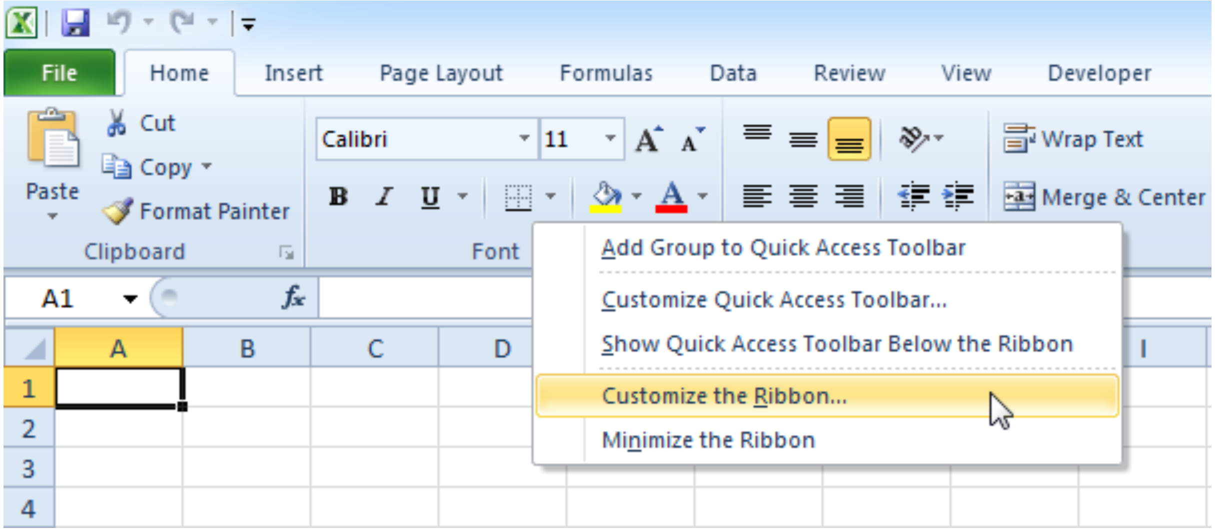
Create a Macro

With **Excel VBA** you can automate tasks in Excel by writing so called macros. In this chapter, learn how to create a simple **macro** which will be executed after clicking on a command button. First, turn on the Developer tab.

