Using Macros in Excel

Contents

1. Overview

A macro allows the user to specify a sequence of steps to be performed whenever the user invokes the macro. Thus, macros are useful for automating any repetitive task, allowing the user to save time and effort. They have a wide variety of applications, such as customizing the layout of worksheets to meet a company standard or helping to clean up data converted from another application into Excel.

Macros are stored in Excel in a programming language called VBA (Visual Basic for Applications). A macro can be created by writing a VBA program or by recording the steps as the user performs them. We will discuss only the latter type of macros here.

Using macros involves the following steps, which will be covered in the remaining document:

a. Displaying the **Developer**[1] tab

b. Recording a macro

c. Playing back a macro

In addition, the difference between absolute and relative references in macros will be illustrated.

In order to understand and follow these instructions, you need:

- Excel 2016 installed on a Windows PC
- A basic knowledge of Excel

2. Displaying the **Developer** Tab

Before using macros, the **Developer** tab must be displayed in the ribbon along the top of the Excel window (as shown in the yellow circle in *Figure 1*). Start Excel to check if the **Developer** tab is already displayed. If so, then go to the next section (*Section 3 – Recording a Macro*); otherwise, follow the instructions here.

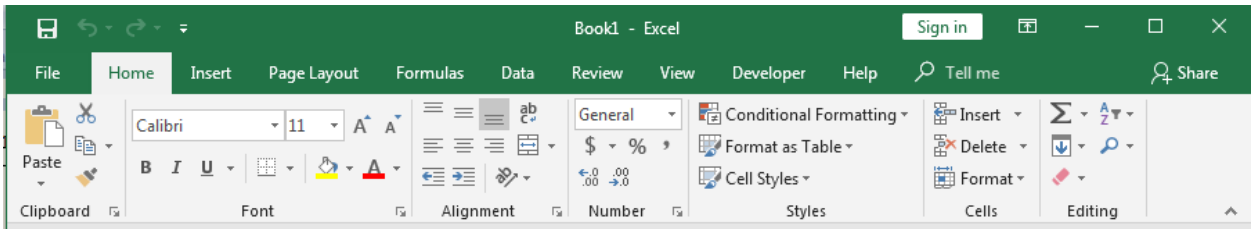

*Figure 1. The **Developer** Tab (circled in yellow)*

To display the **Developer** tab:

[1] The convention used in this document is to show in bold the words that you will find within Excel itself.

1. In the main Excel window, click the **File** tab and then near the bottom left, click **Options**. A window titled **Excel Options** appears.
2. In the list on the left side of this Options window, click **Customize Ribbon**.
3. In the list on the right, under the heading **Main Tabs**, select the **Developer** check box.
4. Press the **OK** button at the bottom of the Options window.

The **Developer** tab will now appear in the ribbon along the top of the Excel window.

3. Recording a Macro

When you record a macro, you are showing Excel what steps you want performed in the future by actually performing them. A sample application of macros will help to illustrate their usage. Imagine that you are an administrator in an elementary school and at the end of each term, you receive student marks from each grade in an Excel file in the format shown in *Figure 2*. You want to add the appropriate headings shown in *Figure 3*. Because you will be doing this for several grades and over several terms, you decide to create a macro to automate this task.

	A	B	C	D	E
1	Trudeau	Phoenix	2704	Math	88
2	Clark	Robin	6350	Math	100
3	Martin	Darryl	8358	Math	59
4	Turner	Lee	5225	Hist	62
5	Pearson	Sam	9590	Math	72
6	Campbell	Frankie	8659	Sci	94
7	Campbell	Frankie	8659	Hist	82
8	Pearson	Sam	9590	Sci	68
9	Chretien	Erin	1002	Music	52
10	Pearson	Sam	9590	Hist	56
11	Turner	Lee	5225	Math	66
12	Pearson	Sam	9590	Music	80

Figure 2. Excel File with Student Marks

	A	B	C	D	E
1	**Last Name**	**First Name**	**Student ID**	**Course**	**Final Mark**
2	Trudeau	Phoenix	2704	Math	88
3	Clark	Robin	6350	Math	100
4	Martin	Darryl	8358	Math	59
5	Turner	Lee	5225	Hist	62
6	Pearson	Sam	9590	Math	72
7	Campbell	Frankie	8659	Sci	94

Figure 3. Headings to be Added to Excel File

To follow along with this example, begin by creating an Excel workbook with student marks data similar to that shown in *Figure 2*. Also, create a copy of this workbook because you will use one workbook for recording the macro and the other for applying the macro.

