EXCEL CHAMPS

Presents

# EXCEL POWER BUNDLE

# USEFUL MACRO CODES

VERSION 3.1

# Awesome

## Thank you for downloading this guide

Hey,

My name is Puneet . I'm a 26 years old guy from India.

I'm on a mission.

And, my mission is to help people & learn Microsoft Excel.

So, that's why I am helping people who want to learn Microsoft Excel.

I have founded ExcelChamps to reach more & more people like you.

I have created this E-Book with love, for people who want to drive their skills to next level.

With love and respect,

Puneet Gogia

Follow the real me on Instagram

# Important Message

This copy is purely dedicated to you. You can use it in several ways. Save it in your laptop, mobile, take a printout, and please, no need to say thanks. 😁

But you can't sell it or you can't make a change in it because all rights of this copy is with ExcelChamps.com.

If want some changes in it or some addition to it, you can mail me.

And, if you like this guide,  don't forget to share it with your buddies. I'm sure they will appreciate it.

## Here's you can share it

f 🐦

# Table of Content

## 1. create a backup of a current workbook

```
Sub FileBackUp()
ThisWorkbook.SaveCopyAs Filename:=ThisWorkbook.Path & _
"" & Format(Date, "mm-dd-yy") & " " & _
ThisWorkbook.name
End Sub
```

## 2. close all workbooks at once

```
Sub CloseAllWorkbooks()
Dim wbs As Workbook
For Each wbs In Workbooks
wbs.Close SaveChanges:=True
Next wb
End Sub
```

## 3. hide all but the active worksheet

```
Sub HideWorksheet()
Dim ws As Worksheet
For Each ws In ThisWorkbook.Worksheets
If ws.Name <> ThisWorkbook.ActiveSheet.Name Then
ws.Visible = xlSheetHidden
End If
Next ws
End Sub
```

## 4. unhide all hidden worksheets

```
Sub UnhideAllWorksheet()
Dim ws As Worksheet
For Each ws In ActiveWorkbook.Worksheets
ws.Visible = xlSheetVisible
Next ws
End Sub
```

## 5. delete all but the active worksheet

```
Sub DeleteWorksheets()
Dim ws As Worksheet
For Each ws In ThisWorkbook.Worksheets
If ws.name <> ThisWorkbook.ActiveSheet.name Then
Application.DisplayAlerts = False
ws.Delete
Application.DisplayAlerts = True
End If
Next ws
End Sub
```

## 6. copy active worksheet into a new workbook

```
Sub CopyWorksheetToNewWorkbook()
ThisWorkbook.ActiveSheet.Copy _
Before:=Workbooks.Add.Worksheets(1)
End Sub
```

## 7. protect all worksheets instantly

```
Sub ProtecAllWorskeets()
Dim ws As Worksheet
Dim ps As String
ps = InputBox("Enter a Password.", vbOKCancel)
For Each ws In ActiveWorkbook.Worksheets
ws.Protect Password:=ps
Next ws
End Sub
```

## 8. convert all formulas into values

```
Sub ConvertToValues()
Dim MyRange As Range
Dim MyCell As Range
Select Case MsgBox("You Can't Undo This Action. " & "Save
Workbook First?", vbYesNoCancel, "Alert")
Case Is = vbYes
ThisWorkbook.Save
Case Is = vbCancel
Exit Sub
End Select
Set MyRange = Selection
For Each MyCell In MyRange
If MyCell.HasFormula Then
MyCell.Formula = MyCell.Value
End If
```

```
Next MyCell
End Sub
```

## 9. remove spaces from selected cells

```
Sub RemoveSpaces()
Dim myRange As Range
Dim myCell As Range
Select Case MsgBox("You Can't Undo This Action. " & "Save
Workbook First?", _
vbYesNoCancel, "Alert")
Case Is = vbYesThisWorkbook.Save
Case Is = vbCancel
Exit Sub
End Select
Set myRange = Selection
For Each myCell In myRange
If Not IsEmpty(myCell) Then
myCell = Trim(myCell)
End If
Next myCell
End Sub
```

