose([逻辑])  
        ,  
        Case "工程"  
            cmbFunc.List = Application.Transpose([工程])  
        ,  
        Case "多维数据集"  
            cmbFunc.List = Application.Transpose([多维数据集])  
        ,  
        Case "用户定义"  
            cmbFunc.List = Application.Transpose([用户定义])  
        Case Else  
            cmbFunc.Clear  
    End Select  
  
    ClearData  
    txtFind.Text = ""  
End Sub
```

当“函数类别”组合框中的内容改变量，用相应的函数名填充“函数名称”组合框，同时清除函数信息文本框中的内容。

单击“查找”按钮的代码：

```
Private Sub cmdFindByName_Click()
```



```

[SearchFuncName] = txtFind.Text

FindData (txtFind.Text)

End Sub

```

单击“通过类别查找”按钮的代码：

```

Private Sub cmdFindByCate_Click()

    txtFind.Text = ""

    [SearchFuncName] = cmbFunc.Text

    FindData (cmbFunc.Text)

End Sub

```

查找数据代码：

```

Sub FindData(strFind As String)
    Dim rngName As Range

    If Len(Trim(strFind)) = 0 Then
        MsgBox "请输入要查找的函数名!"
        ClearData
        cmbCate.Clear
        cmbCate.List = Application.Transpose([Cate])
        cmbFunc.Clear
    Else
        Set rngName = [FuncName].Find(What:=strFind)
        If Not rngName Is Nothing Then
            With rngName
                cmbCate.Value = .Offset(0, 1).Value
                cmbFunc.Value = .Value
                txtName.Text = .Value
                txtCate.Text = .Offset(0, 1).Value
                txtVer.Text = .Offset(0, 2).Value
                txtUse.Text = .Offset(0, 3).Value
                txtSyntax.Text = .Offset(0, 4).Value
                txtPara.Text = .Offset(0, 5).Value
            End With
        Else
            MsgBox "没有找到这个函数, 请检查是否输入了正确的函数名!"
            ClearData
            cmbCate.Clear
            cmbCate.List = Application.Transpose([Cate])
            cmbFunc.Clear
        End If
    End If
End Sub

```

根据传递给子过程的值，在“函数库”工作表中查找到相应的函数，并将函数信息输入到用户窗体上相应的文本框中。

下面的子过程用于清除文本框中的数据：

```

Sub ClearData()

    txtName.Text = ""

```



```

txtCate.Text = ""
txtVer.Text = ""
txtUse.Text = ""
txtSyntax.Text = ""
txtPara.Text = ""
End Sub

```

“详细示例” 按钮代码:

```

Private Sub cmdSample_Click()
    OpenFunctionWB (cmbFunc.Text)
    Unload Me
End Sub

```

打开与函数名相同的函数工作簿，同时卸载用户窗体。

“关闭” 按钮代码:

```

Private Sub cmdClose_Click()
    Unload Me
End Sub

```

其他代码

OpenFunctionWB 过程用于打开指定的工作簿:

```

Sub OpenFunctionWB(strName As String)
    Dim strPath As String
    Dim strTest As String

    If bIsBookOpen(strName & ".xlsx") Then
        Workbooks(strName & ".xlsx").Activate
    Else
        strPath = ThisWorkbook.Path & Application.PathSeparator & _
            "函数示例" & Application.PathSeparator & strName & ".xlsx"

        strTest = ""
        On Error Resume Next
        strTest = Dir(strPath)
        On Error GoTo 0
        If strTest = "" Then
            MsgBox "没有找到这个函数相关的示例工作簿", vbOKOnly, "Excel函数宝典"
        Else
            Application.ScreenUpdating = False
            Workbooks.Open strPath, UpdateLinks:=3
            Application.Goto Sheets(1).Range("A1"), True
            Application.ScreenUpdating = True
        End If
    End If
End Sub

```



bIsBookOpen 函数用于判断工作簿是否已打开：

