COMP1022Q
Introduction to Computing with Excel VBA

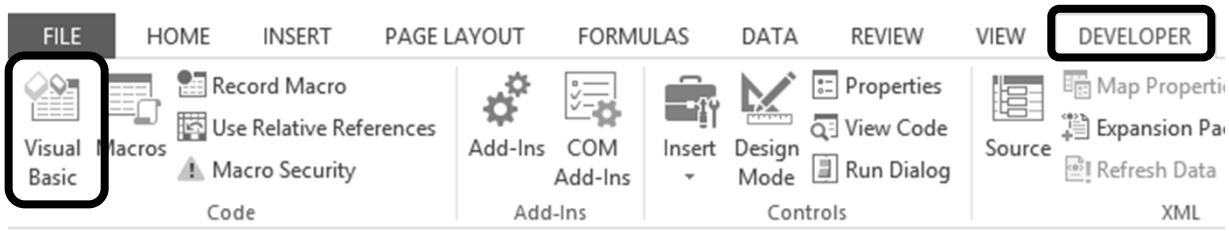
Beginning to Program VBA

Gibson Lam and David Rossiter

Outcomes

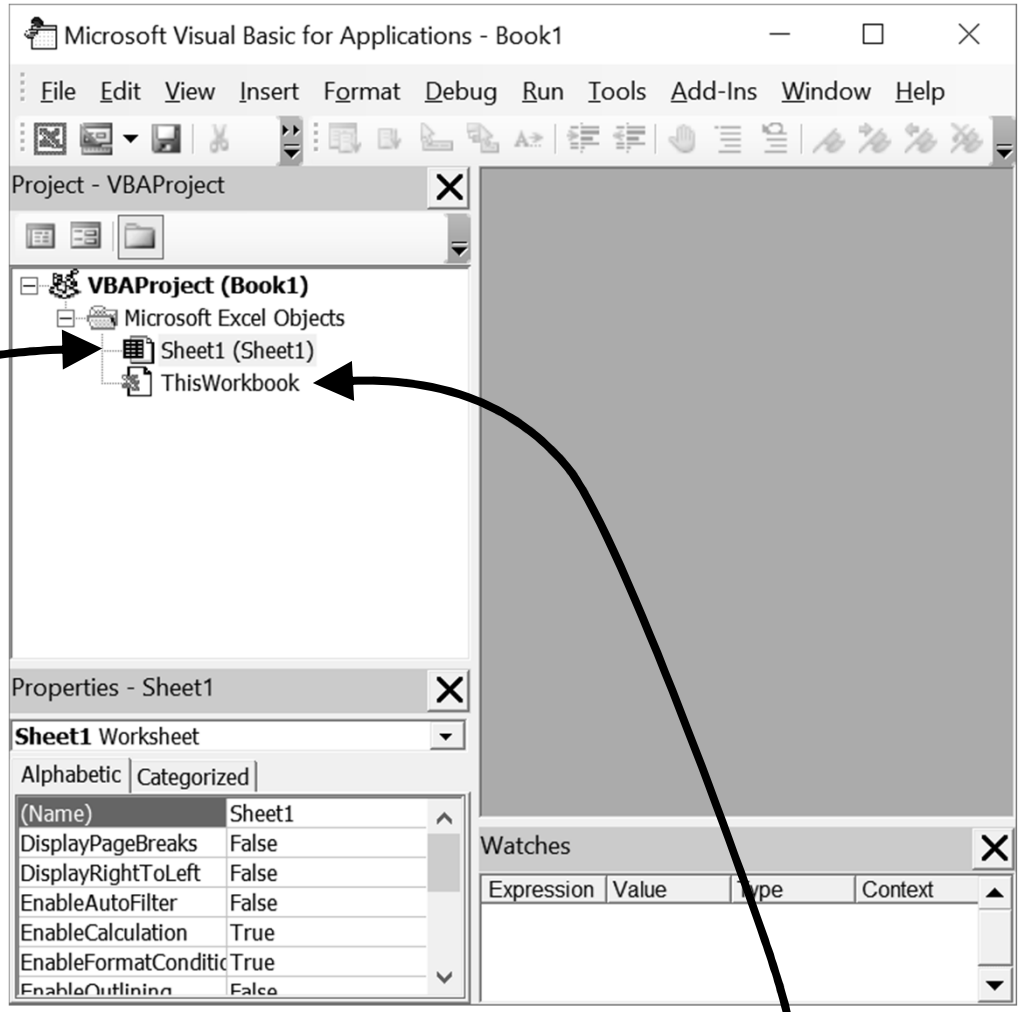
- After completing this presentation, you are expected to be able to:
  1. Access the VBA editor
  2. Describe the basic components in the VBA editor
  3. Create a very simple macro using VBA

