



LET's GRADUATE IN MS-EXCEL



**Workshop
Organised by
Lucknow Branch of ICAI**

**On
Sunday 16th Nov'2007**

**Compiled and presented by
CA. Mukesh Saran
Lucknow**



LET's GRADUATE IN MS-EXCEL

Add-In in Microsoft Excel

(Source: <http://exceltips.com/>)

An Excel Add-In is a file (usually with an **.xla** or **.xll** extension) that Excel can load when it starts up. The file contains code (VBA in the case of an **.xla** Add-In) that adds additional functionality to Excel, usually in the form of new functions.

Add-Ins provide an excellent way of increasing the power of Excel and they are the ideal vehicle for distributing your custom functions. Excel is shipped with a variety of Add-Ins ready for you to load and start using, and many third-party Add-Ins are available.

This article shows you how to write a custom function using Excel VBA and how to save and install it as an Add-In. Custom functions are often referred to as UDFs (User Defined Functions). If you have never built a UDF before, this is a good place to start.

If you are already comfortable with writing custom functions, you can go straight to the section explaining how to save your UDFs as an Add-In. *[Jump to Add-In section]*

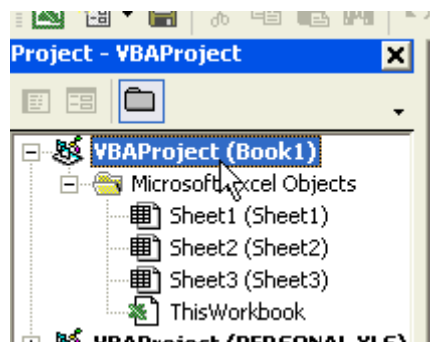
Write the Function

An Add-In can contain as many UDFs as you want, and you can add more later simply by opening and editing the Add-In file.

Step 1: Add a Code Module to a New Workbook

Start Excel or, if you already have Excel open, create a new empty workbook.

Open the Visual Basic Editor from **Tools > Macro > Visual Basic Editor** (Keys: ALT+F11). In the *Project Explorer* pane select **VBAProject (Book1)**. This selects the empty workbook.



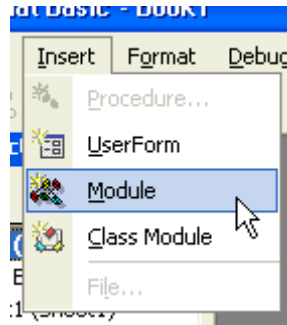
If you were already working in Excel the name might be different - check the name in the title bar of the Excel window if you aren't sure. If the



LET's GRADUATE IN MS-EXCEL

Project Explorer is not visible open it by going to **View > Project Explorer**.

From the Insert menu choose Module. This adds a new empty code module to the selected workbook. You will also see the module appear in the Project Explorer pane.



Step 2: Enter the Code

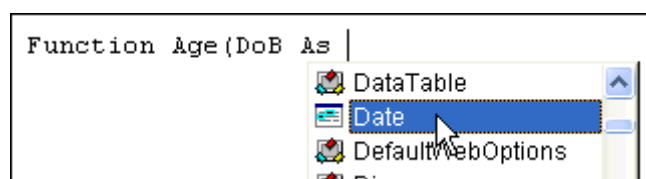
In the code window type the line...

Function Age(DoB As Date)

...and press ENTER. The VB editor automatically enters the line "End Function" and places your cursor in the empty space between them. Typing "Function" defines the code that follows as a function, as opposed to a macro or subroutine, which is denoted by the word "Sub".

Next is the function's name, which can be anything you like. Keep your function names simple and descriptive. Spaces aren't allowed so separate words with underscores (e.g. Age_in_years) or, preferably, with capitals (e.g. AgeInYears).

A function usually needs one or more "arguments". These are entered in brackets after the function name. An argument is a piece of information that the function uses to perform its calculation. When declaring the arguments it is good practice to specify the data type. In this case only one argument is needed, the date of birth that will be used to calculate the person's age. For simplicity, I have called this argument "DoB". Arguments work like variables in macros. When you type the word "As" after the argument's name the VB editor presents you with a list of possible entries. You can type or pick from the list.





LET's GRADUATE IN MS-EXCEL

Between the Function and End Function statements, press TAB (to indent your code for clarity) and type the line...

```
Age = Int((Date - DoB) / 365.25)
```

This tells Excel how to work out the function. It uses two built-in functions, INT (which makes a whole number - or integer - out of the result) and DATE (a visual basic function equivalent to Excel's TODAY() function that returns the current date).

The statement effectively reads... "Take today's date and subtract from it the supplied date of birth. Divide the result by 365.25 and show the answer as a whole number by rounding down."

The finished function should look like this:

```
Function Age(DoB As Date)  
    Age = Int((Date - DoB) / 365.25)  
End Function
```

Step 3: Test the Function

You can try out the function right away. Switch to Excel and in your empty workbook (the same one that you are using to create the function's code module) enter a date into a cell. In another cell enter your function in the same way as you would use one of Excel's built-in functions, e.g. **=Age(A1)**

The screenshot shows an Excel worksheet with columns A, B, C, and D. Cell A1 contains the date 27/09/1950. Cell B1 contains the formula =age(A1) and displays the result 51. The formula bar at the top shows the formula =age(A1).