```
Function bIsBookOpen(ByRef szBookName As String) As Boolean
    On Error Resume Next
    bIsBookOpen = Not (Application.Workbooks(szBookName) Is Nothing)
End Function
```

结语

这个应用程序界面设计和架构的搭建基本完成，主要需要完善的是：

- 收集整理并补充 Excel 函数的应用示例。
- 美化界面并扩展功能。



本章内容 2018 年 2 月 24 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
Excel 实用工具 4：全面获取工作簿的使用信息

4. 全面获取工作簿的使用信息

引言：本章给出的工作簿中的 VBA 代码可以显示出 Excel 工作簿的最重要的度量，例如工作簿的工作表数、已使用的单元格数、公式单元格数、名称单元格数、图表数、表数、代码行数，等等。

在开发一个大且复杂的 Excel 电子表格模型时，应该密切关注该工作簿的大小及其组件：工作簿中有多少个工作表？已经使用了多少个单元格？有多少个公式单元格和常量单元格？有多少个返回值为错误的公式？有多少个定义了名称的区域？有多少个图表、表、数据透视表、VBA 代码行，等等？

这些度量会提供给你关于你的电子表格模型的大小和复杂度的概要信息，向你指出一些潜在改进的地方，帮助你优化性能，减小文件大小。

此时，一段能够自动计算这些最重要的工作簿度量的 VBA 代码是非常有用的。你可以在任何工作簿中使用这段代码，从而获取该工作簿的度量信息。

Excel 工作簿的度量信息

在消息框中将显示下列工作簿的度量信息：

- **工作簿大小：**以 MB 和 B 为单位的工作簿大小
- **工作表数量：**统计工作簿中的工作表数（工作表总数、隐藏的工作表数、深度隐藏的工作表数）
- **已使用的单元格数：**统计工作簿所有工作表已使用的单元格（单元格总数、可见单元格数和不可见单元格数）
- **空单元格数：**空单元格数量
- **常量单元格数：**统计包含常量（数字或文本）的单元格数
- **公式单元格数：**统计包含公式的单元格数（在消息框中，不仅显示公式单元格总数，而且分别显示返回数字、文本、逻辑值或错误的公式单元格数）



- **条件格式单元格数**：统计包含条件格式的单元格数
- **数据有效性单元格数**：统计包含数据有效性的单元格数
- **名称单元格区域数**：统计命名的单元格区域的数量（不包括表/列表对象的名称）
- **数据透视表数**：统计数据透视表数
- **图表数**：统计图表数量（仅统计嵌入到工作表中的图表）
- **其他对象数**：统计诸如形状或图像等对象
- **外部链接数**：统计对其他 Excel 工作簿的链接数（这个数量不是所有包含对其他工作簿的链接的单元格数，而是链接的不同工作簿的数量）
- **VBA 组件数**：统计 VBA 模块、类模块和包含 VBA 代码的工作表对象模块
- **VBA 代码行数**：统计所有 VBA 模块中的 VBA 代码行数

VBA 代码

示例工作簿中的 VBA 代码包含 1 个主过程（WorkbookStatistics）和 3 个辅助函数（CountVBAComponents、LineofCode 和 CountofTables）。通过遍历当前工作簿的所有工作表并利用 Range 对象的.Cells 属性和.SpecialCells 属性来发现大多数的度量信息，代码相当直观。

在示例工作簿中运行代码后的结果如下图 4.1 所示。

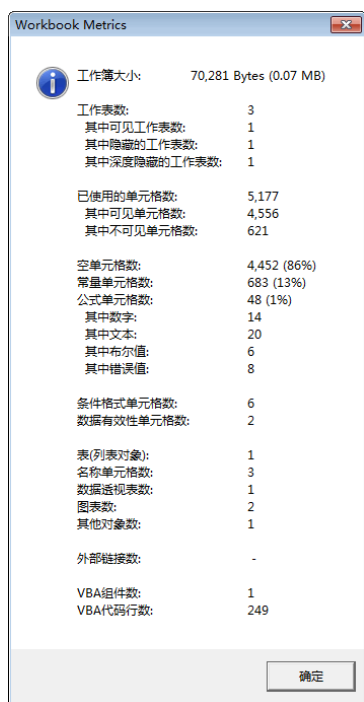


图 4.1



你可以在完美 Excel 公众号中发送消息：**获取工作簿信息** 来下载示例工作簿。

下载示例工作簿后，你可以将工作簿中的代码模块导入到自己的工作簿中，然后运行代码以获取工作簿信息。当然，你也可以进一步完善代码，以获取更多的信息。



本章内容 2018 年 3 月 14 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
Excel 实用工具 5：在 Excel 中创建专业的甘特图

5. 在 Excel 中创建专业的甘特图

这是一个绝对专业级别的甘特图制作工具，而且使用 Excel 来创建甘特图，功能丰富，易于使用，几乎用不着多花时间学习就能上手。

功能区外观如图 5.1 所示。

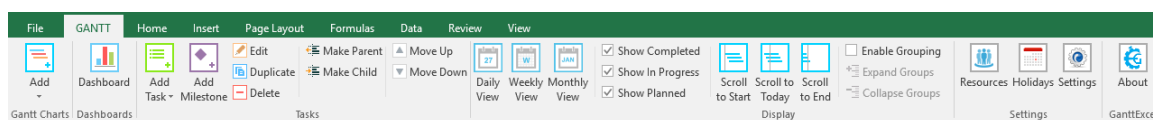


图 5.1

基于时间创建级联关系，以便于查看时间更改会影响到过程的其他部分和最终可交付成果。

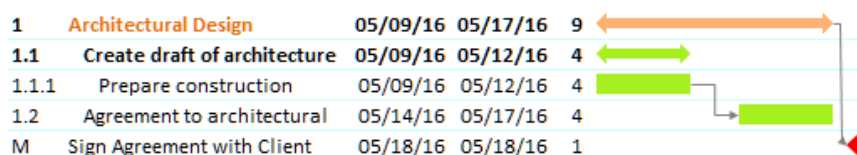


图 5.2

仪表板是轻松沟通关键项目详细信息的最佳方式。

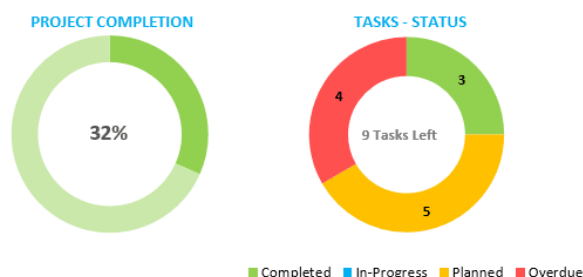


图 5.3



甘特图扩展为预算和费用模板，跟踪项目和任务的成本和支出。



图 5.4

此外，还有丰富的附加功能，让你大开眼界。

真是妙呀！使用 Excel 真的能制作出如此专业的实用工具!!!

耶！这也是一个绝好的学习研究示例!!!

你可以到下面的网站：

<https://ganttxl.com/>

下载免费版体验。

你可以在完美 Excel 公众号中发送消息：**甘特图**来下载工作簿模板。





本章内容 2018 年 3 月 20 日首发于
 [完美 Excel] 微信公众号 [excelperfect](#)
 原标题为
 Excel 实用工具 6：工作时间及薪水计算器

6. 工作时间及薪水计算器

本文介绍一个工作表模板，由 [trumpexcel.com](#) 开发，主要用于计算工作时间及薪水，如图 6.1 所示。

周工作时间表计算器

完美Excel科技

员工姓名: fanjy

部门: Excelperfect

部门经理: Exceler

| 年度 | 月份 | 日期 | 周末 |
|------|-------|----|-----------|
| 2015 | April | 16 | Sat & Sun |

☒ 周末按加班计算

| 上班时间 | 正常工作小时数 | 正常薪水 (每小时) | 加班薪水 (每小时) |
|------|---------|------------|------------|
| 9:00 | 6 | 30 | 50 |

| 星期 | 日期 | 上班 | 下班 | 中断小时 | 正常工作小时 | 加班小时 | 总薪水 |
|-----|----|-----|----|------|--------|------|-----|
| Sun | 16 | | | | | | |
| Mon | 17 | | | | | | |
