1. 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
```

---

2. 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
```

---

3. 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 rows to insert", "Insert Rows")
For j = 1 To i
Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromRightorAbove
Next j
Last: Exit Sub
End Sub
```

---

4. Auto Fit Columns

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

ExcelChamps.com

5. Auto Fit Rows

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

---

6. Remove Text Wrap

```
Sub RemoveTextWrap()
Range("A1").WrapText = False
End Sub
```

---

7. Unmerge Cells

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

---

8. Open Calculator

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

---

9. Add Header/Footer Date

```
Sub DateInHeader()
With ActiveSheet.PageSetup
.LeftHeader = ""
.CenterHeader = "&D"
.RightHeader = ""
.LeftFooter = ""
.CenterFooter = ""
.RightFooter = ""
End With
End Sub
```

---

10. Custom Header/Footer

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

ExcelChamps.com

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

---

11. Highlight Duplicates from Selection

```
Sub HighlightDuplicateValues()
Dim myRange As Range
Dim myCell As Range
Set myRange = Selection
For Each myCell In myRange
If WorksheetFunction.CountIf(myRange, myCell.Value) > 1 Then
myCell.Interior.ColorIndex = 36
End If
Next myCell
End Sub
```

---

12. 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
```

---

13. 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
```

ExcelChamps.com

```
Selection.FormatConditions(1).StopIfTrue = False
End Sub
```

---

14. 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
```

---

15. 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
```

---

16. 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
Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
With Selection.FormatConditions(1)
.Font.Color = RGB(0, 0, 0)
.Interior.Color = RGB(217, 83, 79)
End With
End Sub
```

---

17. Highlight Negative Numbers

```
Sub highlightNegativeNumbers()
Dim Rng As Range
```

ExcelChamps.com

```
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
```

---

18. 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
End If
Next
End With
Next I
```

ExcelChamps.com

```
End Sub
```

---

19. Highlight Cells with Comments

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

---

20. 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"
End If
Next rng
End Sub
```

---

21. 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
```

---

22. Highlight Cells with Errors in the 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
```

---

23. Highlight Cells with a Specific Text in Worksheet

ExcelChamps.com

```
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
```

---

24. Highlight Blank Cells with 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
```

---

25. 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
```

---

26. Highlight Min Value in the Range

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

ExcelChamps.com

27. 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
```

---

28. Highlight Differences in Columns

```
Sub columnDifference()
Range("A1:A10,B1:B10").Select
Selection.ColumnDifferences(ActiveCell).Select
Selection.Style = "Bad"
End Sub
```

---

29. Highlight Differences in Rows

```
Sub rowDifference()
Range("A1:A10,B1:B10").Select
Selection.RowDifferences(ActiveCell).Select
Selection.Style = "Bad"
End Sub
```

---

30. Print Comments

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

---

31. Print 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
```

ExcelChamps.com

```
ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, IgnorePrintAreas:=False
End Sub
```

---

32. Print Selection

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

---

33. Print Custom Pages

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

---

34. 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
```

---

35. Unhide All Hidden Worksheets

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

ExcelChamps.com

36. 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
```

---

37. Protect All Worksheets Instantly

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

---

38. 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
```

---

39. Insert Multiple Worksheets

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

---

40. Protect Worksheet

```
Sub ProtectWS()
```

ExcelChamps.com

```
ActiveSheet.Protect "mypassword", True, True
End Sub
```

---

41. Unprotect Worksheet

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

---

42. 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
```

---

43. Protect All Cells with Formulas

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

---

44. Delete All Blank Worksheets

ExcelChamps.com

```
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
```

---

45. Unhide All Rows and Columns

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

---

46. Save Each Worksheet as a Single PDF

```
Sub SaveWorksheetAsPDF()
Dim ws As Worksheet
For Each ws In Worksheets
    ws.ExportAsFixedFormat xlTypePDF, "C:\ExportedSheets\" & ws.Name & ".pdf"
Next ws
End Sub
```

---

47. Disable Page Breaks

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

---

48. Create a Backup of a Current Workbook

```
Sub FileBackUp()
```

ExcelChamps.com

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

---

49. Close All Workbooks at Once

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

---

50. Copy Active Worksheet into a New Workbook

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

---

51. Send Active Workbook via 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 = "recipient@example.com"
    .Subject = "Workbook Attached"
    .Body = "Please find the attached workbook."
    .Attachments.Add ActiveWorkbook.FullName
    .Display
End With

