

VBA SOURCE CODE BOOK

EXCEL CALENDAR TEMPLATE

SEPTEMBER, 2023

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
27	28	29	30	31	1	2
3	4	5	6	7:00 PM: School Board	8	9
10	11: 7:00 PM: PTA Meeting	12	13	8:00 AM: School Picture Day	14: 10:00 AM: Homecoming	15: 7:00 PM: Homecoming
17: 4:00 PM: Parent-Teacher	18	19	20: 5:00 PM: Fall Festival	21: 4:00 PM: Student Council	22: 7:00 PM: School Dance	23
24: 9:00 AM: Career Day	25: 6:00 PM: College Fair	26: 7:00 PM: School Play	27: 9:00 AM: Volunteer Day	28: 8:00 AM: Field Trip	29: 3:00 PM: Faculty Meeting	30: 5:00 PM: Student Council
						7:00 PM: School Fundraiser 9:00 PM: Teacher

**FREE
TEMPLATE
DOWNLOAD**

CALENDAR ITEM

Item Name	School Board Meeting
Date	9/7/2023
Time	7:00 PM
Duration	2:00
Notes	School Board Meeting notes



Excel For Freelancers

Learn How To Create This
2023 Calendar Template 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®
Most Valuable
Professional



OUR COURSES & PRODUCTS



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](#)

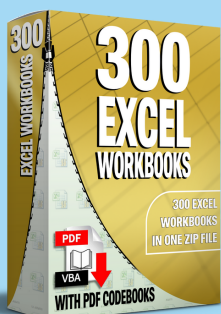
-



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](#)

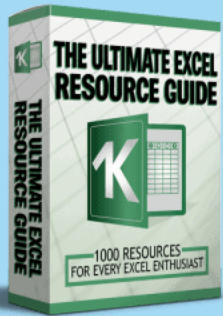
-



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](#)

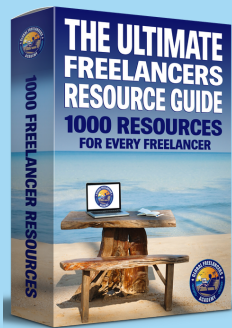
OUR COURSES & PRODUCTS



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](#)

Projects.....	2
VBAProject.....	2
Documents.....	2
Admin.....	2
(Declarations).....	2
Worksheet_SelectionChange [Sub].....	2
Calendar.....	3
(Declarations).....	3
ItemsDB.....	4
(Declarations).....	4
ThisWorkbook.....	5
(Declarations).....	5
Modules.....	6
Admin_Macros.....	6
(Declarations).....	6
Admin_SetPlatformColor [Sub].....	6
Calendar_Macros.....	7
(Declarations).....	7
Calendar_CheckForMove [Sub].....	7
Calendar_Item_Select [Sub].....	7
Calendar_NextMonth [Sub].....	8
Calendar_PrevMonth [Sub].....	8
Calendar_Refresh [Sub].....	8
Calendar_ThisMonth [Sub].....	10
PrintCalendar [Sub].....	10
Item_Macros.....	11
(Declarations).....	11
Item_Delete [Sub].....	11
Item_Load [Sub].....	11
Item_New [Sub].....	11
Item_SaveUpdate [Sub].....	11
Recurring_Macros.....	12
(Declarations).....	12
CreateRecurringItem [Sub].....	12
SendToOutlook_Macros.....	13
(Declarations).....	13
SendAppointmentToOutlook [Sub].....	13

```
1 Option Explicit
2
3 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
4     If Target.CountLarge > 1 Then Exit Sub
5     If Shapes("ColorPalette").Visible = True Then Shapes("ColorPalette").Visible =
        msoFalse
6
7     'Show color palette
8     If Not Intersect(Target, Range("F7,F9")) Is Nothing Then
9         With Shapes("ColorPalette")
10            .Left = Range("F" & Target.Row).Left
11            .Top = Range("F" & Target.Row + 1).Top
12            .Visible = msoCTrue
13        End With
14    End If
15 End Sub
```

