

# USEFUL MACRO CODES

## 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 Worksheet 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 = vbYes  
ThisWorkbook.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  
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
```

## 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

Here are the quick steps to apply this code.

1. Open VBE (ALT + F11).
2. Go to Project Explorer (Ctrl + R, If hidden). Select your workbook & double click on the name of a particular worksheet in which you want to activate the macro.
3. Paste the code into it & Select the “BeforeDoubleClick” from event drop down menu.
4. Close VBE & you are done.

```
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
```

## 17. Welcome Message for User

```
Sub auto_open()  
Dim name As String  
name = Application.UserName
```

```
MsgBox ("Hello " & Application.UserName & ". Welcome  
To ExcelChamps.com")  
End Sub
```

## **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 SelectionWithFormatConditions(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
```

## 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).Se  
    tFirstPriority  
    With Selection.FormatConditions(1)  
        .Font.Color = RGB(0, 0, 0)  
        .Interior.Color = RGB(31, 218, 154)  
    End With
```

## 26. Highlight Greater 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
```

## 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. Delete All Other Worksheets

```
Sub DeleteAllOtherWorksheets()  
Dim xWs As Worksheet  
Application.ScreenUpdating = False  
Application.DisplayAlerts = False  
For Each xWs In Application.ActiveWorkbook.Worksheets  
If xWs.Name <> ActiveSheet.Name Then  
xWs.Delete  
End If  
Next  
Application.DisplayAlerts = True  
Application.ScreenUpdating = True  
End Sub
```

## 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
```



```
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

```
Sub InsertPivotTable()
```

```
'Declare Variables
```

```
Dim PSheet As Worksheet
```

```
Dim DSheet As Worksheet
```

```
Dim PCache As PivotCache
```

```
Dim PTable As PivotTable
```

```
Dim PRange As Range
```

```
Dim LastRow As Long
```

```
Dim LastCol As Long
```

```
'Delete Preivous Pivot Table Worksheet & Insert a New Blank  
Worksheet With Same Name
```

```
On Error Resume Next
```

```
Application.DisplayAlerts = False
```

```
Worksheets("PivotTable").Delete
```

```
Sheets.Add Before:=ActiveSheet
```

```
ActiveSheet.Name = "PivotTable"
```

```
Application.DisplayAlerts = True
```

```
Set PSheet = Worksheets("PivotTable")
```

```
Set DSheet = Worksheets("Data")
```

```
'Define Data Range
```

```
LastRow = DSheet.Cells(Rows.Count, 1).End(xlUp).Row
```

```
LastCol = DSheet.Cells(1, Columns.Count).End(xlToLeft).Column
```

```
Set PRange = DSheet.Cells(1, 1).Resize>LastRow, LastCol)
```

'Define Pivot Cache

```
Set PCache = ActiveWorkbook.PivotCaches.Create _  
(SourceType:=xlDatabase, SourceData:=PRange). _  
CreatePivotTable(TableDestination:=PSheet.Cells(2, 2), _  
TableName:="SalesPivotTable")
```

'Insert Blank Pivot Table

```
Set PTable = PCache.CreatePivotTable _  
(TableDestination:=PSheet.Cells(1, 1),  
TableName:="SalesPivotTable")
```

'Insert Row Fields With

```
ActiveSheet.PivotTables("SalesPivotTable").PivotFields("Year")  
    .Orientation = xlRowField  
    .Position = 1
```

End With

With

```
ActiveSheet.PivotTables("SalesPivotTable").PivotFields("Month")  
    .Orientation = xlRowField  
    .Position = 2
```

End With

'Insert Column Fields With

```
ActiveSheet.PivotTables("SalesPivotTable").PivotFields("Zone")  
    .Orientation = xlColumnField  
    .Position = 1
```

End With

'Insert Data Field

With

```
ActiveSheet.PivotTables("SalesPivotTable").PivotFields("Amount")  
    .Orientation = xlDataField  
    .Position = 1
```

```

.Function = xlSum
.NumberFormat = "#,##0"
.Name = "Revenue "
End With
'Format Pivot Table
ActiveSheet.PivotTables("SalesPivotTable").ShowTableStyleRowStripes = True
ActiveSheet.PivotTables("SalesPivotTable").TableStyle2 =
"PivotStyleMedium9"
End Sub

```

## 50. 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 Pivot Table & Source Worksheet
Set Data_Sheet = ThisWorkbook.Worksheets("PivotTableData3")
Set Pivot_Sheet = ThisWorkbook.Worksheets("Pivot3")

'Enter in Pivot Table Name
PivotName = "PivotTable2"

'Defining Starting Point & Dynamic Range
Data_Sheet.Activate
Set StartPoint = Data_Sheet.Range("A1")
LastCol = StartPoint.End(xlToRight).Column

```

```
DownCell = StartPoint.End(xlDown).Row  
Set DataRange = Data_Sheet.Range(StartPoint, Cells(DownCell,  
LastCol))
```

```
NewRange = Data_Sheet.Name & "!" &  
DataRange.Address(ReferenceStyle:=xlR1C1)
```

```
'Change Pivot Table Data Source Range Address  
Pivot_Sheet.PivotTables(PivotName). _  
ChangePivotCache ActiveWorkbook. _  
PivotCaches.Create(SourceType:=xlDatabase,  
SourceData:=NewRange)
```

```
'Ensure Pivot Table is Refreshed  
Pivot_Sheet.PivotTables(PivotName).RefreshTable
```

```
'Complete Message  
Pivot_Sheet.Activate  
MsgBox "Your Pivot Table is now updated."  
End Sub
```

## 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
```