



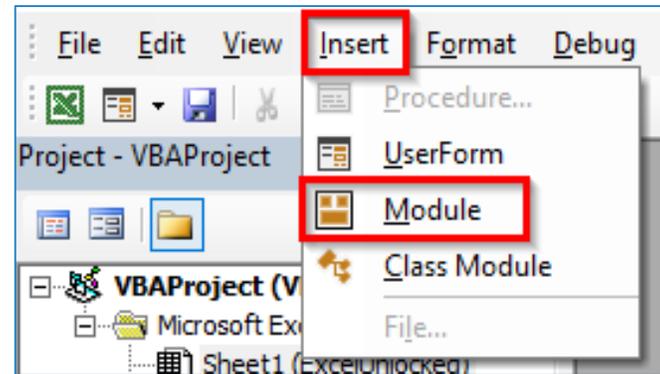
Everyday Use Excel VBA Codes

By : ExcelUnlocked.com

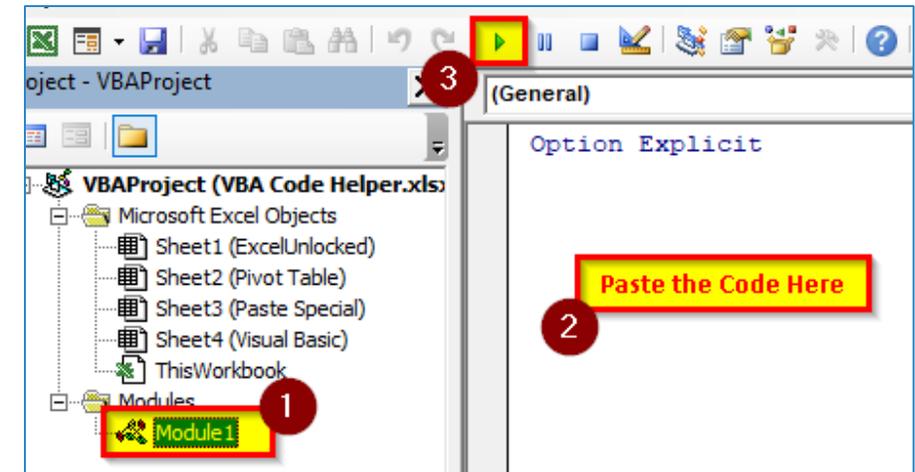


HOW TO READ AND USE THIS DOCUMENT ???

- ▶ Each of the upcoming slides contains VBA codes that would help you to accomplish different tasks in excel.
- ▶ You simply need to copy the VBA Codes and paste it in the VB Editor Window (like the way it is described below):
 - Use keyboard shortcut (Alt + F11) to open the Visual Basic Editor (VBE) Window.
 - In the VBE window, go to 'Insert' tab > 'Module' (see image)
 - Now, copy the code and paste VBA Code in the Code Window.
 - Once your code is pasted, simply use the 'Run/ User Form' button as shown in the image to the right.



Go To Insert > Module



Paste the VBA Code in window and Run it.

INDEX ►

- **Hide all the worksheets except active worksheets in Excel**
- **Unhide or show all the worksheets at one go**
- **Unprotect all the worksheets in excel at one go**
- **Protect all the worksheets in excel at one go**
- **Show or display or turn on the gridlines for all the worksheets at one go**
- **Hide or remove or turn off the gridlines for all the worksheets at one go**
- **Reverse the order of the excel worksheets**
- **Arrange or sort the worksheets in descending order (Z → A)**
- **Arrange or sort the worksheets in ascending order (A → Z)**
- **Create Index of worksheets (without hyperlink)**
- **Create index of worksheets (with hyperlink to sheets)**
- **Create worksheets based on excel list (without hyperlink)**
- **Create worksheets based on excel list (with hyperlink to sheets)**
- **Create worksheets based on excel list (if worksheet is not already created)**
- **Delete worksheets which are mentioned in the excel list**
- **Split each excel worksheet into excel file or workbook**
- **Separate text and number from cell containing both text and numbers**
- **Create folder based on excel list**
- **Create new folder for each worksheet in excel with worksheet name**

HIDE ALL THE WORKSHEETS EXCEPT ACTIVE WORKSHEET IN EXCEL

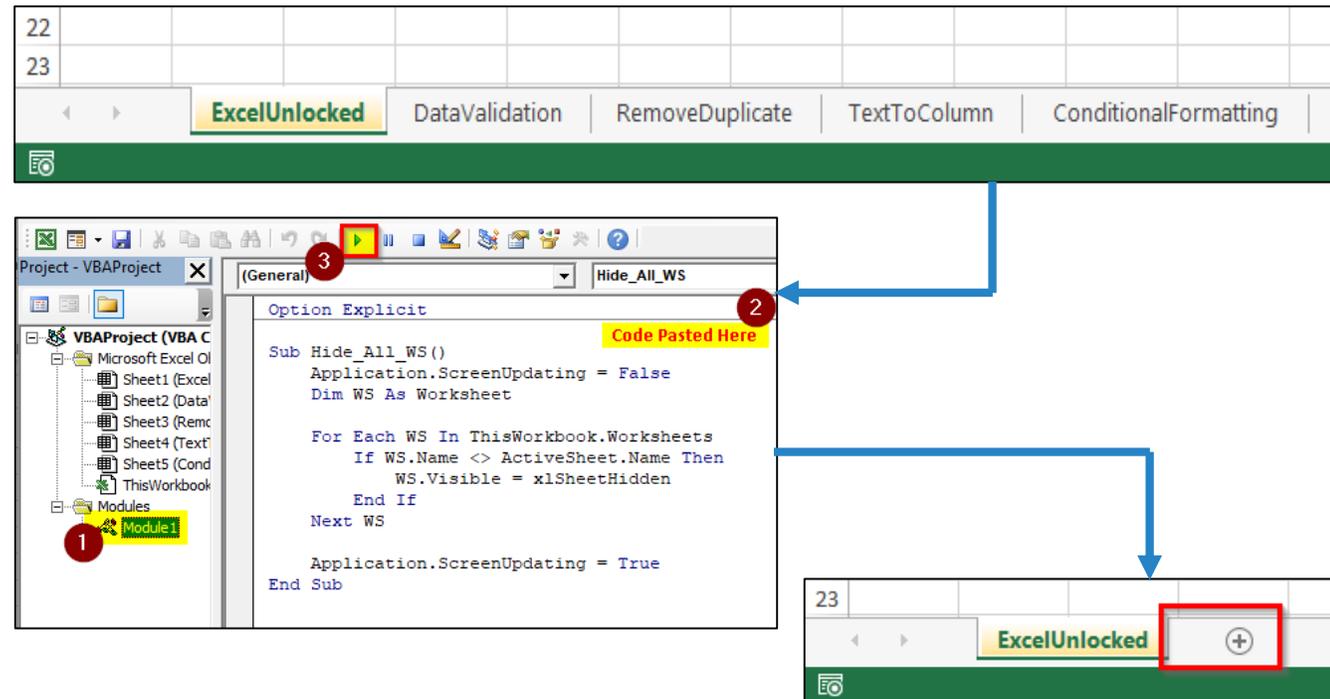
```
Sub Hide_All_WS ()
    Application.ScreenUpdating = False
    Dim WS As Worksheet

    For Each WS In ThisWorkbook.Worksheets
        If WS.Name <> ActiveSheet.Name Then
            WS.Visible = xlSheetHidden
        End If
    Next WS

    Application.ScreenUpdating = True
End Sub
```

Additional Points for Reading

- ❖ Open the Workbook in which you want to hide worksheets except active sheet.
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As soon as you run this code, excel would hide all the worksheets other than active worksheet.