1 Option Explicit

2

1 Option Explicit

2

1 Option Explicit

2

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

```

1 Option Explicit
2 Dim CalRow As Long, CalCol As Long, LastRow As Long, LastResultRow As Long, ResultRow
  As Long
3 Dim ItemNumb As Long, DayItemCount As Long, ItemTop
4 Dim ItemColor As String, ItemID As String, ItemName As String
5 Dim ItemShp As Shape
6 Dim ItemDate As Date, ItemTime As String
7 Dim ItemWidth As Double, ItemLeft As Double
8
9
10 Sub Calendar_CheckForMove()
11   Dim DestRow As Long, DestCol As Long, ItemRow As Long, ItemCol As Long, CountDelay
  As Long
12   With Calendar
13     If .Range("B10").Value = Empty Then Exit Sub
14     ItemID = .Range("B9").Value 'Item ID
15
16     For CountDelay = 1 To 100000
17       DoEvents
18       If .Range("B1").Value = True Then End 'Exit Loop on Move Item Move = True
19
20       With .Shapes("CalItem" & ItemID)
21         If .Left <> Calendar.Range("B15").Value Or .Top <> Calendar.Range("B16").
  Value Then 'Move Detect
22           'Check for incorrect move
23           If .Left < Calendar.Range("D1").Left Or .Left > Calendar.Range("K1").
  Left Or .Top > Calendar.Range("A39").Top Or .Top < Calendar.Range("A4"
  ).Top - 1 Then
24             MsgBox "Plase make sure to move the Calendar Item to a correct
  calendar date on the Calendar"
25             Calendar_Refresh
26             Exit Sub
27           End If
28
29           DestRow = Calendar.Shapes("CalItem" & ItemID).TopLeftCell.Row
30           DestCol = Calendar.Shapes("CalItem" & ItemID).TopLeftCell.Column
31           DestRow = DestRow - (DestRow + 2) Mod 6 'Date Row
32           ItemDate = Calendar.Cells(DestRow, DestCol).Value 'Calendar Date
33           If ItemDate = 0 Then
34             MsgBox "Please move the item to a day with an existing Date"
35             Calendar_Refresh
36             Exit Sub
37           End If
38           Calendar.Range("M5").Value = ItemDate 'new Item dates
39           Item_SaveUpdate 'Save Item Changes
40           Calendar.Range("B1").Value = True 'Set Move Item to true
41           End
42           End If
43         End With
44       Next CountDelay
45       .Range("B1").Value = True 'Set Move Item Movement to True
46     End With
47   End Sub
48
49
50
51 Sub Calendar_Item_Select()
52   With Calendar
53     ItemID = Replace(Application.Caller, Left(Application.Caller, 7), "" ) 'Extract
  Item ID
  