Developer Tab

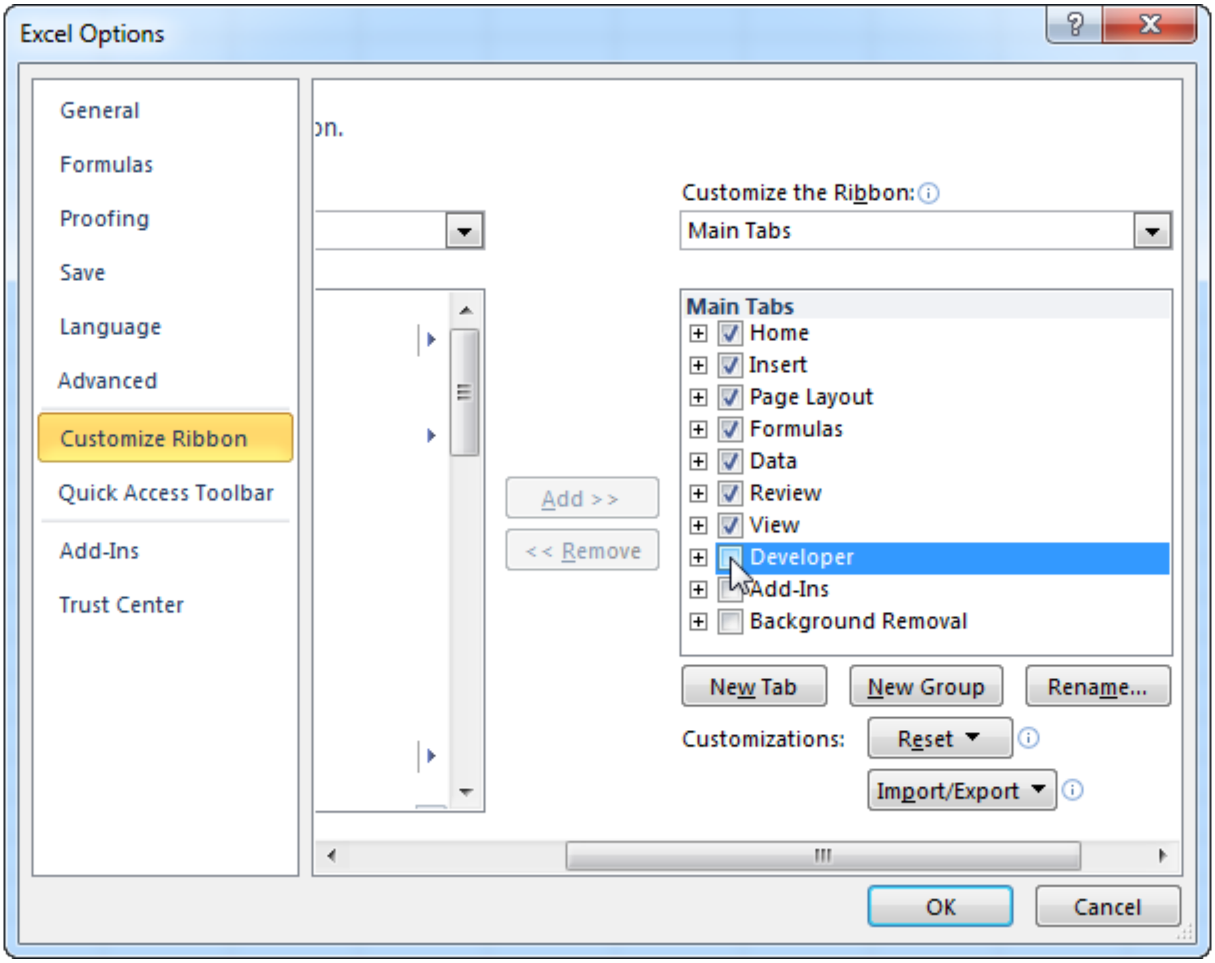
To turn on the Developter tab, execute the following steps.

1. Right click anywhere on the ribbon, and then click Customize the Ribbon.



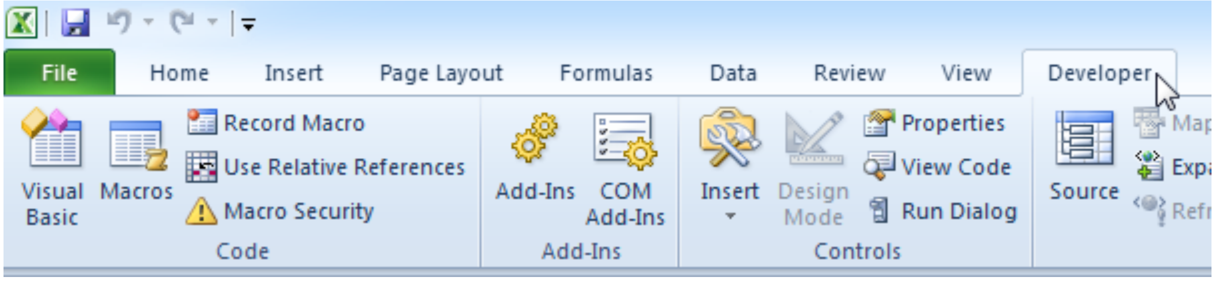
2. Under Customize the Ribbon, on the right side of the dialog box, select Main tabs (if necessary).