The screenshot illustrates the process of running the VBA code. It shows the VBA Editor window with the 'Project - VBAProject' pane on the left, where 'Module1' is selected (indicated by a red circle '1'). The 'Code' pane on the right shows the 'Hide_All_WS' code being pasted into the 'Module1' window (indicated by a red circle '2' and a yellow 'Code Pasted Here' message). The 'Run' button (a green play icon) is highlighted with a red circle '3'. Below the VBA Editor, a small inset shows the Excel worksheet with the 'ExcelUnlocked' sheet selected and a red box around the '+' button in the sheet tab bar, indicating the next step in the process.

UNHIDE OR SHOW ALL THE WORKSHEETS IN EXCEL AT ONE GO

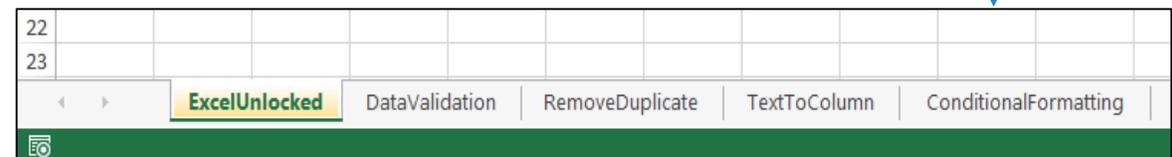
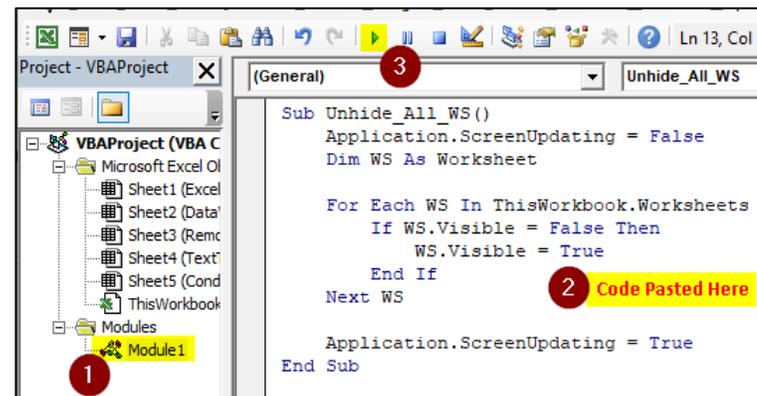
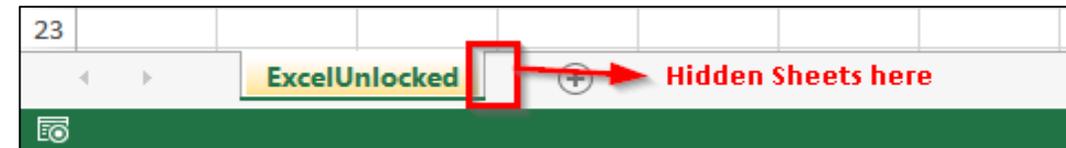
```
Sub Unhide_All_WS()
    Application.ScreenUpdating = False
    Dim WS As Worksheet

    For Each WS In ThisWorkbook.Worksheets
        If WS.Visible = False Then
            WS.Visible = True
        End If
    Next WS

    Application.ScreenUpdating = True
End Sub
```

Additional Points for Reading

- ❖ Open the Workbook in which you want to unhide the worksheets.
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As soon as you run this code, excel would show/unhide all the hidden worksheets.



UNPROTECT ALL THE WORKSHEETS IN EXCEL AT ONE GO USING VBA

```
Sub Unprotect_All_Sh()  
Application.ScreenUpdating = False  
Dim Pwd As String  
Dim WS As Worksheet
```

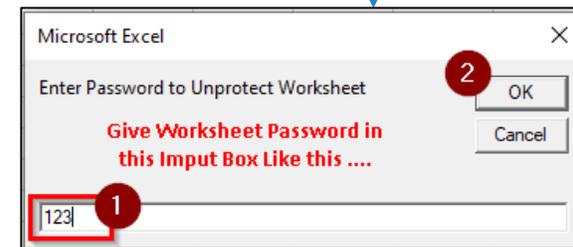
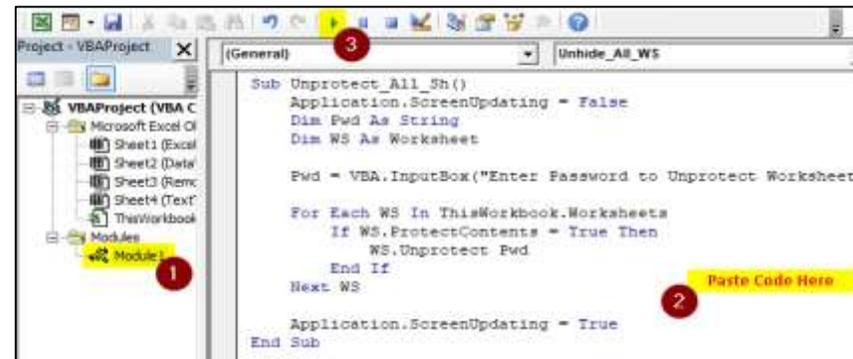
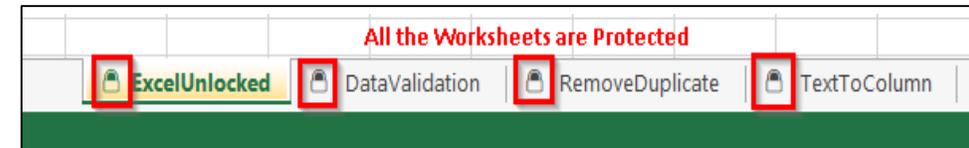
```
Pwd = VBA.InputBox("Enter Password to Unprotect Worksheet")
```

```
For Each WS In ThisWorkbook.Worksheets  
If WS.ProtectContents = True Then  
WS.Unprotect Pwd  
End If  
Next WS
```

```
Application.ScreenUpdating = True  
End Sub
```

Additional Points for Reading

- ❖ Open the Workbook in which you want to unprotect all the worksheets.
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As soon as you run this code, excel would unprotect all the protected worksheets quickly.