```

1 2

```

1 2
54 For Each ItemShp In .Shapes 'Reset All Colors back to default color
55     If InStr(ItemShp.Name, "CalItem" ) > 0 Then ItemShp.Fill.ForeColor.RGB = Admin
        .Range("F7" ).Interior.Color
56 Next ItemShp
57 .Shapes("CalItem" & ItemID).Fill.ForeColor.RGB = Admin.Range("F9" ).Interior.
    Color
58 .Range("B9" ).Value = ItemID
59 'Drag & Drop Settings
60 Calendar.Range("B8" ).Value = ItemID
61 .Range("B15" ).Value = Calendar.Shapes(Application.Caller).Left 'set Selected
    Left Position
62 .Range("B16" ).Value = Calendar.Shapes(Application.Caller).Top ' Set Selected
    Top Position
63 .Range("B1" ).Value = False 'Set Calendar Move to False
64 .Shapes("CalItem" & ItemID).Select
65 Item_Load
66 Calendar_CheckForMove
67 End With
68 End Sub

69
70 Sub Calendar_NextMonth()
71     With Calendar
72         If .Range("B2" ).Value = 7 And .Range("B5" ).Value = 12 Then
73             MsgBox "You are at the last year and first last available"
74             Exit Sub
75         End If
76         If .Range("B5" ).Value = 12 Then 'December Month
77             .Range("B2" ).Value = .Range("B2" ).Value + 1 'Increase Year by 1
78             .Range("B5" ).Value = 1 'Set To January
79         Else 'Not January
80             .Range("B5" ).Value = .Range("B5" ).Value + 1 'Increase month by 1
81         End If
82         Calendar_Refresh 'Refresh Calendar
83     End With
84 End Sub

85
86 Sub Calendar_PrevMonth()
87     With Calendar
88         If .Range("B2" ).Value = 1 And .Range("B5" ).Value = 1 Then
89             MsgBox "You are at the first year and first month available"
90             Exit Sub
91         End If
92         If .Range("B5" ).Value = 1 Then 'January Month
93             .Range("B2" ).Value = .Range("B2" ).Value - 1 'Reduce Year by 1
94             .Range("B5" ).Value = 12 'Set To December
95         Else 'Not January
96             .Range("B5" ).Value = .Range("B5" ).Value - 1 'Reduce month by 1
97         End If
98         Calendar_Refresh 'Refresh Calendar
99     End With
100 End Sub

101
102 Sub Calendar_Refresh()
103     'Clear all existing item shapes from Calendar sheet
104     For Each ItemShp In Calendar.Shapes
105         If InStr(ItemShp.Name, "CalItem" ) > 0 Then ItemShp.Delete
106     Next ItemShp
107     ItemColor = Admin.Range("F7" ).Interior.Color 'Set Item Shape Color
108     ItemNum = 1 'Set default item # to 1
109     ItemWidth = 1 'Set Default to 1

```

```

1
110 With ItemsDB
111     LastRow = .Range("A99999").End(xlUp).Row 'Last Item Row
112     If LastRow < 4 Then Exit Sub
113     Application.ScreenUpdating = False
114     .Range("A3:F" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
115         "I2:J3" ), CopyToRange:=.Range("M2:Q2" ), Unique:=True
116     LastResultRow = .Range("M99999").End(xlUp).Row
117     If LastResultRow < 3 Then Exit Sub
118     If LastResultRow < 4 Then GoTo NoSort
119     With .Sort
120         .SortFields.Clear
121         .SortFields.Add Key:=ItemsDB.Range("O3" ), SortOn:=xlSortOnValues, Order:=
122             xlAscending, DataOption:=xlSortNormal 'Sort Based On Date
123         .SortFields.Add Key:=ItemsDB.Range("P3" ), SortOn:=xlSortOnValues, Order:=
124             xlAscending, DataOption:=xlSortNormal 'Sort Based On Time
125         .SetRange ItemsDB.Range("M3:Q" & LastResultRow) 'Set Range
126         .Apply 'Apply Sort
127     End With
128     NoSort:
129     For ResultRow = 3 To LastResultRow
130         ItemID = .Range("M" & ResultRow).Value 'item ID
131         ItemName = .Range("N" & ResultRow).Value 'Item Name
132         ItemDate = .Range("O" & ResultRow).Value 'Item Date
133         ItemTime = Format(.Range("P" & ResultRow).Value, "h:mm/p" )
134         DayItemCount = Application.WorksheetFunction.CountIf([ItemDate_Results],
135             ItemDate) 'Get # of Items in a single Date
136         If DayItemCount > 5 Then
137             If DayItemCount > 10 Then DayItemCount = 10 'Set Item Limit to 10
138             ItemWidth = 0.5 'Set Item with to 1/2
139         End If
140         For CalRow = 4 To 34 Step 6
141             For CalCol = 4 To 10
142                 If Calendar.Cells(CalRow, CalCol).Value = ItemDate Then 'Day Found
143                     Calendar.Shapes("SampleItemShp" ).Duplicate.Name = "CalItem" &
144                         ItemID
145                     With Calendar.Shapes("CalItem" & ItemID)
146                         .Left = Calendar.Cells(CalRow, CalCol).Left + ItemLeft 'Set Left
147                         Pos.
148                         .Top = Calendar.Cells(CalRow + ItemNumb + ItemTop, CalCol).Top + 1
149                         .Width = Calendar.Cells(CalRow + ItemNumb, CalCol).Width *
150                             ItemWidth
151                         .Height = Calendar.Cells(CalRow + ItemNumb, CalCol).Height
152                         .TextFrame2.TextRange.Text = ItemTime & ": " & ItemName 'Text
153                         inside shape
154                         .Fill.ForeColor.RGB = ItemColor 'Set Item Color
155                         .OnAction = "Calendar_Item_Select" 'Macro to select item
156                     End With
157                     If ItemNumb >= DayItemCount Then
158                         ItemNumb = 1 'Reset
159                         ItemWidth = 1 'Reset Item Width
160                     Else
161                         ItemNumb = ItemNumb + 1 'Increment by 1
162                     End If
163                     If ItemNumb <= 5 Then
164                         ItemLeft = 0 'Set Left Position
165                         ItemTop = 0
166                     Else 'Item Number from 6 to 10
167                         ItemLeft = Calendar.Cells(CalRow + ItemNumb, CalCol).Width / 2
168                         ItemTop = -5