Set OutMail = Nothing
Set OutApp = Nothing
End Sub
```

---

52. Add Workbook to an Email Attachment

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

---

53. Welcome Message on Opening Workbook

ExcelChamps.com

```
Sub auto_open()
MsgBox "Welcome to Excel! Have a great day!"
End Sub
```

---

54. Closing Message on Closing Workbook

```
Sub auto_close()
MsgBox "Goodbye! See you next time."
End Sub
```

---

55. 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 "Unsaved Workbooks: " & i
End Sub
```

---

56. Hide 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 select a PivotTable cell."
    Exit Sub
End If
For Each pf In pt.PivotFields
    pf.Subtotals(1) = False
Next pf
End Sub
```

---

57. Refresh All Pivot Tables

```
Sub RefreshAllPivots()
Dim pt As PivotTable
For Each pt In ActiveWorkbook.PivotTables
    pt.RefreshTable
Next pt
End Sub
```

ExcelChamps.com

---

58. Auto Update Pivot Table Range

```
Sub UpdatePivotTableRange()
Dim Data_Sheet As Worksheet
Dim Pivot_Sheet As Worksheet
Dim StartPoint As Range
Dim DataRange As Range
Dim PivotName As String
Dim NewRange As String
Dim LastCol As Long
Dim LastRow As Long

Set Data_Sheet = ThisWorkbook.Worksheets("PivotData")
Set Pivot_Sheet = ThisWorkbook.Worksheets("Pivot")

PivotName = "PivotTable1"

Data_Sheet.Activate
Set StartPoint = Data_Sheet.Range("A1")
LastCol = StartPoint.End(xlToRight).Column
LastRow = StartPoint.End(xlDown).Row
Set DataRange = Data_Sheet.Range(StartPoint, Cells(LastRow, LastCol))
NewRange = Data_Sheet.Name & "!" & DataRange.Address(ReferenceStyle:=xlR1C1)

Pivot_Sheet.PivotTables(PivotName).ChangePivotCache _
    ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:=NewRange)

Pivot_Sheet.PivotTables(PivotName).RefreshTable
MsgBox "Pivot Table Updated!"
End Sub
```

---

59. Convert All Formulas to Values

```
Sub ConvertToValues()
Dim MyRange As Range
For Each MyRange In Selection
    If MyRange.HasFormula Then
        MyRange.Value = MyRange.Value
    End If
Next MyRange
End Sub
```

---

60. Remove Spaces from Selected Cells

```
Sub RemoveSpaces()
Dim MyRange As Range
For Each MyRange In Selection
    If Not IsEmpty(MyRange) Then
```

ExcelChamps.com

```
        MyRange.Value = Trim(MyRange)
    End If
Next MyRange
End Sub
```

---

61. Convert Text to Uppercase

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

---

62. Convert Text to Lowercase

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

---

63. Convert Numbers to Words

```
Function NumberToWords(ByVal MyNumber As Double) As String
' Simple conversion logic
Dim Units As String, Tens As String
Units = "|One|Two|Three|Four|Five|Six|Seven|Eight|Nine"
Tens = "|Ten|Twenty|Thirty|Forty|Fifty|Sixty|Seventy|Eighty|Ninety"
NumberToWords = "Conversion Logic Needed"
End Function
```

---

64. Sort Data in a Worksheet Automatically

```
Sub AutoSort()
With ThisWorkbook.Sheets("Sheet1").Range("A1:D100")
    .Sort Key1:=.Cells(1, 1), Order1:=xlAscending, Header:=xlYes
End With
End Sub
```

---

65. Print All Workbooks in a Folder

ExcelChamps.com

```
Sub PrintAllWorkbooks()
Dim folderPath As String
Dim filename As String
folderPath = "C:\MyFolder\"
filename = Dir(folderPath & "*.xls*")
Do While filename <> ""
    Workbooks.Open Filename:=folderPath & filename
    ActiveWorkbook.PrintOut Copies:=1
    ActiveWorkbook.Close False
    filename = Dir()
Loop
End Sub
```

---

66. Highlight Cells That Contain Formulas

```
Sub HighlightFormulas()
Dim cell As Range
For Each cell In ActiveSheet.UsedRange
    If cell.HasFormula Then
        cell.Interior.Color = RGB(255, 255, 0)
    End If
Next cell
End Sub
```

---

67. Convert Range into an Image

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

---

68. Insert a Linked Picture

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

---

69. Use Text to Speech

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

---

70. Activate Data Entry Form

ExcelChamps.com

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

---

71. 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 "Invalid value."
End Sub
```

