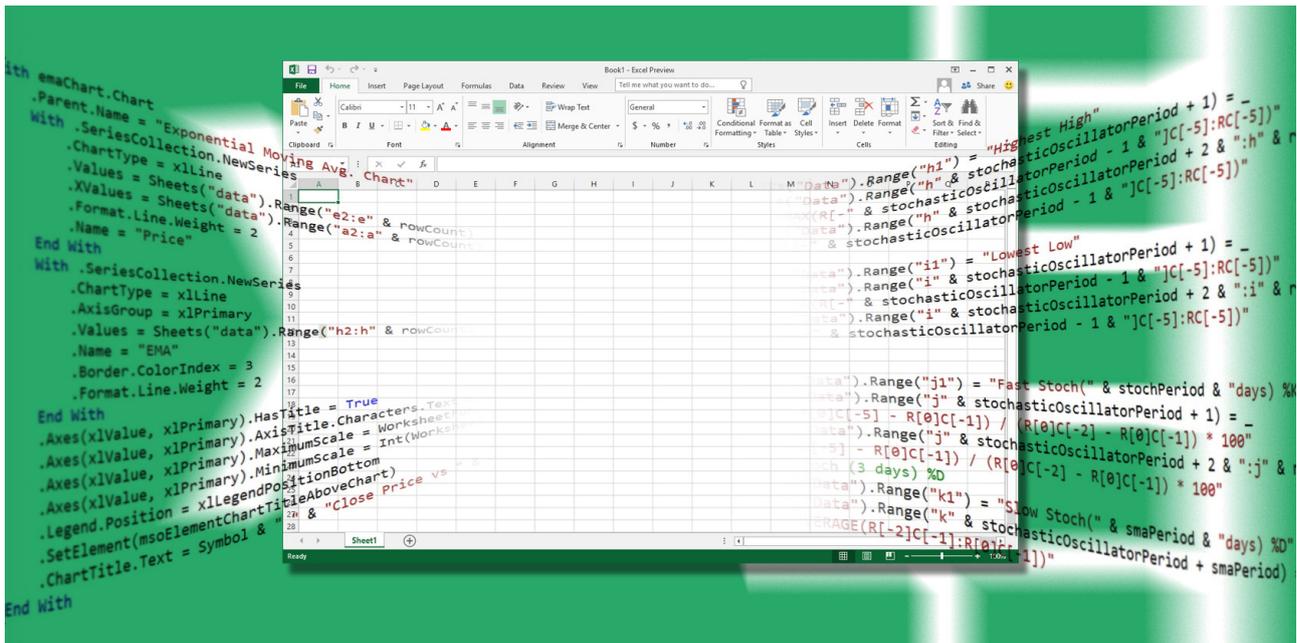


The Excel VBA Programming Tutorial for Beginners

By Brad Jones

Read the original article here: <http://www.makeuseof.com/tag/excel-vba-programming-tutorial/>



If you're using Excel, you have to try this power tool!