3. Check the Developer check box.



4. Click OK.

5. You can find the Developer tab next to the View tab.

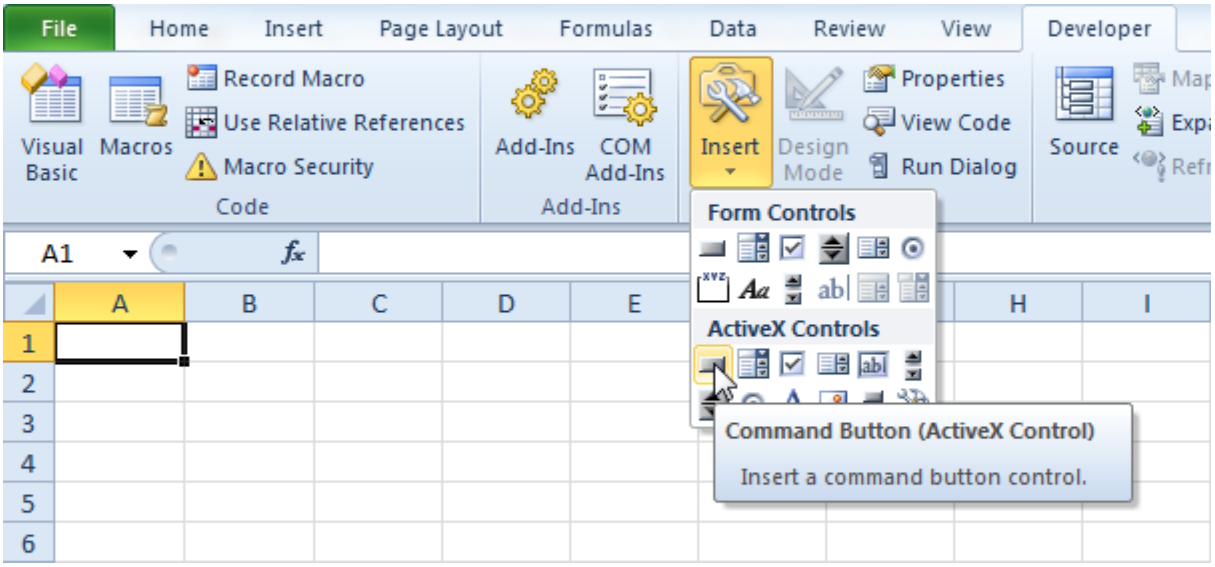


Command Button

To place a command button on your worksheet, execute the following steps.

1. On the Developer tab, click Insert.

2. In the ActiveX Controls group, click Command Button.



Command Button (ActiveX Control)

Insert a command button control.

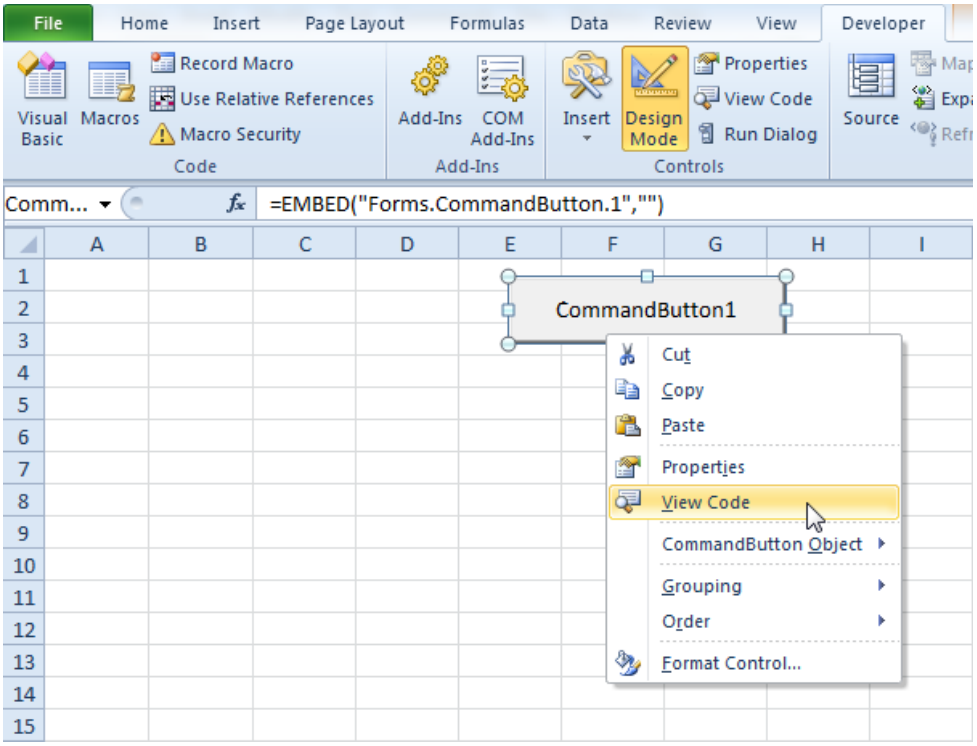
3. Drag a command button on your worksheet.