	A	B	C	D
1	27/09/1950	51		
2				
3				

A UDF is available to all open workbooks whenever its host workbook (the workbook containing the UDF's code module) is open. However, if you try to use the function in a different workbook you can encounter a problem...

The screenshot shows an Excel worksheet with columns A, B, C, and D. Cell A1 contains the date 27/09/1950. Cell B1 contains the formula =age(A1) and displays the error #NAME?. The formula bar at the top shows the formula =age(A1).

	A	B	C	D
1	27/09/1950	#NAME?		
2				
3				

The other workbook can't find the function so the #NAME? error appears. To avoid this happening you could add the host workbook's name to the function to tell Excel where to find it...



LET's GRADUATE IN MS-EXCEL

	A	B	C	D
1	27/09/1950	51		
2				

This works but is very clumsy and demonstrates one of the reasons why the best place for your custom functions is inside an Excel Add-In (with some exceptions...]

Adding Features to the Function

Allowing for the Absence of Data

A disadvantage of many functions is that when you prepare a worksheet in advance of its receiving data (e.g. in a template where you want the functions to be in place ready to calculate the user's input) the functions try to calculate on empty cells, often producing errors or nonsense results. If our Age function tries to calculate an empty cell it still produces a result, in this case 102 (the current year being 2002). This happens because it takes the value of the empty cell to be zero, and interprets day zero as January 0 1900. This is logical to Excel because the first day it knows about is day 1, which is January 1 1900.

Normally you would try to anticipate this by entering your age function as part of an IF statement e.g. **=IF(ISBLANK(A1),"",age(A1))** The IF statement tells Excel to enter a value of "nothing" (represented by "") if the data cell is blank but to calculate the Age function if it contains a value...

	A	B	C	D	E
1					
2					

	A	B	C	D	E
1	27/09/1950	51			
2					

This works fine, but you can save yourself the trouble by incorporating this sort of troubleshooting into the code of the function itself. Modify the function code as follows:



LET's GRADUATE IN MS-EXCEL

```
Function Age(DoB As Date)
    If DoB = 0 Then
        Age = ""
    Else
        Age = Int((Date - DoB) / 365.25)
    End If
End Function
```

Alternatively a custom error message could be included by replacing the pair of quote marks in code line 3 with a message in quotes e.g. **Age = "No Birthdate"**. The result would be...

The screenshot shows an Excel spreadsheet with columns A, B, C, and D, and rows 1, 2, and 3. The formula bar at the top shows the formula `=age(A1)` in cell B1. The cell B1 contains the text "No Birthdate".

	A	B	C	D
1		No Birthdate		
2				
3				

Making the Function More Accurate

The calculation used in the example above is very accurate, but not completely accurate. It works on the principle that there is an average of 365.25 days in a year (usually 365 but 366 every fourth year) so dividing a person's age in days by 365.25 should give their age in years.

This works fine most of the time but it can (rarely) throw up an error. If the person in question has their birthday today and were born on a year that is a multiple of 4 years ago, the calculation will be a year out. A small possibility, but if we're going to do it we might as well do it right!

To calculate someone's age from their date of birth, with complete accuracy use a CASE statement. Excel doesn't have the CASE statement but VBA does. CASE statements are easier than IF statements to figure out when the logic is slightly complex.

Here is the code for my improved function:

```
Function Age(DoB As Date)
    If DoB = 0 Then
        Age = "No Birthdate"
    Else
        Select Case Month(Date)
            Case Is < Month(DoB)
                Age = Year(Date) - Year(DoB) - 1
            Case Is = Month(DoB)
                If Day(Date) >= Day(DoB) Then
                    Age = Year(Date) - Year(DoB)
                Else

```



LET's GRADUATE IN MS-EXCEL

```

        Age = Year(Date) - Year(DoB) - 1
    End If
    Case Is > Month(DoB)
        Age = Year(Date) - Year(DoB)
    End Select
End If
End Function

```

TIP: Select the lines of code in the grey box above, copy them (Keys: CTRL+C) and paste them directly into your VBA code window (Keys: CTRL+V).

How the code works...

Function Age(DoB As Date)	Gives a name to the function and declares that a single argument is needed, which must be a date.
If DoB = 0 Then Age = "No Birthdate"	An IF statement to determine whether there is a value in the data cell. The value of an empty cell is read as zero. If that is true then the function returns the text "No Birthdate".
Else Select Case Month(Date)	If the data cell is not empty, consider the month that it is today...
Case Is < Month(DoB) Age = Year(Date) - Year(DoB) - 1	If today's month is before (i.e. less than) the month of the person's date of birth, they have not had their birthday, so their age is this year minus their birth year minus 1.
Case Is = Month(DoB)	If today's month is the same as the month of the person's date of birth we need to know whether or not they have had their birthday yet so...
If Day(Date) >= Day(DoB) Then Age = Year(Date) - Year(DoB)	If the today is equal to or past the day of their birthday, then they have had their birthday (or it is today) so their age is this year minus their birth year...