```

```
161 | 1 2 3 4 5 6 7  
162 | | | | | | | End If  
163 | | | | | | | End If  
164 | | | | | | | Next CalCol  
165 | | | | | | | Next CalRow  
166 | | | | | | | Next ResultRow  
167 | | | | | | | End With  
168 | | | | | | | Application.ScreenUpdating = True  
169 | | | | | | | End Sub  
170 | | | | | | | Sub Calendar_ThisMonth()  
171 | | | | | | | With Calendar  
172 | | | | | | | .Range("B5" ).Value = Month(Date) 'Set Current Month #  
173 | | | | | | | .Range("B2" ).Value = Admin.Range("Years" ).Find(Year(Date), , xlFormulas,  
174 | | | | | | | xlWhole).Row - 3  
175 | | | | | | | Calendar_Refresh 'Refresh Calendar  
176 | | | | | | | End With  
177 | | | | | | | End Sub  
178 | | | | | | | Sub PrintCalendar()  
179 | | | | | | | Calendar.PrintOut , , , False, True, , , False  
180 | | | | | | | End Sub
```

```

1 Option Explicit
2
3 Dim ItemRow As Long, ItemCol As Long
4
5 Sub Item_Delete()
6     If MsgBox("Are you sure you want to delete this Calendar Item?" , vbYesNo, "Delete
7         Item" ) = vbNo Then Exit Sub
8     With Calendar
9         If .Range("B10" ).Value = Empty Then GoTo NotSaved
10        ItemRow = .Range("B10" ).Value 'Item row
11        ItemsDB.Range(ItemRow & ":" & ItemRow).EntireRow.Delete 'Delete R0w
12        NotSaved:
13        Item_New
14        Calendar_Refresh 'Refresh calendar
15    End With
16 End Sub
17
18 Sub Item_Load()
19     With Calendar
20         If .Range("B10" ).Value = Empty Then
21             MsgBox "Please select a correct calendar item to load"
22             Exit Sub
23         End If
24         ItemRow = .Range("B10" ).Value 'Item Row
25         For ItemCol = 2 To 6
26             .Range("M" & ItemCol + 2).Value = ItemsDB.Cells(ItemRow, ItemCol).Value
27             'Bring Over data
28         Next ItemCol
29     End With
30 End Sub
31
32 Sub Item_New()
33     Calendar.Range("B9,M4:M9" ).ClearContents 'Clear Out
34 End Sub
35
36 Sub Item_SaveUpdate()
37     With Calendar
38         If Application.WorksheetFunction.CountA(.Range("M4:M6" )) < 3 Then
39             MsgBox "Please make sure to add in an Item Name, Date & Time"
40             Exit Sub
41         End If
42         If .Range("B10" ).Value = Empty Then 'New Item
43             ItemRow = ItemsDB.Range("A99999" ).End(xlUp).Row + 1 'First Avail. Row
44             .Range("B9" ).Value = .Range("B11" ).Value 'Set Item ID
45             ItemsDB.Range("A" & ItemRow).Value = .Range("B9" ).Value 'Item ID
46         Else 'Existing Item
47             ItemRow = .Range("B10" ).Value 'Existing Item Row
48         End If
49         For ItemCol = 2 To 6
50             ItemsDB.Cells(ItemRow, ItemCol).Value = .Range("M" & ItemCol + 2).Value
51             'Save/ Update data
52         Next ItemCol
53         If .Range("B12" ).Value = False Then 'Refresh & Set Saved Message except for
54             recurring
55             Calendar_Refresh 'Refresh calendar
56             MsgBox "Item Saved"
57         End If
58     End With
59 End Sub