Assign a Macro

To assign a macro (one or more code lines) to the command button, execute the following steps.

1. Right click CommandButton1 (make sure Design Mode is selected).

2. Click View Code.

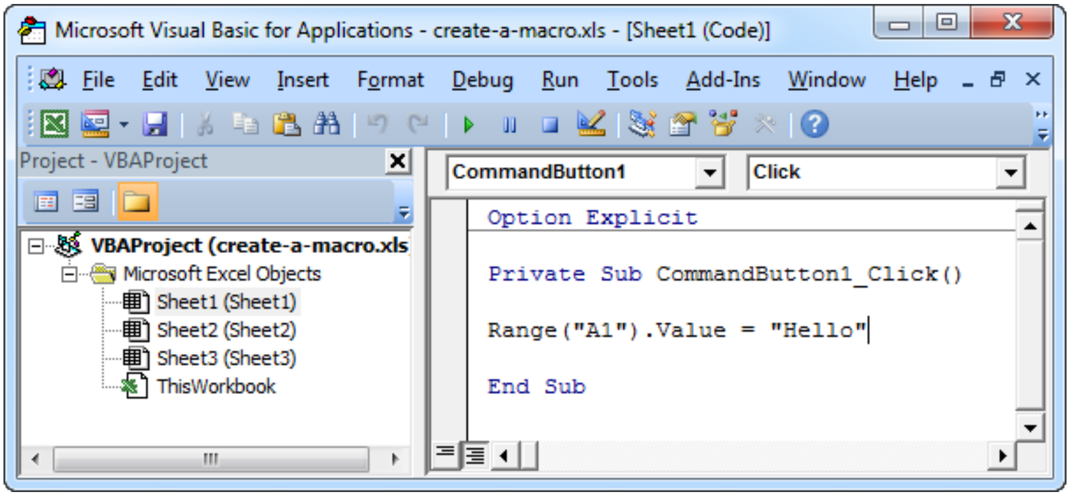


=EMBED("Forms.CommandButton.1","")

The Visual Basic Editor appears.

3. Place your cursor between Private Sub CommandButton1_Click() and End Sub.

4. Add the code line shown below.



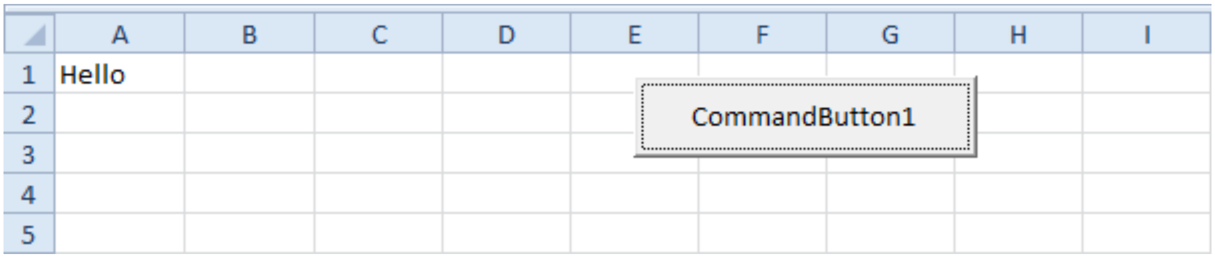
```
Option Explicit

Private Sub CommandButton1_Click()

Range("A1").Value = "Hello"

End Sub
```

Note: the window on the left with the names Sheet1, Sheet2 and Sheet3 is called the Project Explorer. If the Project Explorer is not visible, click View, Project Explorer. To add the Code window for the first sheet, click Sheet1 (Sheet1).