## 10. highlight duplicates from selection

```
Sub HighlightDuplicateValues()
Dim myRange As Range
Dim myCell As Range
```

```
If WorksheetFunction.CountIf(myRange, myCell.Value) > 1 Then
myCell.Interior.ColorIndex = 36
End If
Next myCell
End Sub
```

## 11. hide all pivot table subtotals

```
Sub HideSubtotals()
Dim pt As PivotTable
Dim pf As PivotField
On Error Resume Next
Set pt = ActiveSheet.PivotTables(ActiveCell.PivotTable.name)
If pt Is Nothing Then
MsgBox "You must place your cursor inside of a PivotTable."
Exit Sub
End If
For Each pf In pt.PivotFields
pf.Subtotals(1) = True
pf.Subtotals(1) = False
Next pf
End Sub
```

## 12. refresh all pivot tables

```
Sub RefreshAllPivotTables()
Dim ws As Worksheet
Dim pt As PivotTable
For Each ws In
For Each ws In ThisWorkbook.Worksheets
For Each pt In ws.PivotTables
pt.RefreshTable
Next pt
Next ws
End Sub
```

## 13. resize all charts in a worksheet

```
Sub Resize_Charts()
Dim i As Integer
For i = 1 To ActiveSheet.ChartObjects.Count
With ActiveSheet.ChartObjects(i)
.Width = 300
.Height = 200
End With
Next i
End Sub
```

## 14. highlight the active row and column

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As
Range, Cancel As Boolean)
Dim strRange As String
strRange = Target.Cells.Address & "," & _
Target.Cells.EntireColumn.Address & "," & _
Target.Cells.EntireRow.Address
Range(strRange).Select
End Sub
```

## 15. save selected range as a pdf

```
Sub SaveAsPDF()
Selection.ExportAsFixedFormat Type:=xlTypePDF,
OpenAfterPublish:=True
End Sub
```

## 16. create a table of content

```
Sub TableofContent()
Dim i As Long
On Error Resume Next
Application.DisplayAlerts = False
Worksheets("Table of Content").Delete
Application.DisplayAlerts = True
On Error GoTo 0
ThisWorkbook.Sheets.Add Before:=ThisWorkbook.Worksheets(1)
ActiveSheet.Name = "Table of Content"
For i = 1 To Sheets.Count
With ActiveSheet
.Hyperlinks.Add Anchor:=ActiveSheet.Cells(i, 1), _
Address:="", _
SubAddress:="'" & Sheets(i).Name & "'!A1", _
ScreenTip:=Sheets(i).Name, _
TextToDisplay:=Sheets(i).Name
End With
Next i
End Sub
```

## 15. save selected range as a pdf

```
Sub SaveAsPDF()
Selection.ExportAsFixedFormat Type:=xlTypePDF,
OpenAfterPublish:=True
End Sub
```

## 16. create a table of content

```
Sub TableofContent()
Dim i As Long
On Error Resume Next
Application.DisplayAlerts = False
Worksheets("Table of Content").Delete
Application.DisplayAlerts = True
On Error GoTo 0
ThisWorkbook.Sheets.Add Before:=ThisWorkbook.Worksheets(1)
ActiveSheet.Name = "Table of Content"
For i = 1 To Sheets.Count
With ActiveSheet
.Hyperlinks.Add Anchor:=ActiveSheet.Cells(i, 1), _
Address:="", _
SubAddress:="'" & Sheets(i).Name & "'!A1", _
ScreenTip:=Sheets(i).Name, _
TextToDisplay:=Sheets(i).Name
End With
Next i
End Sub
```

## 17. remove characters from a string

```
Public Function removeFirstC(rng As String, cnt As Long)
removeFirstC = Right(rng, Len(rng) - cnt)
End Function
```

## 18. active workbook in an email

```
Sub Send_Mail()
Dim OutApp As Object
Dim OutMail As Object
Set OutApp = CreateObject("Outlook.Application")
Set OutMail = OutApp.CreateItem(0)
With OutMail
.to = "Sales@FrontLinePaper.com"
.Subject = "Growth Report"
.Body = "Hello Team, Please find attached Growth Report."
.Attachments.Add ActiveWorkbook.FullName
.display
End With
Set OutMail = Nothing
Set OutApp = Nothing
End Sub
```

## 19. convert range into an image

