

Excel VBA and Macros

Learn more about Excel:

Lynda.com – go to <https://skokiellibrary.info/resources/online-training-tutorials/> and click on Lynda.com. Create a free account with your library card number and a password of your choosing. Enter a search term in the search bar and find an interesting class.

Goodwill Free Learning – go to <https://www.gcflearnfree.org/topics/office2016/> and scroll to Excel 2016. Basic to intermediate level lessons.

Books

Alexander, Michael, and John Walkenbach. *Excel VBA Programming for Dummies*. Wiley, 2019. www.dummies.com/go/vbaprogrfd5e code files

Urtis, Tom. *Excel VBA: 24-Hour Trainer*. Wrox, a Wiley Brand, 2015. www.wrox.com/go/excelvba24hour and <http://bit.ly/VBAcode24> code files

Websites

<https://www.excel-easy.com/vba.html> excellent VBA tutorial

<https://docs.microsoft.com/en-us/office/vba/api/overview/excel> VBA reference

<https://www.myonlinetraininghub.com/blog> - paid and free classes and tutorials.

<https://excelchamps.com/blog/useful-macro-codes-for-vba-newcomers/> code snippets to try out

<https://excelchamps.com/blog/vba-to-create-pivot-table/> pivot tables with VBA

<https://chandoo.org/wp/excel-vba/examples/>

<https://chandoo.org/wp/excel-vba/user-forms/>

<https://www.excel-easy.com/vba/userform.html> Create userform tutorial

<https://goalkicker.com/VBABook/> - A good resource for programmers



Identify if there are macros in a workbook

Excel 2016 file extensions we will use today:

.xlsx	Regular Excel file
.xlsm	Excel Macro-enabled file
.xltn	Excel Macro-enabled Template
.xlsb	Excel Binary Workbook – your Personal Macro Workbook is this file type.
.xlam	Excel Add-in
.bas	BASIC file – for exported macros

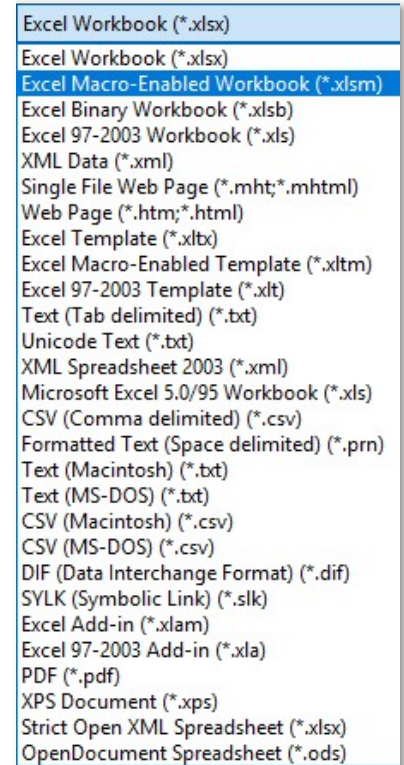


Figure 1 - supported Excel file types

Change the File Explorer to display file extension

1. Open File Explorer
2. On the View tab, put a check mark in the File name extensions box. Press Enter.

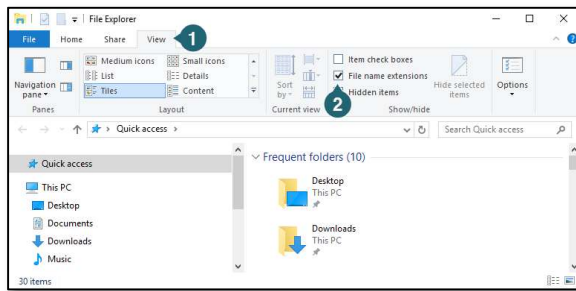


Figure 2 - update View tab to see file extensions

Add in the Developer Tab in Excel

1. File > Options > Customize Ribbon
2. Check the Developer box
3. Click OK

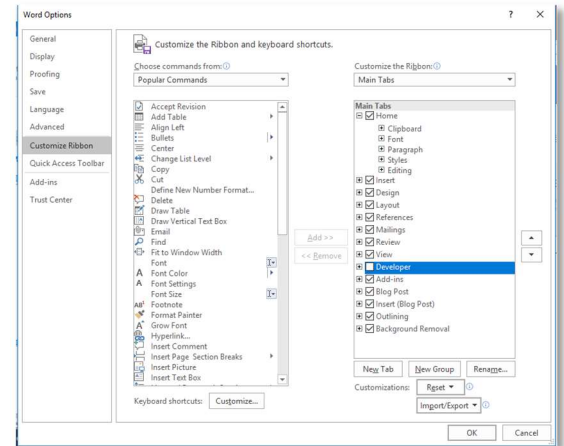


Figure 3 - Add the Developer tab

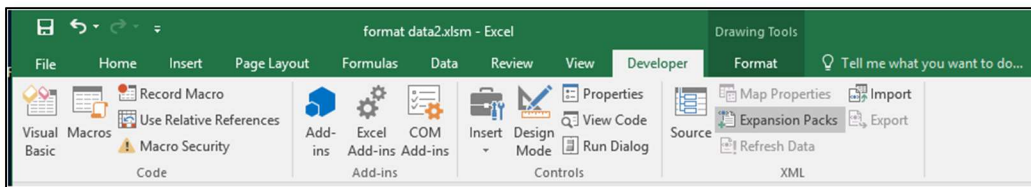


Figure 4 - Find all Macro related controls in the Developer tab





Figure 5 - another way to know the file has macros. Click Enable content if you trust the source.

