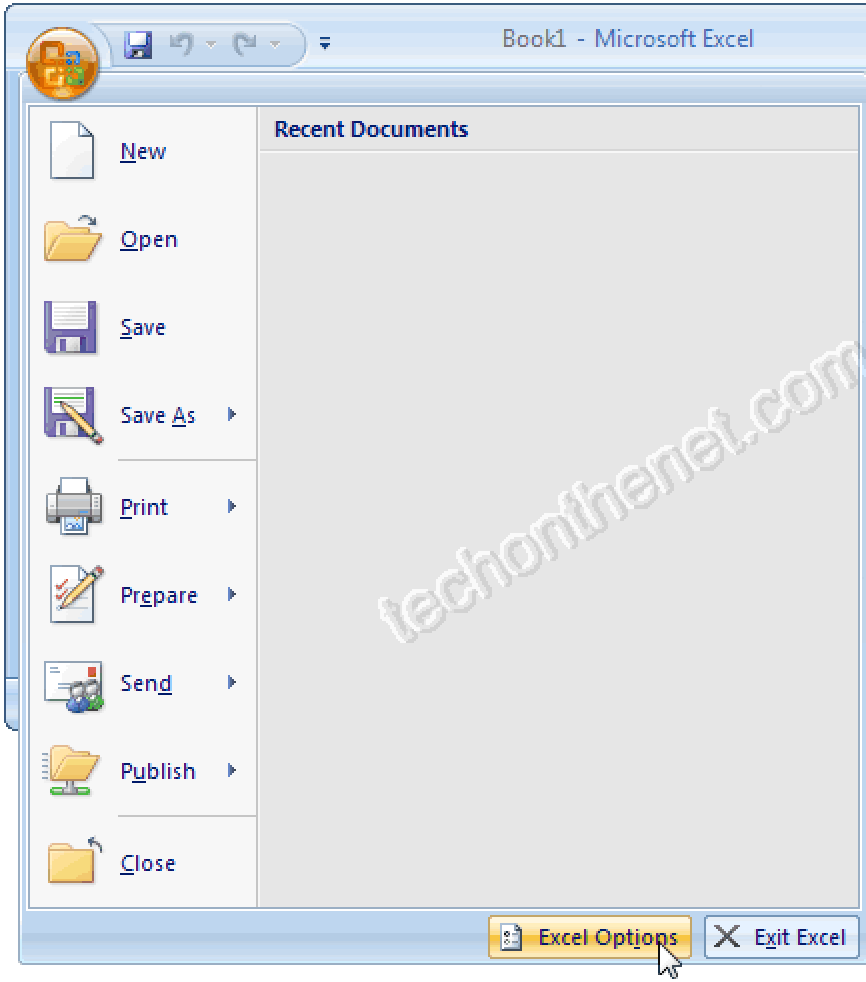
# VBA for MS Excel

Ahmad AL Kawam

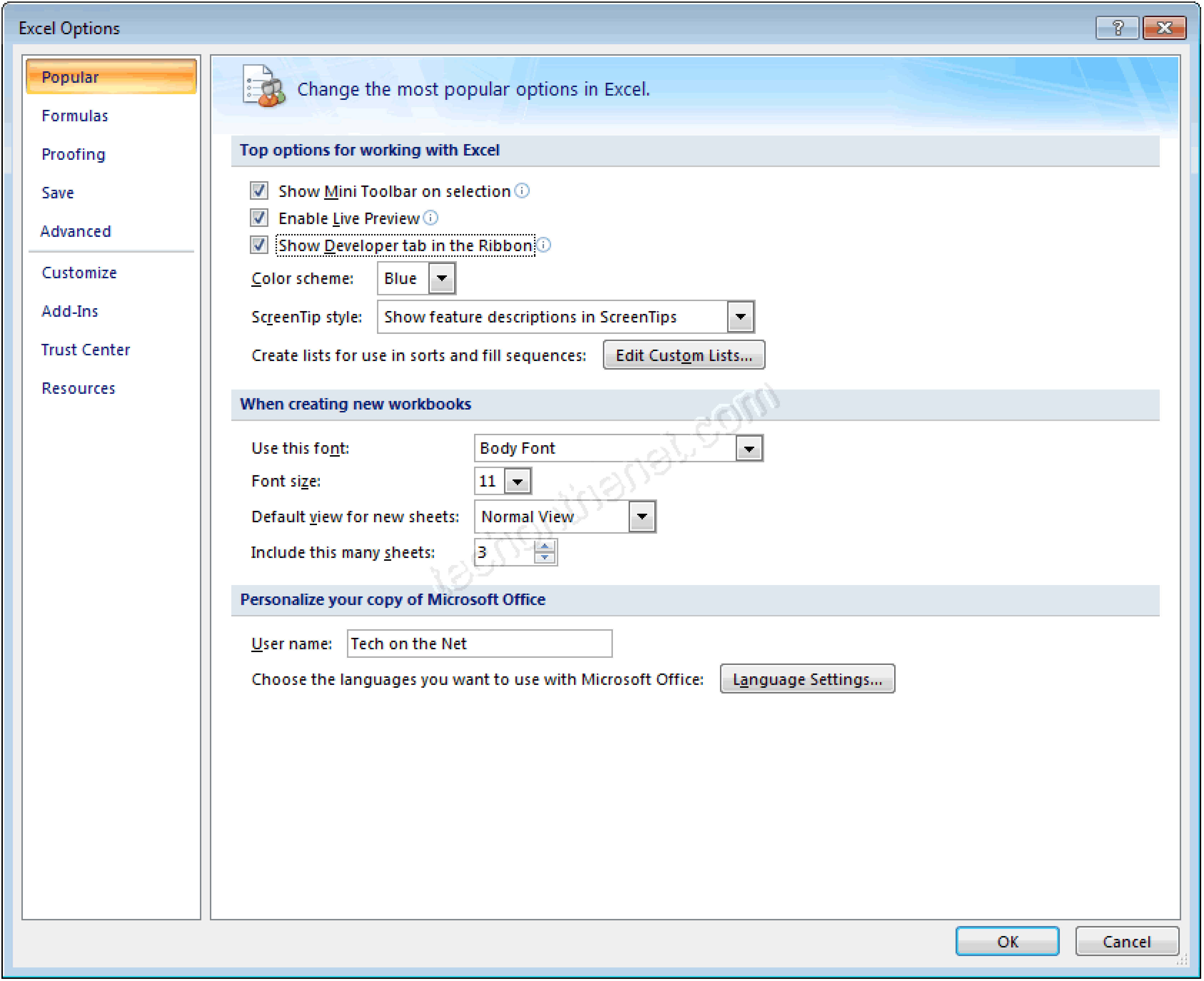
# Open the Visual Basic Editor in Excel 2007

- Click on the Microsoft Office button  in the top left of the Excel window and then click on the Excel Options button



# Open the Visual Basic Editor in Excel 2007

- When the Excel Options window appears, click on the Popular option on the left. Select the option called "**Show Developer tab in the Ribbon**". Then click on the OK button.
-