```

```
1 Option Explicit
2
3 Sub CreateRecurringItem()
4     Dim ItemDate As Date
5     Dim Freq As String, FreqQty As Long, ItemQty As Long, ItemNumb As Long
6     With Calendar
7         If .Range("B10").Value = Empty Then
8             MsgBox "Please make sure to save this item before creating recurring calendar item"
9             Exit Sub
10        End If
11        If Application.WorksheetFunction.CountA(.Range("M12:M15")) < 4 Then
12            MsgBox "Please make sure to fill in all recurring fields"
13            Exit Sub
14        End If
15        FreqQty = .Range("M12").Value 'Frequency Qty
16        Freq = .Range("M13").Value 'Frequency
17        ItemDate = .Range("M14").Value 'Start Date
18        ItemQty = .Range("M15").Value ' Number of Calendar Items
19        .Range("B12").Value = True 'Set Recurring to True
20        For ItemNumb = 1 To ItemQty
21            .Range("M5").Value = ItemDate 'Cal. Item Date
22            .Range("B9").ClearContents 'Clear Existing Item ID
23            Item_SaveUpdate 'Save Calendar Item
24            Select Case Freq
25                Case Is = "Minute(s)"
26                    ItemDate = DateAdd("n", FreqQty, ItemDate)
27                Case Is = "Hour(s)"
28                    ItemDate = DateAdd("h", FreqQty, ItemDate)
29                Case Is = "Day(s)"
30                    ItemDate = DateAdd("d", FreqQty, ItemDate)
31                Case Is = "Week(s)"
32                    ItemDate = DateAdd("ww", FreqQty, ItemDate)
33                Case Is = "Month(s)"
34                    ItemDate = DateAdd("m", FreqQty, Now)
35            End Select
36        Next ItemNumb
37        .Range("B12").Value = False 'Set Recurring to false
38        Calendar_Refresh 'Refresh calendar
39        MsgBox ItemQty & " calendar items have been created"
40    End With
41 End Sub
```



```
1 Option Explicit
2
3 Sub SendAppointmentToOutlook()
4     Dim ItemDate As Date, ItemTime As Date
5     Dim ItemName As String, Notes As String
6     Dim Dur As Double
7     With Calendar
8         ItemName = .Range("M4").Value 'item Name
9         ItemDate = .Range("M5").Value 'Item Date
10        ItemTime = .Range("M6").Value 'Time
11        Dur = .Range("M7").Value * 24 * 60 'Duration (in minutes)
12        Notes = .Range("M8").Value 'Notes
13    End With
14
15
16    Dim OutApp As Object, OutAppt As Object
17    Set OutApp = CreateObject("Outlook.Application")
18    Set OutAppt = OutApp.CreateItem(1)
19    With OutAppt
20        .Subject = ItemName
21        .Start = ItemDate + ItemTime
22        .Duration = Dur
23        .ReminderSet = True
24        .ReminderMinutesBeforeStart = 15
25        .Body = Notes
26        .Save
27        .Display
28    End With
29    Set OutApp = Nothing
30    Set OutAppt = Nothing
31    MsgBox "Calendar Item has been sent to Outlook"
32 End Sub
```