---

72. VBA Code to Search on Google

```
Sub SearchGoogle()
Dim search_string As String
Dim query As String
query = InputBox("Enter your search query", "Google Search")
search_string = Replace(query, " ", "+")
ActiveWorkbook.FollowHyperlink "https://www.google.com/search?q=" & search_string
End Sub
```

---

73. Remove Decimals from Numbers

```
Sub RemoveDecimals()
Dim rng As Range
For Each rng In Selection
    If IsNumeric(rng) Then
        rng.Value = Int(rng)
    End If
Next rng
End Sub
```

---

74. Multiply All Values by a Number

```
Sub MultiplyByNumber()
Dim rng As Range
Dim i As Integer
i = InputBox("Enter number to multiply", "Input Required")
For Each rng In Selection
```

ExcelChamps.com

```
    If IsNumeric(rng) Then
        rng.Value = rng.Value * i
    End If
Next rng
End Sub
```

---

75. Add a Number to All Numbers

```
Sub AddNumber()
Dim rng As Range
Dim i As Integer
i = InputBox("Enter number to add", "Input Required")
For Each rng In Selection
    If IsNumeric(rng) Then
        rng.Value = rng.Value + i
    End If
Next rng
End Sub
```

---

76. Calculate the Square Root of Numbers

```
Sub GetSquareRoot()
Dim rng As Range
For Each rng In Selection
    If IsNumeric(rng) Then
        rng.Value = Sqr(rng)
    End If
Next rng
End Sub
```

---

77. Calculate the Cube Root of Numbers

```
Sub GetCubeRoot()
Dim rng As Range
For Each rng In Selection
    If IsNumeric(rng) Then
        rng.Value = rng.Value ^ (1 / 3)
    End If
Next rng
End Sub
```

---

78. Convert Roman Numbers to Arabic Numbers

```
Sub ConvertToNumbers()
Dim rng As Range
For Each rng In Selection
    If WorksheetFunction.IsNonText(rng) = False Then
```

ExcelChamps.com

```
        rng.Value = WorksheetFunction.Arabic(rng.Value)
    End If
Next rng
End Sub
```

---

79. Remove Negative Signs from Numbers

```
Sub RemoveNegativeSign()
Dim rng As Range
For Each rng In Selection
    If IsNumeric(rng) Then
        rng.Value = Abs(rng.Value)
    End If
Next rng
End Sub
```

---

80. Replace Blank Cells with Zeros

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

---

81. Create a Simple Timer

```
Sub SimpleTimer()
Dim countDown As Date
countDown = Now + TimeValue("00:01:00") ' Set timer for 1 minute
Do Until Now >= countDown
    DoEvents
Loop
MsgBox "Time's up!"
End Sub
```

---

82. Convert Text to Columns Automatically

```
Sub TextToColumnsAuto()
Dim rng As Range
Set rng = ThisWorkbook.Sheets("Sheet1").Range("A1:A100")
rng.TextToColumns Destination:=rng, DataType:=xlDelimited, Comma:=True
End Sub
```

ExcelChamps.com

83. Unprotect All Sheets in a Workbook

```
Sub UnprotectSheets()
Dim ws As Worksheet
For Each ws In ThisWorkbook.Sheets
    ws.Unprotect Password:="password"
Next ws
End Sub
```

---

84. Protect All Sheets in a Workbook

```
Sub ProtectSheets()
Dim ws As Worksheet
For Each ws In ThisWorkbook.Sheets
    ws.Protect Password:="password"
Next ws
End Sub
```

---

85. Automatically Close Workbook After Inactivity

```
Sub AutoClose()
Dim countDown As Date
countDown = Now + TimeValue("00:10:00") ' Set timer for 10 minutes
Do Until Now >= countDown
    If Not Application.Interactive Then Exit Sub
    DoEvents
Loop
ThisWorkbook.Close SaveChanges:=False
End Sub
```

---

86. Export Each Worksheet to a New Workbook

```
Sub ExportSheetsToWorkbooks()
Dim ws As Worksheet
For Each ws In ThisWorkbook.Worksheets
    ws.Copy
    ActiveWorkbook.SaveAs "C:\ExportedSheets\" & ws.Name & ".xlsx"
    ActiveWorkbook.Close False
Next ws
End Sub
```