Change the most popular options in Excel.

**Top options for working with Excel**

☑ Show Mini Toolbar on selection

☑ Enable Live Preview

☑ Show Developer tab in the Ribbon

Color scheme: Blue

ScreenTip style: Show feature descriptions in ScreenTips

Create lists for use in sorts and fill sequences:

**When creating new workbooks**

Use this font: Body Font

Font size: 11

Default view for new sheets: Normal View

Include this many sheets: 3

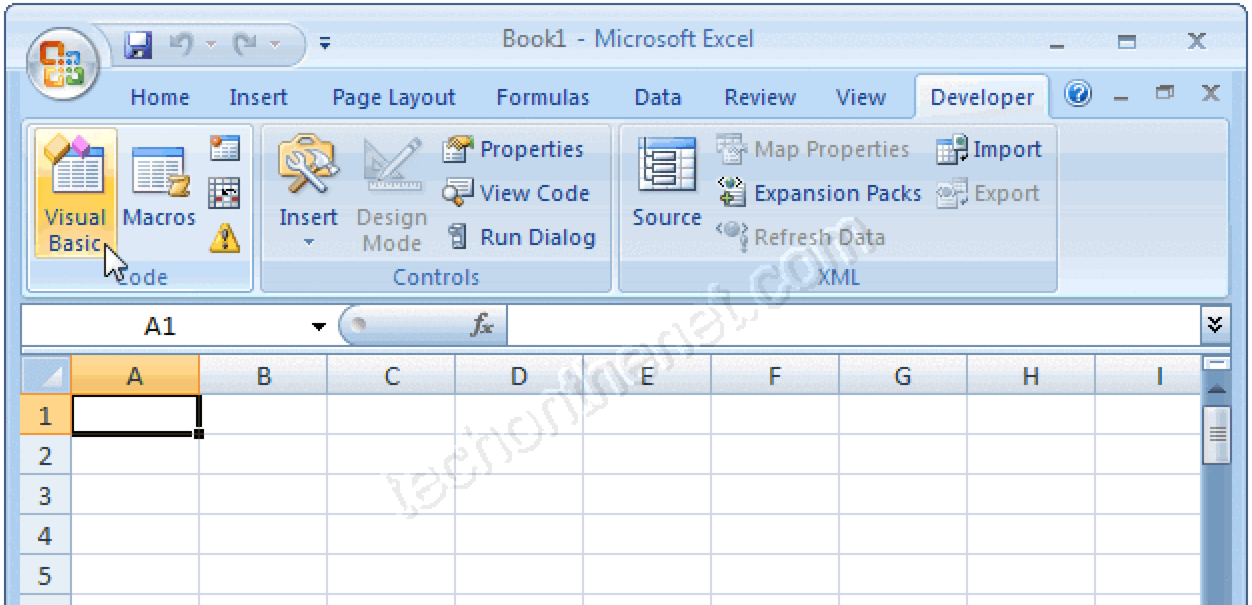
**Personalize your copy of Microsoft Office**