PROTECT ALL THE WORKSHEETS IN EXCEL AT ONE GO USING VBA

```
Sub Protect_All_Sh()
    Application.ScreenUpdating = False
    Dim Pwd As String
    Dim WS As Worksheet

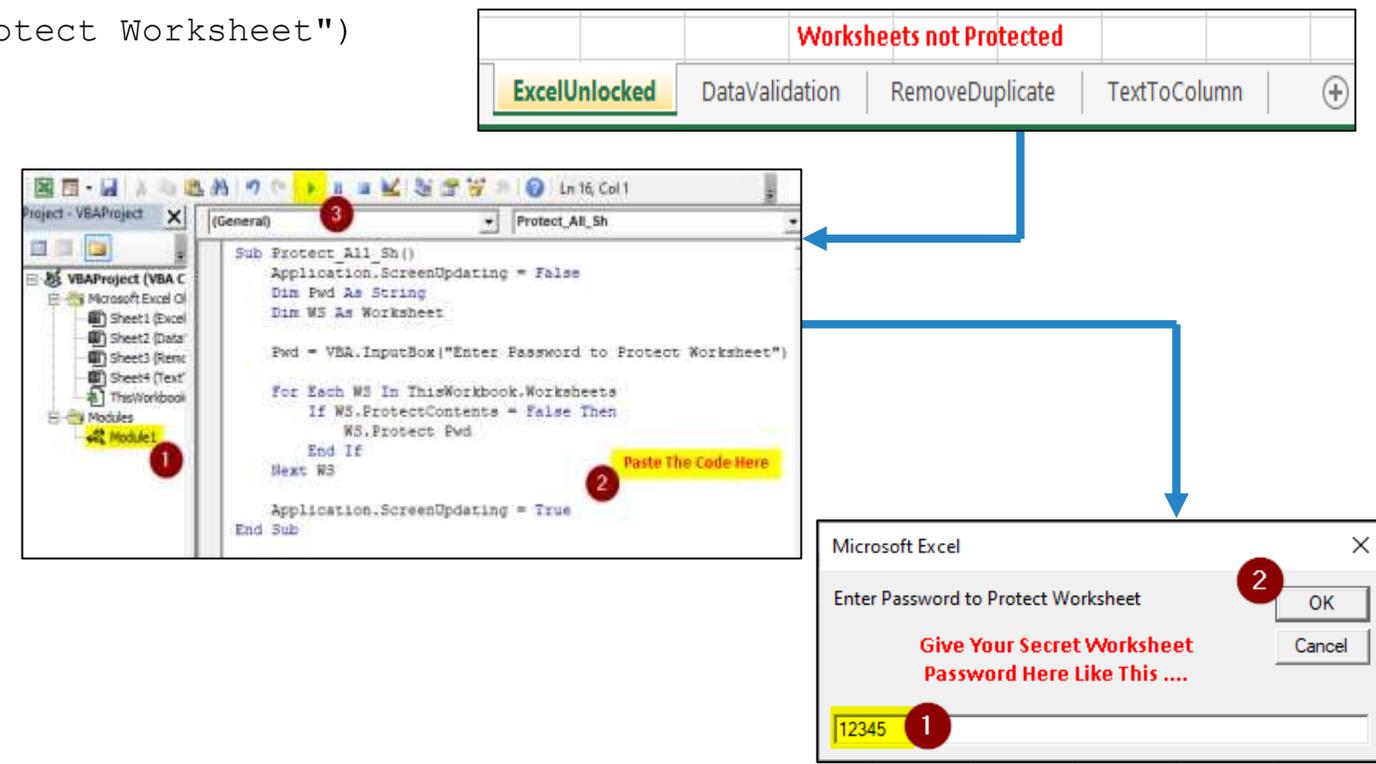
    Pwd = VBA.InputBox("Enter Password to Protect Worksheet")

    For Each WS In ThisWorkbook.Worksheets
        If WS.ProtectContents = False Then
            WS.Protect Pwd
        End If
    Next WS

    Application.ScreenUpdating = True
End Sub
```

Additional Points for Reading

- ❖ Open the Workbook in which you want to unprotect all the worksheets.
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As soon as you run this code, excel would unprotect all the protected worksheets quickly.



The screenshot shows the VBA Editor window with the code from the previous block. A yellow box labeled '1' points to the 'Module1' in the Project Explorer. A yellow box labeled '2' points to the code area with the text 'Paste The Code Here'. A red box labeled '3' points to the 'Protect_All_Sh' subprocedure in the dropdown menu. To the right, a ribbon tab labeled 'ExcelUnlocked' is shown with a red text overlay 'Worksheets not Protected'. Below it, the 'Microsoft Excel' password dialog box is shown with a red box labeled '2' pointing to the 'OK' button and a yellow box labeled '1' pointing to the password input field containing '12345'. Blue arrows indicate the flow from the ribbon tab to the VBA Editor and then to the password dialog box.

SHOW OR DISPLAY OR TURN ON GRIDLINES FOR ALL THE WORKSHEETS AT ONE GO

```
Sub Show_Gridlines ()
    Application.ScreenUpdating = False
    Dim WS As Worksheet

    For Each WS In ThisWorkbook.Worksheets
        WS.Activate
        ActiveWindow.DisplayGridlines = True
    Next WS

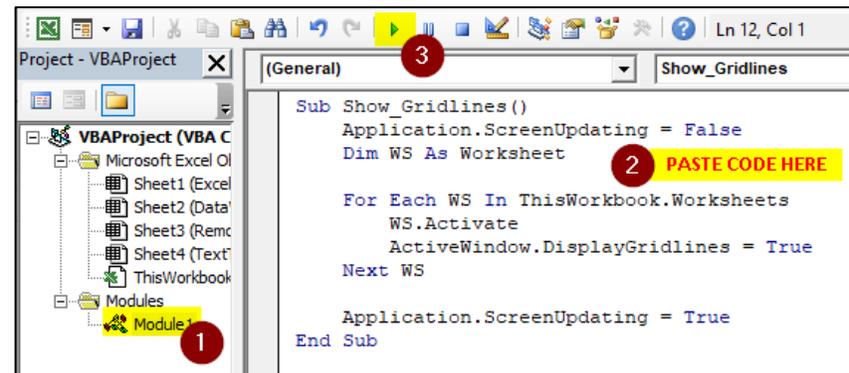
    Application.ScreenUpdating = True
End Sub
```

Additional Points for Reading

- ❖ Open the workbook in which you want to show or display the excel gridlines.
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As soon as you run this code, excel would display all the hidden gridlines.

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

GRIDLINES ARE NOT VISIBLE IN ALL THE WORKSHEET



	A	B	C	D	E
1					
2					
3					
4					
5					

GRIDLINES ARE VISIBLE IN ALL THE WORKSHEETS.

HIDE OR REMOVE OR TURN OFF GRIDLINES FOR ALL THE WORKSHEETS AT ONE GO

```
Sub Hide_Gridlines ()
    Application.ScreenUpdating = False
    Dim WS As Worksheet

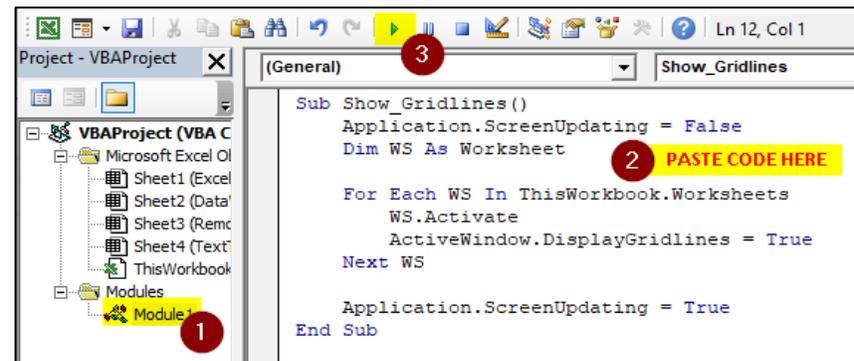
    For Each WS In ThisWorkbook.Worksheets
        WS.Activate
        ActiveWindow.DisplayGridlines = False
    Next WS

    Application.ScreenUpdating = True
End Sub
```

Additional Points for Reading

- ❖ Open the workbook in which you want to hide the excel gridlines from all the worksheets.
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As soon as you run this code, excel would hide all the visible excel gridlines.

	A	B	C	D	E
1					
2					
3					
4					
5					



The screenshot shows the VBA Editor interface. On the left, the Project Explorer shows a VBA Project with a folder 'Modules' containing 'Module1' (1). The main editor shows the code for 'Show_Gridlines' (3). A yellow highlight is placed over the text 'PASTE CODE HERE' (2) in the code editor.