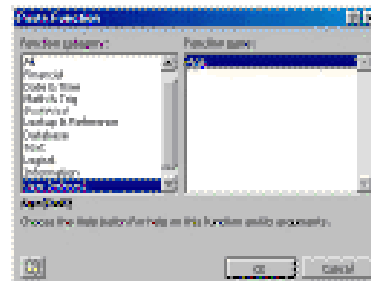
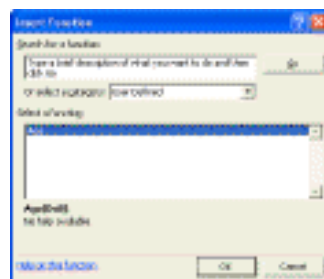
LET's GRADUATE IN MS-EXCEL

Else Age = Year(Date) - Year(DoB) - 1 End If	...otherwise, they have not had their birthday so their age is this year minus their birth year minus 1.
Case Is > Month(DoB) Age = Year(Date) - Year(DoB)	If today's month is after (i.e. greater than) the month of the person's date of birth, they have had their birthday, so their age is this year minus their birth year.
End Select End If End Function	Closes the CASE statement, the IF statement and the Function.

This calculation may seem rather complex but you only have to type it once! When you have created your function all you ever have to type is its name.

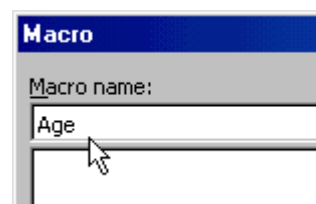
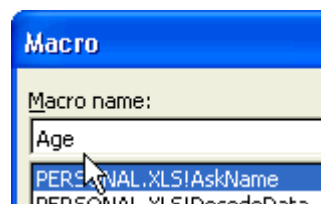
Step 1: Adding a Description to the Function

When a function is inserted via the function wizard (i.e. the Paste Function tool) the user sees a description of the function that helps them choose the correct one. This isn't the case with custom functions



If you want a description you have to add it yourself. Here's how...

Go to Tools > Macro > Macros to open the Macro dialog box. You will see all available macros listed but no custom functions. In the Macro name: text box type the name of the function you want to describe...

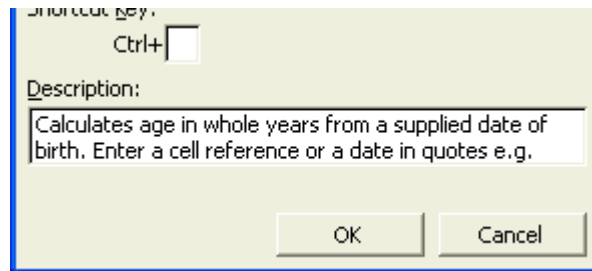




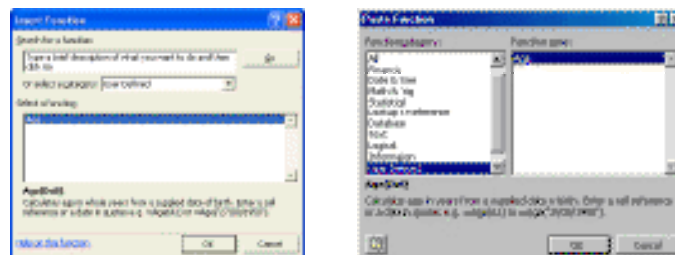
LET's GRADUATE IN MS-EXCEL

Click the [Options] button to open the Macro Options dialog box. (If the Options button is greyed out the function name you entered has not been recognised. Check your typing.)

Type a description for your function in the Description: text box...



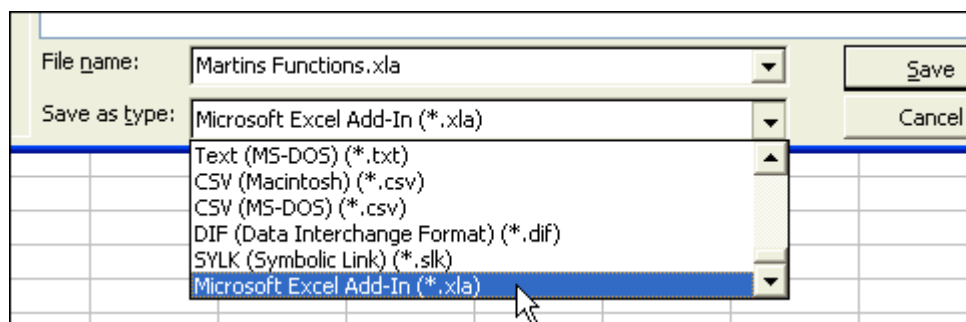
Click [OK] to close the Macro Options dialog and then **click [Cancel]** to close the Macro dialog. The description has now been assigned to the function. You can repeat the procedure to amend the description if required. This description is now shown when the custom function is selected in the Function Wizard



Step 2: Save the Workbook as an Add-In

The workbook containing your code module now has to be saved as an **Excel Add-In (*.xla)** file.