```
Sub PasteAsPicture()
Application.CutCopyMode = False
Selection.Copy
ActiveSheet.Pictures.Paste.Select
End Sub
```

## 20. insert a linked picture

```
Sub LinkedPicture()
Selection.Copy
ActiveSheet.Pictures.Paste(Link:=True).Select
End Sub
```

## 21. highlight top 10 values

```
Sub TopTen()
Selection.FormatConditions.AddTop10
Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
With Selection.FormatConditions(1)
.TopBottom = xlTop10Top
.Rank = 10
.Percent = False
End With
With Selection.FormatConditions(1).Font
.Color = -16752384
.TintAndShade = 0
End With
With Selection.FormatConditions(1).Interior
.PatternColorIndex = xlAutomatic
.Color = 13561798
.TintAndShade = 0
End With
Selection.FormatConditions(1).StopIfTrue = False
End Sub
```

## 22. add serial numbers

```
Sub AddSerialNumbers()
Dim i As Integer
On Error GoTo Last
i = InputBox("Enter Value", "Enter Serial Numbers")
For i = 1 To i
ActiveCell.Value = i
ActiveCell.Offset(1, 0).Activate
Next i
Last:
Exit Sub
End Sub
```

## 23. insert multiple worksheets

```
Sub InsertMultipleSheets()
Dim i As Integer
i = InputBox("Enter number of sheets to insert.", "Enter Multiple Sheets")
Sheets.Add After:=ActiveSheet, Count:=i
End Sub
```

## 24. highlight named ranges

```
Sub HighlightRanges()
Dim RangeName As Name
Dim HighlightRange As Range
On Error Resume Next
For Each RangeName In ActiveWorkbook.Names
```

```
Set HighlightRange = RangeName.RefersToRange
HighlightRange.Interior.ColorIndex = 36
Next RangeName
End Sub
```

## 25. highlight greater than values

```
Sub HighlightGreaterThanValues()
Dim i As Integer
i = InputBox("Enter Greater Than Value", "Enter Value")
Selection.FormatConditions.Delete
Selection.FormatConditions.Add Type:=xlCellValue,
Operator:=xlGreater, Formula1:=i
Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
With Selection.FormatConditions(1)
.Font.Color = RGB(0, 0, 0)
.Interior.Color = RGB(31, 218, 154)
End With
End Sub
```

## 26. highlight lower than values

```
Sub HighlightLowerThanValues()
Dim i As Integer
i = InputBox("Enter Lower Than Value", "Enter Value")
Selection.FormatConditions.Delete
Selection.FormatConditions.Add Type:=xlCellValue,
Operator:=xlLower, Formula1:=i
```

## 27. protect worksheet

```
Sub ProtectWS()
ActiveSheet.Protect "mypassword", True, True
End Sub
```

## 28. unprotect worksheet

```
Sub UnprotectWS()
ActiveSheet.Unprotect "mypassword"
End Sub
```

## 29. convert text to upper case

```
Sub ConvertUpperCase()
Dim rng As Range
For Each rng In Selection
rng = UCase(rng)
Next rng
End Sub
```

## 30. convert text to lower case

```
Sub ConvertLowerCase()
Dim rng As Range
For Each rng In Selection
rng = LCase(rng)
Next rng
End Sub
```

## 31. insert multiple columns

```
Sub InsertMultipleColumns()
Dim i As Integer
Dim j As Integer
ActiveCell.EntireColumn.Select
On Error GoTo Last
i = InputBox("Enter number of columns to insert", "Insert
Columns")
For j = 1 To i
Selection.Insert Shift:=xlToRight,
CopyOrigin:=xlFormatFromRightorAbove
Next j
Last:
Exit Sub
End Sub
```

## 32. insert multiple rows

```
Sub InsertMultipleRows()
Dim i As Integer
Dim j As Integer
ActiveCell.EntireRow.Select
On Error GoTo Last
i = InputBox("Enter number of columns to insert", "Insert Columns")
For j = 1 To i
Selection.Insert Shift:=xlToDown,
CopyOrigin:=xlFormatFromRightorAbove
```