EXERCISE 1 - RECORD YOUR FIRST MACRO

Recording a macro is similar to recording a video on your phone. Get everything up, press “record”, name the macro, do stuff, and press “Stop”. Then you can review the results (in Exercise 2).

Open **VBA01.xls** workbook and follow the on-screen directions

Start recording a macro in one of three ways:

1. In the status bar at the bottom left corner of the screen, you'll see the word Ready. Left click the icon (see Figure 6)
2. Or View > Macros > Record macro.
3. Or Developer > Code > Record Macro.



Figure 6 - bottom left of screen

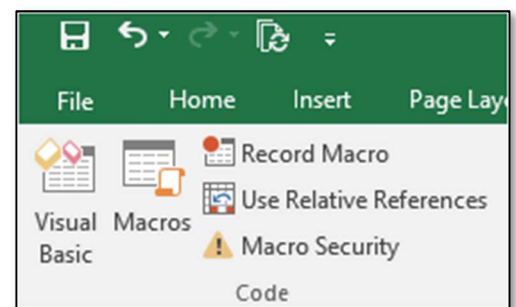


Figure 7 - in the Developer tab

Record Macro Dialog box:

Rename the default name with something meaningful

- No spaces or periods
- Upper and lower case matters!
- First character must be a letter
- Can't use #, \$, &, @, ^, *, !
- CamelCase is what I prefer.

Can leave the shortcut key blank for now. (Click Edit button to assign a shortcut later.)

Store macro in Personal or This Workbook

Description – Give a little overview about what the macro does. Very useful to maintain code.

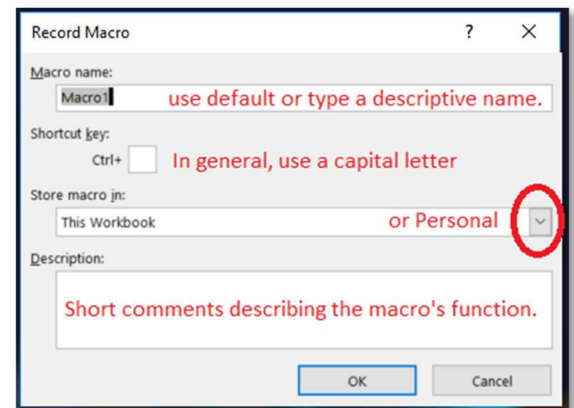


Figure 8 - the Record Macro Dialog box

EXERCISE 2 – VIEW AND MODIFY VBA CODE

VBE – Visual Basic Editor is where you can read, test, and modify VBA code.

Get to the VBE by doing one:

- Alt+F11 (or FN+Alt+F11 if you are on a laptop or Mac)
- Developer > Code > Visual Basic

(Example may differ from your program)

Project Explorer

The framework of the open programs and associated modules. Files are collapsible – click on the + or – just left of an object to display its contents. If you already have a Personal Macro Workbook, it will also appear here.

Handle Modules within the VBE

To create a module:

1. Click on the desired workbook
2. Click on Insert > Module
3. Click on that module to open it in the Code window. Type or paste code. Or File > Import a .bas module from a trusted source.

To delete a Module:

1. Right click on the module you want to delete
2. Choose “Remove Module”. It will give you a choice to back it up first, or not.

To export a Module:

1. Right click on the module you'd like export (won't delete original module).
2. Click on Export. Choose a location for the .bas file.
3. You can open the .bas file with the Notepad or other text editor.

Modules hold three types of code:

1. Declarations (of variables)
2. Sub procedures – aka “macros”.
3. Function procedures.

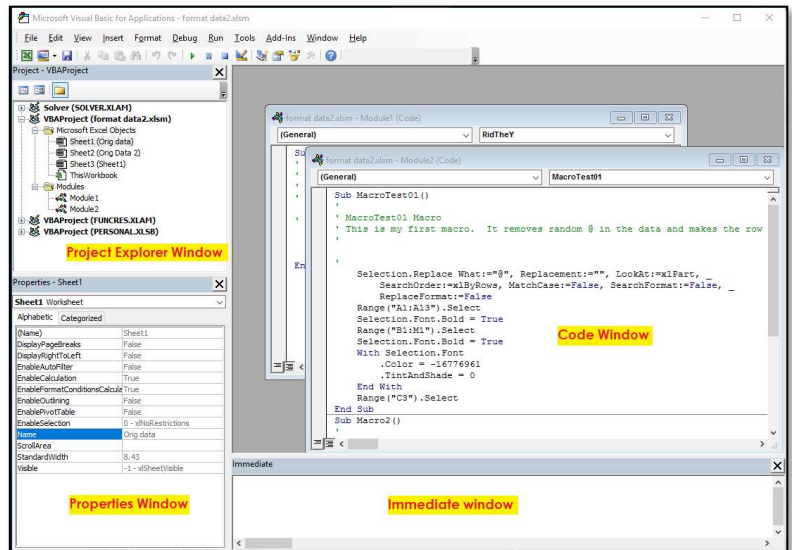


Figure 9 - the Visual Basic Editor or VBE

Code Window

The Code window can have zero, one, or many modules open at once. To make a comment, type a single apostrophe, ' Comments turn **green**. Certain reserved words are **blue**.

