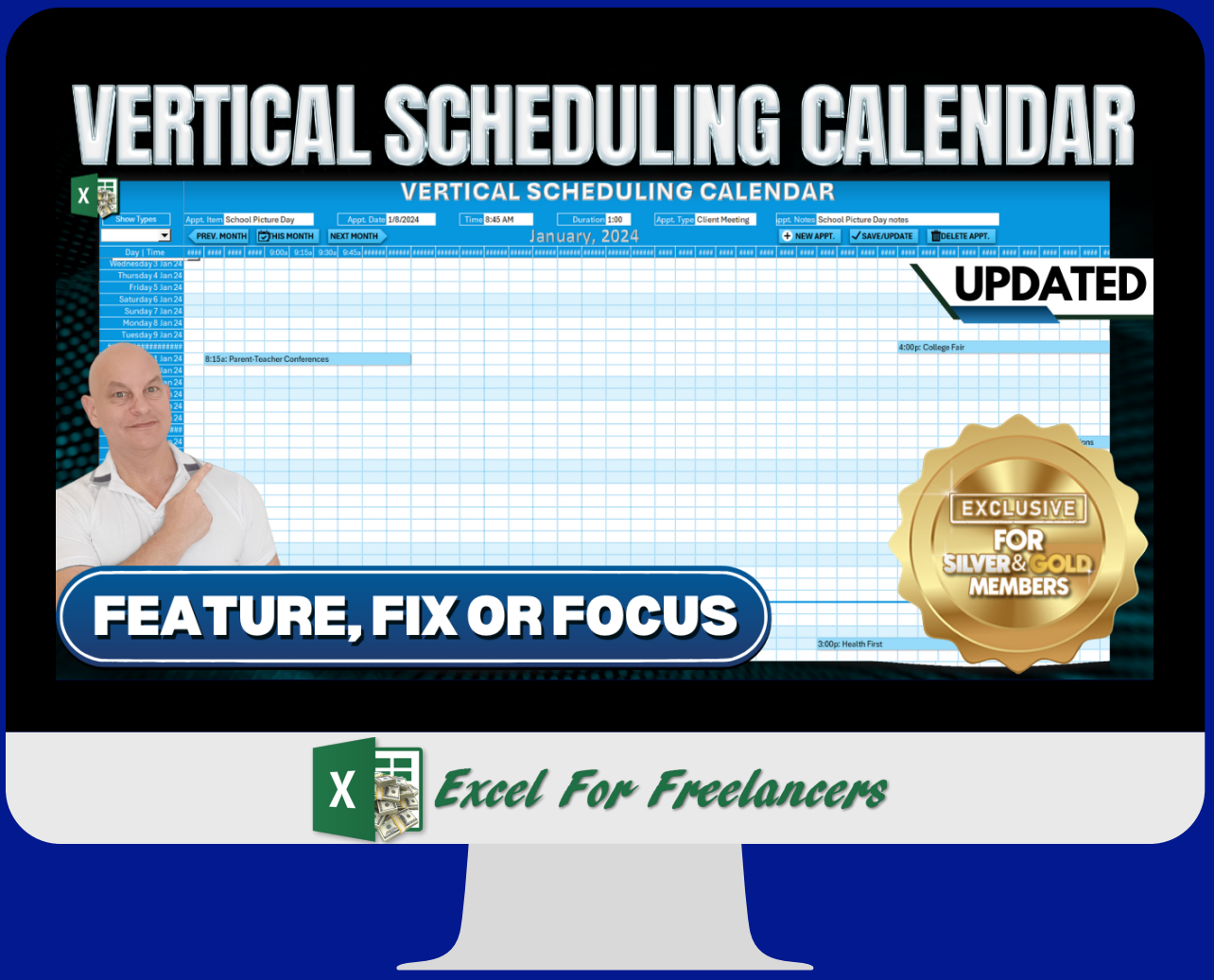
VBA SOURCE CODE BOOK



Create This Vertical Scheduling Calendar In Excel

DOWNLOAD APPLICATION

VIEW TRAINING

by: Randy Austin

ABOUT THE AUTHOR

A 4-time Microsoft MVP & lifetime Excel enthusiast, Randy Austin founded Excel For Freelancers in 2017. Excel For Freelancers quickly became the most prominent resource Excel for developers to learn how to turn their passion for Excel into profits by building & selling their own excel-based applications for passive & recurring income.

With over 471,000 YouTube subscribers, 35,448,742 video views, 430+ comprehensive training videos, and a thriving 65,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

Get free content, training, and downloads just by clicking any of the free resources below:

WEBSITE

YOUTUBE

FACEBOOK

TWITTER

DISCORD

INSTAGRAM

TELEGRAM

RUMBLE

Microsoft® MVP Most Valuable Professional

OUR COURSES & PRODUCTS

MENTORSHIP EXPRESS

This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

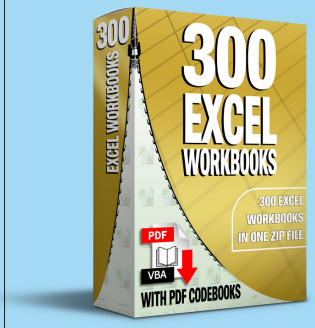
Click here to learn more

16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

Click here to learn more

This incredible 13-hour freelancer masterclass will teach anyone how to be a successful freelancer with my proven 9-Phase 'Financial Freedom Roadmap' and includes 30+ downloads and exercises.