User name: Tech on the Net

Choose the languages you want to use with Microsoft Office:

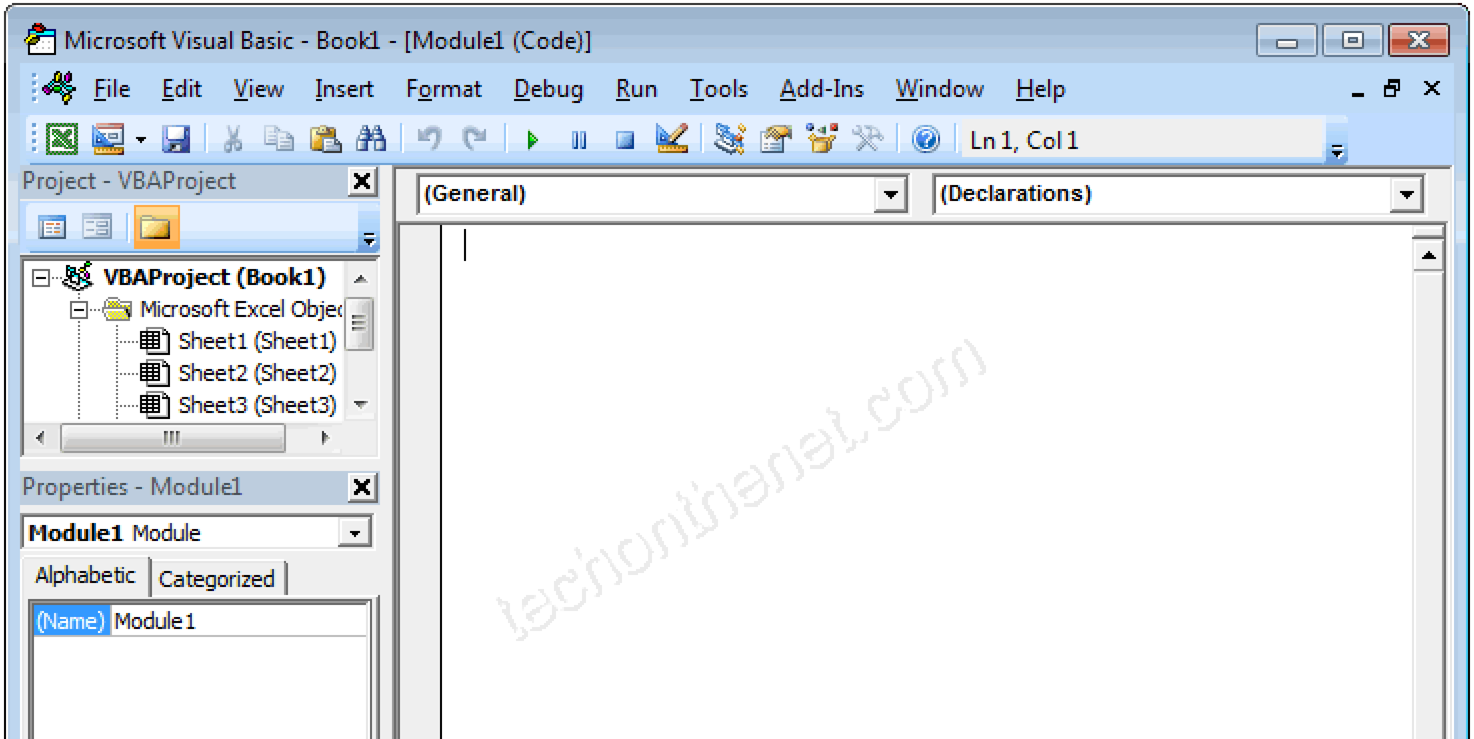
# Open the Visual Basic Editor in Excel 2007

- Select the **Developer tab** from the toolbar at the top of the screen. Then click on the **Visual Basic** option in the **Code group.**



# Open the Visual Basic Editor in Excel 2007

- Now the Microsoft Visual Basic editor should appear and you can view your VBA code.