In the Excel window go to **File > Save** to open the **Save As** dialog. Enter a name for your Add-In file (the usual file naming rules apply) and use the Save as type: option to change the file type to **Microsoft Excel Add-In (*.xla)**.





LET's GRADUATE IN MS-EXCEL

Before **clicking [OK]** check the location in which you are saving the Add-In file. This will vary depending upon your particular set-up. On my computers the locations are:

Excel XP(2002) on Windows XP:

C:\Documents and Settings\UserName\Application Data\Microsoft\AddIns

Excel 2000 on Windows 98:

C:\Windows\Application Data\Microsoft\AddIns

Excel 97 on Windows 98:

C:\Program Files\Microsoft Office\Office\Library

You can store your Add-In anywhere you like but, if you want it to be listed along with the built-in ones, you should save it into the correct location. Excel XP and Excel 2000 will automatically take you to the correct folder but Excel 97 does not.

TIP: To check the save location, first set the Save as type: to Microsoft Excel Add-In then open the Save in: drop-down list to reveal the path to the save folder.

You can now close the original workbook (**Book 1**). You do not need to save the changes if prompted.

Step 3: Add a Description to the Add-In

It is a good idea to add a description to the Add-In itself. This description will be displayed in the Add-Ins dialog box when you choose an Add-In to install.

First, use the file manager to locate your Add-In file. Right-click on the file icon and choose Properties from the context menu. In the file properties dialog click the Summary tab. Type a description of your Add-In in the Comments: text box. If you wish you can also type a name for your Add-In in the Title: text box. This is useful if you have chosen a short or cryptic name for your *.xla file but would like to show a more descriptive name in the Add-Ins dialog. I could give my Add-In file the filename mgfunctions.xla but assign it the title Martin's Functions.

Finally **click [OK]** to accept your changes. Your Add-In is now ready for installation, and can be distributed to other users if required.

Step 4: Install the Add-In

If Excel has not been shut down since you created your Add-In (or since one was copied to the computer's hard disk) restart Excel to make sure that it refreshes its list of available Add-Ins.

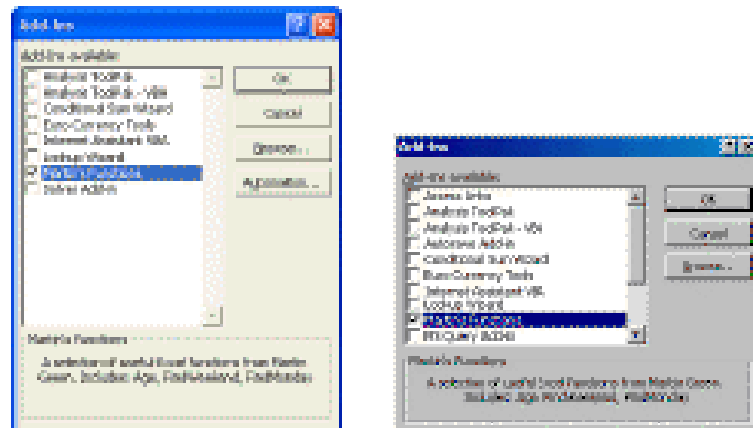
Go to **Tools > Add-Ins** to open the Add-Ins dialog. If you have stored your Add-In in the default location you will see its name displayed in the Add-Ins available: window (if you have stored your Add-In in a different



LET's GRADUATE IN MS-EXCEL

folder, use the **[Browse]** button to find it. Click on your Add-In's name to see its description at the bottom of the dialog box.

To install your Add-In place a tick in the check-box next to your Add-In's name and **click [OK]**.



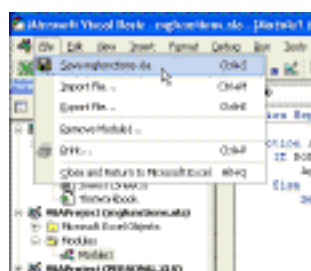
As soon as the Add-In is installed its functions will be available in Excel. Find them in the User Defined section of the Function Wizard (Paste Function Tool) or simply type them into a cell as you would any built-in function. The Add-In will remain installed until you return to the Add-Ins dialog and uninstall it by removing the tick from the check-box.

Making Additions and Changes to an Add-In

Your Add-In file can contain as many modules and custom functions as you want. You can add them at any time.

If your Add-In is installed you will see it listed in the Project Explorer pane of the VB editor. Locate the module containing your functions and make whatever additions and changes you want. If your Add-In is not installed, find the Add-In file and **double-click** it to open it in Excel. You will not be able to see it in the Excel window but it will appear in the VB editor's Project Explorer.

Remember to save your changes! Do this from the VB editor window with **File > Save**





LET's GRADUATE IN MS-EXCEL

Adding Comments to Your Code

It's easy to add notes (properly called "comments") to your VBA code. Enter an apostrophe (i.e. single quote mark) and then type. The apostrophe tells the code compiler to ignore everything that follows it on the same line so it isn't treated as executable code and the compiler doesn't try to run it. Anything you type after the apostrophe is "commented out" and is coloured green in the VB editor's code window...

```
Function Age (DoB As Date)
' =====
' Calculates age in whole years from a supplied
' Author: Martin Green (martin@fontstuff.com)
' =====
    If DoB = 0 Then
        Age = "" 'Returns nothing if no date supplied
    Else
        Age = Int ((Date - DoB) / 365.25)
```

You can use this technique to add explanatory notes to your code (ever returned to some code you wrote a while ago and wondered what on earth it means?).