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

REVERSE THE ORDER OF WORKSHEETS (FIRST SHEET BECOMES LAST AND SO ON)

```

Sub Revrs_WS_Order()
    Application.ScreenUpdating = False
    Dim WS_Cnt As Long
    Dim i As Long
    Dim r As Long

    WS_Cnt = Worksheets.Count
    r = 1

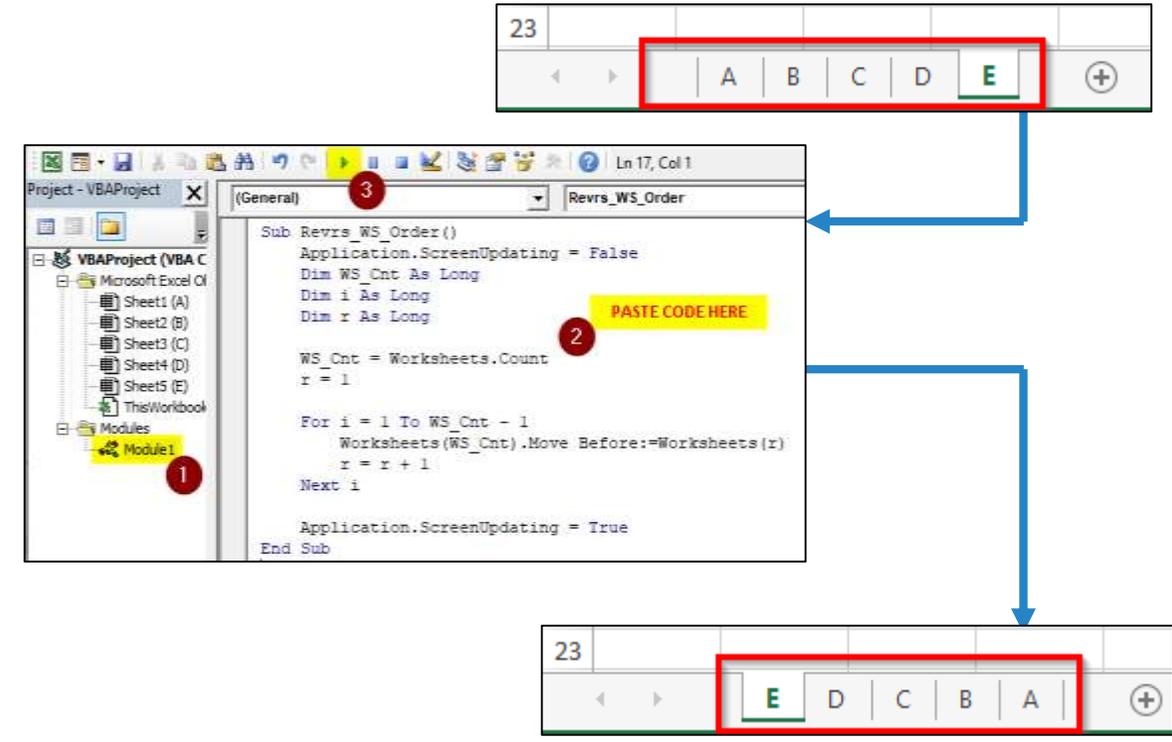
    For i = 1 To WS_Cnt - 1
        Worksheets(WS_Cnt).Move Before:=Worksheets(r)
        r = r + 1
    Next i

    Application.ScreenUpdating = True
End Sub

```

Additional Points for Reading

- ❖ Open the workbook in which you want to reverse the worksheet order.
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As soon as you run this code, excel would move first sheet to last and likewise arrange other worksheets.



ARRANGE OR SORT THE WORKSHEETS IN DESCENDING ORDER (Z TO A)

```

Sub Sort_WS_Descending_ZA()
    Application.ScreenUpdating = False
    Dim Sh_Cnt As Long
    Dim i As Long
    Dim r As Long
    Dim z As Long

    Sh_Cnt = Worksheets.Count
    z = 1

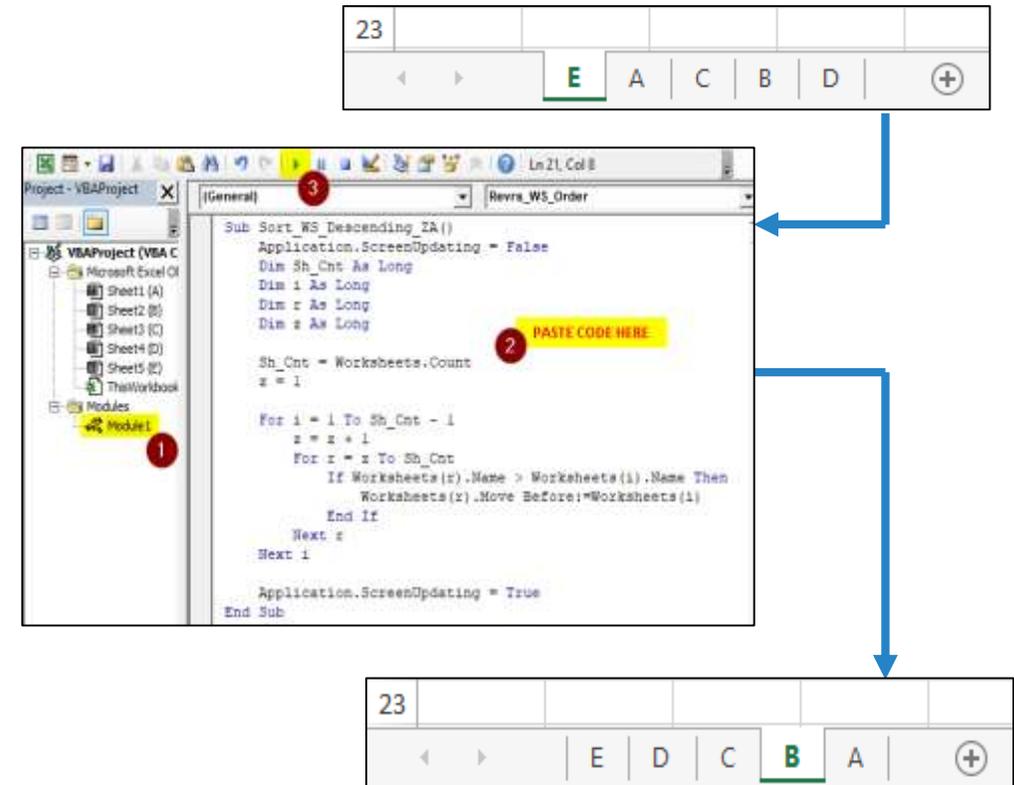
    For i = 1 To Sh_Cnt - 1
        z = z + 1
        For r = z To Sh_Cnt
            If Worksheets(r).Name > Worksheets(i).Name Then
                Worksheets(r).Move Before:=Worksheets(i)
            End If
        Next r
    Next i

    Application.ScreenUpdating = True
End Sub

```

Additional Points for Reading

- ❖ Open the workbook in which you want to arrange the worksheets in descending order.
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As soon as you run this code, excel would arrange the worksheets from Z to A (based on the sheet names)



The screenshot illustrates the process of running the VBA code. It shows the VBE Editor window with the code pasted into the 'Module1' window. A yellow box labeled 'PASTE CODE HERE' is positioned over the code. The 'Project - VBAProject' window shows the workbook structure with sheets A, B, C, D, and E. The 'Worksheet: Module1' window shows the code being executed. The 'Worksheet: Revra_WS_Order' window shows the result of the code, where the sheets are now ordered E, D, C, B, A. The 'Worksheet: Revra_WS_Order' window also shows the code being executed. The 'Worksheet: Revra_WS_Order' window shows the result of the code, where the sheets are now ordered E, D, C, B, A. The 'Worksheet: Revra_WS_Order' window shows the result of the code, where the sheets are now ordered E, D, C, B, A.

