## **What is VBA in Excel?**

VBA stands for **Visual Basic for Applications**.

It is a programming language developed by Microsoft and has been incorporated into several major Office applications, such as Word, Excel, Outlook, and Access.

Visual Basic for Applications (VBA) is the programming language in Excel that can help you automate tasks, create your own functions (pretty cool), and even create applications.

Using VBA, you can do the following:

* **Automate repetitive Excel tasks**, such as formatting, filtering, and data entry. For example, if I get some data set regularly and I need to do the same steps to clean the data, I can create a VBA code that does all that as soon as the code is run.
* **Create custom Excel functions** (UDFs) to perform specialized calculations. While Excel does have a lot of functions, in case there is something you need that cannot be done with the built-in Excel function, you can create your own User Defined Functions using VBA.
* **Use Loops** that allow you to go through cells/rows/columns/sheets/workbooks/charts and do the specified task. For example, I can use VBA to go through all the worksheets in the workbook and change the formatting in each of the worksheets.
* **Work with Other MS Applications**. For example, I can write a VBA code to quickly import data from multiple Microsoft Word documents into an Excel sheet.

[**Excel VBA**](https://www.youtube.com/playlist?list=PLm8I8moAHiH2n5HC4ZXBgS-cBLjxWDreu).

**The Concept of Object Oriented Programming**

VBA is an Object Oriented Programming (OOP) language, which means that VBA works with objects such as Workbooks, worksheets, cells, rows, columns, charts, etc.

So when you are writing a code in VBA, you start with an object, and then you specify what change you want to make.

Let me give you an example.

Let’s say I want to change the color of cell A1 in the sheet named Sheet1 in the workbook named Example.xlsx

Below is the VBA code that will do this:

Sub ChangeCellColor()

Application.Workbooks("Example.xlsx").Sheets("Sheet1").Range("A1").Interior.Color = RGB(255, 0, 0)

End Sub

Here is what is happening in this code:

1. We start with the main object, which is the Excel application.
2. Within the Excel application, we then go to the workbook called Example.xlsx
3. Within that workbook, we then go to the sheet with the name Sheet1
4. Within the sheet, we then refer to the cell where we want to make the color change, which is cell A1
5. Within the cell, we refer to the interior object that will allow us to make changes inside the cell
6. Finally, we refer to the color property of the cell Interior object that allows us to change the color to red.

This is a really simple example to explain object-oriented programming, but it encapsulates the essence of how it works.

We always start with the main object, and then we keep zeroing in till we reach the object in which we want to make the change (which was the cell interior in the above example).

Once we have the object in which we want to make the change, we then use the property/method of that object to make the change. For example, in this case, we used the color property to change the color of the cell.

To give you an example outside the programming world, imagine You have sent a letter to your friend. Now, for that letter to be delivered, the post office is going to first check which country the letter needs to be delivered.

Once the letter is in the country, they would check which state or city the letter needs to be delivered.

When the letter is in that city, the delivery person will check the locality, followed by the street address and the house number.

This is just like our example above, where we first started with the Excel application, then we zeroed into the workbook, then the worksheet, then the cell in the worksheet, and then the color interior object that we needed to change.