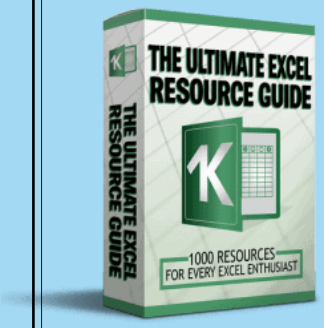
Click here to learn more



Incredible Package of 300 of my BEST Applications with PDF VBA Codebooks packed into a SINGLE ZIP File which also includes the "300 Workbook Library".

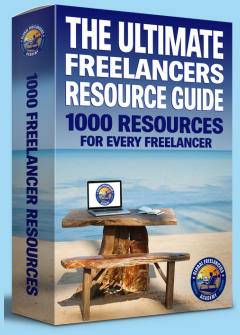
Click here to learn more

OUR COURSES & PRODUCTS



With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

Click here to learn more



Freelancing essentials from freelance tools to freelance templates, in this Ultimate Freelancer's Resource Guide for freelancers at any stage in their career. With 1,000 Live Links, and a single click-to-update application, you will always have the most current and up-to-date information at your fingertips.

Click here to learn more

Revolutionize the way you work with Excel and take productivity to the next level with the Excel AI Toolpack - the FIRST AI tool designed for ANY Windows Desktop version of Excel. This incredible add-in combines FIVE powerful AI tools, transforming your Excel into an intelligent powerhouse!