```
Next j
Last:
Exit Sub
End Sub
```

## 33. auto fit columns

```
Sub AutoFitColumns()
Cells.Select
Cells.EntireColumn.AutoFit
End Sub
```

## 34. auto fit rows

```
Sub AutoFitRows()
Cells.Select
Cells.EntireRow.AutoFit
End Sub
```

## 35. remove text wrap

```
Sub RemoveWrapText()
Cells.Select Selection.WrapText = False
Cells.EntireRow.AutoFit
Cells.EntireColumn.AutoFit
End Sub
```

## 36. unmerge cells

```
Sub UnmergeCells()
Selection.UnMerge
End Sub
```

## 37. change chart type

```
Sub ChangeChartType()
ActiveChart.ChartType = xlColumnClustered
End Sub
```

## 38. paste chart as an image

```
Sub ConvertChartToPicture()
 ActiveChart.ChartArea.Copy
 ActiveSheet.Range("A1").Select
 ActiveSheet.Pictures.Paste.Select
End Sub
```

## 39. add chart title

```
Sub AddChartTitle()
Dim i As Variant
i = InputBox("Please enter your chart title", "Chart Title")
On Error GoTo Last
ActiveChart.SetElement (msoElementChartTitleAboveChart)
ActiveChart.ChartTitle.Text = i
Last:
Exit Sub
End Sub
```

## 40. reverse text

```
Public Function rvrse(ByVal cell As Range) As String
rvrse = VBA.strReverse(cell.Value)
End Function
```

## 41. sort worksheets

```
Sub SortWorksheets()
Dim i As Integer
Dim j As Integer
Dim iAnswer As VbMsgBoxResult
iAnswer = MsgBox("Sort Sheets in Ascending Order?" & Chr(10) _
& "Clicking No will sort in Descending Order", _
vbYesNoCancel + vbQuestion + vbDefaultButton1, "Sort Worksheets")
For i = 1 To Sheets.Count
For j = 1 To Sheets.Count - 1
If iAnswer = vbYes Then
If UCase$(Sheets(j).Name) > UCase$(Sheets(j + 1).Name) Then
Sheets(j).Move After:=Sheets(j + 1)
End If
ElseIf iAnswer = vbNo Then
If UCase$(Sheets(j).Name) < UCase$(Sheets(j + 1).Name) Then
Sheets(j).Move After:=Sheets(j + 1)
End If
End If
Next j
Next i
End Sub
```

## 42. add workbook to a mail attachment

```
Sub OpenWorkbookAsAttachment()
Application.Dialogs(xlDialogSendMail).Show
End Sub
```

## 43. activate r1c1 reference style

```
Sub ActivateR1C1()
If Application.ReferenceStyle = xlA1 Then
Application.ReferenceStyle = xlR1C1
Else
Application.ReferenceStyle = xlR1C1
End If
End Sub
```

## 44. Activate A1 Reference Style

```
Sub ActivateA1()
If Application.ReferenceStyle = xlR1C1 Then
Application.ReferenceStyle = xlA1
Else
Application.ReferenceStyle = xlA1
End If
End Sub
```

## 45. Open Calculator

```
Sub OpenCalculator()
Application.ActivateMicrosoftApp Index:=0
End Sub
```

## 46. use text to speech

```
Sub Speak()
Selection.Speak
End Sub
```

## 47. activate user form

```
Sub DataForm()
ActiveSheet.ShowDataForm
End Sub
```

## 48. insert timestamp

```
Sub TimeStamp()
Dim i As Integer
For i = 1 To 24
ActiveCell.FormulaR1C1 = i & ":00"
ActiveCell.NumberFormat = "[$-409]h:mm AM/PM;@"
ActiveCell.Offset(RowOffset:=1, ColumnOffset:=0).Select
Next i
End Sub
```

## 49. create pivot table

## 50. Update Pivot Table Range

## 51. welcome message