ARRANGE OR SORT THE WORKSHEETS IN ASCENDING ORDER (A TO Z)

```

Sub Sort_WS_Ascending_AZ()
    Application.ScreenUpdating = False
    Dim Sh_Cnt As Long
    Dim i As Long
    Dim r As Long
    Dim z As Long

    Sh_Cnt = Worksheets.Count
    z = 1

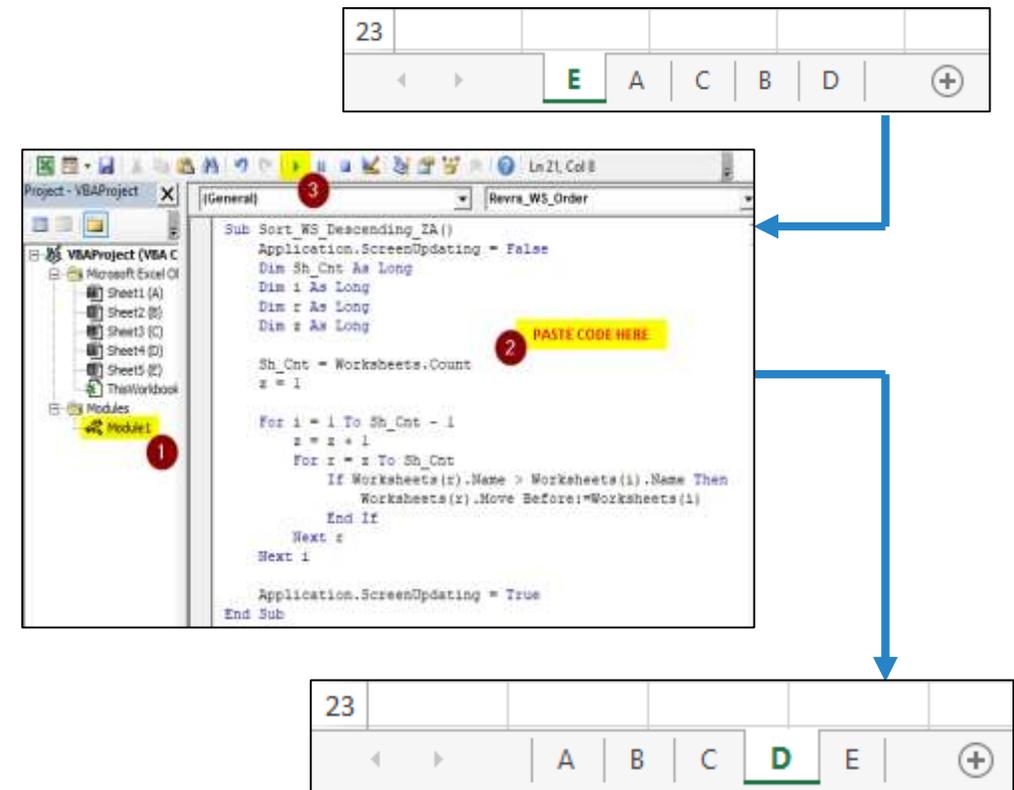
    For i = 1 To Sh_Cnt - 1
        z = z + 1
        For r = z To Sh_Cnt
            If Worksheets(r).Name < Worksheets(i).Name Then
                Worksheets(r).Move Before:=Worksheets(i)
            End If
        Next r
    Next i

    Application.ScreenUpdating = True
End Sub

```

Additional Points for Reading

- ❖ Open the workbook in which you want to arrange the worksheets in ascending order.
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As soon as you run this code, excel would arrange the worksheets from A to Z (based on the sheet names)



The screenshot illustrates the process of running the VBA code. It shows the VBE Editor window with the code pasted into the 'Module1' window. The worksheet tabs at the bottom of the Excel window show the order: E, A, C, B, D. The VBE Editor window has a yellow highlight on 'PASTE CODE HERE' and a red circle around the code. The worksheet tabs at the bottom of the Excel window show the order: A, B, C, D, E.

CREATE INDEX OF WORKSHEETS USING VBA (WITHOUT HYPERLINK)

```

Sub Index_Without_Link()
    Application.ScreenUpdating = False
    Dim WS_Cnt As Long
    Dim i As Long
    Dim Sh_Name As String

    Sh_Name = InputBox("Give a Name to Index Worksheet")
    WS_Cnt = Worksheets.Count
    Worksheets.Add Before:=Sheets(1)
    Worksheets(1).Name = Sh_Name

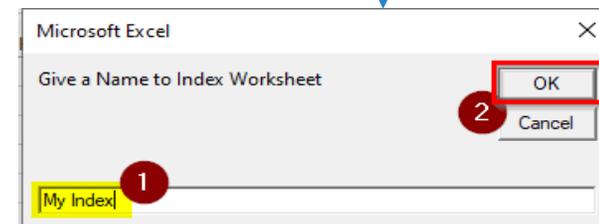
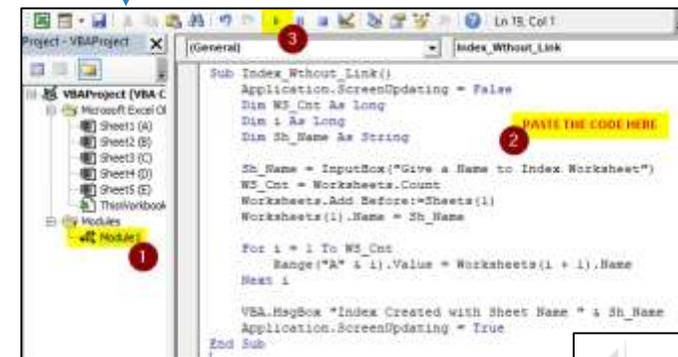
    For i = 1 To WS_Cnt
        Sheets(1).Range("A" & i).Value = Worksheets(i + 1).Name
    Next i

    VBA.MsgBox "Index Created with Sheet Name " & Sh_Name
    Application.ScreenUpdating = True
End Sub

```

Additional Points for Reading

- ❖ Open the workbook in which you want to create index of worksheets (without hyperlink).
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ You would be asked to write the Index Worksheet name. Enter the name of your choice and press OK to created an Index.



	A	B
1	A	
2	B	WORK SHEET
3	C	INDEX
4	D	CREATED
5	E	
6		

CREATE INDEX OF WORKSHEETS USING VBA (WITH HYPERLINK TO SHEETS)

```

Sub Index_With_Link()
Application.ScreenUpdating = False
Dim WS_Cnt As Long
Dim i As Long
Dim Sh_Name As String

Sh_Name = InputBox("Give a Name to Index Worksheet")
WS_Cnt = Worksheets.Count
Worksheets.Add Before:=Sheets(1)
Worksheets(1).Name = Sh_Name

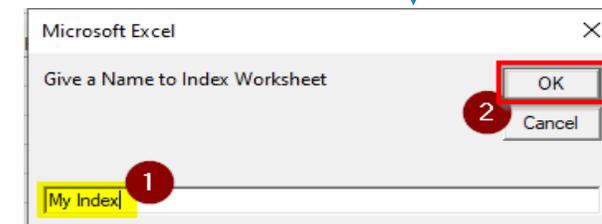
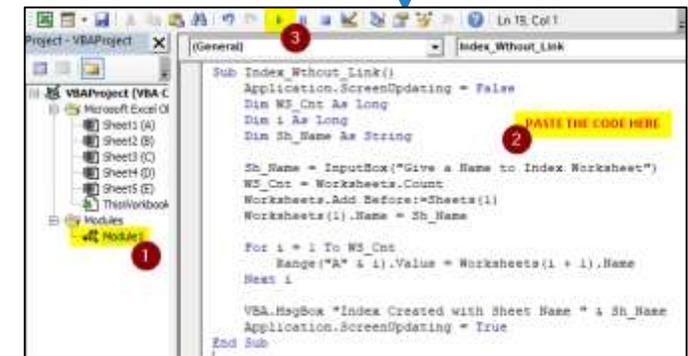
For i = 1 To WS_Cnt
    Sheets(1).Range("A" & i).Value = Worksheets(i + 1).Name
    Sheets(1).Range("A" & i).Hyperlinks.Add _
        Anchor:=Sheets(1).Range("A" & i), Address:="", _
        SubAddress:="" & Sheets(1).Range("A" & i).Value & "!A1"
Next i

VBA.MsgBox "Index Created with Sheet Name - " & Sh_Name
Application.ScreenUpdating = True
End Sub

```

Additional Points for Reading

- ❖ Open the workbook in which you want to create index of worksheets (with hyperlink).
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ You would be asked to write the Index Worksheet name. Enter the name of your choice and press OK to created an Index.