The VBA Editor



- The VBA editor is where you see the VBA code included in the Excel file
- To see it, click the 'Developer' tab and then select 'Visual Basic'
- Alternatively, do *Alt-F11*
- However, the 'Developer' tab is **not** shown by default; to show it, select 'File > Options > Customize Ribbon', and ensure 'Developer' has a tick on it in the list on the right hand side

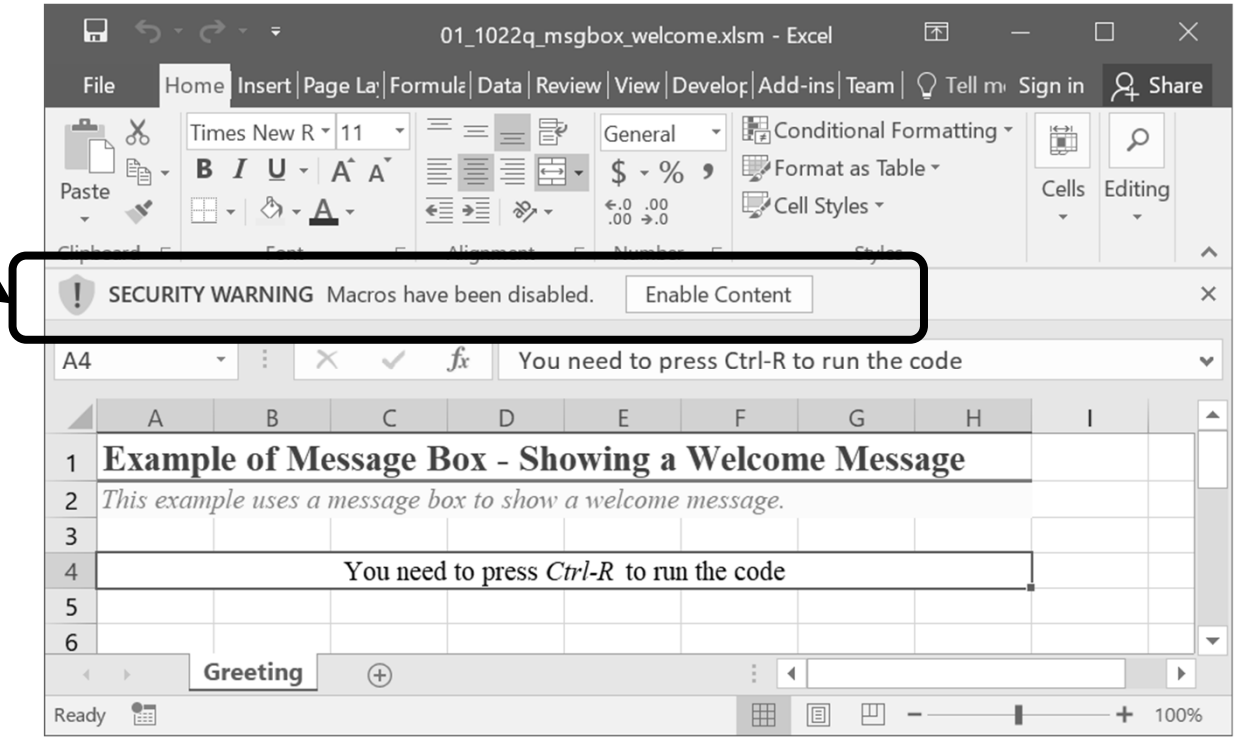
The VBA Editor



- The name of each worksheet is shown
- Each worksheet can have its own VBA code
- If you double click the name of the worksheet, the VBA code for that worksheet will be shown (if there is any)
- 'ThisWorkbook' (in other words, the Excel file) is also shown, this is for code which applies to the whole file (instead of just one worksheet)