Click here to learn more

Table of Contents

```
1   Option Explicit
2
3   Private Sub Worksheet_Change(ByVal Target As Range)
4     If Not Intersect(Target, Range("F8" )) Is Nothing And Range("F8" ).Value <> Empty
      Then
5       Calendar.Range("D6:D371" ).NumberFormat = Target.Value 'Change Date Format
6     End If
7     If Not Intersect(Target, Range("F9" )) Is Nothing And Range("F9" ).Value <> Empty
      Then Calendar_ChangeView 'View Change
8   End Sub
9
10  Private Sub Worksheet_SelectionChange(ByVal Target As Range)
11    If Shapes("ColorPalette" ).Visible = True Then Shapes("ColorPalette" ).Visible =
      msoFalse
12
13    'On Appt. Type Color Selection
14    If Not Intersect(Target, Range("C15:C33" )) Is Nothing Then
15      With Shapes("ColorPalette" )
16        .Left = Range("C"  & Target.Row).Left 'Set Left Position
17        .Top = Range("C"  & Target.Row + 1).Top 'Top Position
18        .Visible = msoCTrue
19      End With
20    End If
21
22
23    'On workday selection
24    If Not Intersect(Target, Range("C5:C11" )) Is Nothing Then
25      If Target.Value = Chr(111) Then Target.Value = Chr(254) Else Target.Value = Chr(
        111)
26      Range("F4" ).Select
27    End If
28
29  End Sub
```

```
1   Option Explicit
2
```

```
1   Option Explicit
2
3   Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     'Check To make sure calendar exists, if not, copy from Developers sheet and paste
      it in
5     Dim Cal As Shape
6     On Error Resume Next
7     Set Cal = Shapes("Calendar" )
8     On Error GoTo 0
9     If Cal Is Nothing Then ReplaceCalendar 'Shape Deleted
10    If Shapes("Calendar" ).Visible = True Then CalendarHide 'Run macro to hide calendar
11
12    If Not Intersect(Target, Range("N3" )) Is Nothing Then 'Change this to any cell (or
      cells) you would like to have the Pop-Up Calendar Appear
13      CheckForSheet
14      CalendarShow
15    Else:
16      CheckForSheet
17      CalendarHide
18    End If
19  End Sub
```

```
1    Option Explicit
2
```

```
1    Option Explicit
2
```

```
1   Option Explicit
2
3   Sub Admin_SetTypeColor()
4       ActiveCell.Interior.Color = Admin.Shapes(Application.Caller).Fill.ForeColor.RGB
5       Admin.Shapes("ColorPalette" ).Visible = msoFalse
6   End Sub
```

```
Option Explicit

Dim ApptRow As Long, ApptCol As Long

Sub Appt_Delete()
  If MsgBox("Are you sure you want to delete this appointment?" , vbYesNo, "Delete Appointment" ) = vbNo Then Exit Sub
  With Calendar
    If .Range("B3" ).Value = Empty Then GoTo NotSaved
    ApptRow = .Range("B3" ).Value 'APpt DB Row
    ApptsDB.Range(ApptRow & ":"  & ApptRow).EntireRow.Delete
    NotSaved:
    Appt_New
    Calendar_Refresh 'Run macro to refresh calendar
  End With
End Sub

Sub Appt_Load()
  With Calendar
    If .Range("B3" ).Value = Empty Then
      MsgBox "Please select a correct appointment to load"
      Exit Sub
    End If
    Application.ScreenUpdating = False
    ApptRow = .Range("B3" ).Value 'Appt DB Row
    For ApptCol = 2 To 7
      .Range(ApptsDB.Cells(1, ApptCol).Value).Value = ApptsDB.Cells(ApptRow, ApptCol ).Value 'Bring Over Data
    Next ApptCol
    Application.ScreenUpdating = True
  End With
End Sub

Sub Appt_New()
  Calendar.Range("B2,G3:J3,N3:O3,R3:S3,W3,AA3:AC3,AG3:AO3" ).ClearContents 'Clear Out Existing Cells
End Sub


Sub Appt_SaveUpdate()
  With Calendar
    If Application.WorksheetFunction.CountA(.Range("G3,N3,R3,W3" )) < 4 Then
      MsgBox "Please fill in the required fields, Item, Date, Time & Duration"
      Exit Sub
    End If
    Application.ScreenUpdating = False
    If .Range("B3" ).Value = Empty Then 'New Appt
      ApptRow = ApptsDB.Range("A99999" ).End(xlUp).Row + 1  'First Avail. Row
      .Range("B2" ).Value = .Range("B4" ).Value 'Next Appt ID
      ApptsDB.Range("A"  & ApptRow).Value = .Range("B2" ).Value 'Appt ID
    Else 'Existing
      ApptRow = .Range("B3" ).Value
    End If

    For ApptCol = 2 To 7
      ApptsDB.Cells(ApptRow, ApptCol).Value = .Range(ApptsDB.Cells(1, ApptCol).Value ).Value 'Bring Over Data
    Next ApptCol
    Calendar_Refresh 'Run macro to refresh calendar
    Application.ScreenUpdating = True
```

```
        End With
End Sub
```

```
Option Explicit

Dim ApptRow As Long, ApptCol As Long, LastRow As Long, LastResultRow As Long
Dim DurCols As Long, ResultRow As Long, TypeRow As Long
Dim ApptColor As String, ApptID As String, ApptName As String, ApptType As String
Dim ApptShp As Shape
Dim ApptDate As Date, StartDate As Date
Dim ApptLeft As Double, ApptTime As Double, ApptDur As Double, SchedDur As Double,
SchedStart As Double


Sub Calendar_Appt_Select()
  With Calendar
    ApptID = Replace(Application.Caller, "CalAppt" , "" ) 'Extract Appt ID
    .Range("B2" ).Value = ApptID
    .Range("B13" ).Value = .Shapes(Application.Caller).Left 'Set Left Pos. of
    Current Selected Shape
    .Range("B14" ).Value = .Shapes(Application.Caller).Top 'Set Top Pos of Sel.
    Shape
    .Range("B15" ).Value = False 'Set Move Shape To False
    .Shapes(Application.Caller).Select 'Select On shape
    Appt_Load
    Appt_CheckForMove
  End With
End Sub

Sub Calendar_ChangeView()
  With Calendar
    If [ViewType] = "Month"  Then