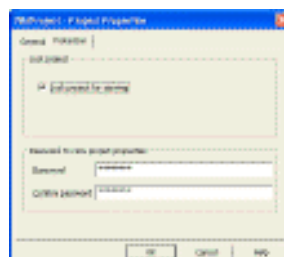
You can also temporarily comment out lines of code so that they don't run - useful for testing your code. If you think you don't need a line of code, don't delete it but comment it out first. Then test your code to see if it still works. If everything is OK you can now delete the line. If your code fails, simply remove the apostrophe and your line of code is restored.

Password Protecting Your Code

VBA allows you to password-protect your code, whether it is the macro code behind a regular workbook or the code modules of an Add-In. Password-protecting code prevents unauthorised viewing of your code for purposes of security or copyright.

To password-protect your Add-In locate it in the VB Editor. Right-click its name in the Project Explorer pane and choose VBAProject Properties... from the context menu. In the VBAProject - Project Properties dialog click the Protection tab.

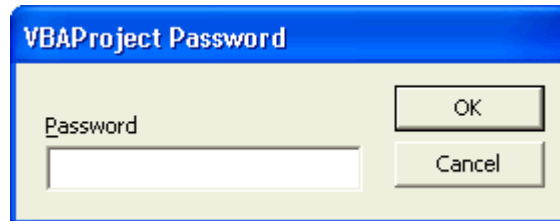
Place a tick in the Lock project for viewing check box, then enter and confirm your password. **Click [OK]** and go to **File > Save** to save your changes.





LET'S GRADUATE IN MS-EXCEL

The password-protection will take effect from the next time Excel is opened. Any attempt to view the Add-In's code module will result in a request for the password...



Assigning Add-In Macros to Custom Toolbar Buttons and Menu Items

Add-Ins aren't just for containing functions. They may contain useful macros. You may want to assign a macro that is contained in an Add-In to a custom toolbar button or custom menu item.

Normally to do this you would right-click on your new button or menu item and choose Assign Macro and pick your macro from the list provided. However, the names of macros contained in Add-Ins are not shown in the list. Don't worry! Just type the name of the macro in the Macro Name: box at the top of the dialog. Excel will find it and the macro will run as expected when the button is clicked or the menu item chosen.

A Final Word of Caution!

A custom function that is located in a code module within a workbook will go wherever the workbook goes. In other words if you open the workbook file on a different machine, or e-mail it to someone else, the function travels with the workbook and will always be available.

If your workbook refers to a custom function contained in an Add-In, the workbook will only be able to calculate the function when the Add-In is present. If you mail the workbook to someone else you will have to mail them the Add-In too!

If you want to use custom functions contained in an Add-In in a the code of another workbook, you will have to set a reference to the Add-In. Read the article [How to Use Your Excel Add-In Functions in VBA](#) for details.



Tips in Microsoft Excel

(source BCAS Referencer – 2007-08)

The question most frequently asked is always been “How can I learn Advanced Excel?” The short answer is “Use more of Excel”. You would then have many questions, which means so many opportunities to learn. To solve these questions and thereby self learn Excel, here is the route to follow:

1. The First place to look for help on Excel is in Excel itself. **Press F1**, type out your question to get the reply. The help files are quite exhaustive and there are good chances that the solution to your query would be found on your desktop itself.
2. Come Home to Excel: There is a wealth of information on Excel at Microsoft Office Excel Home Page:
<http://office.microsoft.com/en-us/FX010858001033.aspx>
3. If this does not work, try searching newsgroups at **http://groups.google.com/advanced_group_search** to find out if a similar question has already been asked and replied (new to newsgroups?, click here to know all about them **http://www.tushar-mehta.com/misc_tutorials/oe_ng/index.html**
4. Search if a similar question has been put up on Microsoft Knowledge database at:
<http://search.support.microsoft.com/search/?adv=1>
5. Search at web sites listed in Part IV – Links to Excel resources for tips and/or (free) add-ins to do what you want.
6. If you still cant find the solution, post a question to a newsgroup to know the netiquettes of posting
<http://www.cpearson.com/excel/newposte.htm>

Tips and Tricks from the perspective of its utility to Chartered Accountants. The tips and tricks are in the form of solution to questions whose answer you always wanted to know.