---

87. Create a Directory from VBA

```
Sub CreateDirectory()
Dim path As String
path = "C:\NewFolder"
If Dir(path, vbDirectory) = "" Then
```

ExcelChamps.com

```
    MkDir path
End If
End Sub
```

---

88. Add a Watermark to a Worksheet

```
Sub AddWatermark()
ActiveSheet.Shapes.AddTextEffect(msoTextEffect1, "Confidential", _
    "Arial", 50, msoFalse, msoFalse, 100, 100).Select
With Selection.ShapeRange.Fill
    .Visible = msoTrue
    .ForeColor.RGB = RGB(217, 217, 217)
    .Transparency = 0.5
End With
End Sub
```

---

89. Print All Workbooks in a Folder

```
Sub PrintAllWorkbooks()
Dim folderPath As String
Dim filename As String
folderPath = "C:\MyFolder\"
filename = Dir(folderPath & "*.xls*")
Do While filename <> ""
    Workbooks.Open Filename:=folderPath & filename
    ActiveWorkbook.PrintOut Copies:=1
    ActiveWorkbook.Close False
    filename = Dir()
Loop
End Sub
```

---

90. Count Words in the Entire Worksheet

```
Sub Word_Count_Worksheet()
Dim WordCnt As Long
Dim rng As Range
Dim S As String
Dim N As Long
For Each rng In ActiveSheet.UsedRange.Cells
    S = Application.WorksheetFunction.Trim(rng.Text)
    N = 0
    If S <> vbNullString Then
        N = Len(S) - Len(Replace(S, " ", "")) + 1
    End If
    WordCnt = WordCnt + N
Next rng
MsgBox "There are total " & Format(WordCnt, "#,##0") & " words in the active worksheet."
End Sub
```

ExcelChamps.com

91. Remove the Apostrophe from Numbers

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

92. Remove Characters from Selected Cells

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

93. Insert A-Z Alphabets in a Range

```
Sub AddsAlphabets1()
Dim i As Integer
For i = 65 To 90
    ActiveCell.Value = Chr(i)
    ActiveCell.Offset(1, 0).Select
Next i
End Sub

Sub AddsAlphabets2()
Dim i As Integer
For i = 97 To 122
    ActiveCell.Value = Chr(i)
    ActiveCell.Offset(1, 0).Select
Next i
End Sub
```

94. Sort Data in a Worksheet Automatically

```
Sub AutoSort()
With ThisWorkbook.Sheets("Sheet1").Range("A1:D100")
    .Sort Key1:=.Cells(1, 1), Order1:=xlAscending, Header:=xlYes
End With
End Sub
```

ExcelChamps.com

95. Save Selected Range as a PDF

```
Sub SaveSelectedRangeAsPDF()
Dim Rng As Range
Set Rng = Selection
Rng.ExportAsFixedFormat Type:=xlTypePDF, Filename:="C:\SelectedRange.pdf"
End Sub
```

---

96. Highlight Cells with Specific Text

```
Sub HighlightSpecificText()
Dim Rng As Range
Dim Text As String
Text = InputBox("Enter text to highlight:", "Find Text")
For Each Rng In ActiveSheet.UsedRange
    If Rng.Value = Text Then
        Rng.Interior.Color = RGB(255, 255, 0)
    End If
Next Rng
End Sub
```

---

97. Unmerge and Fill Blank Cells with Original Value

```
Sub UnmergeAndFill()
Dim Rng As Range
For Each Rng In Selection
    If Rng.MergeCells Then
        Rng.UnMerge
        Rng.Value = Rng.Cells(1, 1).Value
    End If
Next Rng
End Sub
```

---

98. Protect Workbook Structure

```
Sub ProtectWorkbook()
ThisWorkbook.Protect Password:="mypassword", Structure:=True
End Sub
```

---

99. Unprotect Workbook Structure

```
Sub UnprotectWorkbook()
ThisWorkbook.Unprotect Password:="mypassword"
End Sub
```

ExcelChamps.com

100. Convert Date into Time Only

```
Sub RemoveDateKeepTime()
Dim Rng As Range
For Each Rng In Selection
    If IsDate(Rng) Then
        Rng.Value = Rng.Value - VBA.Fix(Rng.Value)
    End If
Next
Selection.NumberFormat = "hh:mm:ss AM/PM"
End Sub
```

ExcelChamps.com