      .Rows("6:371" ).EntireRow.RowHeight = 15.5
      .Shapes("MonthGrp" ).Visible = msoCTrue
      .Shapes("WeekGrp" ).Visible = msoFalse
    Else 'Week View
      .Range("6:371" ).EntireRow.RowHeight = 68
      .Shapes("MonthGrp" ).Visible = msoFalse
      .Shapes("WeekGrp" ).Visible = msoCTrue
    End If
  End With
End Sub

Sub Calendar_GoToMonth()
  Dim MonthStart As Date
  Dim CalRow As Long
  With Calendar
    MonthStart = DateSerial(.Range("B5" ).Value, .Range("B6" ).Value, 1)
    CalRow = DatePart("y" , MonthStart) + 5 'Calendard Row based on the day # of the
    year
    If CalRow > 5 Then ActiveWindow.ScrollRow = CalRow
  End With
End Sub

Sub Calendar_GoToWeek()
  Dim WeekStart As Date
  Dim CalRow As Long
  With Calendar
    WeekStart = .Range("B8" ).Value 'Week Start
    CalRow = DatePart("y" , WeekStart) + 5 'Calendard Row based on the day # of the
    year
    If CalRow > 5 Then ActiveWindow.ScrollRow = CalRow
  End With
1
```

```
End Sub


Sub Calendar_NextMonth()
  With Calendar
    If .Range("B6" ).Value = 12 Then 'On December Go to Next year and January
      .Range("B5" ).Value = .Range("B5" ).Value + 1   'Increase year by 1
      .Range("B6" ).Value = 1 'Set To January
      Calendar_Refresh 'Refresh Calendar
    Else:
      .Range("B6" ).Value = .Range("B6" ).Value + 1
    End If
    Calendar_GoToMonth 'Go to specific month
  End With
End Sub

Sub Calendar_NextWeek()
  With Calendar
    If Year(.Range("B8" ).Value) <> Year(.Range("B8" ).Value + 7) Then  'On Next
    year week, go to next year
      .Range("B5" ).Value = .Range("B5" ).Value + 1   'Increase year by 1
      .Range("B6" ).Value = 1 'Set To January
      Calendar_Refresh 'Refresh Calendar
    End If
    .Range("B8" ).Value = .Range("B8" ).Value + 7
    Calendar_GoToWeek 'Go to specific week
  End With
End Sub

Sub Calendar_PrevMonth()
  With Calendar
    If .Range("B6" ).Value = 1 Then 'On Janauary Go to Prev Year & December
      .Range("B5" ).Value = .Range("B5" ).Value - 1   'Reduce year by 1
      .Range("B6" ).Value = 12 'Set To December
      Calendar_Refresh 'Refresh Calendar
    Else:
      .Range("B6" ).Value = .Range("B6" ).Value - 1
    End If
    Calendar_GoToMonth 'Go to specific month
  End With
End Sub


Sub Calendar_PrevWeek()
  With Calendar
    If Year(.Range("B8" ).Value) <> Year(.Range("B8" ).Value - 7) Then  'On Previous
    year week, go to previous year
      .Range("B5" ).Value = .Range("B5" ).Value - 1   'Reduce year by 1
      .Range("B6" ).Value = 12 'Set To December
      Calendar_Refresh 'Refresh Calendar
    End If
    .Range("B8" ).Value = .Range("B8" ).Value - 7
    Calendar_GoToWeek 'Go to specific week
  End With
End Sub

Sub Calendar_Refresh()
  'Clear All existing appt shapes
  For Each ApptShp In Calendar.Shapes
    If InStr(ApptShp.Name, "CalAppt" ) > 0 Then ApptShp.Delete
```

```
114         Next ApptShp
115
116         'Run advanced filter for appts. database
117         With ApptsDB
118           LastRow = .Range("A99999" ).End(xlUp).Row  'Last Row of data
119           If LastRow < 4 Then Exit Sub
120           .Range("A3:G"  & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
              "I2:K3" ), CopyToRange:=.Range("M2:R2" ), Unique:=True
121           LastResultRow = .Range("M99999" ).End(xlUp).Row
122           If LastResultRow < 3 Then Exit Sub
123           Application.ScreenUpdating = False ' Turn Off Screen updating
124           SchedDur = [Duration] 'Schedule duration
125           SchedStart = [StartTime] 'Schedule Start Time
126           StartDate = [StartYearDate] 'Start Year Date
127           For ResultRow = 3 To LastResultRow
128             ApptID = .Range("M"  & ResultRow).Value 'Appt ID
129             ApptName = .Range("N"  & ResultRow).Value 'Appointment Name
130             ApptType = .Range("O"  & ResultRow).Value 'Appt Type
131             ApptDate = .Range("P"  & ResultRow).Value 'Appt Date
132             ApptTime = .Range("Q"  & ResultRow).Value 'Appt Time
133             ApptDur = .Range("R"  & ResultRow).Value 'Appt Duration
134             DurCols = Int(ApptDur / SchedDur) 'Set & of columns for duration
135             On Error Resume Next
136             TypeRow = Admin.Range("ApptTypes" ).Find(ApptType, , xlValues, xlWhole).Row
                'Get Type Row
137             On Error GoTo 0
138             If TypeRow <> 0 Then ApptColor = Admin.Range("C"  & TypeRow).Interior.Color
                Else ApptColor = ""  'Get Interior Color
139             ApptRow = ApptDate - StartDate + 6
140             ApptCol = (ApptTime - SchedStart) / SchedDur + 5
141             If ApptCol > 53 Then GoTo NextAppt
142             Calendar.Shapes("SampleApptShp" ).Duplicate.Name = "CalAppt"  & ApptID
143             With Calendar.Shapes("CalAppt"  & ApptID)
144               .Left = Calendar.Cells(ApptRow, ApptCol).Left
145               .Top = Calendar.Cells(ApptRow, ApptCol).Top + 1
146               If ApptCol + DurCols > 53 Then DurCols = 53 - ApptCol
147               .Width = Range(Calendar.Cells(ApptRow, ApptCol), Calendar.Cells(ApptRow,
                  ApptCol + DurCols)).Width 'Set Appt Width