	A	B
1	A	Index Created for All Worksheets along with Hyperlink
2	B	
3	C	
4	D	
5	E	
6		

CREATE WORKSHEETS BASED ON EXCEL LIST (WITHOUT HYPERLINK)

```
Sub Create_WS_Wthot_Link()
    Application.ScreenUpdating = False
    Dim LstCell As Long
    Dim i As Long

    LstCell = ActiveSheet.Range("A1").End(xlDown).Row

    For i = 1 To LstCell
        Worksheets.Add After:=Worksheets(Sheets.Count)
        ActiveSheet.Name = Worksheets(1).Range("A" & i).Value
    Next i

    Application.ScreenUpdating = True
End Sub
```

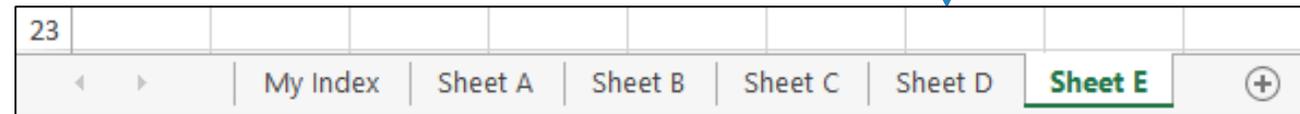
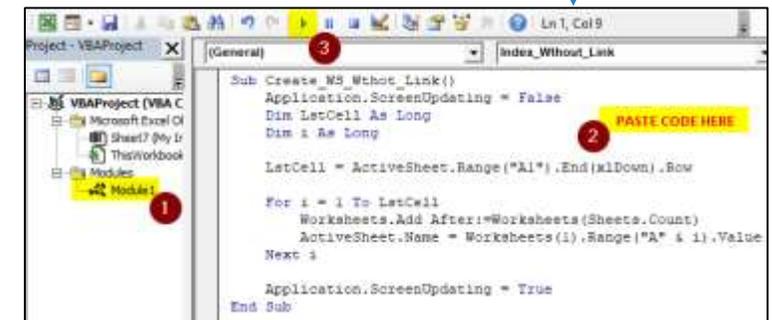
Make sure that the excel list is in cell A1, Also the worksheet containing list should be an active worksheet.

Additional Points for Reading

- ❖ Open the workbook in which you want to create worksheets based on excel list (without hyperlink).
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ You would notice that excel creates different worksheets for each of the names in excel sheet.

	A	B	C	D
1	Sheet A			
2	Sheet B			
3	Sheet C			
4	Sheet D			
5	Sheet E			
6				

Excel List Based on which we want to create Worksheets.
Make sure that Names starts from Cell A1



CREATE WORKSHEETS BASED ON EXCEL LIST (WITH HYPERLINK TO SHEETS)

```
Sub Create_WS_With_Link()
    Application.ScreenUpdating = False
    Dim LstCell As Long
    Dim i As Long

    LstCell = Range("A1").End(xlDown).Row

    For i = 1 To LstCell
        Worksheets.Add After:=Worksheets(Sheets.Count)
        ActiveSheet.Name = Worksheets(1).Range("A" & i).Value
        Worksheets(1).Range("A" & i).Hyperlinks.Add _
            Anchor:=Worksheets(1).Range("A" & i), Address:="", _
            SubAddress:="" & Worksheets(1).Range("A" & i) _
            .Value & "!A1"
    Next i

    Worksheets(1).Activate
    Application.ScreenUpdating = True
End Sub
```

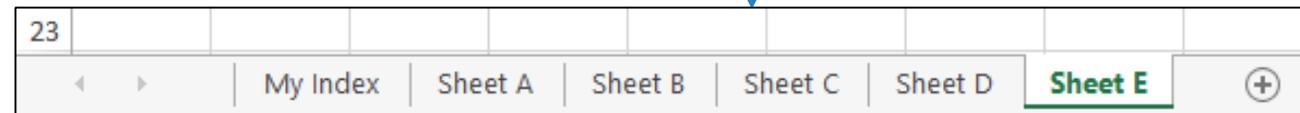
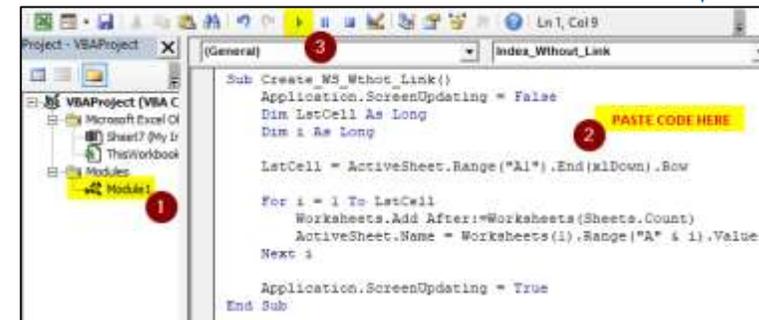
Make sure that the excel list is in cell A1, Also the worksheet containing list should be an active worksheet.

Additional Points for Reading

- ❖ Open the workbook in which you want to create worksheets based on excel list (with hyperlink).
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ You would notice that excel creates different worksheets for each of the names in excel sheet. It also creates hyperlink in Index.

	A	B	C	D
1	Sheet A			
2	Sheet B			
3	Sheet C			
4	Sheet D			
5	Sheet E			
6				

Excel List Based on which we want to create Worksheets. Make sure that Names starts from Cell A1



CREATE WORKSHEETS BASED ON EXCEL LIST IF WORKSHEET IS NOT ALREADY CREATED

```
Sub Crt_Sh_If_DoentExst ()
Application.ScreenUpdating = False
Dim i As Long
Dim WS As Worksheet
Dim LstCell As Long

LstCell = Worksheets(1).Range("A1").End(xlDown).Row

For i = 1 To LstCell
    For Each WS In ThisWorkbook.Worksheets
        If Worksheets(1).Range("A" & i).Value = WS.Name _
            Then GoTo NextOne
    Next WS
    Worksheets.Add After:=Worksheets(Worksheets.Count)
    ActiveSheet.Name = Worksheets(1).Range("A" & i).Value
NextOne:
    Next i

Worksheets(1).Activate
Application.ScreenUpdating = True
End Sub
```

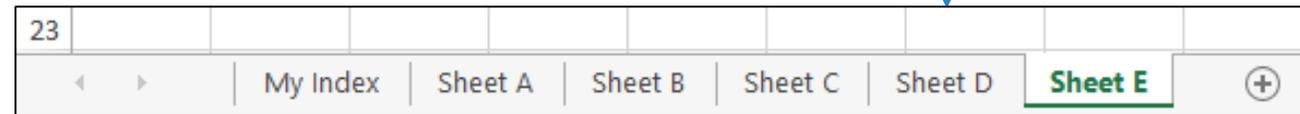
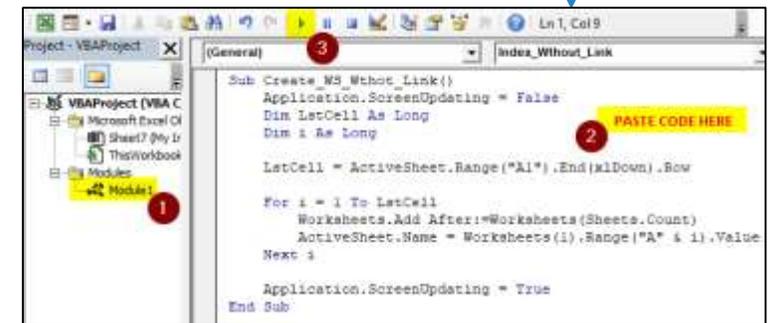
Make sure that the excel list is in cell A1, Also the worksheet containing list should be an active worksheet.