Our First VBA Code

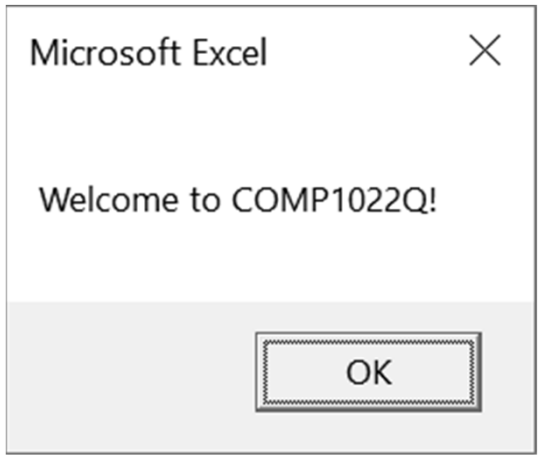
- Lets look at our first VBA code
- When we run it, we see this:
- Excel will show this warning whenever it loads a file which has some VBA code



- This gives you a chance to refuse to run the VBA code - you have to click ‘Enable Content’ before it can work

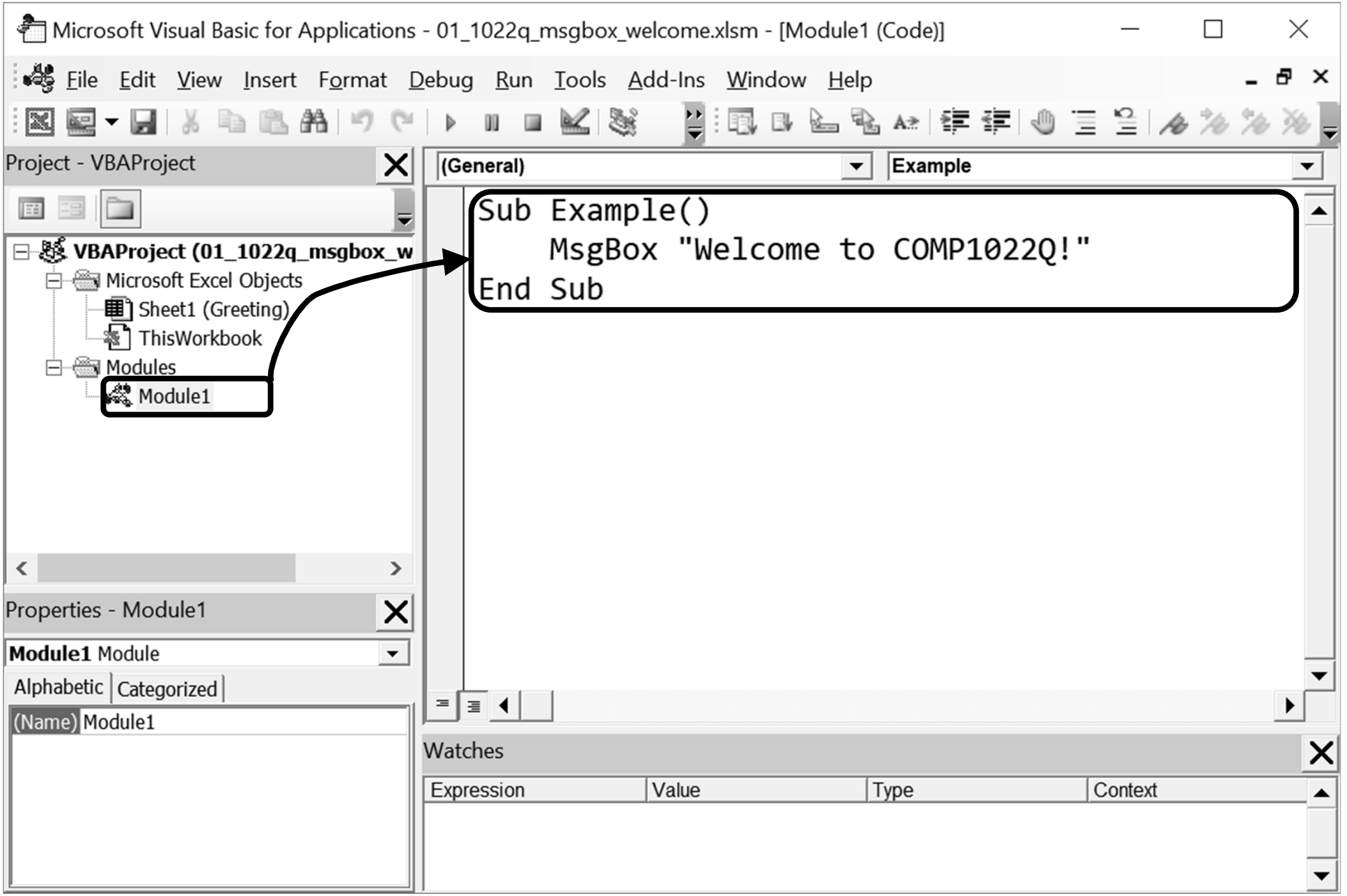
Running the VBA Code

- As soon as you give permission for the VBA code to run, the VBA can run
- In our first example, you can press Ctrl-R to run the code which shows a message:



- The code has been written in a *macro* called *Example*, inside a component called *Mobule1*

Our First VBA Code

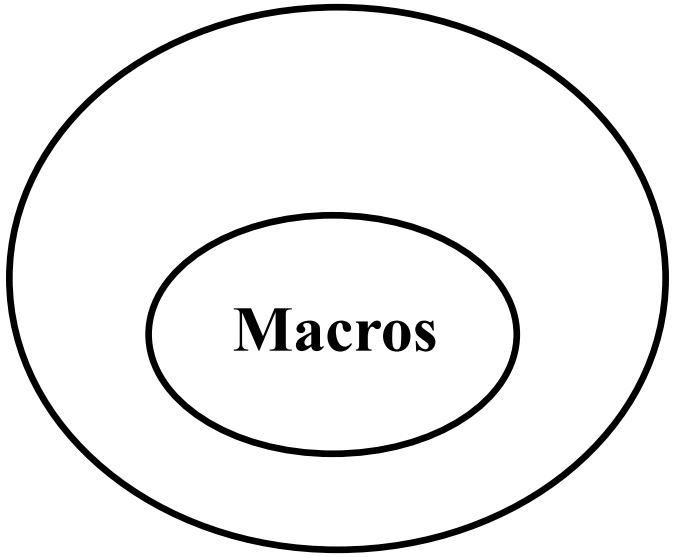


Macros

- *Macros* are pieces of VBA code that you can use many times
- A typical way to create a macro is to ‘record’ it
- If you know VBA programming you can also ‘write’ a macro, by typing the code
- In the rest of this presentation we will demonstrate how to write a simple macro
- Later in the course, we will also discuss how to record and customize a macro

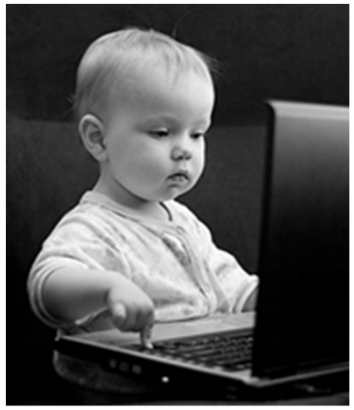
VBA Programming and Macros

- Basically, VBA programming can do anything, if you know how to program it
- VBA code can create/delete/modify almost anything at any time
- You might have a few lines of VBA code, or thousands of lines



- The general idea of a macro is to do something once, and that operation could be needed lots of times
- A macro is VBA code
- Often, a macro is a few lines of code
- You can record a macro or write the code for a macro, or both
- Usually a macro is triggered by a combination of keys e.g. Ctrl-Shift-A