| Tue | 18 | | | | | | |
| Wed | 19 | | | | | | |
| Thu | 20 | | | | | | |
| Fri | 21 | | | | | | |
| Sat | 22 | | | | | | |
| | | 周汇总 | | | | 0.0 | 0.0 |
| | | | | | | 0.0 | 0.0 |

图 6.1（视频截图，完整视频详见[完美 Excel](#) 微信公众号）

从演示中可以看出，这是一个动态的时间记录表模板。在选择年度和月份及日期后，下方的周记录表会自动调整，并根据周末的时间将其所在行自动添加背景色。在输入上下班及中断小时数后，会自动计算正常工作小时、加班小时及总薪水。



如果是周末，则根据是否选取了“周末按加班计算”复选框来计算相应的薪水。

除提供计算周工作时间的模板外，还提供了计算双周工作时间、月工作时间的模板。

要注意的是，除选择和输入部分外，不要破坏工作表结构。

表头部分。可以修改为自己公司名称、员工姓名、部门、部门经理，如图 6.2 所示。

周工作时间表计算器

完美Excel科技

员工姓名:

fanjy

部门:

Excelperfect

部门经理:

Exceler

图 6.2

选择年、月、日，以及设置周末，如图 6.3 所示。时间修改后，下图 5 中的记录表的日期相应自动调整。

| 年度 | 月份 | 日期 | 周末 |
|------|-------|----|-----------|
| 2018 | March | 15 | Sat & Sun |

从下拉列表中选择年份、月份、日期、周末

Sat & Sun

Sat & Sun

Sun & Mon

Mon & Tue

Tue & Wed

Wed & Thu

Thu & Fri

Fri & Sat

Mon Only

☒ 周末按加班计算

是否按加班计算周末工作时间

图 6.3

设置上班时间、每日正常工作小时数、正常工作时间的每小时薪水和加班的每小时薪水，如图 6.4 所示。

| 上班时间 | 正常工作小时数 | 正常薪水 (每小时) | 加班薪水 (每小时) |
|------|---------|---------------|---------------|
| 9:00 | 6 | 30 | 50 |

图 6.4



记录区。前两列根据图 3 选择的日期时间自动调整，中间的三列为输入区，后面的三列自动计算，最后一行自动汇总时间。如图 6.5 所示。

| 星期 | 日期 | 上班 | 下班 | 中断小时 | 正常工作小时 | 加班小时 | 总薪水 | |
|-----|----|-------|-------|------|--------|------|-------|-------|
| Thu | 15 | 9:00 | 18:00 | 2.0 | 4.0 | 3.0 | 270.0 | |
| Fri | 16 | 9:00 | 18:00 | 3.0 | 3.0 | 3.0 | 240.0 | |
| Sat | 17 | 10:00 | 17:00 | 1.0 | 4.0 | 2.0 | 300.0 | |
| Sun | 18 | | | | | | | |
| Mon | 19 | | | | | | | |
| Tue | 20 | | | | | | | |
| Wed | 21 | | | | | | | |
| | | 周汇总 | | | 6.0 | 11.0 | 8.0 | 810.0 |

图 6.5

这个工作表模板综合应用了 Excel 公式、数据有效性、名称、条件格式、工作表控件等技术，有兴趣的朋友可以在完美 Excel 公众号中发送消息：**工时表计算器**来下载工作簿模板。

也可以到下面的网址下载：

<https://trumpexcel.com/excel-timesheet-calculator-template/>





本章内容 2018 年 4 月 4 日首发于
[完美 Excel] 微信公众号 [excelperfect](#)
原标题为
[Excel 实用工具 7: Excel 日历模板](#)

7. Excel 日历模板

本章介绍的这两款 Excel 模板，由 [trumpexcel.com](#) 开发，使用 Excel 公式创建动态的日历模板，基于所选择的月份或年度动态更新。模板容易更新且在需要时能够直接打印出来查看。为便于理解，我对模板进行了适当的修改。

月日历模板

下图 7.1 为月日历模板的演示：

| | A | B | C | D | E | F | G | H | I | J |
|----|----|----|---|---|---|---|---|---|---|---|
| 1 | 年度 | 20 | | | | | | | | |
| 2 | 月份 | 6月 | | | | | | | | |
| 3 | | | | | | | | | | |
| 4 | | | | | | | | | | |
| 5 | | | | | | | | | | |
| 6 | | | | | | | | | | |
| 7 | | | | | | | | | | |
| 8 | | | | | | | | | | |
| 9 | | | | | | | | | | |
| 10 | | | | | | | | | | |
| 11 | | | | | | | | | | |

2017年6月

| 星期一 | 星期二 | 星期三 | 星期四 | 星期五 | 星期六 | 星期日 |
|-----|-----|-----|-----|-----|-----|-----|
| | | | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | | |
| | | | | | | |

图 7.1（视频截图，完整视频详见[完美 Excel 微信公众号](#)）

在单元格 B1 中输入年份，在单元格 B2 中使用下拉列表选择月份，月日历会自动更新。由于我们指定了假期，因此对应于假期的日期会突出显示。

年日历模板

下图 7.2 为年日历模板演示：



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
|----|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | + 2017 | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | | | | | | | | | | |

图 7.2（视频截图，完整视频详见完美 Excel 微信公众号）

与月日历模板相同，在单元格 B1 中输入年份，日历会自动更新，并且会突出显示对应于指定假期的日期。