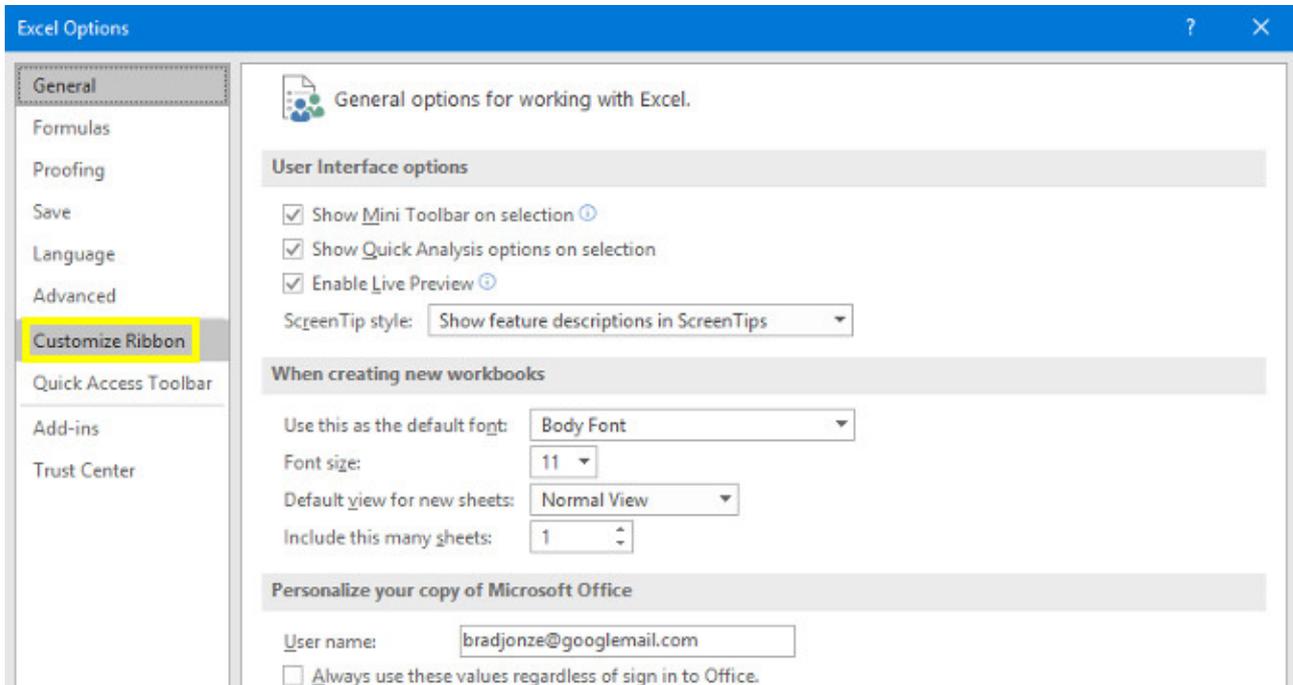
Visual Basic for Applications (VBA) is the Microsoft Office programming language that allows you to create macros and userforms, add a message box, execute code inside a document in response to a trigger, and much more. With VBA you can supercharge your Excel spreadsheets. And you just have to **learn a little bit about coding**.

This guide will help you try your hand at VBA with a simple project: a button that converts the value of a chosen cell from GBP to USD. We will introduce you to the ways that VBA and Excel can intersect. This short tutorial will put you on a path toward creating your own more complex projects.

Here's how to get started with VBA in **Excel 2016 (CA/UK)**.

Access Developer Controls

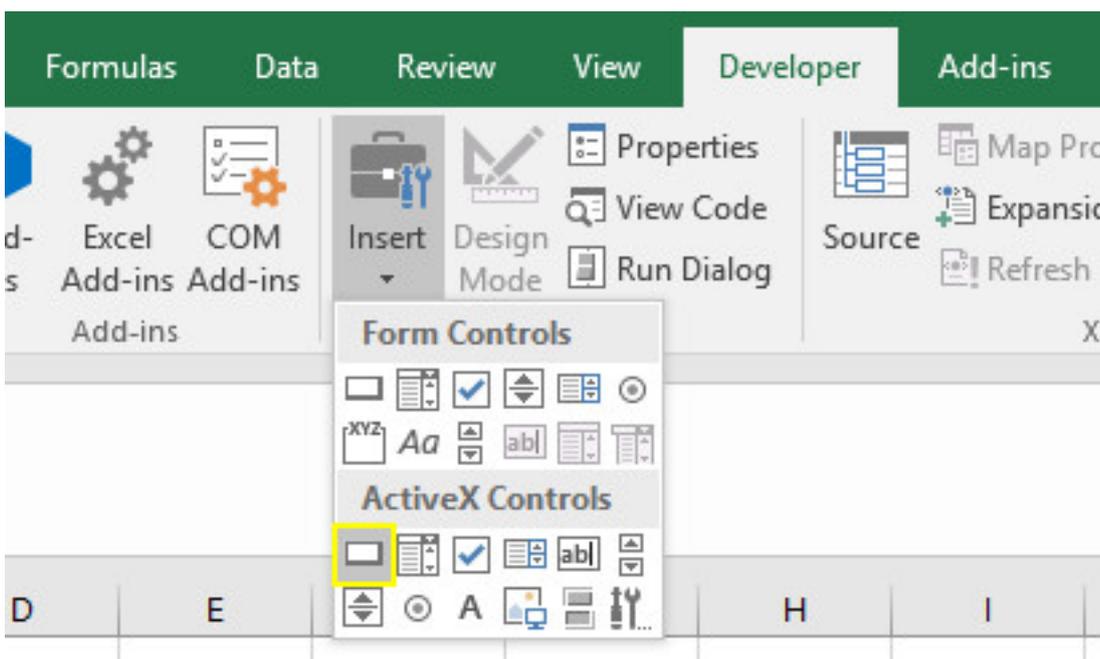
Before we can dive into VBA, it might be necessary to open Excel and adjust the settings to display the **Developer** tab as part of the Ribbon. To do so, head to **File > Options > Customize Ribbon**.