5. Close the Visual Basic Editor.

6. Click the command button on the sheet (make sure Design Mode is deselected).

Result:



	A	B	C	D	E	F	G	H	I
1	Hello								
2									
3									
4									
5									

Congratulations. You've just created a macro in Excel!

If Then Statement

If Then Statement | Else Statement

Use the **If Then statement** in **Excel VBA** to execute code lines if a specific condition is met.

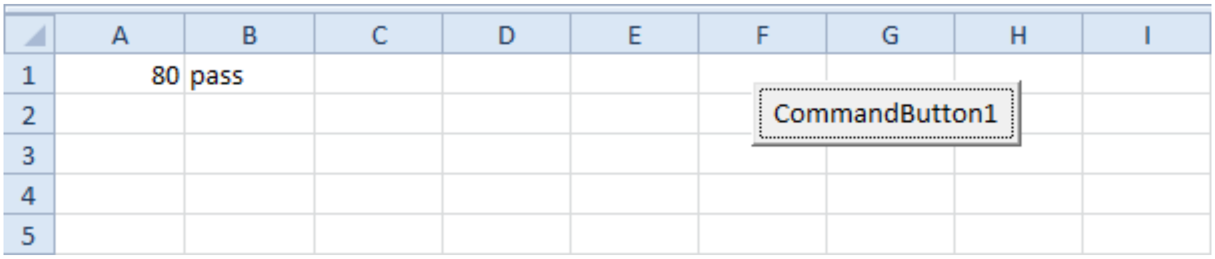
If Then Statement

Place a command button on your worksheet and add the following code lines:

```
Dim score As Integer, result As String
score = Range("A1").Value

If score >= 60 Then result = "pass"

Range("B1").Value = result
```

Explanation: if score is greater than or equal to 60, Excel VBA returns pass.

Result when you click the command button on the sheet:



	A	B	C	D	E	F	G	H	I
1	80	pass							
2									
3									
4									
5									

Note: if score is less than 60, Excel VBA places the value of the empty variable result into cell B1.

Else Statement

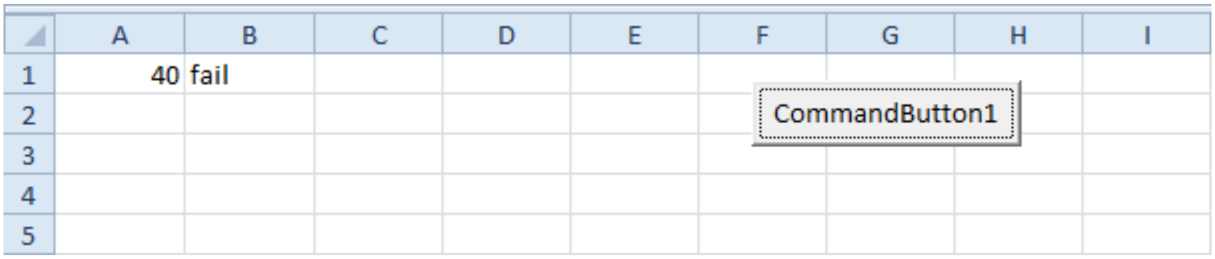
Place a command button on your worksheet and add the following code lines:

```
Dim score As Integer, result As String
score = Range("A1").Value

If score >= 60 Then
   result = "pass"
Else
```

```
    result = "fail"
End If

Range("B1").Value = result
```

Explanation: if score is greater than or equal to 60, Excel VBA returns pass, else Excel VBA returns fail.

Result when you click the command button on the sheet:



	A	B	C	D	E	F	G	H	I
1	40	fail							
2									
3									
4									
5									

Note: only if you have one code line after Then and no Else statement, it is allowed to place a code line directly after Then and to omit (leave out) End If (first example). Otherwise start a new line after the words Then and Else and end with End If (second example).