148               .Height = Calendar.Cells(ApptRow, ApptCol).Height 'Set Height - 1
149               .TextFrame2.TextRange.Text = Format(ApptTime, "h:mma/p" ) & ": "  &
                  ApptName 'Text inside shape
150               If ApptColor <> ""  Then .Fill.ForeColor.RGB = ApptColor 'Set Shape color
151               .OnAction = "Calendar_Appt_Select"  'Macro to run
152             End With
153             NextAppt:
154           Next ResultRow
155           Application.ScreenUpdating = True 'Turn on Screen updating
156         End With
157       End Sub
158
159       Sub Calendar_ThisMonth()
160         With Calendar
161           .Range("B6" ).Value = Month(Date) 'Set Current Month
162           If Year(Date) <> .Range("B5" ).Value Then
163             .Range("B5" ).Value = Year(Date) 'Set to current year
164             Calendar_Refresh 'Refresh Calendar
165           End If
166           Calendar_GoToMonth 'Go to specific month
167         End With
168       End Sub
```

```
169
170     Sub Calendar_ThisWeek()
171       Calendar.Range("B8" ).Value = .Range("B9" ).Value
172       Calendar_GoToWeek 'Go to this week
173     End Sub
```

```
1   Option Explicit
2
3   Sub Appt_CheckForMove()
4     Dim DestRow As Long, DestCol As Long, ApptRow As Long, ApptCol As Long, CountDelay
      As Long
5     Dim ApptID As String, DestDate As String
6     Dim DestTime As Date
7     With Calendar
8       ApptID = .Range("B2" ).Value 'Select Appt ID
9       ApptRow = .Range("B3" ).Value 'Select Appt DB Row
10      For CountDelay = 1 To 100000
11        DoEvents
12        If .Range("B15" ).Value = True Then GoTo EndDragDrop 'Exit loop move Appt =
          True
13        With .Shapes("CalAppt"  & ApptID)
14          If .Left <> Calendar.Range("B13" ).Value Or .Top <> Calendar.Range("B14" ).
            Value Then 'Shape has been moved
15            'Check to see if shape is moved within proper grid area
16            If .Left < Calendar.Range("E1" ).Left Or .Left > Calendar.Range("BB1" ).
              Left Or .Top < Calendar.Range("E6" ).Top - 1 Or Calendar.Range("D"  & .
              TopLeftCell.Row).Value = Empty Then
17              MsgBox "Please move the Appt shape within the given Calendar"
18              GoTo EndDragDrop
19            End If
20
21            DestDate = Calendar.Range("D"  & .TopLeftCell.Row).Value 'Destination
              Date
22            DestTime = Calendar.Cells(5, .TopLeftCell.Column).Value 'Destination
              Time
23            ApptsDB.Range("D"  & ApptRow).Value = DestDate 'Update Date
24            ApptsDB.Range("E"  & ApptRow).Value = DestTime 'Update Time
25            GoTo EndDragDrop
26          End If
27        End With
28      Next CountDelay
29      EndDragDrop:
30      Calendar_Refresh 'Refresh Calendar
31      Appt_Load 'Refresh Appt data
32      .Range("B15" ).Value = True
33    End With
34  End Sub
```

```
Dim SelCell As Range
Dim DayName As String



'''''''''''''''''Color Calendar Background'''''''''''''''''''''
Sub CalCol()
  With ActiveSheet.Shapes.Range(Array("CalBack" , "Settings" )).Select
    With Selection.ShapeRange.Fill
      .ForeColor.RGB = ActiveSheet.Shapes(Application.Caller).Fill.ForeColor.RGB
    End With
    ActiveSheet.Range(Sheets("CalPopUp" ).Range("A7" ).Value).Select
  End With
End Sub
Sub CalendarHide()
  Dim DayNum As Long
  'Hide Calendar, Reset Day Colors
  On Error GoTo NoCal
  ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
  Sheets("CalPopUp" ).Range("A7" ).Value = ""
  If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
    For DayNum = 1 To 42
      DayName = DayNum & "Day"
      With ActiveSheet.Shapes(DayName)
        .Fill.ForeColor.RGB = RGB(255, 255, 255)
        .TextFrame2.TextRange.Font.Bold = msoFalse
      End With
    Next DayNum
  End If
  Exit Sub
  NoCal: 'If calendar has been removed by accident, paste in backup calendar from CalPopUp Sheet
End Sub

Sub CalendarShow()
  With ActiveSheet
    Set SelCell = ActiveCell
    'Check if active cell is a valid date
    If IsDate(SelCell.Value) = True Then
      Sheets("CalPopUp" ).Range("A1" ).Value = SelCell.Value
    Else: 'If No Date or incorrect Date user current date
      Sheets("CalPopUp" ).Range("A1" ).Value = "=Today()"
    End If
    'Clear all shapes to white (if calendar is visible)
    If ActiveSheet.Shapes("Calendar" ).Visible = True Then
      For DayNum = 1 To 42
        DayName = DayNum & "Day"
        With ActiveSheet.Shapes(DayName)
          .Fill.ForeColor.RGB = RGB(255, 255, 255)
          .TextFrame2.TextRange.Font.Bold = msoFalse
        End With
      Next DayNum
    End If

    Sheets("CalPopUp" ).Range("A3" ).Value = Month(Sheets("CalPopUp" ).Range("A1" ).Value) 'Set Month
    Sheets("CalPopUp" ).Range("A2" ).Value = Year(Sheets("CalPopUp" ).Range("A1" ).Value) 'Set Year
    DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
    ' UnGroupCal
1 2
```

```
58          If InStr(.Shapes("1Day" ).DrawingObject.Formula, "]" ) <> 0 Then   'Run Workbook Link Remover and Cell Link Replacement
59            MacroLinkRemover
60            CalFormulaReplacement
61          End If
62          'GroupCal
63          On Error GoTo NoCal