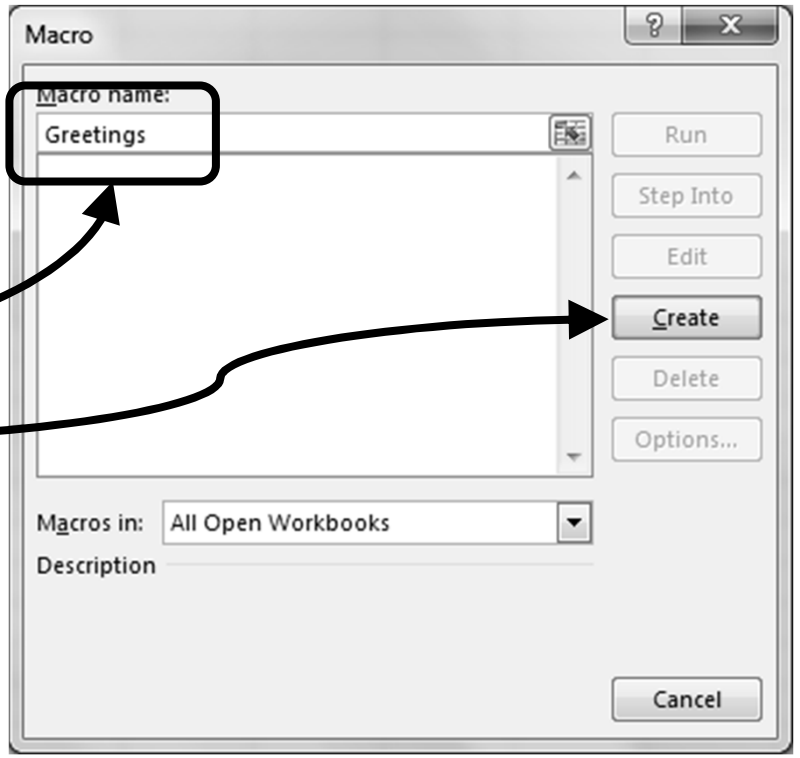
Running Macros



- After creating a macro, you can run the macro from the VBA editor
- Alternatively you can associate a shortcut key with a macro, i.e. if you press a specific shortcut key the corresponding macro will be run
- In the next few slides we will look at the steps to create a macro which contains a single line of VBA code, and then we will assign a shortcut key to that macro