1. Fill Blank Cells in a list?

Many times in a list, especially so when it is imported from a database, cells down below that have same headings or subheadings are left blank for the sake of easier readability. For e.g. See the screenshot below



LET's GRADUATE IN MS-EXCEL

	A	B	C	D
1	Pro_Id	Date	Challan_No	Qty
2	A001	01/04/2004	191001	170
3			50341	-143
4			298243	178
5	A002	02/04/2004	449553	114
6			442305	132
7	A003	01/04/2004	479232	100
8			7249	185
9		02/04/2004	203712	-43
10			431623	139

However this creates a difficulty when such lists are required to be sorted and/or subtotaled. To fill the cells down below with the subheadings of the cells above them, do the following:

1. Select the list
2. **Click Edit >Go to..**
3. In the **[Go To]** dialogue box that appears, click the **'Special'** button
4. In the **[Special]** dialogue box, select **'blanks'**. This will highlight only the blank cells in the list, with focus on the 1st blank cell. See figure

	A	B	C	D
1	Pro_Id	Date	Challan_No	Qty
2	A001	01/04/2004	191001	170
3			50341	-143
4			298243	178
5	A002	02/04/2004	449553	114
6			442305	132
7	A003	01/04/2004	479232	100
8			7249	185
9		02/04/2004	203712	-43
10			431623	139

5. In the 1st blank cell, i.e. the cell with the focus, enter the formula **=A2** (see figure) and press **CTRL+ENTER**.

	A	B	C	D
1	Pro_Id	Date	Challan_No	Qty
2	A001	01/04/2004	191001	170
3	=A2		50341	-143
4			298243	178
5	A002	02/04/2004	449553	114
6			442305	132
7	A003	01/04/2004	479232	100
8			7249	185
9		02/04/2004	203712	-43
10			431623	139



LET's GRADUATE IN MS-EXCEL

6. As a result, the blank cells down below get filled up with the subheadings immediately above them. See figure

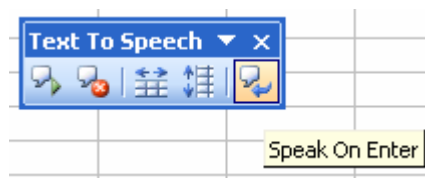
	A	B	C	D
1	Pro_Id	Date	Challan_No	Qty
2	A001	01/04/2004	191001	170
3	A001	01/04/2004	50341	-143
4	A001	01/04/2004	298243	178
5	A002	02/04/2004	449553	114
6	A002	02/04/2004	442305	132
7	A003	01/04/2004	479232	100
8	A003	01/04/2004	7249	185
9	A003	02/04/2004	203712	-43
10	A003	02/04/2004	431623	139

2. Make Excel read out the contents of a worksheet?

1. Choose **Tools > Speech > Show Text To Speech** toolbar. The toolbar is displayed as below.



2. To make Excel read out a range of cells, select the range first, and then click **<Speak Cells>** button (the first button on the toolbar). To read the data as it's entered, click **<Speak On Enter>** button (the last button on the toolbar).



3. Create Drop Down Lists?

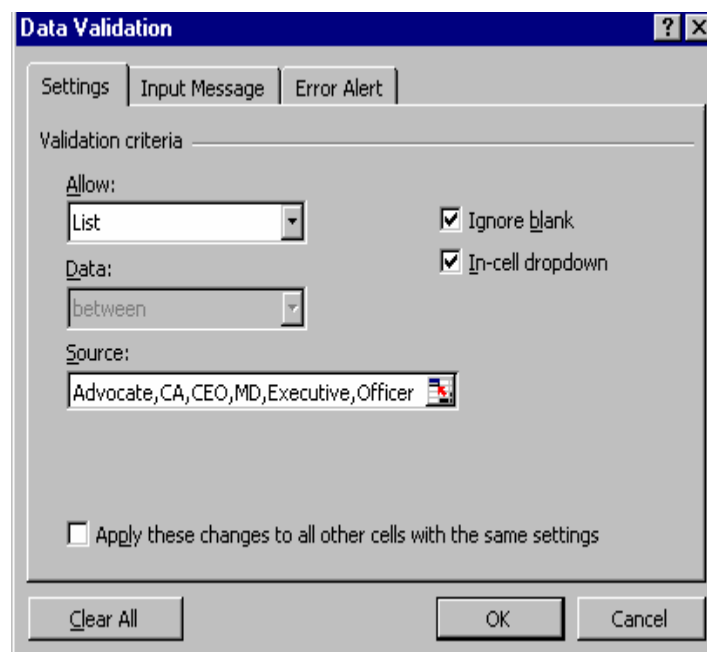
Haven't you always wanted to create those lovely drop down list boxes to enter data in lists or forms for uniformity and standardisation? (See Figure)

	A	B	C	D
1	Shah	Nikunj	S	
2				Advocate
3				CA
4				CEO
5				MD
				Executive Officer



LET's GRADUATE IN MS-EXCEL

These are called data validation list boxes. Here's how create them. In our example, let us consider that the user has to choose a designation from Chartered Accountant, Advocate, Managing Director, CEO, Executive and Officer. The designation is to be entered in column D. To create the List box, select the entire column D and click on **Data > Validation...** This will activate the **[Data Validation]** dialogue box. This dialogue box has three tabs viz., **[Settings]**, **[Input Message]** and **[Error Alert]**. The **[Settings tab]**, prompts you for the validation criteria. In the first list box captioned Allow, select the option à List. Now in the third list box captioned Source, enter the designation list, separated by comma (See Figure).