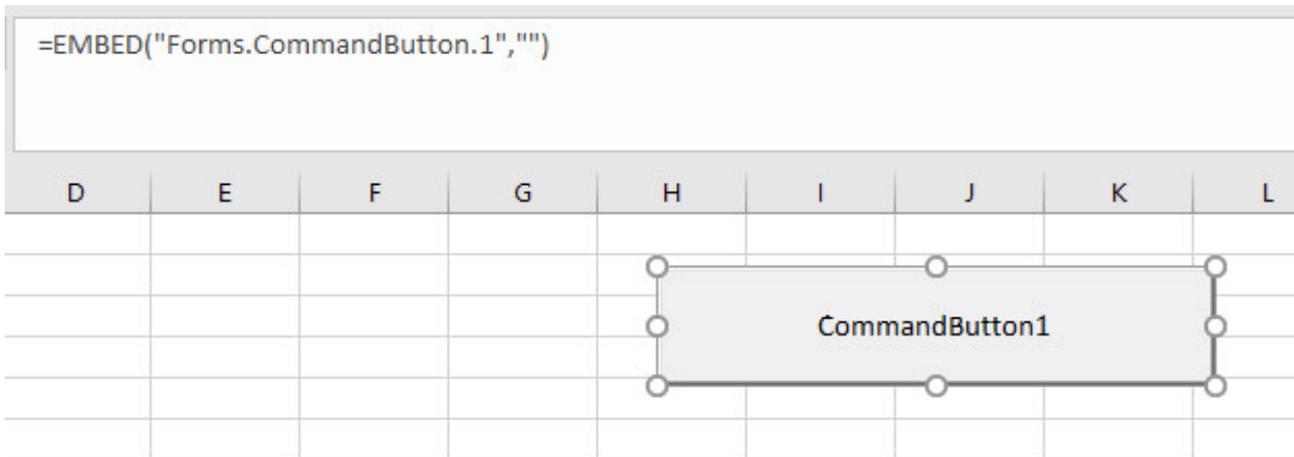
Create a Button

To create our currency converter, we first need to insert the button element. In a second step, we'll attach our VBA code to that button.

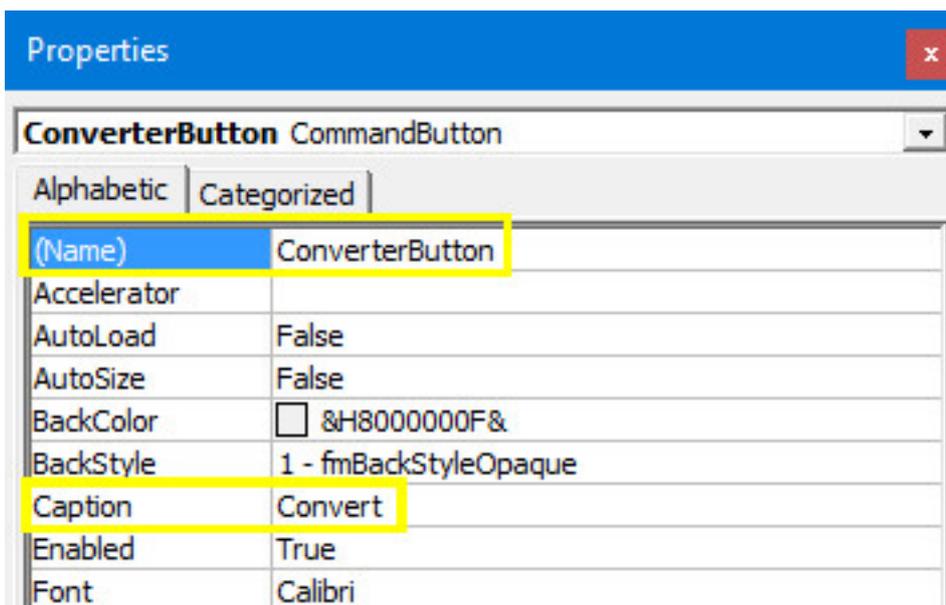
Open a new Excel spreadsheet, then navigate to the **Developer** tab. Use the **Insert** dropdown in the **Controls** section to select an **ActiveX Command Button**.