The basic steps in recording a macro are: (1) start recording, (2) perform the actions to be reproduced later, and (3) stop recording.

1. Open one of the workbooks containing the student marks data that you have created. To start recording, click the **Developer** tab. Then, on the left side of the ribbon, click **Record Macro**. A **Record Macro** window appears, prompting you for the following information:

    a. **Macro name**: addHeadings[2] [Enter a suitable name for the macro's function.]

    b. **Shortcut key**: **Ctrl +** h [Optional. This is a fast way to later invoke this macro, but you must be careful not to accidentally override a pre-existing Excel shortcut that you use, e.g. "Ctrl + s" for saving a worksheet.]

    c. **Store macro in**: select **Personal Macro Workbook** [The choice depends on whether you expect your recorded steps to be used in other workbooks. The default option is **This Workbook**, which means that you would be able to use the macro only within the particular workbook where you record the macro. We will choose **Personal Macro Workbook** since we want to use this macro in other Excel files as well.]

    d. **Description**: This macro adds headings to student marks for one grade. [Optional. Enter text that makes it easier to recognize the function of the macro by others – or even yourself as you may forget this at a later time.]

    e. Press the **OK** button.

2. From this point until you stop recording, everything you do will be recorded by Excel as part of the macro.

[2] Underlined words indicate what you will be entering into Excel. These are followed by an explanation in brackets.

a. Insert a new empty row at the top of the worksheet.

b. Type the bolded headings shown in *Figure 3* into the new row.

c. Add a border line underneath the headings (as shown in *Figure 3*).

3. To stop the recording, click **Stop Recording** in the ribbon under the **Developer** tab.

Your macro is now saved and available for use in any Excel workbook. You will test the macro in the next section by applying it.

4. Playing Back a Macro

Playing back a macro means that every step you did while recording the macro will now be re-done – or "played back" – exactly as you did it then. To test the macro that you created in the previous section, start with the Excel workbook containing the original data (without headings). Press Ctrl-h (or whatever the keyboard shortcut you created for the macro) to invoke the macro. This will cause the heading row that you created earlier to be inserted at the top of the workbook.

There is another way to invoke a macro, which you will need to use when you do not create a keyboard shortcut for your macro. This also allows you to view all the macros that are available. Click the **Developer** tab, then from the left side of the ribbon click **Macros**. A new window appears with a list of macros, from which you can select the one we created and then press the **Run** button.

5. Absolute versus Relative References

So far, we have added headings only in the first row of a workbook using a macro. But what if the courses differed in the number of students and the student marks were listed under each course separately (*Figure 4*)? You want a row of headings under each course title (*Figure 5*). No matter where you place your cursor in the workbook, playing back the macro we created in the previous sections will cause a row of headings to be placed at the top only. This is because Excel macros, by default, use absolute references. This means that the macro actions will always occur in the exact cells where you recorded them (e.g. A1, B1 … in our previous example). But here we want the macro to add headings in row 2, row 11, and so on (*Figure 5*).

	A	B	C	D
1	HISTORY			
2	Chretien	Erin	1002	5
3	Harper	Bobby	5908	6
4	Clark	Robin	6350	9
5	Martin	Darryl	8358	7
6	Pearson	Sam	9590	5
7	Trudeau	Ashley	9681	8
8				
9	MATH			
10	Diefenbaker	Alex	1002	6
11	Trudeau	Phoenix	2704	8
12	St. Laurent	Carol	3258	6
13	Turner	Lee	5225	6
14	Harper	Bobby	5908	9
15	Clark	Robin	6350	10
16	Mackenzie	Pat	8358	5
17	Campbell	Frankie	8659	9
18	Pearson	Sam	9590	7
19	Trudeau	Ashley	9681	5
20				
21	SCIENCE			
22	Chretien	Erin	1002	9
23	Trudeau	Phoenix	2704	8
24	Mulroney	Jamie	3258	9