# Variables and Data Types

- To use some values in code, you must first create them
- In the world of computer programming, a variable is a value you ask the computer to temporarily store in its memory while the program is running.

# Declaring a Variable

- In order to reserve that storage area, you have to let the computer know. Letting the computer know is referred to as *declaring* the variable. To declare a variable, you start with the **Dim** word.

```
Dim VariableName As DataType
```

# Declaring a Variable

- There are rules you should follow when naming your variables:
    - The name of a variable must begin with a letter or an underscore
    - After starting with a letter or an underscore, the name can be made of letters, underscores, and digits in any order
    - The name of a variable cannot have a period
    - The name of a variable can have up to 255 characters.
    - The name of a variable must be unique in the area where it is used

# Data Types

- **Byte:**
  - To declare a variable that would hold natural numbers that range from 0 to 255, use the **Byte** data type. Here is an example:

```
Sub Exercise()
Dim Value As Byte
Value = 246
End Sub
```

# Data Types

- **Integer:**
  - ▫ To declare a variable that would hold a number that ranges from -32768 to 32767, use the **Integer**data type
- **Long:**
  - ▫ A long integer is a number that can be used for a variable involving greater numbers than integers. To declare a variable that would hold such a large number, use the **Long** data type
- **Double:**
  - ▫ If you want to use a decimal number that requires a good deal of precision, declare a variable using the **Double** data type.

# Data Types

- **String:**
  - A string is a character or a combination of characters that constitute text of any kind and almost any length. To declare a string variable, use the **String** data type. Here is an example:

```
Sub Exercise()
        Dim CountryName As String
        CountryName = "Brésil"
End Sub
```

# VBA in Excel/ Worksheets

- In Microsoft Excel, a spreadsheet is called a worksheet
- A workbook is a series of worksheets that are treated as a group.
- A worksheet is an object of type **Worksheet**.
- Another name for the collection that contains the worksheets is called **Sheets**.
- In most cases, you can use either of these two collections. Each worksheet is an object of type **Worksheet**.

# Referencing a Worksheet

- worksheets of a document are part of the workbook that is opened
- the **Workbook** class is equipped with a property named **Worksheets** or **Sheets**.
- Therefore, after identifying the workbook, use the period operator to access the Worksheets or the Sheets property.
- Each worksheet can be located based on an indexed property named **Item**

```
Sub Exercise()
        Workbooks.Item(1).Sheets.Item(2)
End Sub
```

# Referencing a Worksheet

- Other ways to reference a worksheet:
  - Omitting Item():

```
Sub Exercise()
        Workbooks.Item(1).Worksheets(2)
End Sub
```

  - Using the sheet name:

```
Sub Exercise()

Workbooks.Item(1).Sheets.Item("Sheet3")
End Sub
```

# The Cells of a Worksheet
# Cell Referencing

- You can identify a cell using the **Range** object. To do this, in the parentheses of the **Range**object, pass a string that contains the name of the cell. Here is an example that refers to the cell located as D6:
- Sub Exercise() Workbooks.Item(1).Worksheets.Item("Sheet1").Range("D6")
- End Sub

# Cell Referencing

- To get a reference to a cell, declare a variable of type Range. To initialize the variable, identify the cell and assign it to the variable using the Set operator. Here is an example:
- Sub Exercise()
- Dim Cell As Range
- Set Cell = Workbooks.Item(1).Worksheets.Item("Sheet1").Range("D6")
- End Sub
- Multi-Cell Ranges:
- Sub Exercise()
- Range("D2:B5, F8:I14")
- End Sub

# Cell Contents

- To set a Cell’s contents to a certain value, just use the ‘=‘ operator:

```
Sub Example()
Sheets.Item(1).Range("A1") = “Hello"
End Sub
```

- To insert a formula, just ad an ‘=‘ before the formula’s name:

```
Sub Example()
Sheets.Item(1).Range(“B:1") = "=sum(A1:A10)"
End Sub
```

# Cell Background Color

- To change a cell’s backgroud color, use the interior.color property:

```
Sub Example()
Sheets.Item(1).Range("A1").Interior.Color = RGB(255,0,0)
End Sub
```