

Class: MSc

Subject : Application of IT- Basics and Advance Excel

Chapter: Unit 4 Chapter 1

Chapter Name: Functions

# Functions

- *A Function procedure is a VBA code that performs calculations and returns a value (or an array of values).*
- *Using a Function procedure, you can create a function that you can use in the worksheet (just like any regular Excel function such as SUM or VLOOKUP).*
- *When you have created a Function procedure using VBA, you can use it in three ways:*
  1. *As a formula in the worksheet, where it can take arguments as inputs and returns a value or an array of values.*
  2. *As a part of your VBA subroutine code or another Function code.*
  3. *In Conditional Formatting.*

# Function Vs. Subroutine

- A 'Subroutine' allows you to execute a set of code while a 'Function' returns a value (or an array of values).
- To give you an example, if you have a list of numbers (both positive and negative), and you want to identify the negative numbers, here is what you can do with a function and a subroutine.
- A subroutine can loop through each cell in the range and can highlight all the cells that have a negative value in it. In this case, the subroutine ends up changing the properties of the range object (by changing the color of the cells).
- With a custom function, you can use it in a separate column and it can return a TRUE if the value in a cell is negative and FALSE if it's positive. With a function, you can not change the object's properties.

# ***Creating a Function***

- The below code creates a function that will extract the numeric parts from an alphanumeric string.

```
Function GetNumeric(CellRef As String) as Long
Dim StringLength As Integer
StringLength = Len(CellRef)
For i = 1 To StringLength
If IsNumeric(Mid(CellRef, i, 1)) Then Result = Result & Mid(CellRef, i, 1)
Next i
GetNumeric = Result
End Function
```

- When you have the above code in the module, you can use this function in the workbook.

# ***Explanation***

- The first line of the code starts with the word – Function.
- This word tells VBA that our code is a function (and not a subroutine). The word Function is followed by the name of the function – GetNumeric. This is the name that we will be using in the worksheet to use this function.
- The name of the function cannot have spaces in it. Also, you can't name a function if it clashes with the name of a cell reference. For example, you can not name the function ABC123 as it also refers to a cell in Excel worksheet.
- You shouldn't give your function the same name as that of an existing function. If you do this, Excel would give preference to the in-built function.
- You can use an underscore if you want to separate words. For example, Get\_Numeric is an acceptable name.
- The function name is followed by some arguments in parenthesis. These are the arguments that our function would need from the user. These are just like the arguments that we need to supply to Excel's inbuilt functions. For example in a COUNTIF function, there are two arguments (range and criteria)
- Note that the function is specified as the 'String' data type. This would tell VBA that the result of the formula would be of the String data type.

# ***Explanation***

- The second line of the code declares a variable 'StringLength' as an Integer data type. This is the variable where we store the value of the length of the string that is being analyzed by the formula.
- The third line declares the variable Result as a String data type. This is the variable where we will extract the numbers from the alphanumeric string.
- The fourth line assigns the length of the string in the input argument to the 'StringLength' variable. Note that 'CellRef' refers to the argument that will be given by the user while using the formula in the worksheet.
- Sixth, seventh, and eighth lines are the part of the For Next loop. The loop runs for as many times as many characters are there in the input argument. This number is given by the LEN function and is assigned to the 'StringLength' variable.
- So the loop runs from '1 to Stringlength'.
- Within the loop, the IF statement analyzes each character of the string and if it's numeric, it adds that numeric character to the Result variable. It uses the MID function in VBA to do this.

# Using User Defined Functions

## ➤ **Using UDFs in Worksheets**

*All you need to do is enter the functions name, and it shows up in the intellisense.*

## ➤ **Using User Defined Functions in VBA Procedures and Functions**

*When you have created a function, you can use it in other sub-procedures as well.*

*If the function is Public, it can be used in any procedure in the same or different module. If it's Private, it can only be used in the same module.*

*Below is a function that returns the name of the workbook.*

```
Function WorkbookName() As String  
    WorkbookName = ThisWorkbook.Name  
End Function
```

```
Sub ShowWorkbookName()  
    MsgBox WorkbookName  
End Sub
```

*The below procedure call the function and then display the name in a message box.*