```
Sub auto_open()
MsgBox "Welcome To ExcelChamps & Thanks for downloading
this file."
End Sub
```

## 52. closing message

```
Sub auto_close()
MsgBox "Bye Bye! Don't forget to check other cool stuff on
excelchamps.com"
End Sub
```

## 53. convert date into day

```
Sub date2day()
Dim tempCell As Range
Selection.Value = Selection.Value
For Each tempCell In Selection
   If IsDate(tempCell) = True Then
      With tempCell
      .Value = Day(tempCell)
      .NumberFormat = "0"
      End With
   End If
Next tempCell
End Sub
```

## 54. convert date into year

```
Sub date2year()

Dim tempCell As Range

Selection.Value = Selection.Value

For Each tempCell In Selection

If IsDate(tempCell) = True ThenWith tempCell

.Value = Year(tempCell)

.NumberFormat = "0"

End With

End If

Next tempCell

End Sub
```

## 55. remove time from date

```
Sub RemoveTime()

Dim Rng As Range

For Each Rng In Selection

If IsDate(Rng) = True Then

Rng.Value = VBA.Int(Rng.Value)

End If

Next

Selection.NumberFormat = "dd-mmm-yy"

End Sub
```

## 56. remove date from date & time

```
Sub removeDate()

Dim Rng As Range

For Each Rng In Selection
```

```
If IsDate(Rng) = True Then
Rng.Value = Rng.Value - VBA.Fix(Rng.Value)
End If
Next
Selection.NumberFormat = "hh:mm:ss am/pm"
End Sub
```

## 57. add header/footer date

```
Sub dateInHeader()
With ActiveSheet.PageSetup
.LeftHeader = ""
.CenterHeader = "&D"
.RightHeader = ""
.LeftFooter = ""
.CenterFooter = ""
.RightFooter = ""
End With
ActiveWindow.View = xlNormalView
End Sub
```

## 58. custom header/footer

```
Sub customHeader()
Dim myText As String
myText = InputBox("Enter your text here", "Enter Text")
    With ActiveSheet.PageSetup
        .LeftHeader = ""
        .CenterHeader = myText
```

```
.RightHeader = ""
.LeftFooter = ""
.CenterFooter = ""
.RightFooter = ""
End With
End Sub
```

## 59. disable/enable get pivot data

```
Sub activateGetPivotData()
Application.GenerateGetPivotData = True
End Sub

 Sub deactivateGetPivotData()
Application.GenerateGetPivotData = False
End Sub
```

## 60. convert to upper case

```
Sub convertUpperCase()
Dim Rng As Range
For Each Rng In Selection
   If Application.WorksheetFunction.IsText(Rng) Then
      Rng.Value = UCase(Rng)
   End If
Next
End Sub
```

## 61. convert to lower case

```
Sub convertLowerCase()
Dim Rng As Range
For Each Rng In Selection
If Application.WorksheetFunction.IsText(Rng) Then
Rng.Value = LCase(Rng)
End If
Next
End Sub
```

## 62. convert to upper case

```
Sub convertProperCase()
Dim Rng As Range
For Each Rng In Selection
  If WorksheetFunction.IsText(Rng) Then
    Rng.Value = WorksheetFunction.Proper(Rng.Value)
  End If
Next
End Sub
```

## 63. convert to upper case

```
Sub convertTextCase()
Dim Rng As Range
For Each Rng In Selection
  If WorksheetFunction.IsText(Rng) Then
  Rng.Value = UCase(Left(Rng, 1)) & LCase(Right(Rng, Len(Rng) - 1))
  End If
```

```
Next rng
End Sub
```

## 64. remove a character from selection

```
Sub removeChar()
    Dim Rng As Range
    Dim rc As String
    rc = InputBox("Character(s) to Replace", "Enter  Value")
        For Each Rng In Selection
            Selection.Replace What:=rc, Replacement:=""
        Next
End Sub
```

## 65. relative to absolute reference

```
Sub relToAbs()
For Each c In Selection
    If c.HasFormula = True Then
        c.Formula = Application.ConvertFormula(c.Formula, _
          xlA1, xlA1, xlAbsolute)
    End If
Next c
End Sub
```

## 66. remove apostrophe from a number