Drag the button out to an appropriate size and place it somewhere convenient – you can easily change this later on.



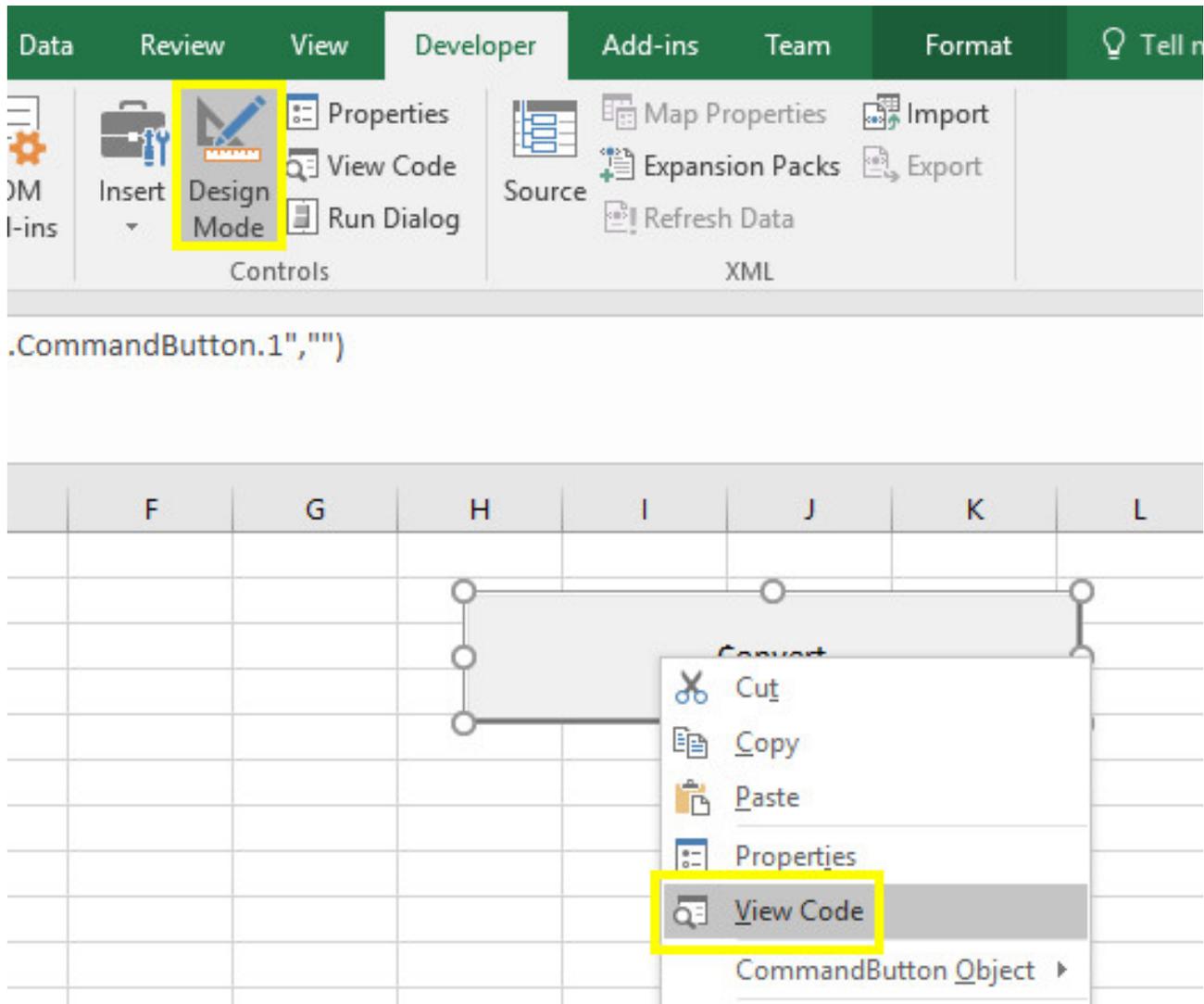
Now we'll attach the code. Right-click the button and select **Properties**. We'll make two changes; we're going to change the **Name** that we'll use to refer to the button **while coding**, and the **Caption** that displays text on the button itself. You can choose whatever you like for these labels, but remember that you'll need to swap out ConverterButton for whatever you use in its place while we're adjusting the code.