Figure 4. Student Marks by Course

	A	B	C	D
1	HISTORY			
2	**Last Name**	**First Name**	**Student ID**	**Final Mark**
3	Chretien	Erin	1002	59
4	Harper	Bobby	5908	60
5	Clark	Robin	6350	93
6	Martin	Darryl	8358	70
7	Pearson	Sam	9590	56
8	Trudeau	Ashley	9681	84
9				
10	MATH			
11	**Last Name**	**First Name**	**Student ID**	**Final Mark**
12	Diefenbaker	Alex	1002	64
13	Trudeau	Phoenix	2704	88
14	St. Laurent	Carol	3258	69

Figure 5. Headings by Course

The solution to this issue is “relative references”. With relative references, the actions are all executed relative to where the cursor is when the macro is invoked (as opposed to where you recorded the macro). If you were to place your cursor in cell A10 or A22 (*Figure 4*), the headings would appear in row 10 or row 22, respectively, instead of only row 1.

To use relative references, you must specify that you want to use relative references before you begin recording the macro. The remaining steps are very similar to those in the previous sections of the document. To follow the instructions below, create a new workbook with data similar to *Figure 4*.

To create a macro that uses relative references:

1. Click the **Developer** tab, then from the left side of the ribbon, click **Use Relative References** (*Figure 6*).

2. Click **Record Macro**. Refer to *Step 1* in *Section 3 (“Recording a Macro”)* for instructions on how to fill out the information in the **Record Macro** window.

3. Insert a new empty row after any course title.

4. Type the bolded headings into the new row (as shown in *Figure 5*).

5. Add a border line below and above the headings (as shown in *Figure 5*).

6. Click **Stop Recording** in the **Developer** tab ribbon.

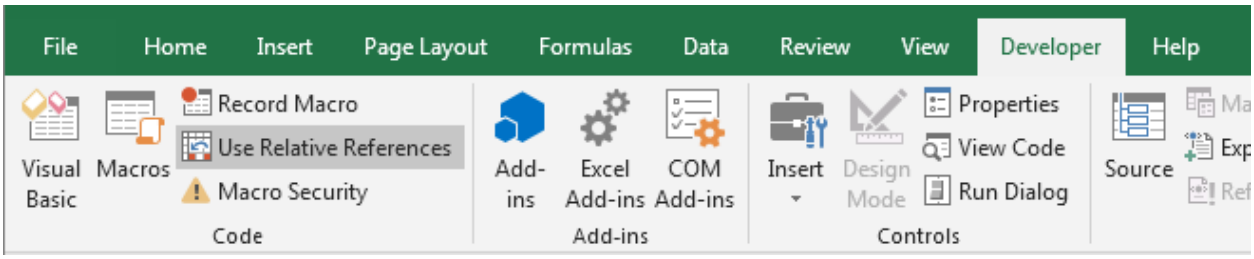

Figure 6. Relative References (circled in blue)

To test the new macro, place the cursor in your workbook in the row after any course name that does not yet have the heading row (e.g. A10 or A22) and then play back the macro (refer to *Section 4 – "Playing Back a Macro"*). A row of headings should appear that is exactly like the one you recorded, except it will be where you had placed the cursor in the current workbook and not in the original position where you recorded the macro.

6. Final Thoughts

Macros can be much more powerful than illustrated in the examples in this document. However, as useful as macros are, they can also be finicky because the results you get are sometimes not what you expect. For example, while recording a macro, you may do a copy and paste operation, but when you replay the macro later on, this will not work because the system clipboard no longer contains the same data that it did when you recorded the macro.

The topic of security with respect to macros was not discussed here. Saving a macro within a single workbook instead of the global **Personal Macro Workbook** will cause a warning when saving or opening the workbook. For this and more information on macros from the Microsoft website, google "site:microsoft.com excel macros 2016". One webpage with useful information, though only for Excel 2013, is "https://www.universalclass.com/articles/computers/excel/excel-2013-how-to-use-macros.htm". There are many other useful websites as well.

To do even more with macros, learning VBA will permit you to do many useful actions, including loops and conditional execution of steps.

Enjoy exploring!