64          If DayName = "Day"  Then DayName = Day(Date) & "Day"  'Set Default Day
65          With ActiveSheet.Shapes(DayName)
66            .Fill.ForeColor.RGB = RGB(252, 213, 180)
67            .TextFrame2.TextRange.Font.Bold = msoTrue
68          End With
69          On Error GoTo NoCal
70          .Shapes("Calendar" ).Visible = msoCTrue
71          .Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" , "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible = False '
72          .Shapes("Calendar" ).Left = SelCell.Left
73          .Shapes("Calendar" ).Placement = xlMove
74          .Shapes("Calendar" ).ZOrder msoBringToFront
75          .Shapes("Calendar" ).Top = SelCell.Offset(1, 0).Top
76          If Sheets("CalPopUp" ).Range("A6" ).Value > 0 Then
77            .Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" , "40Day" , "41Day" , "42Day" )).Visible = True
78          Else:
79            .Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" , "40Day" , "41Day" , "42Day" )).Visible = False
80          End If
81          Sheets("CalPopUp" ).Range("A7" ).Value = SelCell.Address
82          ActiveCell.Select
83        End With
84        Exit Sub
85        NoCal:
86        MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the calendar over from another sheet and paste into this sheet"
87      End Sub
88      Sub CalFormulaReplacement()
89        With ActiveSheet
90          Dim DayNum, ColNum, RowNum As Long
91          Dim Shp As Shape
92          ColNum = 2
93          RowNum = 1
94          For DayNum = 1 To 42
95            .Shapes(DayNum & "Day" ).DrawingObject.Formula = "=CalPopUp!"  & .Cells(RowNum , ColNum).Address
96            ColNum = ColNum + 1
97            If ColNum = 9 Then
98              ColNum = 2
99              RowNum = RowNum + 1
100           End If
101         Next DayNum
102         .Shapes("Month" ).DrawingObject.Formula = "=CalPopUp!$A$4"
103         .Shapes("Year" ).DrawingObject.Formula = "=CalPopUp!$A$2"
104       End With
105     End Sub
106
107     Sub CheckForSheet()
108       'Checks for existance of Calendar Pop-up Worksheet
109       Dim ws As Worksheet
110       On Error GoTo CreateWS
111       Set ws = ActiveWorkbook.Sheets("CalPopUp" )
```

```
    Exit Sub
    CreateWS:
    CreateCalSht
End Sub


'Create Calendar Sheet on First Run of Calendar
Sub CreateCalSht()
    Dim ColCnt, RowCnt, DayCnt, CalCol As Long
    Dim ws, ActSht As Worksheet
    Set ActSht = ActiveSheet
    'On Error GoTo NoCal
    ActiveSheet.Shapes("Calendar" ).Copy
    Set ws = ThisWorkbook.Sheets.Add(After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.
    Count))
    ws.Name = "CalPopUp"
    ActSht.Activate

    'Reassign Shape Links & Macros
    With ActiveSheet
      UnGroupCal
      '.Unprotect
      .Shapes("PrevYr" ).OnAction = "'"  & ActiveWorkbook.Name & "'!PrevYear"
      .Shapes("NextYr" ).OnAction = "'"  & ActiveWorkbook.Name & "'!NextYear"
      .Shapes("NextRec" ).OnAction = "'"  & ActiveWorkbook.Name & "'!NextMonth"
      .Shapes("NextTri" ).OnAction = "'"  & ActiveWorkbook.Name & "'!NextMonth"
      .Shapes("PrevRec" ).OnAction = "'"  & ActiveWorkbook.Name & "'!PrevMonth"
      .Shapes("PrevTri" ).OnAction = "'"  & ActiveWorkbook.Name & "'!PrevMonth"
      .Shapes("SetBtn" ).OnAction = "'"  & ActiveWorkbook.Name & "'!ShowSettings"
      .Shapes("Month" ).DrawingObject.Formula = "=CalPopUp!A4"
      .Shapes("Year" ).DrawingObject.Formula = "=CalPopUp!A2"
      DayCnt = 1
      For RowCnt = 1 To 6
        For ColCnt = 2 To 8
          .Shapes(DayCnt & "Day" ).DrawingObject.Formula = "=CalPopUp!"  & .Cells(
          RowCnt, ColCnt).Address 'Assigned Linked Cell
          .Shapes(DayCnt & "Day" ).OnAction = "'"  & ActiveWorkbook.Name &
          "'!SelectDay"  'Assign Macro
          DayCnt = DayCnt + 1
        Next ColCnt
      Next RowCnt

      'Assign Color Macros
      For CalCol = 1 To 9
        .Shapes("CalCol"  & CalCol).OnAction = "'"  & ActiveWorkbook.Name &
        "'!CalCol"  'Assign Color Macro
      Next CalCol
    End With

    With Sheets("CalPopUp" )
      .Paste
      .Visible = xlSheetHidden

      'Add in Formulas and Details
      .Range("A1" ).Value = Date 'Set Current Date
      .Range("A2" ).Value = Year(Date) 'Set Current Year
      .Range("A3" ).Value = Month(Date) 'Set Current Month #
      .Range("A4" ).Value = "=INDEX(CalMonths,A3,)"
      .Range("A5" ).Value = "=A4&"  & Chr(34) & " "  & Chr(34) & "&CalYear"
      .Range("A6" ).Value = "=SUM(B6:H6)"
```

```
168         .Range("A8" ).Value = "January"
169         .Range("A8" ).AutoFill Destination:=.Range("A8:A19" ), Type:=xlFillDefault
170         .Range("A20" ).Value =
            "=IFERROR(INDIRECT(ADDRESS(SUMPRODUCT((B1:H6=A1)*ROW(B1:H6))+6,SUMPRODUCT((B1:H6=A
            1)*COLUMN(B1:H6)),1,1)),"  & Chr(34) & Chr(34) & ")"
171         'Set Defined Names
172         ActiveWorkbook.Names.Add Name:="CalMonths" , RefersTo:="=CalPopUp!$A$8:$A$19"
173         ActiveWorkbook.Names.Add Name:="CalYear" , RefersTo:="=CalPopUp!$A$2"
174
175         'Add in Calendar Formulas
176
177         .Range("B1" ).Value =
            "=IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=1,DATE(CalYear,MATCH($A$4,Ca