Additional Points for Reading

- ❖ Open the workbook in which you want to create worksheets based on excel list (without hyperlink).
- ❖ Copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ You would notice that excel creates different worksheets for each of the names in excel sheet.

	A	B	C	D
1	Sheet A			
2	Sheet B			
3	Sheet C			
4	Sheet D			
5	Sheet E			
6				

Excel List Based on which we want to create Worksheets. Make sure that Names starts from Cell A1



DELETE WORKSHEET WHICH ARE MENTIONED IN AN EXCEL LIST

```
Sub Del_WS_From_List()
    Application.ScreenUpdating = False
    Application.DisplayAlerts = False
    Dim WS As Worksheet
    Dim i As Long
    Dim LstCell As Long

    LstCell = Worksheets(1).Range("A1").End(xlDown).Row

    For i = 1 To LstCell
        For Each WS In ThisWorkbook.Worksheets
            If Worksheets(1).Range("A" & i).Value = WS.Name
                Then WS.Delete
            Next WS
        Next i

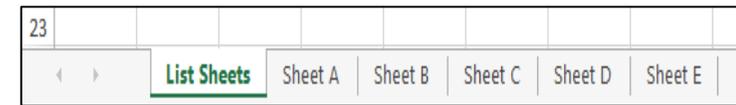
        Application.DisplayAlerts = True
        Application.ScreenUpdating = True
    End Sub
```

Make sure that the excel list is in cell A1, Also the worksheet containing list should be an active worksheet.

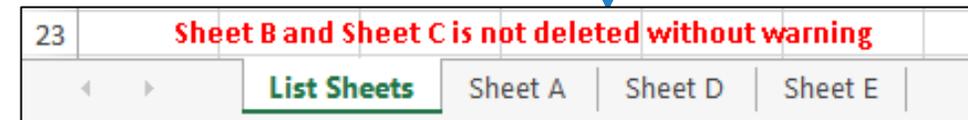
Additional Points for Reading

- ❖ You have an excel list containing names of the worksheets, and you want to delete those worksheets.
- ❖ Simply, copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As a result, excel would delete the worksheet based on the excel list

	A	B
1	Sheet B	
2	Sheet C	
3		



The purpose here is to delete 'Sheet B' and 'Sheet C' and keep the others.



SPLIT EACH EXCEL WORKSHEET INTO SEPARATE EXCEL FILE OR WORKBOOK

```

Sub Split_WS_Sep_File()
    Application.ScreenUpdating = False
    Application.DisplayAlerts = False
    Dim WS As Worksheet
    Dim Folder As String

    Folder = ThisWorkbook.Path

    ThisWorkbook.Save

    For Each WS In ThisWorkbook.Worksheets
        WS.Copy
        ActiveWorkbook.SaveAs Folder & "\" & WS.Name & ".xlsx"
        ActiveWorkbook.Close
    Next WS

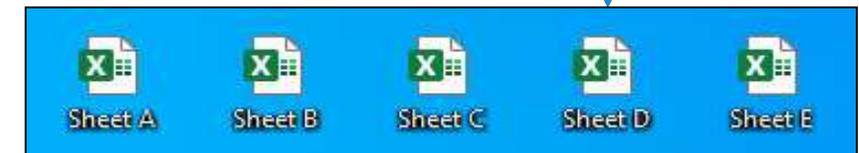
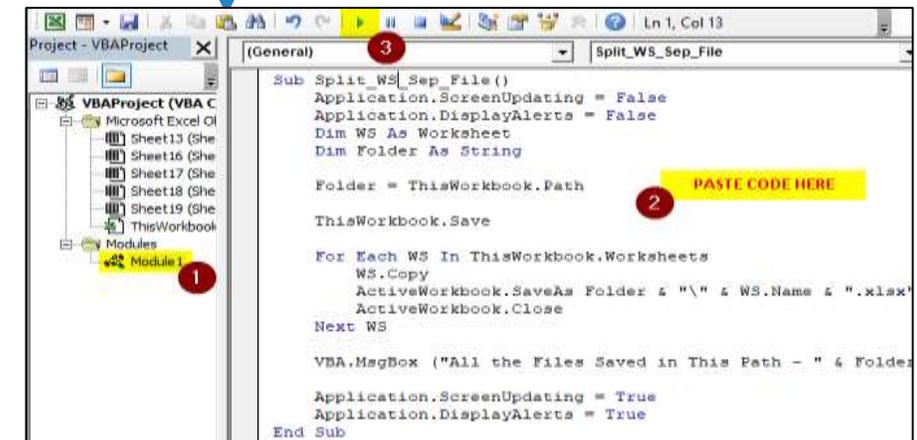
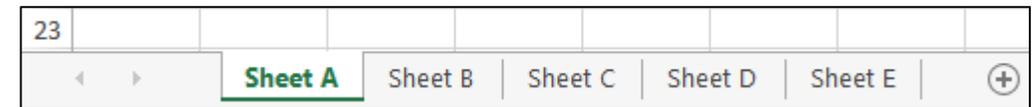
    VBA.MsgBox ("All the Files Saved in This Path - " & Folder)

    Application.ScreenUpdating = True
    Application.DisplayAlerts = True
End Sub

```

Additional Points for Reading

- ❖ You have different worksheets in excel workbook and you want to create a separate file or workbooks for each worksheets.
- ❖ Simply, copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As a result, excel would create new workbook for each worksheet in excel at the same location/folder.



SEPARATE TEXT AND NUMBERS FROM CELL CONTAINING BOTH TEXT & NUMBERS

```

Sub Extrct_Text_Num()
Application.ScreenUpdating = False
Dim Num As String, Text As String, Glb_Num As String,
Glb_Text As String, Lst_Cell As Long, i As Long, r As
Long, Char_Len As Long
Lst_Cell = Range("A1").End(xlDown).Row
For i = 1 To Lst_Cell
Char_Len = VBA.Len(Range("A" & i).Value)
For r = 1 To Char_Len
If IsNumeric(Mid(Range("A" & i).Value, r, 1)) Then
Num = Mid(Range("A" & i).Value, r, 1)
Else
Text = Mid(Range("A" & i).Value, r, 1)
End If
Glb_Num = Glb_Num & Num: Glb_Text = Glb_Text & Text
Num = "": Text = ""
Next r
Range("B" & i).Value = Glb_Text
Range("C" & i).Value = Glb_Num
Glb_Num = "": Glb_Text = ""
Next i
Application.ScreenUpdating = True
End Sub