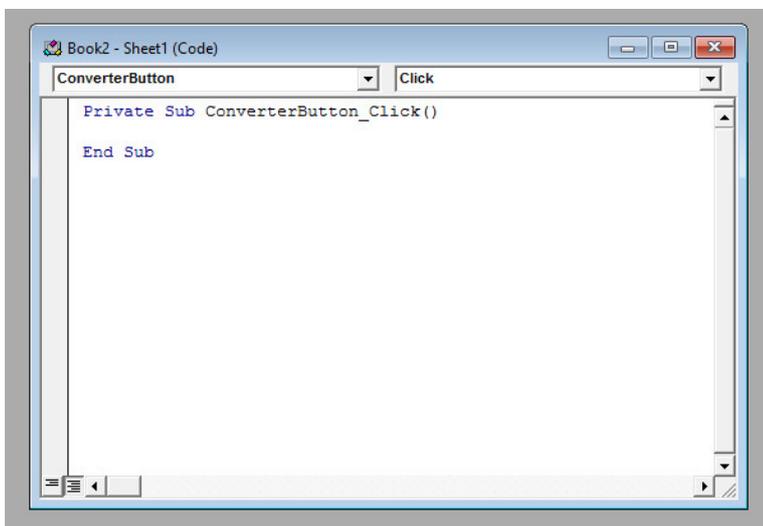
Now it's time to give the button some functionality.

Add Some Code

Coding with VBA takes place in a separate environment to the standard Excel interface. To access it, make sure **Design Mode** is active in the **Developer** tab, then right-click the button we created and select **View Code**.



You'll see a window like the one below:

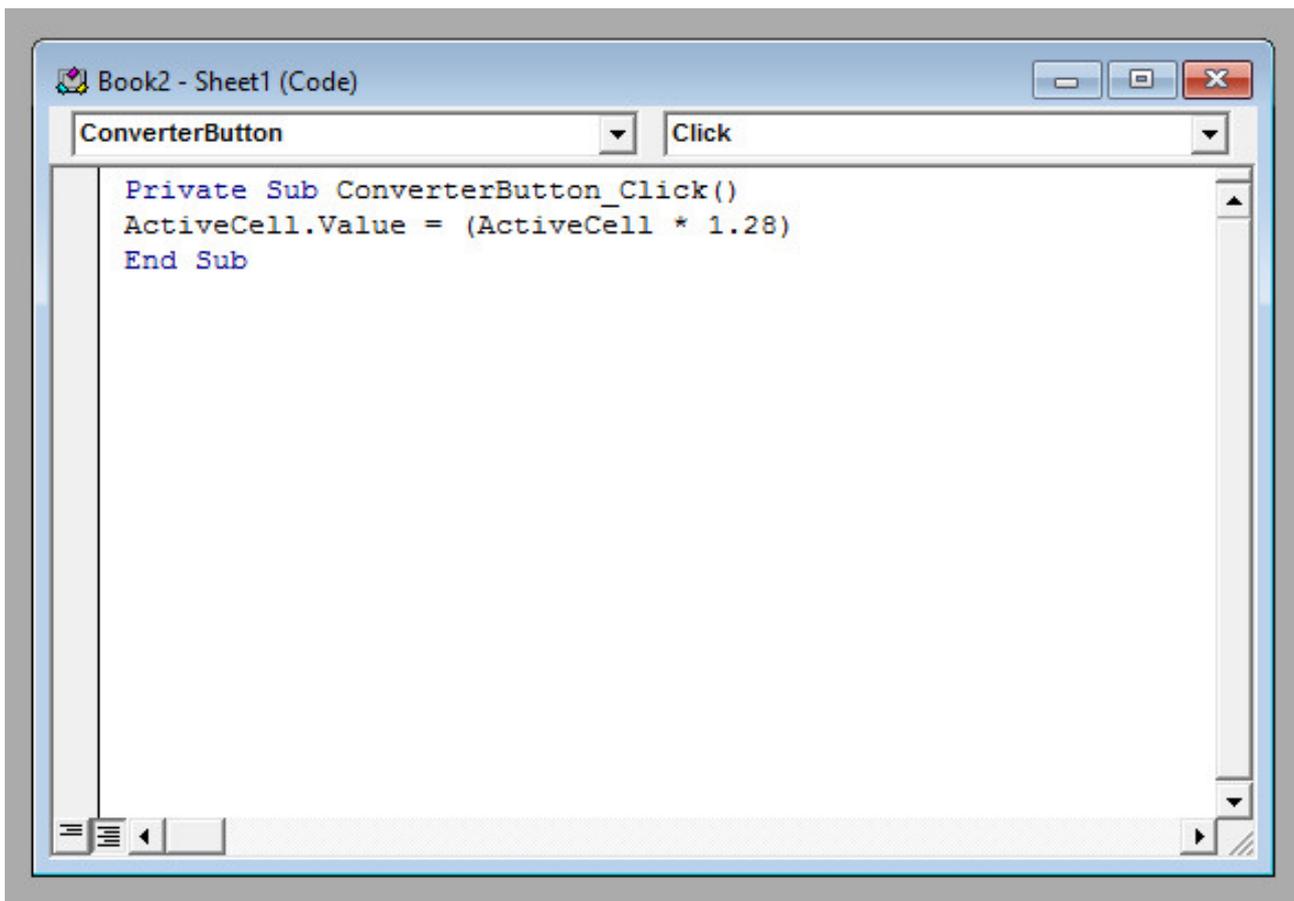


The start and the end of our code is already in place – the two blue pieces of text bookend our function, while the text in black states that we’re stipulating the action that should take place when the user **clicks on the button** we created. If you chose a different name to ConverterButton, you should see the respective term in your version of this window.

To carry out the currency conversion procedure, we’ll use the following line of code between the two that have already been created for us:

```
ActiveCell.Value = (ActiveCell * 1.28)
```

To break it down further, this piece of code states that the new value of the cell that the user has selected will be the current value multiplied by 1.28 – the exchange rate from GBP to USD. Here’s how that looks in the VBA window:



Next, close the VBA editor window and head back to Excel.

Test Your Work

It's now time to see whether our code works – but there's an important step to take before we can do that. We need to disable **Design Mode** to stop any further modifications to the button, and make it functional.



Next, enter a number into a cell, select that cell, and click your button to see it work its magic. Hopefully, you'll see the value increase by around a quarter, meaning that the conversion has been carried out correctly.

Next Steps

Now that you've created a button and used it to execute VBA code in Excel, you can use the same basic method to carry out all kinds of different projects. You might want to create a simulated die that returns a random value when you press the button, perhaps as **part of a larger game**. Alternatively, you could create a button that checks the contents of a specified cell against another one elsewhere in the same document.

Embarking on projects like these exposes you to the different features of VBA. Our currency converter is about as simple as it gets – but it's the first step towards bigger things. If you're just starting out, pick projects that relate your learning to a basic goal or task you're curious about. Step by step, you will become more familiar with how VBA works.

Do you have a question about this VBA project? Ask for assistance or offer up some help in the comments!

Read more stories like this at