日历模板制作技术

优秀的作品不仅方便我们拿来就用，也是学习的好示例。下面，简单地介绍一下这两款日历模板是如何制作的。

- 在“假期”工作表中，在列 B 中应输入假期的日期，并且是正确的日期格式。
- 工作簿中有一个名为“Backend”的隐藏的工作表，存储着月份及对应的数字。
- 模板中使用了由 DATE 函数、WEEKDAY 函数、DAY 函数、ISNUMBER 函数、COLUMNS 函数和 VLOOKUP 函数组成的公式。
- 使用了条件格式来突出显示假期的日期。
- 使用数据有效性来创建选取月份的下拉列表。
- 使用了自定义数字格式来显示日历月名。

这个工作表模板综合应用了 Excel 公式、数据有效性、条件格式件等技术，以及单元格格式设置技巧，有兴趣的朋友可以在完美 Excel 公众号中发送消息：



日历模板

下载工作簿模板。

也可以到下面的网址下载：

<https://trumpexcel.com/calendar-template/>





本章内容 2018 年 4 月 11 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 实用工具 8：一款实用的自定义功能区加载项](#)

8. 一款实用的自定义功能区加载项

这款加载项适用于 Excel 2010 及以上版本，可用于直接在 Excel 中添加、编辑和验证功能区 XML 代码。由一个窗体组成，在窗体中可以编辑 customUI 和 customUI14 XML 文本。其界面如图 8.1 所示。

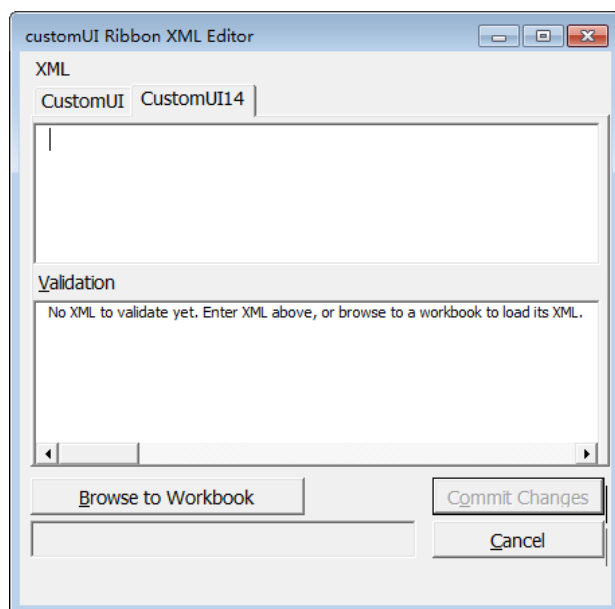


图 8.1

功能概述

- 在独立的窗体中装载、修改和验证 Ribbon XML
- 编辑工作簿和加载项功能区
- 能用于仅验证功能区 XML 而无须装载文件
- 处理 customUI 和 customUI14 XML
- 显示验证错误消息并高亮显示无效的 XML
- 适用于 Excel 工作簿和已安装的加载项



- 可以取消某项操作并恢复之前的所有 XML
- 目标文件在进行任何更改之前进行备份
- 窗体和字体都可调整大小

下载并安装

- 可在本文后面给出的资源链接或公众号回复消息，下载压缩文件。文件中包含着自定义功能区编辑器和 2 个 xsd 文件。
- 将这三个文件解压到你的 Addins 文件夹。这两个 XSD 文件是必需的，用于 XML 验证。
- 在 Excel 中安装加载项。

使用

- 安装这个加载项后，Excel 功能区“开发工具”选项卡中将会出现一个名为“ysb customUI editor”的组，如图 8.2 所示。

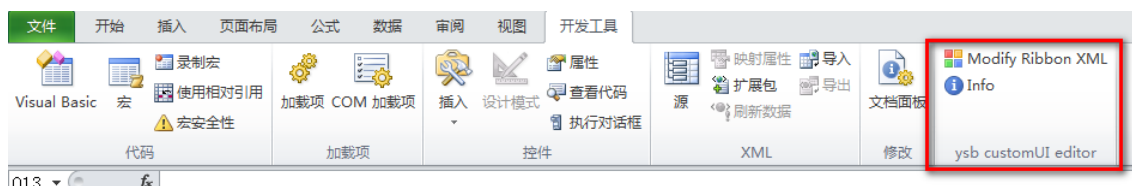


图 8.2

- 单击“Modify Ribbon XML”打开图 8.1 所示的窗体。
- 此时，可以导航到工作簿或加载项来处理 Ribbon XML，或者在 customUI 或 customUI14 选项卡下简单地输入一些 XML。
- 如果选择了一个已安装的加载项，那么可以确认编辑器在导入其 XML 之前关闭了该加载项。
- 可以在编辑区中修改 XML 从而修改工作簿中的功能区。
- 如果输入了无效的 XML，那么在验证区域将列出错误。
- “Commit Changes”按钮仅适用于：浏览到工作簿或加载项，并作出了修改，并且已验证为有效的 XML，按下该按钮后，新的 XML 将应用于该工作簿或加载项。
- 一旦修改了工作簿，系统就会询问是否打开它。已安装的加载项会自动重新安装。



XML 验证

当输入或粘贴 XML 后，会自动验证，所有错误都会列于验证区。上文已提到，验证必需将 customUI.xsd 和 customUI14.xsd 文件与加载项放置在相同的文件夹下。

双击验证区中列出的错误行列号，即可选择 XML 中的错误代码。

安全

该加载项在浏览打开文件时的第一件事就是将其先备份一份文件到相同的文件夹中，因此，如果编辑的文件出现意外崩溃，可以使用备份文件恢复。

其他

正如本文开头所说，这款加载项适用于编辑、添加和验证功能区代码，即已经包含有 XML 的工作簿。如果要从头创建自定义功能区，则应选择其他工具。

资源下载

这个加载项是完全开放的，其中的代码均公开，有兴趣的朋友可以在[完美 Excel](#) 公众号中发送消息：

自定义功能区加载项

下载这个加载项。

也可以到下面的网址下载：

<http://yoursumbuddy.com/ribbon-customui-xml-editor/>





本章内容 2018 年 5 月 9 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 实用工具 9：一款定制右键菜单的加载项](#)

9. 一款定制右键菜单的加载项

在 [yoursumbuddy.com](#) 中，发布了一款可以让你修改右键菜单的加载项 **MenuRighter**。你几乎可以添加任何在其他右键菜单或者 Excel 2003 经典菜单中发现的控件，例如“编辑”菜单、“冻结窗格”按钮，甚至是你自己的加载项菜单。你还可以重新排列右键菜单、删除按钮或者从子菜单中上移按钮，例如，可以复制“按选定单元格的值筛选”到单元格上下文菜单的顶层。

MenuRighter 让你可以通过右键菜单，例如单元格、绘图对象和数据透视表，以及 Excel 2003 顶层菜单如“文件”和“编辑”、以及你创建的任何自定义命令栏来搜索。然后，你可以将它们插入到几乎任何右键菜单中。

工作原理

从 Excel 2007 开始，功能区替换了旧的 Excel 2003 文件、编辑等菜单结构，但 2003 的菜单和控件仍然存在于后台中。在现在的 Excel 版本的右键菜单仍然与 2003 大致相同，当然添加了一些新的控件。这一切意味着我们可以编程来使用菜单系统对右键菜单进行修改，这就是为什么 MenuRighter 加载项能够工作。

下载并安装

- 可在本文后面给出的资源链接或公众号回复消息，下载压缩文件。
- 将文件解压到你的 Addins 文件夹。
- 在 Excel 中安装加载项。

使用

安装这个加载项后，Excel 功能区“开发工具”选项卡中将会出现一个名为“MenuRighter”的组，如图 9.1 所示。





图 9.1

单击“MenuRighter”打开图 9.2 所示的窗体。

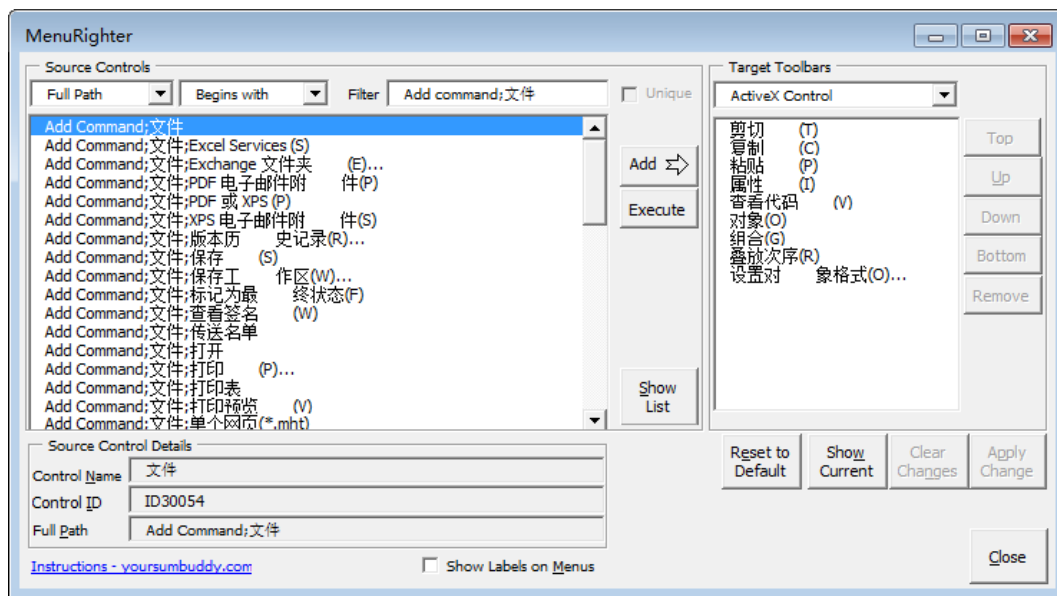


图 9.2

你可以通过拖拉来调整这个窗体的大小。

正如图所示，源控件位于左侧，可以选择命令并添加到右键菜单。目标控件位于右侧，可以选择菜单来添加，也可以删除和重新排列已有的菜单按钮。

首先，看看源控件，如图 9.3 所示。



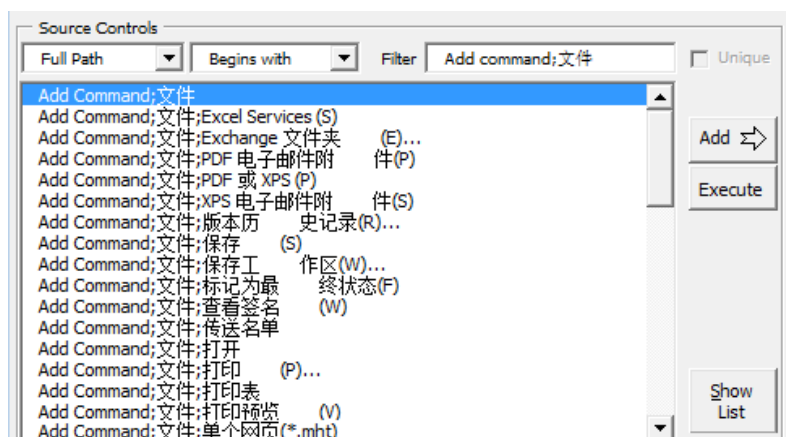


图 9.3

