

Macros & VBA

Cheat Sheet

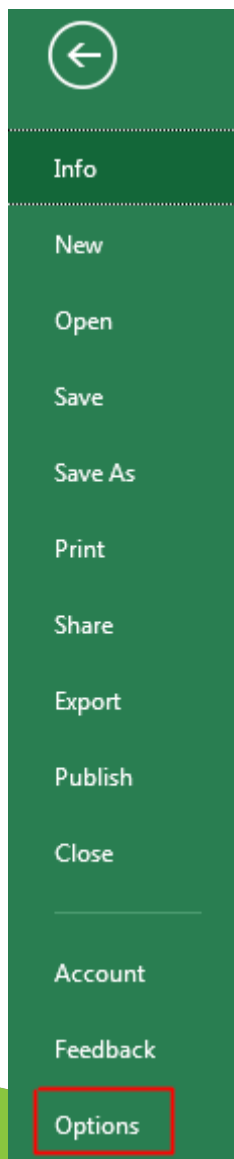


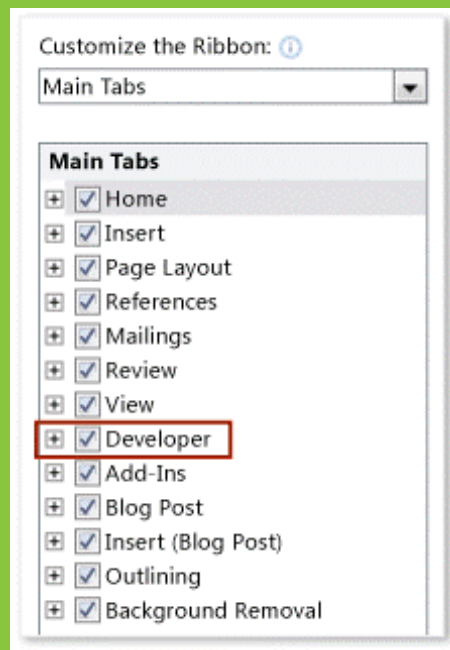
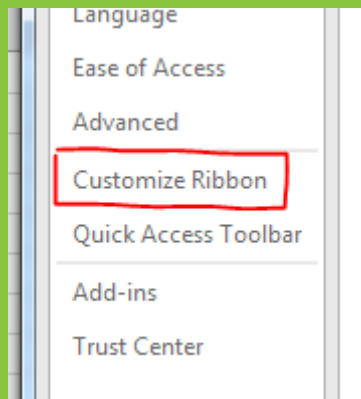
EXCEL DEVELOPER TAB



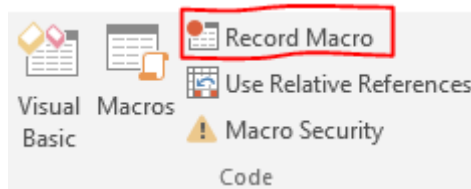
To get access to the Excel Developer tab (and all the VBA goodies!) you must activate the tab by customizing the Excel ribbon.

FILE > OPTIONS > CUSTOMIZE RIBBON > MAIN TABS > DEVELOPER



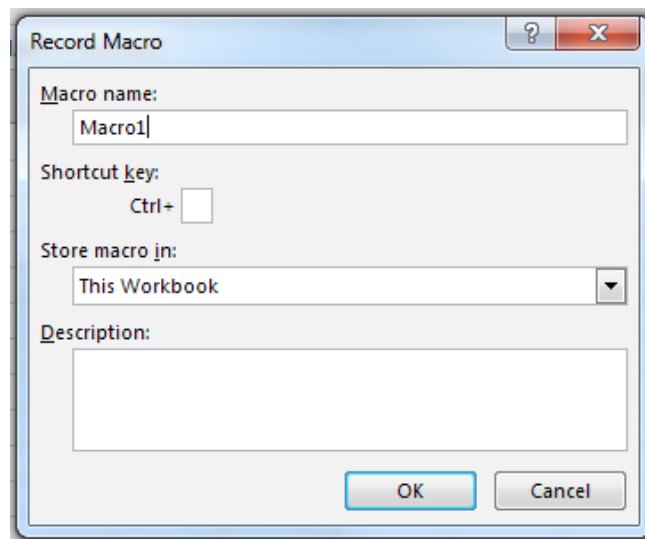


EXCEL MACRO RECORDER



» The Excel Macro Recorder allows you to record your steps as you work in Excel. The recording - made by Excel - can help you automate repetitive tasks.