```
Sub removeApostrophes()
    Selection.Value = Selection.Value
End Sub
```

```
Next rng
End Sub
```

## 67. highlight negative numbers

```
Sub highlightNegativeNumbers()
Dim Rng As Range
For Each Rng In Selection
    If WorksheetFunction.IsNumber(Rng) Then
        If Rng.Value < 0 Then
        Rng.Font.Color = -16776961
        End If
    End If
Next
End Sub
```

## 68. highlight specific text

```
Sub highlightValue()
    Dim myStr As String
    Dim myRg As Range
    Dim myTxt As String
    Dim myCell As Range
    Dim myChar As String
    Dim I As Long
    Dim J As Long
    On Error Resume Next
    If ActiveWindow.RangeSelection.Count > 1 Then
      myTxt = ActiveWindow.RangeSelection.AddressLocal
```

```
Else
      myTxt = ActiveSheet.UsedRange.AddressLocal
    End If
LInput:
    Set myRg = Application.InputBox("please select the data
range:", "Selection Required", myTxt, , , , , 8)
    If myRg Is Nothing Then Exit Sub
    If myRg.Areas.Count > 1 Then
        MsgBox "not support multiple columns"
        GoTo LInput
    End If
    If myRg.Columns.Count <> 2 Then
        MsgBox "the selected range can only contain two columns "
        GoTo LInput
    End If
    For I = 0 To myRg.Rows.Count - 1
        myStr = myRg.Range("B1").Offset(I, 0).Value
        With myRg.Range("A1").Offset(I, 0)
            .Font.ColorIndex = 1
            For J = 1 To Len(.Text)
                If Mid(.Text, J, Len(myStr)) = myStr Then .Characters(J,
Len(myStr)).Font.ColorIndex = 3
            Next
        End With
    Next I
End Sub
```

## 69. remove decimals to numbers

```
Sub removeDecimals()
Dim lnumber As Double
Dim lResult As Long
Dim rng As Range
    For Each rng In Selection
    rng.Value = Int(rng)
    rng.NumberFormat = "0"
    Next rng
End Sub
```

## 70. multiply all the values with a number

```
Sub multiplyWithNumber()
Dim rng As Range
Dim c As Integer
c = InputBox("Enter number to multiple", "Input Required")
For Each rng In Selection
    If WorksheetFunction.IsNumber(rng) Then
        rng.Value = rng * c
    Else
    End If
Next rng
End Sub
```

## 71. add a number in all the numbers

```
Sub addNumber()
Dim rng As Range
Dim i As Integer
```

```
i = InputBox("Enter number to multiple", "Input Required")
For Each rng In Selection
    If WorksheetFunction.IsNumber(rng) Then
        rng.Value = rng + i
    Else
    End If
Next rng
End Sub
```

## 72. calculate square root

```
Sub getSquareRoot()
Dim rng As Range
Dim i As Integer
For Each rng In Selection
    If WorksheetFunction.IsNumber(rng) Then
        rng.Value = Sqr(rng)
    Else
    End If
Next rng
End Sub
```

## 73. calculate cube root

```
Sub getCubeRoot()
Dim rng As Range
Dim i As Integer
For Each rng In Selection
    If
```

```
 rng.Value = rng ^ (1 / 3)
    Else
    End If
Next rng
End Sub
```

## 74. highlight cells with comments

```
Sub highlightCommentCells()
    Selection.SpecialCells(xlCellTypeComments).Select
    Selection.Style = "Note"
End Sub
```

## 75. highlight cells with misspelled words

```
Sub HighlightMisspelledCells()
    Dim rng As Range
    For Each rng In ActiveSheet.UsedRange
        If Not Application.CheckSpelling(word:=rng.Text) Then
            rng.Style = "Bad"
        End If
    Next rng
End Sub
```

## 76. highlight alternate rows in the selection

```
Sub highlightAlternateRows()
Dim rng As Range
For Each rng In Selection.Rows
   If rng.Row Mod 2 = 1 Then
      rng.Style = "20% - Accent1"
```

