COMP1022Q
Introduction to Computing with Excel VBA

A First Look at Excel VBA

David Rossiter

The VBA 'Trigger'

- There has to be some kind of 'trigger' for VBA code to begin

- This signal is called an *event*

- There are lots of different possible events
  - An example of a keyboard event:
    - Pressing Ctrl-z on the keyboard
  - An example of a mouse event:
    - Clicking on something

Basic Idea of Excel VBA

The User   **Excel File**   **VBA**

1. The user does something
2. An event is triggered
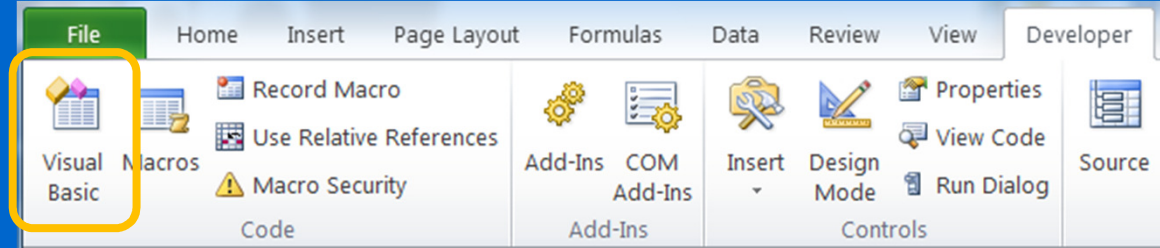3. The relevant VBA *subroutine* is executed

- A *subroutine* is a group of VBA code
- A subroutine which handles an event is called an *event handler*
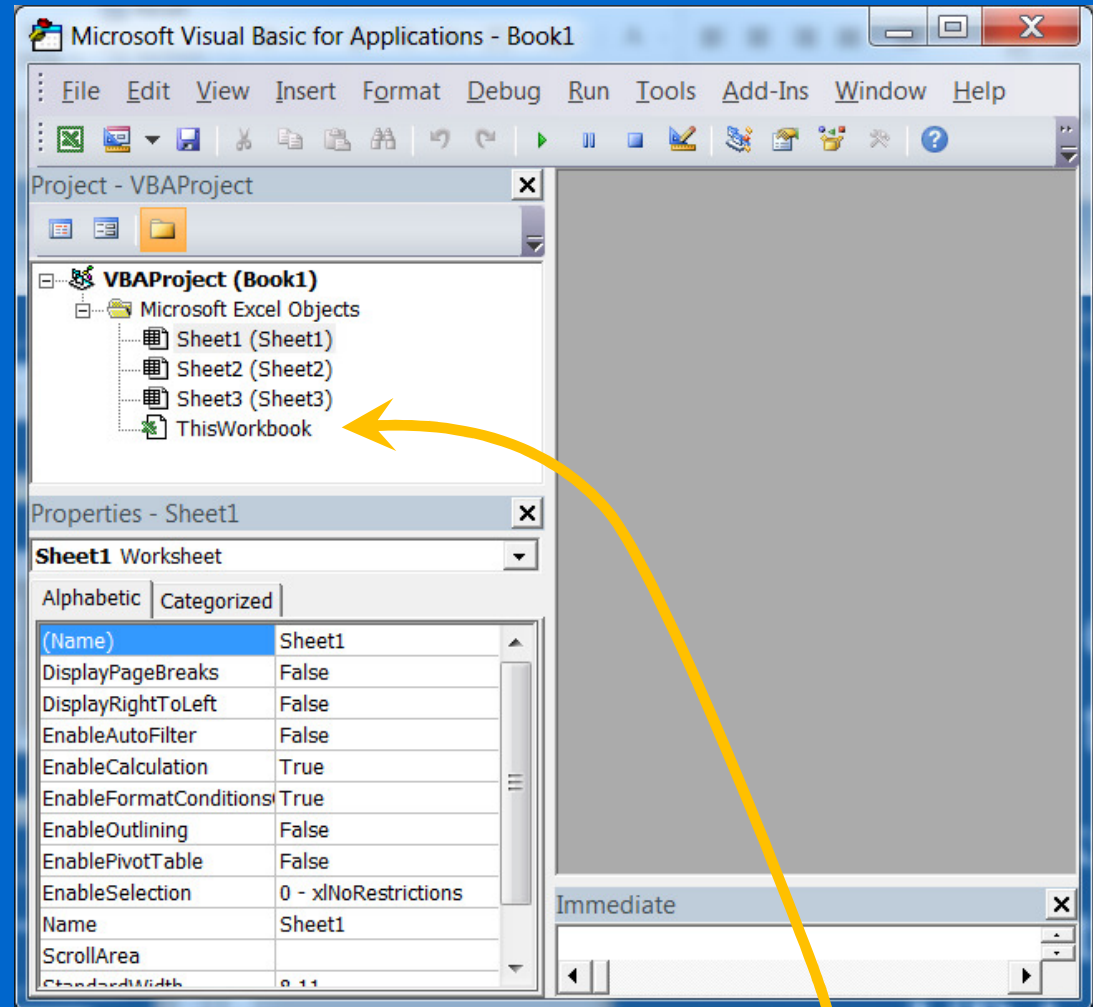
The VBA Editor