顶部的两个下拉列表框和一个搜索框，可以筛选要添加的菜单项。开始筛选出 Excel 2003 的添加命令工具栏和文件菜单，这个列表框包含了所有在该菜单中发现的控件，例如打开、关闭等。

假设你想将冻结窗格按钮添加到行菜单中，因为搜索框允许使用通配符，只需在搜索框中输入“*冻结”。

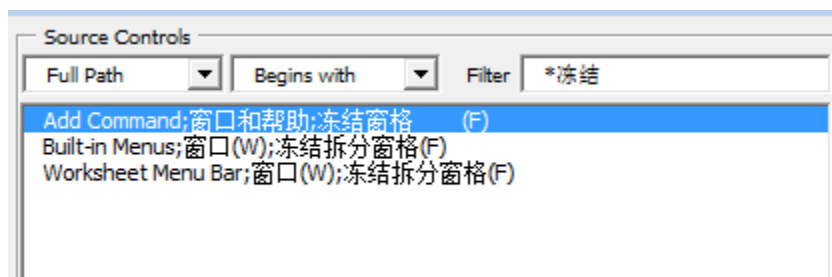


图 9.4

在源控件部分的右边有 4 个控件：

- **Unique** 复选框仅应用于在第一个下拉列表框中选择了 **Control Name** 时。默认情况下，意味着仅显示每个控件的一个实例
- **Add** 按钮将选定的控件添加到目标工具栏。
- **Execute** 按钮试图执行在列表中当前选择的命令。如果它不适用于当前的 Excel 情形，将被禁用，即使启用它也不会执行任何操作。
- **Show List** 命令在新的工作表中显示当前已筛选的可用命令完整的列表，这只是搜索要添加的菜单的一种不同方式。如果使用的是 Excel 2013 或更高版本，则应在查看后关闭此工作表。



使用分号分隔符作为菜单层级之间的分隔，例如“Add Command;文件;关闭”，类似于 Windows 中的路径分隔符。之所以选择分号，是因为它不在菜单或控件名中使用。你可以避免在使用通配符的情况下使用它们，但是如果输入完整的路径，你需要包括它们。

目标控件部分有一个下拉列表，在其中可以修改快捷菜单（工具栏）的名称，例如单元格、行和数据透视表上下文菜单。下面是一个列表框，显示所选择的快捷菜单的所有控件。

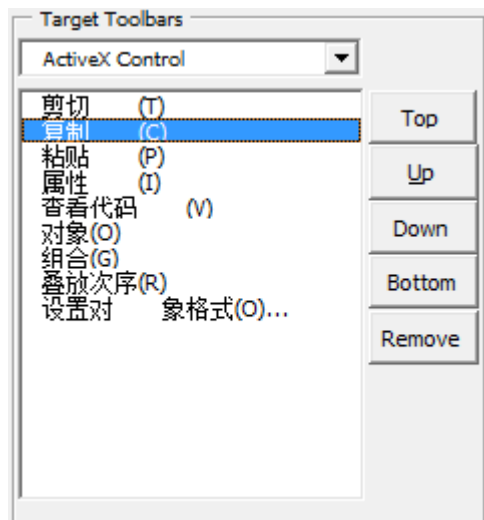


图 9.5

使用 Top、Bottom、Up 和 Down 来重新设置当前所选定的控件的位置。Delete 按钮删除当前选择的控件，双击也可以删除。

Reset to Default 按钮重置当前所选择的目标工具栏为 Excel 默认工具栏，这意味着对所选快捷菜单所做的任何更改都将被删除。

Show Current 按钮弹出当前所选择的右键菜单配置，在应用之前不会显示你已经做出的更改。

Clear Changes 按钮清除你对列表框所做的任何尚未应用的更改。

Apply Changes 按钮应用你在列表框中所做的更改。

单击这些控件中的任一个都会弹出实际的目标菜单，你可以立即使用它！或以后使用。

Show Labels On Menus 复选框，使用该菜单的名称和 ID 为每个右键菜单添加一个控件。



更多的使用信息，详见接下来的资源下载链接。

资源下载

这个加载项是完全开放的，其中的代码均公开，有兴趣的朋友可以在完美 Excel 公众号中发送消息：

自定义右键菜单

下载这个加载项，然后细细研究，从中学习到开发的方法和技巧。

也可以到下面的网址下载：

<http://yoursumbuddy.com/menurighter/>





本章内容 2018 年 5 月 17 日首发于
[完美 Excel] 微信公众号 [excelperfect](#)
原标题为
[Excel 实用工具 10: Excel 工作任务清单模板](#)

10. Excel 工作任务清单模板

本章介绍的这款 Excel 模板，仍然是 [trumpexcel.com](#) 开发的，使用 Excel 技术制作工作任务清单模板，可以很方便地列举待办事项清单，记录并跟踪完成情况。为便于理解，我对模板进行了适当的修改。

清单模板使用演示

在下图 10.1 所示的区域中输入待办工作清单。

| 待办清单 | | | | |
|------|----------|-----------|-----|------|
| | | | | 0.0% |
| 编号 | 日期 | 工作任务 | 优先级 | 状态 |
| 1 | 2018/4/8 | 编写施工技术方案 | 紧急 | |
| 2 | 2018/4/8 | 参加办公会 | 紧急 | |
| 3 | 2018/4/8 | 编写行业标准 | 较紧急 | |
| 4 | 2018/4/8 | 准备明天的会议材料 | 紧急 | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |

图 10.1

在图 10.2 所示的区域中给每项工作赋予优先级，包括紧急、较紧急和一般。

| 待办清单 | | | | |
|------|----------|-----------|-----|------|
| | | | | 0.0% |
| 编号 | 日期 | 工作任务 | 优先级 | 状态 |
| 1 | 2018/4/8 | 编写施工技术方案 | 紧急 | |
| 2 | 2018/4/8 | 参加办公会 | 紧急 | |
| 3 | 2018/4/8 | 编写行业标准 | 较紧急 | |
| 4 | 2018/4/8 | 准备明天的会议材料 | 一般 | |
| 5 | | | | |
| 6 | | | | |

图 10.2



在状态栏中选择完成情况，如下图 10.3 所示。如果已经完成，工作任务上会添加删除线，该行会添加淡绿色，上方的进度条会显示相应的完成百分比。



图 10.3（视频截图，完整视频详见[完美 Excel 微信公众号](#)）

清单模板制作技术

优秀的作品不仅方便我们拿来就用，也是学习的好示例。

下面，简单地介绍一下这款清单模板是如何制作的。

在工作表中隐藏了列 G 至列 I，其中的内容如图 10.4 所示。

| | G | H | I | J |
|----|-----------|-----|-------|---|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | 状态 | 优先级 | 权重 | |
| 5 | 已完成 | 紧急 | 5 | |
| 6 | 处理中 | 较紧急 | 3 | |
| 7 | 未开始 | 一般 | 1 | |
| 8 | | | | |
| 9 | 工作任务总数 | | 4 | |
| 10 | 已完成任务占的分数 | | 15 | |
| 11 | 总分 | | 18 | |
| 12 | 完成百分比 | | 83.3% | |
| 13 | | | | |
| 14 | | | | |
| 15 | | | | |
| 16 | | | | |
| 17 | | | | |
| 18 | 已完成任务占的分数 | | 1 | |
| 19 | 总分 | | 10 | |
| 20 | 完成百分比 | | 10.0% | |
| 21 | | | | |

图 10.4



在单元格区域 G5:G7 中放置着工作状态列表，H5:H7 中放置着工作优先级列表，用于在列 D 和列 E 中创建下拉列表。这两列中的下拉列表是使用 Excel 的数据有效性功能创建的。