For example, you may have a block of data that needs to be formatted. You can manually format the data or record the actions of formatting the data and automate the experience for future data.



» **Macro Name:** The name can be up to 255, must start with an alpha character and have no spaces included.



Shortcut Key: (OPTIONAL) Using a shortcut key to run your macro is great, but there are other ways to run macros as well, such as buttons on a worksheet, buttons in the Excel Quick Access Toolbar or Events tied to actions you take in Excel.



Store Macro In: The VBA code generated by the Macro Recorder can be Stored in the *Active Workbook* (This Workbook) or the *Personal Macro Workbook* (Your Computer).



Description: Leave notes about what actions the macro will perform for yourself and others that may interact with the macro.

EXCEL VBA VARIABLES



Excel VBA variables are a named storage location. A value can be stored within the variable for later use or modified for during the execution of VBA code.

Variable names must begin with an alphabetic character and can be up to 255 characters in length.

Declaring a variable contains 3 parts:

1. Start with the *Dim* statement
2. A *unique name* identifying the variable
3. Declare the *Data Type* of the variable

```
Dim userInput As String
```

The most common Data Types include;

- › **String:** String Data Types can contain a combination of alpha-numeric data, as well as special character (i.e. Spaces, comma, etc.)
- › **Integer:** Integer type variables can store integers between -32,768 and 32,767
- › **Long:** Or, Long Integer, can store integer values between:
-2,147,483,648 and 2,147,483,647
- › **Boolean:** Boolean is a logic Data Type can store one of two values, TRUE or FALSE

TRY IT OUT...

```
Public Sub FunWithVariables()
```

```
    ' declaring a string variable  
    Dim userName As String
```

```
    ' store a text value in the variable  
    userName = "George"
```

```
    ' assign the active cell the value from the variable  
    ActiveCell.Value = userName
```

```
End Sub
```

CODE SAMPLES



Here are a few of the common VBA codes that you can use straight away by pasting them in your VBA editor!

Public Sub UserRenameWorksheet()

```
' declare variable to store input from user
  Dim userInputWS As String

' create input box
  userInputWS = InputBox("Enter a name for the
    Active Worksheet")

' change the name of the active sheet
  ' based on user input
  ActiveSheet.Name = userInputWS

' message box informing user of what they have done
  MsgBox ("You have rename the worksheet "
    & userInputWS)
```

End Sub

Public Sub AddListHeaders()

```
' add new worksheet and name sheet
Worksheets.Add
ActiveSheet.Name = "Sales Data"
```

```
Range("A1").Select
```

```
' add headers
    ActiveCell.Value = "EMPLOYEE"
    ActiveCell.Offset(0, 1).Value = "PRODUCT CATEGORY"
    ActiveCell.Offset(0, 2).Value = "MONTH"
    ActiveCell.Offset(0, 3).Value = "SALES AMOUNT"
```

```
' select header range
    Range("A1:D1").Select
```

```
' format the headers selection
    With Selection
        .Interior.Color = RGB(0, 112, 192)
        .Font.Color = RGB(255, 255, 255)
        .Font.Bold = True
        .Borders(xlEdgeBottom).Weight = xlMedium
    End With
```

```
' change column width
    Columns("A:D").AutoFit
```

```
End Sub
```

Public Sub AddMonthlyWorksheets()

‘ declare integer variable for loop
 Dim i As Integer

‘ start i = 1
 i = 1

‘ loop 12 times. Once for each month
 Do While i <= 12
 Worksheets.Add after:=ActiveSheet

‘ name the new sheet to current month

‘ first time the loop runs MonthName(i) = JANUARY (i=1)
 ActiveSheet.Name = MonthName(i) ‘ MonthName
returns the month based on the numeric value passed

‘ Add header to Worksheet
 Range("A1").Value = UCase(ActiveSheet.Name)
 & " BUDGET"

‘ Format Header
 With Selection
 .Interior.Color = RGB(0, 112, 192)
 .Font.Color = RGB(255, 255, 255)
 .Font.Bold = True
 .Borders(xlEdgeBottom).Weight = xlMedium
 End With

```
' change column width  
Columns("A:A").AutoFit
```

```
' increment i for the next loop  
i = i + 1  
Loop
```

```
End Sub
```

**JOIN OUR FREE WEBINAR ON
MACROS & VBA >>**

**ACCESS OUR ONLINE COURSE:
XTREME MACROS & VBA >>**