            lMonths,0),1),"  & Chr(34) & Chr(34) & ")"
178         .Range("C1" ).Value = "=IF(B1<>"  & Chr(34) & Chr(34) &
            ",B1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=2,DATE(CalYear,MATCH($A
            $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
179         .Range("D1" ).Value = "=IF(C1<>"  & Chr(34) & Chr(34) &
            ",C1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=3,DATE(CalYear,MATCH($A
            $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
180         .Range("E1" ).Value = "=IF(D1<>"  & Chr(34) & Chr(34) &
            ",D1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=4,DATE(CalYear,MATCH($A
            $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
181         .Range("F1" ).Value = "=IF(E1<>"  & Chr(34) & Chr(34) &
            ",E1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=5,DATE(CalYear,MATCH($A
            $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
182         .Range("G1" ).Value = "=IF(F1<>"  & Chr(34) & Chr(34) &
            ",F1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=6,DATE(CalYear,MATCH($A
            $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
183         .Range("H1" ).Value = "=IF(G1<>"  & Chr(34) & Chr(34) &
            ",G1+1,IF(WEEKDAY(DATE(CalYear,MATCH($A$4,CalMonths,0),1))=7,DATE(CalYear,MATCH($A
            $4,CalMonths,0),1),"  & Chr(34) & Chr(34) & "))"
184         .Range("B2" ).Value = "=H1+1"
185         .Range("C2" ).Value = "=B2+1"
186         .Range("C2" ).AutoFill Destination:=.Range("C2:H2" ), Type:=xlFillDefault
187         .Range("B2:H2" ).AutoFill Destination:=.Range("B2:H4" ), Type:=xlFillDefault
188         .Range("B5" ).Value = "=IF(OR(H4="  & Chr(34) & Chr(34) & ",MONTH(H4+1)<>$A$3),"
             & Chr(34) & Chr(34) & ",H4+1)"
189         .Range("C5" ).Value = "=IFERROR(IF(MONTH(B5+1)<>$A$3,"  & Chr(34) & Chr(34) &
            ",B5+1),"  & Chr(34) & Chr(34) & ")"
190         .Range("B6" ).Value = "=IFERROR(IF(OR(H5="  & Chr(34) & Chr(34) &
            ",MONTH(H5+1)<>$A$3),"  & Chr(34) & Chr(34) & ",H5+1),"  & Chr(34) & Chr(34) &
            ")"
191         .Range("C6" ).Value = "=IFERROR(F(MONTH(I5+1)<>$A$3,"  & Chr(34) & Chr(34) &
            ",I5+1),"  & Chr(34) & Chr(34) & ")"
192         .Range("C5:C6" ).AutoFill Destination:=.Range("C5:H6" ), Type:=xlFillDefault
193
194         'Set format to Single Day
195         .Range("B1:H6" ).NumberFormat = "d"
196
197         'Add in relative Day #'s
198         .Range("B7" ).Value = "1"
199         .Range("C7" ).Value = "2"
200         .Range("B8" ).Value = "8"
201         .Range("C8" ).Value = "9"
202         .Range("B7:C8" ).AutoFill Destination:=.Range("B7:H8" ), Type:=xlFillDefault
203         .Range("B7:H8" ).AutoFill Destination:=.Range("B7:H12" ), Type:=xlFillDefault
204         GroupCal
205     End With
206     Exit Sub
```

```
    NoCal:
    MsgBox "The Pop-up Calendar does not exist on this worksheet. Please copy the calendar over from another sheet and paste into this sheet"
End Sub
Sub GroupCal()
    ActiveSheet.Shapes.Range(Array("NextTri" , "NextRec" )).Group.Select
    Selection.ShapeRange.Name = "NextMonth"
    ActiveSheet.Shapes.Range(Array("PrevTri" , "PrevRec" )).Group.Select
    Selection.ShapeRange.Name = "PrevMonth"
    ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
        , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" , "1Day" , _
        "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
        "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" , "18Day" _
        , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
        "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" , "Sa" , _
        "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" , "CalCol3" , _
        "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth" , _
        "NextMonth" , "NextYr" , "PrevYr" )).Visible = msoCTrue
    ActiveSheet.Shapes.Range(Array("Settings" , "40Day" , "41Day" , "39Day" , "38Day" _
    , "42Day" , "37Day" , "36Day" , "CalBack" , "Month" , "Year" , "CalBorder" , "1Day" , _
    "3Day" , "14Day" , "7Day" , "4Day" , "2Day" , "5Day" , "8Day" , "10Day" , "6Day" , _
    "13Day" , "11Day" , "9Day" , "12Day" , "15Day" , "17Day" , "20Day" , "21Day" , "18Day" _
    , "16Day" , "19Day" , "22Day" , "24Day" , "26Day" , "27Day" , "25Day" , "23Day" , _
    "28Day" , "29Day" , "31Day" , "34Day" , "35Day" , "32Day" , "30Day" , "33Day" , "Sa" , _
    "Fr" , "Th" , "We" , "Tu" , "Mo" , "Su" , "SetBtn" , "CalCol1" , "CalCol2" , "CalCol3" , _
    "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" , "PrevMonth" , _
    "NextMonth" , "NextYr" , "PrevYr" )).Select
    Selection.ShapeRange.Group.Select
    Selection.ShapeRange.Name = "Calendar"
    Selection.Name = "Calendar"
    Selection.Placement = xlMove
    ActiveSheet.Shapes("Calendar" ).Placement = 2
End Sub



Sub MacroLinkRemover()
    'PURPOSE: Remove an external workbook reference from all shapes triggering macros
    'Source: www.ExcelForFreelancers.com
    Dim Shp As Shape
    Dim MacroLink, NewLink As String
    Dim SplitLink As Variant