从图 10.4 中看出，我们赋予了不同优先级工作的权重，放置在单元格区域 I5:I7 中，你可以根据实际情况修改。

在单元格 H9 中计算填写的工作任务总数。

单元格 H10 中计算已完成任务占的分数。例如上面的 4 项任务，已经完成了 1 项紧急任务，则分数为 $1*5=5$ 分。

单元格 H11 计算总分，总分为所有任务按权重计算的分数之和。例如 4 项任务，3 项为紧急任务，1 项为较紧急任务，则总分为 $3*5+1*3=18$ 分。

单元格 H12 计算完成百分比，即完成任务所占分数除以总分。例如上面完成了 1 项紧急任务，则完成百分比为 $5/19=27.78\%$ 。

然后，根据计算的结果，使用条形图来绘制进度图，根据完成百分比而动态显示进度情况。

单元格 H9 中的公式：

=COUNTIF(\$C\$5:\$C\$29,"<>")

单元格 H10 中的公式：

=COUNTIFS(\$D\$5:\$D\$29,\$H\$5,\$E\$5:\$E\$29,\$G\$5)*\$I\$5+COUNTIFS(\$D\$5:\$D\$29,\$H\$6,\$E\$5:\$E\$29,\$G\$5)*\$I\$6+COUNTIFS(\$D\$5:\$D\$29,\$H\$7,\$E\$5:\$E\$29,\$G\$5)*\$I\$7

单元格 H11 中的公式：

=COUNTIF(\$D\$5:\$D\$29,\$H\$5)*\$I\$5+COUNTIF(\$D\$5:\$D\$29,\$H\$6)*\$I\$6+COUNTIF(\$D\$5:\$D\$29,\$H\$7)*\$I\$7

单元格 H12 中的公式：

=IFERROR(H10/H11,0)

在工作清单列表区域 A5:E29 中，使用了条件格式，根据工作完成状态来添加行背景色和工作任务添加删除线。



清单模板下载

这款工作表模板也是综合应用了 Excel 公式、数据有效性、条件格式、Excel 图表等技术，以及单元格格式美化技巧，有兴趣的朋友可以在完美 Excel 公众号中发送消息：

工作清单模板

下载工作簿模板。

也可以到下面的网址下载：

<https://trumpexcel.com/excel-to-do-list-template-download/>



本章内容 2018 年 5 月 27 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 实用工具 11：任务矩阵模板](#)

11. 任务矩阵模板

本来想着自己有时间再开发一套这样的模板的，想不到在 [trumpexcel.com](#) 中已经有了。这个 Excel 模板帮助你管理每天/每周的工作任务，非常容易使用。

在这里，将该模板中文化，你可以根据本文后面的信息下载该模板。同时，将运用原理与大家分享。在学习的过程中，发现了几个小 Bug，也一同修复。

任务矩阵原理

你一定听说过斯蒂芬·柯维的《高效能人士的 7 个习惯》一书，可能很多人都看过，一部分人甚至研读过并按照书上的方法来培养工作和生活中的好习惯。

在书中，斯蒂芬讲述了时间管理技术，建议将任务分成 4 类：

- 重要且紧急的
- 重要但不紧急的
- 不重要但紧急的
- 既不重要也不紧急的

我们要将精力主要集中在第 1 类，接着是第 2 类，最后才是第 3 类和第 4 类。





图 11.1

本章介绍的 Excel 任务矩阵模板就是基于上面的时间管理原理。

任务矩阵模板的使用

下图 11.2 演示了如何使用任务矩阵模板。



图 11.2（视频截图，完整视频详见[完美 Excel 微信公众号](#)）

单击工作表中的“添加新任务”，会弹出如图 11.3 所示的对话框。





图 11.3

在对话框中输入任务名称、是否重要、是否紧急，单击“添加任务”按钮即可将任务添加到工作表中相应的目录下，如图 11.4 所示。注意，除任务状态外，其他的 3 个必需填写或选择。

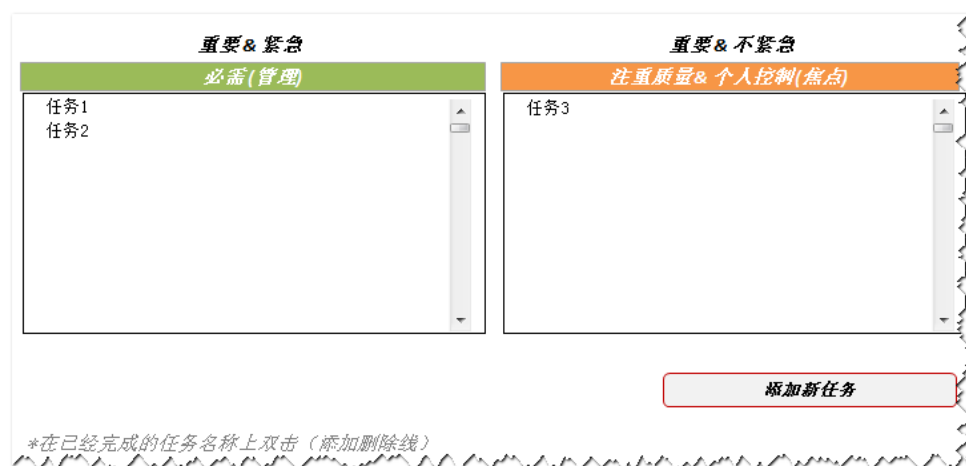


图 11.4

如果某项任务完成，只需在该任务名称上双击鼠标，Excel 会自动在其上添加删除线，如图 11.5 所示。任务 2 已完成，在其上双击鼠标，添加删除线。

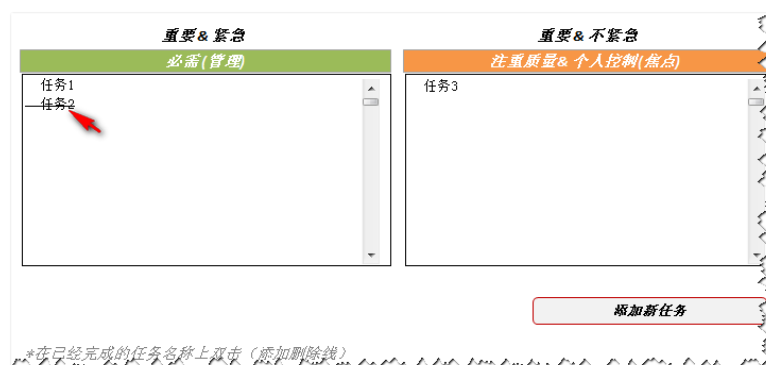


图 11.5



单击“移除已完成的任务”，会将清单中已添加删除线的任务清除，只留下还没有完成的任务。

下载信息

这是一款优秀的作品，将书中的原理程序化，直接用于实践。这样的作品不仅方便我们使用，提高效率，也是学习的好示例。

这款工作表模板是综合应用了 Excel 公式、数据有效性、条件格式、VBA 等技术，以及单元格格式美化技巧，有兴趣的朋友可以在[完美 Excel](#) 公众号中发送消息：

任务矩阵模板

下载该模板的中文版（修正了原模板中的一些 Bug）。

也可以到下面的网址下载英文原版：

<https://trumpexcel.com/free-excel-templates-excel-list/>



本章内容 2018 年 6 月 2 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 实用工具 12：随机分组模板](#)

12. 随机分组模板

本章继续分享 [trumpexcel.com](#) 中的优秀 Excel 模板。

我们经常会碰到这样的需求，需要将一群人随机分成几组。本文介绍的模板，可以帮助你容易地创建随机分组。你需要输入人员列表以及要创建的分组数，剩下的就是让 Excel 为你自动分配了，如下图 12.1 所示。

| | A | D | E | F | G | H | I | J | K | L | M |
|----|-----|---|-----|---|-----|-----|-----|-----|-----|---|---|
| 1 | 姓名 | | 分组数 | | 1 | 2 | 3 | 4 | 5 | | |
| 2 | 韦小宝 | | 7 | | 郭靖 | 小龙女 | 杨过 | 赵敏 | 袁承志 | | |
| 3 | 陈近南 | | | | 韦小宝 | 石破天 | 风清扬 | 陈家洛 | 张无忌 | | |
| 4 | 令狐冲 | | | | 段誉 | 黄蓉 | 袁紫衣 | 胡斐 | 令狐冲 | | |
| 5 | 风清扬 | | | | 陈近南 | | | | | | |
| 6 | 杨过 | | | | | | | | | | |
| 7 | 小龙女 | | | | | | | | | | |
| 8 | 张无忌 | | | | | | | | | | |
| 9 | 赵敏 | | | | | | | | | | |
| 10 | 郭靖 | | | | | | | | | | |
| 11 | 黄蓉 | | | | | | | | | | |