In the Input message tab, you may enter a message like {Please Select A Designation from the List}. This may help the person who enters data as regards what is to be entered in that particular cell. In the Error Alert tab, you may enter a message like {Invalid Designation, Only an item from the list is allowed!}. This message is flashed whenever the data entry person enters a designation that is not one from the List. If you do not specify the Error Alert Message, Excel displays its standard Error Alert Message. That's it! Click **<OK>**, and you are done. Now in every cell of column D, a list box appears as under. You may select the appropriate designation, by simply clicking the drop down list box arrow. For Key Board addicts, who find it irritating to reach for the mouse every time the designation is to be entered, just go the cell in column D, and hit the **[ALT + Down Arrow Key]**

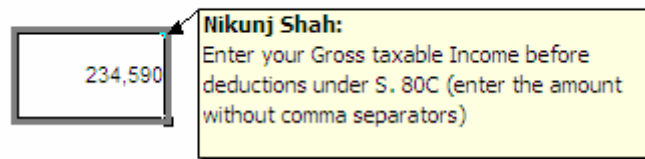


LET's GRADUATE IN MS-EXCEL

4. Suggest what is to be entered in a cell? (Cell comment / Data validation Input message)

You can suggest the user as regards what is to be entered in the cell by inserting a comment in the cell. Select the cell wherein you wish to see the message, right click and select **<Insert Comment>**. Type the instruction in the comment box. That's it. You are done. See figure

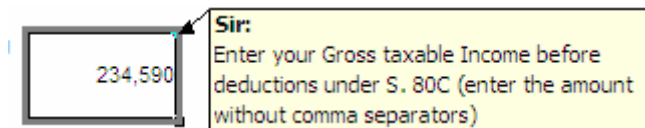
Personal tax calculator



Note that the cell comment is displayed only when you move the cursor over the cell.

5. Change the User Name in the cell comment?

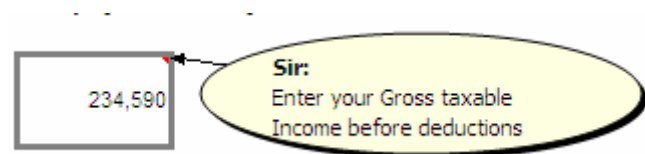
Instead of showing the user name at the start of a comment, you can change to something generic, such as "Sir". To do this Choose **Tools > Options**. Select the **[General]** tab. Delete the existing User Name, and type a new entry. Click **OK**. See figure.



However, note that this change affects the User Name in all Microsoft Office programs.

6. Change the shape of the Comment?

Make your worksheet a little more interesting, by changing the default comment shape. Right-click the cell which contains the comment. Choose **<Edit Comment>**. Click on the border of the comment, to select it. On the Drawing toolbar, click the Draw button. Choose Change AutoShape, and choose a category. Click on a shape to select it. When finished, click outside the comment. See figure

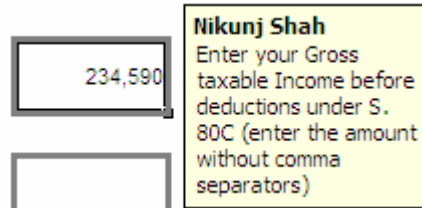




LET's GRADUATE IN MS-EXCEL

7. Is there an alternative to Cell comment?

An alternative to giving Cell comment is to give a Data Validation Input Message. Note that this message is displayed when you tab to the cell. Select the cell in which you want to see the message. Choose **Data>Data Validation**. Click on the **Input Message** tab to activate it. Add a check mark to 'Show input message when cell is selected'. Type your message heading text in the Title box. This text will appear in bold print at the top of the message. Type your message in the Input message box. Click **OK**. See figure.

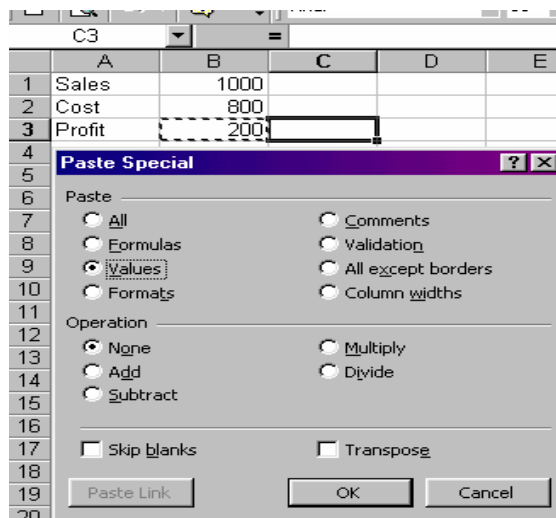


8. Copy Paste only Values from one cell to another?

Consider this example. Cell **B3** contains the Formula **=B1-B2** (see figure below).

		B3	=	=B1-B2
	A	B	C	
1	Sales	1000		
2	Cost	800		
3	Profit	200		
4				

This formula evaluates to 200. This '200' is the value of Cell B3. To copy and paste this Value i.e. 200, to the adjoining cell C3, go to cell B3, hit **[CTRL + C]**, or right your mouse and select **<Copy>** from the shortcut menu that pops up. Go to cell C3 and click **Edit > Paste Special**. A **[Paste Special]** dialogue box shall appear (See figure).