```
Next rng
End Sub
```

## 77. protect all the cells with formulas

```
Sub lockCellsWithFormulas()
With ActiveSheet
  .Unprotect
  .Cells.Locked = False
  .Cells.SpecialCells(xlCellTypeFormulas).Locked = True
  .Protect AllowDeletingRows:=True
End With
End Sub
```

## 78. add a-z alphabets in a range

```
Sub addcAlphabets()
Dim i As Integer
        For i = 65 To 90
            ActiveCell.Value = Chr(i)
            ActiveCell.Offset(1, 0).Select
        Next i
End Sub
Sub addsAlphabets()
Dim i As Integer
        For i = 97 To 122
            ActiveCell.Value = Chr(i)
            ActiveCell.Offset(1, 0).Select
        Next i
End Sub
```

## 79. count open unsaved workbooks

```
Sub VisibleWorkbooks()
Dim book As Workbook
Dim i As Integer
For Each book In Workbooks
    If book.Saved = False Then
        i = i + 1
    End If
Next book
MsgBox i
End Sub
```

## 80. delete all blank worksheets

```
Sub deleteBlankWorksheets()
Dim Ws As Worksheet
On Error Resume Next
Application.ScreenUpdating = False
Application.DisplayAlerts = False
    For Each Ws In Application.Worksheets
        If Application.WorksheetFunction.CountA(Ws.UsedRange) = 0 Then
            Ws.Delete
        End If
    Next
Application.ScreenUpdating = True
Application.DisplayAlerts = True
End Sub
```

## 81. convert roman numbers into arabic number

```
Sub convertToNumbers()
Dim rng As Range
Selection.Value = Selection.Value
   For Each rng In Selection
      If Not WorksheetFunction.IsNonText(rng) Then
         rng.Value = WorksheetFunction.Arabic(rng)
      End If
   Next rng
End Sub
```

## 82. use goal seek

```
Sub GoalSeekVBA()
Dim Target As Long
On Error GoTo Errorhandler
Target = InputBox("Enter the required value", "Enter Value")
Worksheets("Goal_Seek").Activate
With ActiveSheet
.Range("C7").GoalSeek _
Goal:=Target, _
ChangingCell:=Range("C2")
End With
Exit Sub
Errorhandler:
MsgBox ("Sorry, value is not valid.")
End Sub
```

## 83. unhide all rows and columns

```
Sub UnhideRowsColumns()
Columns.EntireColumn.Hidden = False
Rows.EntireRow.Hidden = False
End Sub
```

## 84. save each worksheet as a single pdf

```
Sub SaveWorkshetAsPDF()
Dim ws As Worksheet
For Each ws In Worksheets
ws.ExportAsFixedFormat xlTypePDF, “ENTER-FOLDER-NAME-HERE" & ws.Name & ".pdf"
Next ws
End Sub
```

## 85. count/highlight cells with error in entire worksheet

```
Sub highlightErrors()
Dim rng As Range
Dim i As Integer
    For Each rng In ActiveSheet.UsedRange
        If WorksheetFunction.IsError(rng) Then
            i = i + 1
            rng.Style = "bad"
        End If
    Next rng
MsgBox "There are total " & i & " error(s) in this worksheet."
End Sub
```

## 86. count/highlight cells with error in entire worksheet

```
Sub highlightSpecificValues()
Dim rng As Range
Dim i As Integer
Dim c As Variant
c = InputBox("Enter Value To Highlight")
For Each rng In ActiveSheet.UsedRange
If rng = c Then
rng.Style = "Note"
i = i + 1
End If
Next rng
```

```
MsgBox "There are total " & i &" "& c & " in this worksheet."
End Sub
```

## 87. highlight all the cells in the worksheet which are blank but have an invisible space

```
Sub blankWithSpace()
Dim rng As Range
    For Each rng In ActiveSheet.UsedRange
        If rng.Value = " " Then
            rng.Style = "Note"
        End If
    Next rng
End Sub
```

## 88. highlight max value in the range

```
Sub highlightMaxValue()
Dim rng As Range
For Each rng In Selection
If rng = WorksheetFunction.Max(Selection) Then
rng.Style = "Good"
End If
Next rng
End Sub
```

## 89. highlight min value in the range