A

ActiveCell, 6
 Add, 9
 Admin, 6, 8, 10
 Admin_SetPlatformColor, 6
 AdvancedFilter, 9
 Application, 6-12
 Apply, 9

B

Body, 13

C

CalCol, 7, 9, 10
 Calendar, 7-13
 Calendar_CheckForMove, 7, 8
 Calendar_Item_Select, 7
 Calendar_NextMonth, 8
 Calendar_PrevMonth, 8
 Calendar_Refresh, 7, 8, 10-12
 Calendar_ThisMonth, 10
 Caller, 6-8
 CalRow, 7, 9, 10
 Cells, 7, 9, 11
 Clear, 9
 ClearContents, 11, 12
 Color, 6, 8
 Column, 7
 CopyToRange, 9
 CountA, 11, 12
 CountDelay, 7
 CountIf, 9
 CountLarge, 2
 CreateItem, 13
 CreateObject, 13
 CreateRecurringItem, 12
 CriteriaRange, 9

D

DataOption, 9
 DateAdd, 12
 DayItemCount, 7, 9
 Delete, 8, 11
 DestCol, 7
 DestRow, 7
 Display, 13
 DoEvents, 7
 Duplicate, 9
 Dur, 13
 Duration, 13

E

Empty, 7, 11, 12
 EntireRow, 11
 Explicit, 2-7, 11-13

F

Fill, 6, 8, 9
 Find, 10
 ForeColor, 6, 8, 9
 Format, 9
 Freq, 12
 FreqQty, 12

H

Height, 9

I

InStr, 8

Interior, 6, 8
 Intersect, 2
 Item_Delete, 11
 Item_Load, 8, 11
 Item_New, 11
 Item_SaveUpdate, 7, 11, 12
 ItemCol, 7, 11
 ItemColor, 7-9
 ItemDate, 7, 9, 12, 13
 ItemDate_Results, 9
 ItemID, 7-9
 ItemLeft, 7, 9
 ItemName, 7, 9, 13
 ItemNumb, 7-9, 12
 ItemQty, 12
 ItemRow, 7, 11
 ItemsDB, 9, 11
 ItemShp, 7, 8
 ItemTime, 7, 9, 13
 ItemTop, 7, 9
 ItemWidth, 7-9

K

Key, 9

L

LastResultRow, 7, 9
 LastRow, 7, 9
 Left, 2, 7-9

M

Month, 10
 MsgBox, 7, 8, 11-13
 msoCTrue, 2
 msoFalse, 2, 6

N

Name, 8, 9
 NoSort, 9
 Notes, 13
 NotSaved, 11
 Now, 12

O

OnAction, 9
 Order, 9
 OutApp, 13
 OutAppt, 13

P

PrintCalendar, 10
 PrintOut, 10

R

Range, 2, 7-13
 ReminderMinutesBeforeStart, 13
 ReminderSet, 13
 Replace, 7
 ResultRow, 7, 9, 10
 RGB, 6, 8, 9
 Row, 2, 7, 9-11

S

Save, 13
 ScreenUpdating, 9, 10
 SendAppointmentToOutlook, 13
 SetRange, 9
 Shape, 7
 Shapes, 2, 6-9
 Sort, 9

SortFields, 9
 SortOn, 9
 Start, 13
 Subject, 13

T

Target, 2
 Text, 9
 TextFrame2, 9
 TextRange, 9
 Top, 2, 7-9
 TopLeftCell, 7

U

Unique, 9

V

Value, 7-13
 vbNo, 11
 vbYesNo, 11
 Visible, 2, 6

W

Width, 9
 Worksheet_SelectionChange, 2
 WorksheetFunction, 9, 11, 12

X

xlAscending, 9
 xlFilterCopy, 9
 xlFormulas, 10
 xlSortNormal, 9
 xlSortOnValues, 9
 xlUp, 9, 11
 xlWhole, 10

Y

Year, 10

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