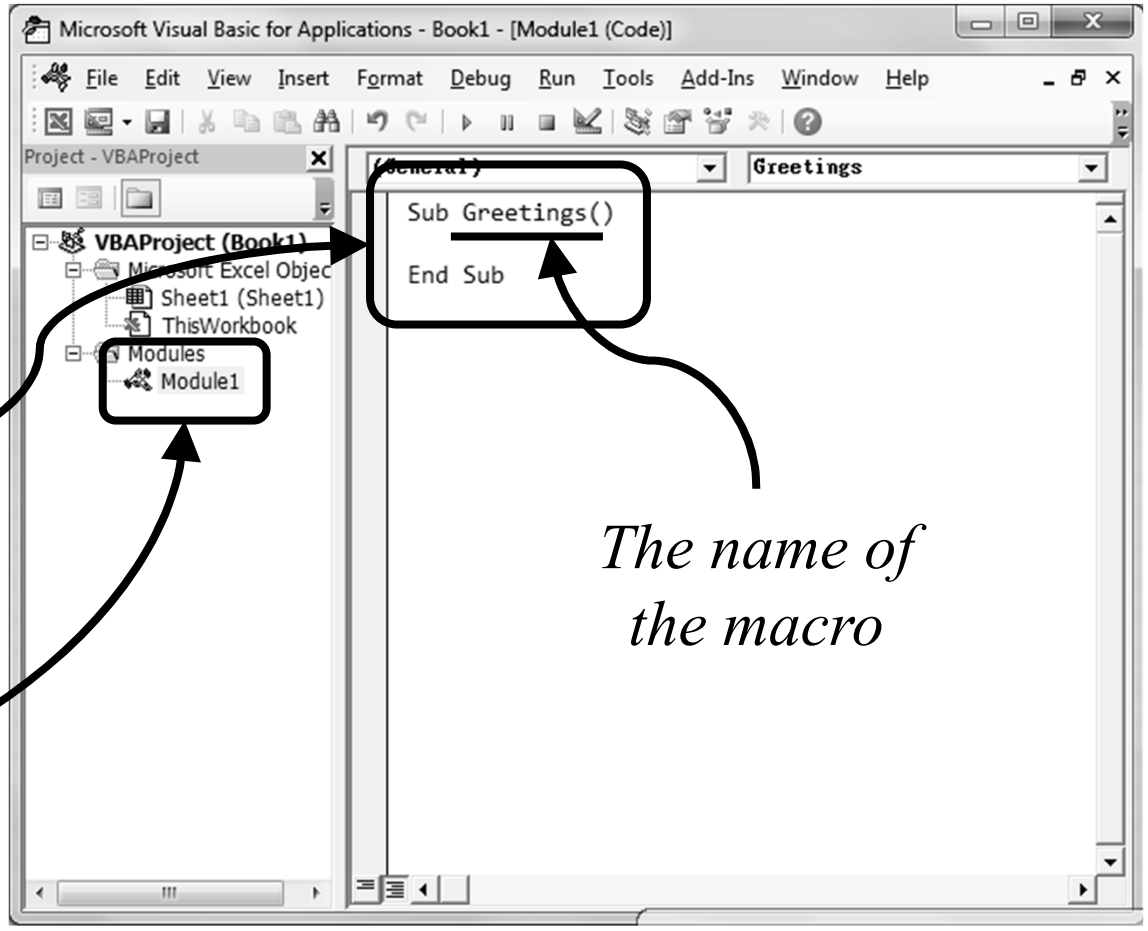
Creating an Empty Macro

- To create a macro, go to the *Developer* tab and then click on *Macros*
- In the macro window, enter the name of the macro and then click on *Create*
- There are some limitations to the name of a macro, e.g. the macro name cannot contain spaces



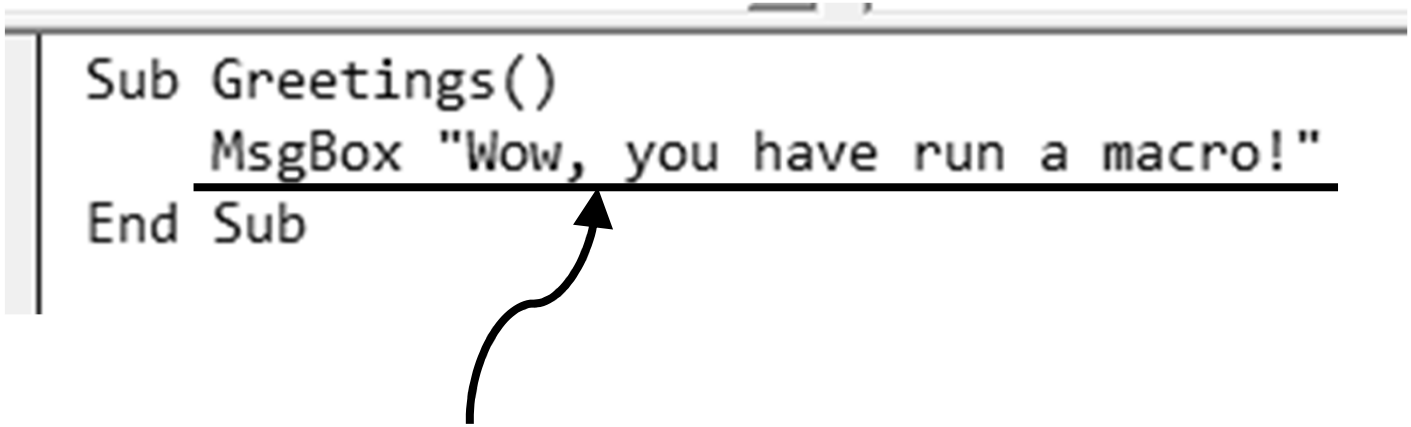
The Macro Subroutine

- A subroutine is a way to group VBA code
- After creating the macro we can see the macro subroutine in the VBA editor
- The macro is stored in a place called *Module1*