- The VBA editor is where you see the VBA code included in the Excel file

- To see it, select Developer > 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' is selected in the list on the right hand side
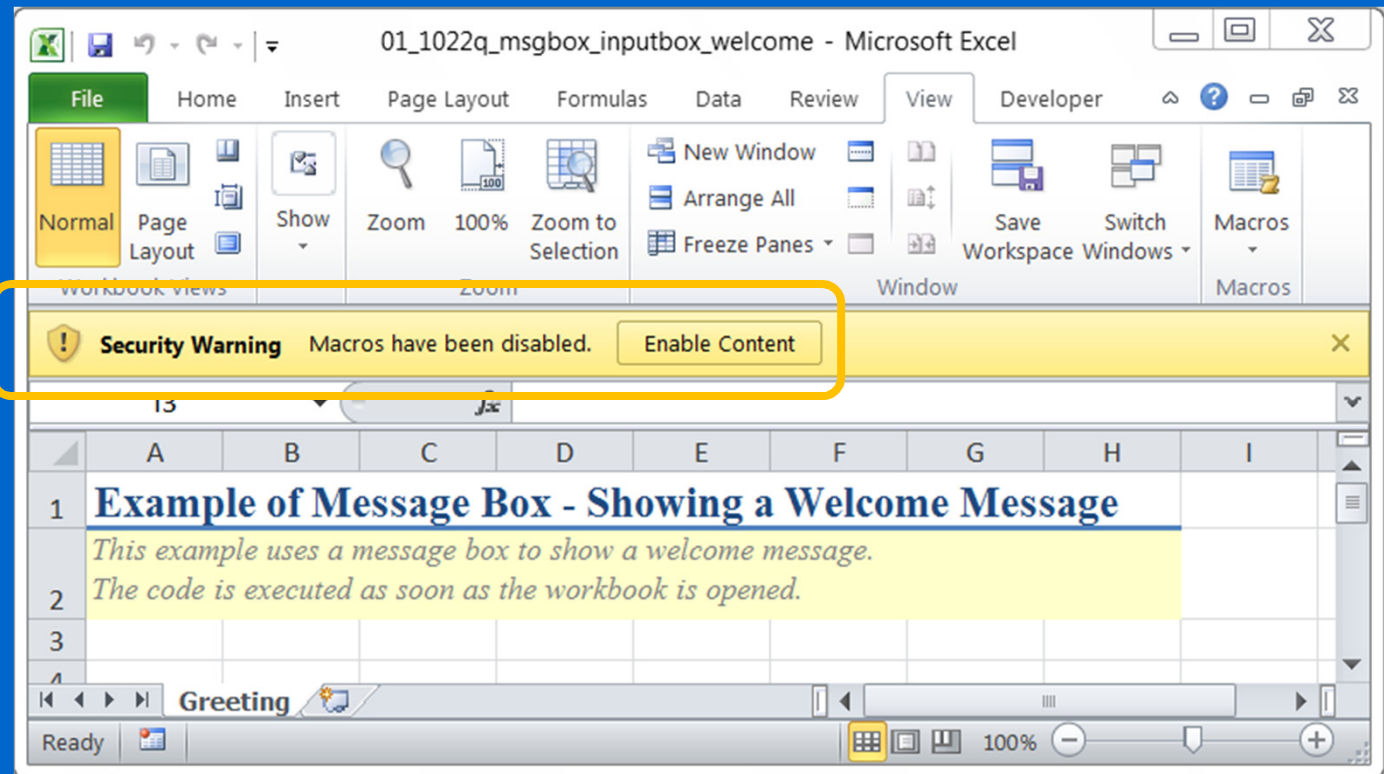
The VBA Editor

- 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
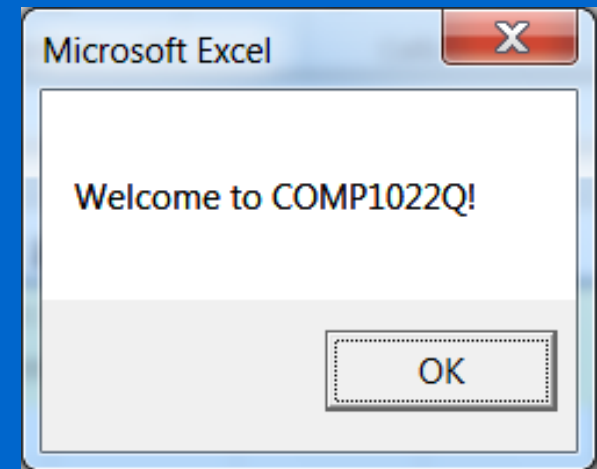
- When we run it, we see this:

- 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 be executed, the VBA can be executed

- In our first example, a message immediately appears:

- This is because an event took place, and the appropriate VBA event handler was executed

- The event was 'the workbook was opened' (in other words, the Excel file was opened)

Our First VBA Code