Most code is **black** (if it is ok) but can turn **red** if there is a syntax error.

Background highlighted in yellow means it is the execution point, used in debugging.

Background highlighted in maroon is a break point, also for debugging.

Properties Window

Here you can update macro names and properties Used in Userforms.

VBE menu and toolbar

Notice the Debug and Run commands.

Standard Visual Basic Editor Shortcut Keys

F stands for the Function key, usually at the very top row of the keyboard. If you are on a PC or Mac, you may have to press an FN key by the Ctrl key, at the same time as the specific function key.

Shortcut	What it does
Alt+F11	Toggles between the VBE and Excel windows.
Shift+F10	Displays the active window's shortcut menu (replicates right clicking).
Ctrl+R	Opens Project Explorer.
F4	Opens the Properties window.
F2	Opens Object Browser.
F1	Opens VBA Help.
F7	Activates the open module window.

Compiles the current Visual Basic project.

F5	Runs the current procedure or continues after pausing.
Ctrl+Break	Halts the currently running procedure.
F8	Goes into debug mode and executes one line at a time.
Ctrl+F8	Executes code up until the cursor.
Shift+F8	Steps over the current line while in debug mode.
F9	Toggles a breakpoint for the currently selected line.
Ctrl+Shift+F9	Clears all breakpoints.
Alt+D+L	Compiles the current Visual Basic project.

- From Excel Programming for Dummies



EXERCISE 3 – SAVE FILE AS .XLSM

Share your macro

By default, newly created macros are attached to one workbook.

<http://bit.ly/PersMacro>

Personal Macro Workbook

If you want to use your macro in many workbooks, you can copy it to Personal.xlsm on your computer (or network profile) File will be at:

`C:\Users\YourUser\AppData\Roaming\Microsoft\Excel\XLSTART`

(Hidden Items needs to be selected in your files explorer.) If you don't have a Roaming folder use the Local folder. Rest of the path should be the same.

Copy from one worksheet to another worksheet

In the VBE, click Project Explorer. Click the module which holds the code you want to copy and drag it to the destination workbook. Or Copy and Paste the macro to it's new module in the same Workbook or into another Workbook. (Delete the original macro if that's logical.)

Export module

In the VBE, select the module you want to share. File > Export press enter. Save the .bas type file to a known location, such as a USB drive. To import the macro to another workbook, in the VBE, File > Import and select the .bas file.

Save as an Excel Add-in

<https://trumpexcel.com/excel-add-in/>

Trust settings

When you open an .xlsm file, normally you will get the yellow warning banner asking if you want to trust the file, and another warning banner asking if you want to enable the macro. If you are unsure of the second question, don't enable the macro, but go to the VBE (`Alt+F11`) and examine the code. If you have a regular set of files with macros that you use, you can set the Trust Settings for them at File > Option > Trust center. See <http://bit.ly/VBAtrust> for full details.

Open VBA02.xlsx and do:

Exercise 4 - Copy macro

Exercise 5 – save macro to PERSONAL.XLSM



Three types of modules (from Excel VBA programming for Dummies):

1. **Declarations** – one or more information statements that you provide to VBA, such as Data Types.
2. **Sub procedures**: a set of programming instructions that, when executed, performs some action. They begin and end with Sub ... End Sub.
3. **Function procedures**: a set of programming instructions that returns a single value (similar in concept to a worksheet function, such as SUM). They begin and end with Function ... End Function.

In VBA03.xlsx, **Exercise 6 – Create a function**

Automatic procedures and events

An event is something that happens in Excel. A workbook is opened, a window is activated, data is entered into a cell. There are workbook and worksheet events. Code is written in an event-handler procedure and is stored in the Code window of the Object module, not in the regular VBA module.

In VBA04.xlsx, **Exercise 7 – Create an Event-Handler Procedure.**

Where to go from here?

As discussed at the beginning of class, I have skipped over:

- Variables, constants, data types
- Declaring and scope of variables – DIM
- Processing loops
- Arrays
- Objects and collections
- R1C1 notation and properties
- Charts and Pivot Tables
- Event handler, in depth <- Lynda.com course is excellent
- Security, Trust Center
- Using VBA with other Office applications

Consider taking the Lynda.com online class through the library's website. And try out <https://www.excel-easy.com/vba.html> excellent tutorial

mnelson@skokielibrary.info

[Linkedin.com/in/martha-nelson](https://www.linkedin.com/in/martha-nelson)



5215 Oakton Street / Skokie, IL 60077 / 847-673-7774 / www.skokielibrary.info