| 12 | 段誉 | | | | | | | | | | |
| 13 | 陈家洛 | | | | | | | | | | |
| 14 | 袁承志 | | | | | | | | | | |
| 15 | 胡斐 | | | | | | | | | | |
| 16 | 袁紫衣 | | | | | | | | | | |
| 17 | 石破天 | | | | | | | | | | |
| 18 | | | | | | | | | | | |

图 12.1（视频截图，完整视频详见[完美 Excel](#) 微信公众号）

在单元格区域 A2:A17 中是人员姓名，你可以在末尾添加更多的人名。在单元格 E2 中是想要创建的分组数，其于此分组数，在列 G 至列 P 中创建分组。本模板支持最多 10 个分组。

在单元格 E2 中输入想要创建的分组数，单击“创建随机分组”，就可在右侧的单元格中显示各组。

工作原理

工作表中使用了辅助列，即列 B 和列 C。将区域 A1:C17 转换成表，Excel 自动将



其命名为 Table1。这样，就可以自动调整表中的数据而保持结构不变，即可以随意增加或删除人员姓名。

在列 B 中的公式为：

=RANDBETWEEN(1,COUNTA([姓名])) + ROW()/100

其中，COUNTA 函数统计列 A 中的人数，RANDBETWEEN 函数生成在 1 和总人数之间的随机数。由于 RANDBETWEEN 函数可能会产生重复数，因此添加 ROW()/100 来使随机数唯一，如图 12.2 所示。

| | | | | | | | | |
|----|-----|-------|----|--|-----|---|-----|-----|
| B2 | | fx | | =RANDBETWEEN(1,COUNTA([姓名])) + ROW()/100 | | | | |
| | A | B | C | D | E | F | G | H |
| 1 | 姓名 | 唯一值 | 排序 | | 分组数 | | 1 | 2 |
| 2 | 韦小宝 | 6.02 | 15 | | 5 | | 袁紫衣 | 令狐冲 |
| 3 | 陈近南 | 15.03 | 3 | | | | 小龙女 | 黄蓉 |
| 4 | 令狐冲 | 12.04 | 7 | | | | 风清扬 | 段誉 |
| 5 | 风清扬 | 15.05 | 2 | | | | 韦小宝 | |
| 6 | 杨过 | 6.06 | 14 | | | | | |
| 7 | 小龙女 | 13.07 | 6 | | | | | |
| 8 | 张无忌 | 11.08 | 10 | | | | | |
| 9 | 赵敏 | 11.09 | 9 | | | | | |
| 10 | 郭靖 | 11.1 | 8 | | | | | |
| 11 | 黄蓉 | 4.11 | 16 | | | | | |
| 12 | 段誉 | 14.12 | 4 | | | | | |
| 13 | 陈家洛 | 10.13 | 11 | | | | | |
| 14 | 袁承志 | 13.14 | 5 | | | | | |
| 15 | 胡斐 | 6.15 | 13 | | | | | |
| 16 | 袁紫衣 | 9.16 | 12 | | | | | |
| 17 | 石破天 | 16.17 | 1 | | | | | |

创建随机分组

图 12.2

在单元格 C2 中的公式为：

=RANK([@唯一值],[唯一值])

给列 B 中的值进行排序。由于列 B 中的所有值都是唯一的，因此该公式生成的列 C 中的唯一整数列表是从 1 至人员数之间的整数值。

| C2 | | fx | | =RANK([@唯一值],[唯一值]) | | | | |
|----|-----|-------|----|---------------------|-----|---|-----|-----|
| | A | B | C | D | E | F | G | H |
| 1 | 姓名 | 唯一值 | 排序 | | 分组数 | | 1 | 2 |
| 2 | 韦小宝 | 6.02 | 15 | | 5 | | 袁紫衣 | 令狐冲 |
| 3 | 陈近南 | 15.03 | 3 | | | | 小龙女 | 黄蓉 |
| 4 | 令狐冲 | 12.04 | 7 | | | | 风清扬 | 段誉 |
| 5 | 风清扬 | 15.05 | 2 | | | | 韦小宝 | |
| 6 | 杨过 | 6.06 | 14 | | | | | |
| 7 | 小龙女 | 13.07 | 6 | | | | | |
| 8 | 张无忌 | 11.08 | 10 | | | | | |
| 9 | 赵敏 | 11.09 | 9 | | | | | |
| 10 | 郭靖 | 11.1 | 8 | | | | | |

创建随机分组

图 12.3



在单元格 G1 中的公式为：

=IF(COLUMNS(\$G\$1:G1)>\$E\$2,"",COLUMNS(\$G\$1:G1))

将公式拖至单元格 P1。公式返回当前列与列 G 间隔的列数值，因此单元格 G1 值是 1，H1 是 2，依此类推。数值不会超过单元格 E2 中的值，超过的单元格中返回空。

| G1 | | | =IF(COLUMNS(\$G\$1:G1)>\$E\$2,"",COLUMNS(\$G\$1:G1)) | | |
|----|-----|-------|--|---|-----|
| | A | B | C | D | E |
| 1 | 姓名 | 唯一值 | 排序 | | 分组数 |
| 2 | 韦小宝 | 6.02 | 15 | | 5 |
| 3 | 陈近南 | 15.03 | 3 | | |
| 4 | 令狐冲 | 12.04 | 7 | | |
| 5 | 风清扬 | 15.05 | 2 | | |
| 6 | 杨过 | 6.06 | 14 | | |
| 7 | 小龙女 | 13.07 | 6 | | |
| 8 | 张无忌 | 11.08 | 10 | | |

图 12.4

在单元格 G2 中的公式为：

**=IFERROR(IF(G\$1<>"",INDEX(Table1[姓 名],INDEX(Table1[排
序],G\$1+\$E\$2*(ROWS(\$F\$2:F2)-1))),""),"")**

将其拖拉至 G2:P17，根据生成的随机数提取列 A 中的值。

由于 RANDBETWEEN 函数是易失的，因此每当工作表中有变化时都会自动刷新，这可能不是我们想看到的，因此，我们将计算设置为“手动重算”，如图 12.5 所示。



图 12.5



现在，按 F9 键才会刷新工作簿，随机变化分组名。我们可以在工作表中添加按钮并指定 VBA 代码，通过单击来刷新工作表。

VBA 代码很简单：