```
Sub highlightMaxValue()
Dim rng As Range
    For Each rng In Selection
        If rng = WorksheetFunction.Min(Selection) Then
```

```
 rng.Style = "Good"
      End If
   Next rng
End Sub
```

## 90. highlight unique values

```
Sub highlightUniqueValues()
Dim rng As Range
Set rng = Selection
rng.FormatConditions.Delete
Dim uv As UniqueValues
Set uv = rng.FormatConditions.AddUniqueValues
uv.DupeUnique = xlUnique
uv.Interior.Color = vbGreen
End Sub
```

## 91. show progress on status bar

```
Sub progressStatusBar()
Application.StatusBar = "Start Printing the Numbers"
For icntr = 1 To 5000
Cells(icntr, 1) = icntr
Application.StatusBar = " Please wait while printing the numbers "
& Round((icntr / 5000 * 100), 0) & "%"
Next
Application.StatusBar = ""
End Sub
```

## 92. disable page breaks

```
Sub DisablePageBreaks()
Dim wb As Workbook
Dim wks As Worksheet
Application.ScreenUpdating = False
    For Each wb In Application.Workbooks
        For Each Sht In wb.Worksheets
            Sht.DisplayPageBreaks = False
        Next Sht
    Next wb
Application.ScreenUpdating = True
End Sub
```

## 93. highlight difference in columns

```
Sub columnDifference()
    Range("H7:H8,I7:I8").Select
    Selection.ColumnDifferences(ActiveCell).Select
    Selection.Style = "Bad"
End Sub
```

## 94. highlight difference in rows

```
Sub rowDifference()
    Range("H7:H8,I7:I8").Select
    Selection.RowDifferences(ActiveCell).Select
    Selection.Style = "Bad"
End Sub
```

## 95. print comments

```
Sub printComments()
    With ActiveSheet.PageSetup
        .printComments = xlPrintSheetEnd
    End With
End Sub
```

## 96. print with narrow margin

```
Sub printNarrowMargin()
    With ActiveSheet.PageSetup
        .LeftMargin = Application.InchesToPoints(0.25)
        .RightMargin = Application.InchesToPoints(0.25)
        .TopMargin = Application.InchesToPoints(0.75)
        .BottomMargin = Application.InchesToPoints(0.75)
        .HeaderMargin = Application.InchesToPoints(0.3)
        .FooterMargin = Application.InchesToPoints(0.3)
    End With
    ActiveWindow.SelectedSheets.PrintOut Copies:=1,
Collate:=True, IgnorePrintAreas:=False
End Sub
```

## 97. print selection

```
Sub printSelection()
Selection.PrintOut Copies:=1, Collate:=True
End Sub
```

## 98. Print custom pages

```
Sub printCustomSelection()
Dim startpage As Integer
Dim endpage As Integer
startpage = InputBox("Please Enter Start Page number.", "Enter Value")
If Not WorksheetFunction.IsNumber(startpage) Then
    MsgBox "Invalid Start Page number. Please try again.", "Error"
    Exit Sub
End If
endpage = InputBox("Please Enter End Page number.", "Enter Value")
If Not WorksheetFunction.IsNumber(endpage) Then
    MsgBox "Invalid End Page number. Please try again.", "Error"
    Exit Sub
End If
Selection.PrintOut From:=startpage, To:=endpage, Copies:=1, Collate:=True
End Sub
```

## 99. remove negative numbers

```
Sub removeNegativeSign()
Dim rng As Range
Selection.Value = Selection.Value
    For Each rng In Selection
        If WorksheetFunction.IsNumber(rng) Then
```

```
 rng.Value = Abs(rng)
        End If
    Next rng
End Sub
```

## 100. replace blank cells with zeros

```
Sub replaceBlankWithZero()
Dim rng As Range
Selection.Value = Selection.Value
    For Each rng In Selection
        If rng = "" Or rng = " " Then
                rng.Value = "0"
        Else
        End If
    Next rng
End Sub
```

# THANK YOU

for reading this e-book