    For Each Shp In ActiveSheet.Shapes 'Loop through each shape in worksheet
```

```
    'Grab current macro link (if available)
    On Error GoTo NextShp
    MacroLink = Shp.OnAction

    'Determine if shape was linking to a macro
    If MacroLink <> ""  And InStr(MacroLink, "!" ) <> 0 Then
      'Split Macro Link at the exclaimation mark (store in Array)
      SplitLink = Split(MacroLink, "!" )

      'Pull text occurring after exclaimation mark
      NewLink = SplitLink(1)

      'Remove any straggling apostrophes from workbook name
      If Right(NewLink, 1) = "'"  Then
        NewLink = Left(NewLink, Len(NewLink) - 1)
      End If

      'Apply New Link
      Shp.OnAction = NewLink
    End If
    NextShp:
  Next Shp
End Sub
Sub NextMonth()
  'Next Month button
  If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
    DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
    With ActiveSheet.Shapes(DayName)
      .Fill.ForeColor.RGB = RGB(255, 255, 255)
      .TextFrame2.TextRange.Font.Bold = msoFalse
    End With
  End If
  With Sheets("CalPopUp" )
    If .Range("A3" ).Value = 12 Then
      .Range("A3" ).Value = 1
      .Range("A2" ).Value = .Range("A2" ).Value + 1
    Else:
      .Range("A3" ).Value = .Range("A3" ).Value + 1
    End If
    If .Range("A6" ).Value > 0 Then
      ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
      "40Day" , "41Day" , "42Day" )).Visible = True
    Else:
      ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
      "40Day" , "41Day" , "42Day" )).Visible = False
    End If
  End With
End Sub

Sub NextYear()
  ThisWorkbook.Sheets("CalPopUp" ).Range("A2" ).Value = ThisWorkbook.Sheets(
  "CalPopUp" ).Range("A2" ).Value + 1
End Sub
Sub PrevMonth()
  'Previous Month Button
  If Sheets("CalPopUp" ).Range("A20" ).Value <> Empty Then
    DayName = Sheets("CalPopUp" ).Range("A20" ).Value & "Day"
    With ActiveSheet.Shapes(DayName)
      .Fill.ForeColor.RGB = RGB(255, 255, 255)
```

```
1 2 3
308             .TextFrame2.TextRange.Font.Bold = msoFalse
309           End With
310         End If
311         With Sheets("CalPopUp" )
312           If .Range("A3" ).Value = 1 Then
313             .Range("A3" ).Value = 12
314             .Range("A2" ).Value = .Range("A2" ).Value - 1
315           Else:
316             .Range("A3" ).Value = .Range("A3" ).Value - 1
317           End If
318           If .Range("A6" ).Value > 0 Then
319             ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
                "40Day" , "41Day" , "42Day" )).Visible = True
320           Else:
321             ActiveSheet.Shapes.Range(Array("36Day" , "37Day" , "38Day" , "39Day" ,
                "40Day" , "41Day" , "42Day" )).Visible = False
322           End If
323         End With
324       End Sub
325
326       Sub PrevYear()
327         ThisWorkbook.Sheets("CalPopUp" ).Range("A2" ).Value = ThisWorkbook.Sheets(
            "CalPopUp" ).Range("A2" ).Value - 1
328       End Sub
329       Sub ReplaceCalendar() 'Shape Deleted
330         ThisWorkbook.Sheets("CalPopUp" ).Shapes("Calendar" ).Copy 'Copy From Developers
            sheet
331         ActiveCell.Select 'Select the active Cell
332         ActiveSheet.Paste 'Paste in Calendar
333       End Sub
334
335       '''''''''''''''''''''''''''''''''''''''
336       '''''Select Day Of The Month
337       '''''''''''''''''''''''''''''''''''''''
338       Sub SelectDay()
339         Dim DayNumb As Long, RowNumb As Long, ColNumb As Long
340         DayNumb = Replace(Application.Caller, "Day" , "" )
341         RowNumb = Application.WorksheetFunction.RoundUp(DayNumb / 7, 0)
342         ColNumb = DayNumb Mod 7 + 1
343         If ColNumb = 1 Then ColNumb = 8
344         'On Error Resume Next
345         If ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value = Empty Then Exit Sub
346         ActiveSheet.Range(ThisWorkbook.Sheets("CalPopUp" ).Range("A7" ).Value).Value =
            ThisWorkbook.Sheets("CalPopUp" ).Cells(RowNumb, ColNumb).Value
347         ActiveSheet.Shapes("Calendar" ).Visible = msoFalse
348         ActiveCell.Offset(0, 1).Select
349       End Sub
350       Sub ShowSettings()
351         'Show or Hide Calendar Settings Panel
352         If ActiveSheet.Shapes.Range(Array("Settings" )).Visible = True Then
353           ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
              "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
              = False
354         Else:
355           ActiveSheet.Shapes.Range(Array("Settings" , "CalCol1" , "CalCol2" , "CalCol3" ,
              "CalCol4" , "CalCol5" , "CalCol6" , "CalCol7" , "CalCol8" , "CalCol9" )).Visible
              = True
356         End If
357       End Sub
358
```

```
359     Sub UnGroupCal()
360       On Error Resume Next
361       ActiveSheet.Shapes("Calendar" ).Ungroup
362       ActiveSheet.Shapes("NextMonth" ).Ungroup
363       ActiveSheet.Shapes("PrevMonth" ).Ungroup
364       On Error GoTo 0
365     End Sub
```

Thank You!

This source code was created and made available to help you gain a better understanding of how VBA is used to create amazing Excel-based applications.

Thank you so much for your continued shares, likes and support. It really helps.

Excel For Freelancers