```
Sub refresh()  
    Worksheets("Team Generator").Calculate  
End Sub
```

下载信息

这款模板并不复杂，只是运用了几个 Excel 公式以及单元格格式美化技巧，但是能够解决实际问题。有兴趣的朋友可以在[完美 Excel](#) 公众号中发送消息：

随机分组模板

下载该模板的中文版。

也可以到下面的网址下载英文原版：

<https://trumpexcel.com/random-group-generator-template/>



本章内容 2018 年 6 月 10 日首发于
[完美 Excel]微信公众号 [excelperfect](#)
原标题为
[Excel 实用工具 13: 每日事项清单——与日历结合](#)

13. 每日事项清单——与日历结合

下面继续分享 [trumpexcel.com](#) 中的优秀 Excel 模板，本章结合实际稍作修改。

在 [10: Excel 工作任务清单模板](#) 中，我们分享了一款模板，可以方便地列举待办事项清单，记录并跟踪完成情况。本章将事项清单与日历结合，可以直观地输入每日事项清单，查看每月事项清单。

如图 13.1 所示，单击年和月输入框左右两侧的箭头可以调整年和月。



图 13.1（视频截图，完整视频详见[完美 Excel](#) 微信公众号）

在所选日期所在的单元格中双击，将打开输入事项框，可以添加当日待办事项。添加待办事项后，所在单元格中的日期显示为蓝色，添加的待办事项被记录在相应的工作表中，如图 13.2 所示。



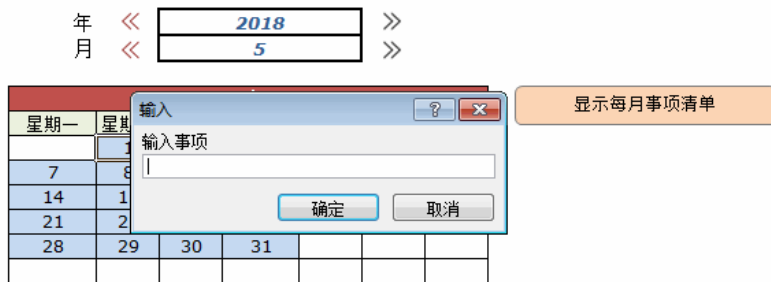


图 13.2（视频截图，完整视频详见完美 Excel 微信公众号）

可以选取日期后输入多个事项，VBA 代码中已限定为 1000 项任务。

要获得当月所有的事项列表，可单击“显示每月事项清单”，如图 13.3 所示。

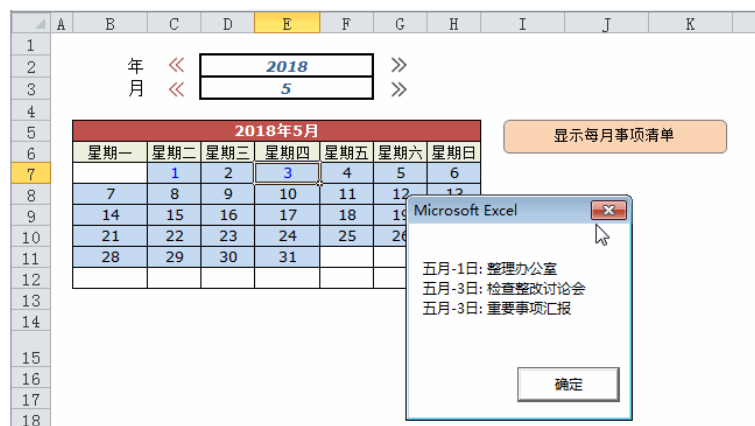


图 13.3（视频截图，完整视频详见完美 Excel 微信公众号）

所有事项都被记录在工作表 comments 中，当事项已完成时，你可以删除该项事项。

下载信息

这款模板运用公式创建日历，使用 VBA 代码来增减日期并完成事项的添加和显示。有兴趣的朋友可以在完美 Excel 公众号中发送消息：

日历清单模板

下载该模板的中文版。

也可以到下面的网址下载英文原版：

<https://trumpexcel.com/calendar-to-do-list-template/>



关于完美 Excel

完美 Excel 是一个坚持分享 Excel 与 VBA 技术知识的微信公众号，自 2017 年 5 月 15 日开始，每天推送一篇 Excel 与 VBA 技术和应用方面的文章。目前，共有 670 余篇实用文章可供广大 Excel 爱好者和使用者学习交流。这本电子书就是根据完美 Excel 上发表的 Excel 实用工具系列文章整理而成的。

每天清晨，完美 Excel 微信公众号：**excelperfect** 都会推送一篇关于 Excel 与 VBA 的相关文章。如果你有兴趣学习 Excel 和 VBA 的相关知识和实战技巧，可以关注完美 Excel 微信公众号，绝对不会让你失望！

可以通过下列方法关注[完美 Excel]微信公众号：

方法 1—在通讯录中搜索“完美 Excel”或者“**excelperfect**”后点击关注。

方法 2—扫一扫下面的二维码



完美 Excel 微信公众号使用指南

下图 1 是完美 Excel 微信公众号的界面。公众号名称显示在屏幕正上方，屏幕底部显示有“菜单栏”，目前设置的菜单为“技术精粹”、“VBA 精选”、“联系 me”。在底部左侧的小键盘图标为消息框入口，单击可进入消息框界面给完美 Excel 公众号发送消息。



图 1

下图 2、图 3、图 4 分别为底部 3 个菜单的子菜单。目前，菜单“技术精粹”中设置有“2018 年文章合集”、“480 篇文章合集”、“又一波 20 个函数”、“快速学会 30 个函数”、“玩转数据验证”等 5 个子菜单；菜单“VBA 精选”中设置有“Excel VBA 编程基础”、“最最基础入门篇”、“VBA 学习经验”等 3 个子菜单；菜单“联系 me”中设置有“投稿须知”、“网站集粹”、“个人声明”、“坚持的美好”、“2019 年目标”等 5 个子菜单。





图 2



图 3





图 4

单击这些子菜单会进入详细的文章页面或者文章整理的入口页面，方便读者浏览或查阅本公众号的文章。同时，这些子菜单会随着完美 Excel 微信公众号内容的增加而适时调整。

可以单击底部左侧的小键盘图标，进入发送消息界面，如图 5 所示。在文本框中输入想要发送的文字，单击底部的“发送”按钮，就可以将消息发送给完美 Excel 微信公众号。

大家应留意完美 Excel 微信公众号推送的文章中的一些信息，例如，我会在百度网盘中存放一些文档资料或者示例工作簿文件，并在文章中给出进入百度网盘下载的文本信息，你只需在发送消息框中输入我给出的文本，单击发送后，就会收到一条关于下载链接和密码的信息。单击链接并按提示输入密码后，即可获得



相关的文档资料或示例工作簿文件了。



图 5

例如，在图 5 所示的界面中输入“Excel 动画图 2”后，会自动收到图 6 所示的信息，根据信息即可获取这个 Excel 动画图表文件。



图 6



希望大家在完美 Excel 微信公众号中能够学习到所需要的知识，获取到所需要的 Excel 应用技巧，提高自己的水平。但愿在今后的日子里，完美 Excel 微信公众号能够真正帮助大家发挥 Excel 的威力，为大家解决问题，提高工作效率。