An Example Macro

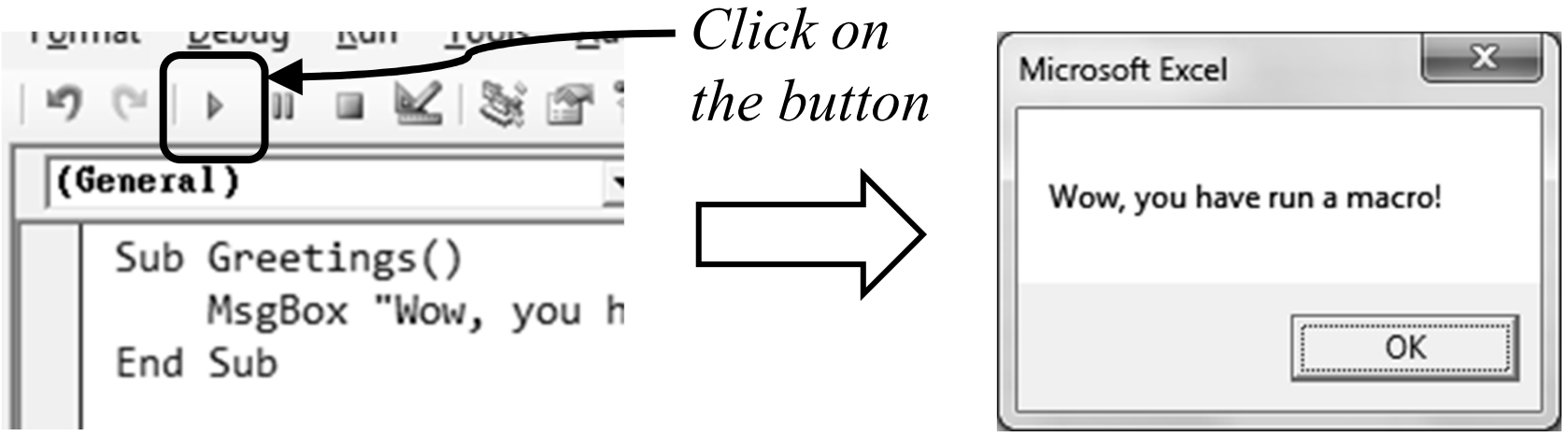
- Let's click inside the subroutine (i.e. inside the `Sub...End Sub`) and write one line of VBA code, like this:



- This line of VBA code shows a simple message box

Testing the Macro

- We can quickly run the macro by using the ▶ button
- To do that, we need to
  - put the text cursor inside the macro subroutine (i.e. click anywhere inside the subroutine code), then:
  - click on the ▶ button



Assigning a Shortcut Key to the Macro 1/2

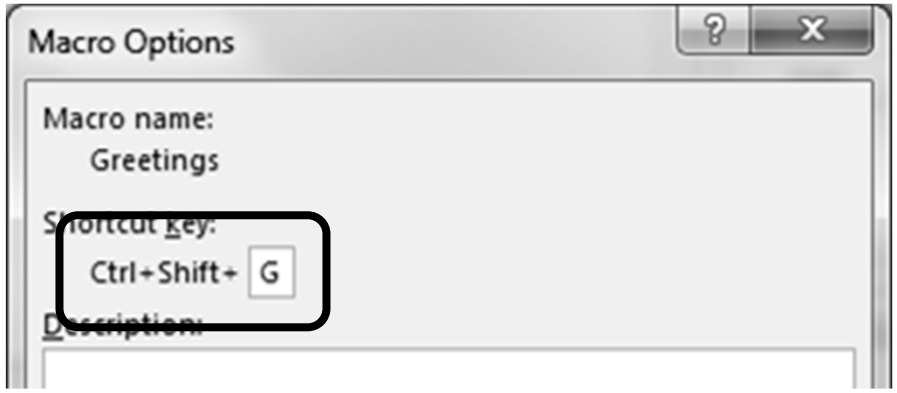
- We can assign a shortcut key to the macro so that Excel runs the macro when we press that shortcut key
- First, we need to open the macro window again, select the macro and then click on *Options*

Assigning a Shortcut Key to the Macro 2/2

- Inside the macro options window, enter the key we want the macro to be associated with
- The key can be combined with the control key or the control and shift keys, for example:

Enter the small letter 'g' to associate Ctrl-g with the macro

Enter the big letter 'G' to associate Ctrl-Shift-g with the macro



Running the Macro Code

- Let's assume we have used *Ctrl-g* as the shortcut key associated with our macro
- Then to run the macro we can press *Ctrl-g* anytime when we are on the worksheet