```

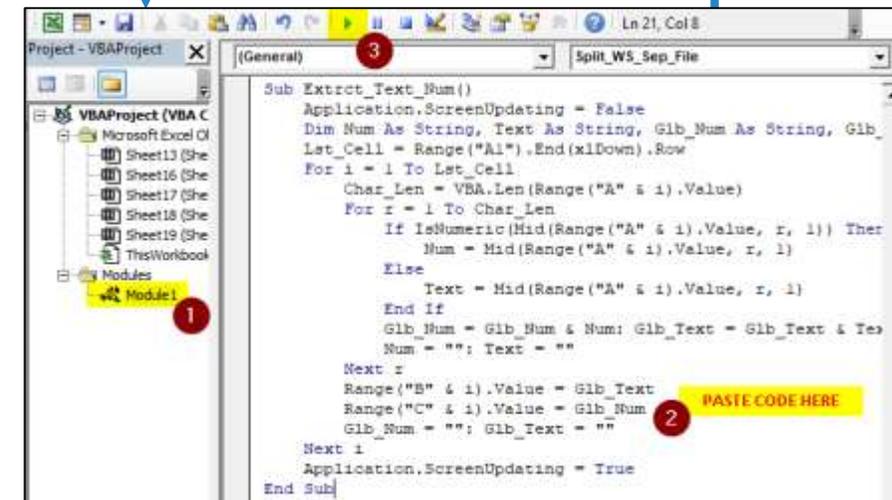
Additional Points for Reading

- ❖ You have some values containing text and number combination in column A, and you want to separate the text and numbers.
- ❖ Simply, copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As a result, excel would separate the text and names in separate columns. You can copy them and use it wherever you like.

	A	B	C
1	12Excel345		
2	W456rap543		
3	V456B123A		
4			

Text and Numbers are same cell

	A	B	C
1	12Excel345	Excel	12345
2	W456rap543	Wrap	456543
3	V456B123A	VBA	456123



The screenshot shows the VBA Editor with the code from the previous block pasted into the code window. Red circles 1, 2, and 3 highlight the Project Explorer, the code area, and the Paste Code Here button respectively.

Make sure that the excel list is in cell A1, Also the worksheet containing list should be an active worksheet.

CREATE FOLDER BASED ON THE EXCEL LIST IN THE COMPUTER

```
Sub Create_Folder()
    Application.ScreenUpdating = False
    Dim LstCell As Long
    Dim i As Long

    LstCell = ActiveSheet.Range("A1").End(xlDown).Row

    For i = 1 To LstCell
        On Error Resume Next
        VBA.MkDir ThisWorkbook.Path & "\" & Range("A" &
i).Value
    Next i

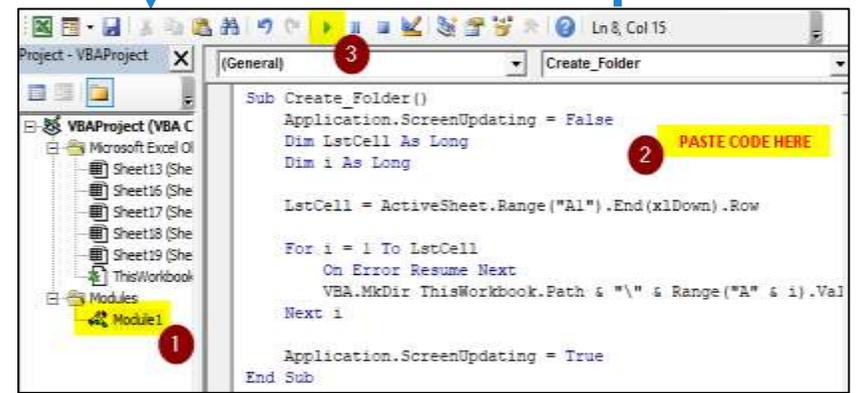
    Application.ScreenUpdating = True
End Sub
```

Make sure that the excel list is in cell A1, Also the worksheet containing list should be an active worksheet.

Additional Points for Reading

- ❖ You have an excel list in column A (list starting from A1) in Excel, and you want to create a separate folders based on this list.
- ❖ Simply, copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As a result, excel would create folders based on this list at the location/path of the excel.

	A	B
1	Folder 1	Purpose is
2	Folder 2	to create
3	Folder 3	folder based
4	Folder 4	on this list
5		



CREATE NEW FOLDER FOR EACH WORKSHEET IN EXCEL WITH WORKSHEET NAME

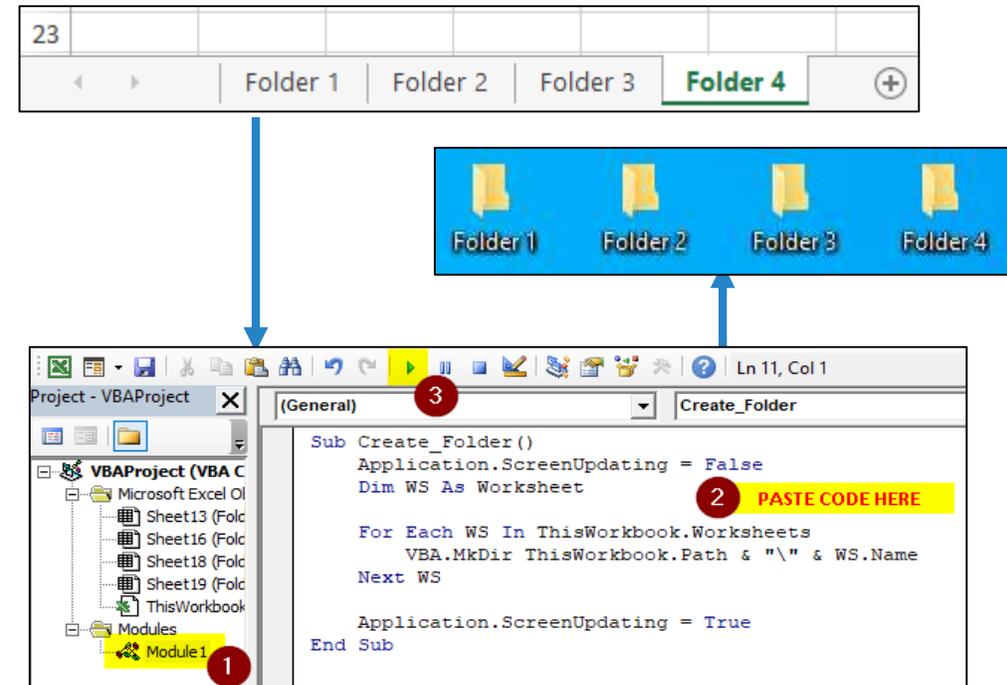
```
Sub Create_Folder()
    Application.ScreenUpdating = False
    Dim WS As Worksheet

    For Each WS In ThisWorkbook.Worksheets
        VBA.MkDir ThisWorkbook.Path & "\" & WS.Name
    Next WS

    Application.ScreenUpdating = True
End Sub
```

Additional Points for Reading

- ❖ The purpose here is to create folders for each of the worksheets in Excel (Folder name = Worksheet Name).
- ❖ Simply, copy the Code from here and paste it in the VBE Editor Window as explained [here](#) and Run it.
- ❖ As a result, excel would create folders for each of the worksheets in Excel with folder name as worksheet name.



The screenshot illustrates the execution of the VBA code. In the VBA Editor, the 'Project - VBAProject' window shows 'Module1' selected (1). The code is pasted into the 'Create_Folder' module (2). The 'Run' button is clicked (3). The result is a folder bar in Excel showing 'Folder 1', 'Folder 2', 'Folder 3', and 'Folder 4'.

THANK YOU FOR VISITING EXCELUNLOCKED