LET's GRADUATE IN MS-EXCEL

From the various options in the dialogue box, select the third option **[Values]** and click on **<OK>**. It's done! Cell C3 now contains the value 200. In the same way you can copy and paste only the Formats (especially the conditional Formatting applied to a cell), Comments, Validation, etc, by selecting the appropriate button from the [Paste Special] dialogue box.

9. Convert rows into columns?

Well, consider this. You have created a sheet wherein headings like LName, FName, MName, Address1, Address2, etc, are arranged in rows while records appear as columns besides these headings (see Figure below).

	A	B	C	D	E	F
1	LName	Shah	Jain	Vora	Shah	Anchalia
2	Fname	Nikunj	Ajay	Nimesh	Shital	Manoj
3	MName	S	B	I	V	M
4	Address1	abc	abc	abc	abc	abc
5	Address2	abc	abc	abc	abc	abc
6	City	abc	abc	abc	abc	abc
7	Pincode	123	123	123	123	123

Now, after entering around thirty to fourty records, you realise the obvious drawbacks of this layout and wish to rearrange the entire sheet by making the headings as columns and records below them as rows. This is what you can do. Select all cells, including the heading, and copy them. Open a fresh worksheet and select cell **A1**. Click **Edit>Paste Special...** In the **[Paste Special]** dialogue box simply check the **Transpose checkbox** and click **<OK>**. That's it. The sheet now looks as desired (see figure below).

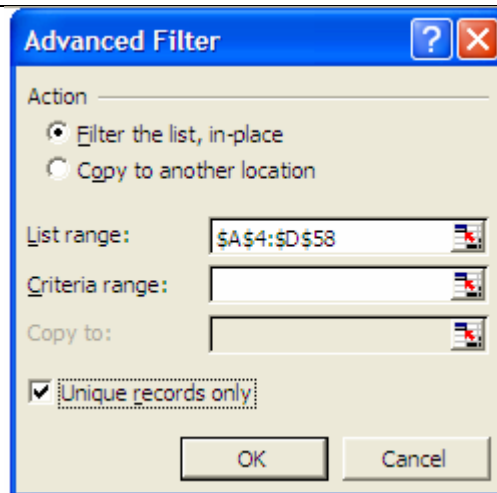
	A	B	C	D	E	F
1	LName	FName	MName	Address	Address2	City
2	Shah	Nikunj	S	abc	abc	abc
3	Jain	Ajay	B	abc	abc	abc
4	Vora	Nimesh	I	abc	abc	abc
5	Shah	Shital	V	abc	abc	abc
6	Anchalia	Manoj	M	abc	abc	abc

10. Remove duplicates / Filter Unique Records?

You can use an Advanced Filter to extract a list of unique items in the list. To do this, click on **Data > filter > Advance filter**. In the **[Advanced Filter]** dialogue box that opens, check out the **'Unique records only'** option (See figure below). That's it. You are done.



LET's GRADUATE IN MS-EXCEL



11. Filter for more than one criterion?

Use 'Criteria range:' in the [Advanced Filter] dialogue box to filter a list that meets a given criteria. You can use more than one criteria to filter a list as well.

To filter a list for records that meet ALL the criteria, write all conditions in one row. This shall filter the list for records that meet all the conditions. For example writing the criteria in this way

F	G	H
Customer	Product	Total
CityMall	Cakes	>500

shall filter the list for records where -

Customer = CityMall AND Product = Cakes AND Total = greater than 500.

To filter a list for records that meet ANY of the criteria, write conditions in different rows. This shall filter the list for records that meet any of the conditions. For example writing the criteria in this way

J	K	L
Customer	Product	Total
CityMall		
	Cakes	
		>500

shall filter the list for records where -

Customer = CityMall OR Product = Cakes OR Total = greater than 500.

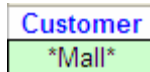


LET's GRADUATE IN MS-EXCEL

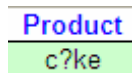
12. Filter for records containing text string / Use Wildcards in Criteria?

Use wildcard characters to filter for a text string in a cell.

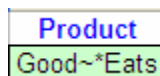
The * wildcard - The asterisk (*) wildcard character represents any number of characters in that position, including zero characters. In this example, any customer whose name contains "Mall" will pass through the filter.



The ? wildcard - The question mark (?) wildcard character represents one character in that position. In this example any 4-letter product that begins with 'c', and ends with 'ke' (eg. Coke, Cake) will pass through the filter.

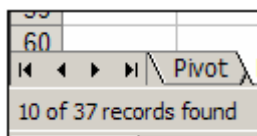


The ~ wildcard - The tilde (~) wildcard character lets you search for characters that are used as wildcards. In this example, only the product named Good*Eats, will pass through the filter.

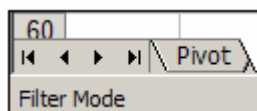


13. See the Count of Filtered Records in Status Bar?

Normally, after you have applied an AutoFilter, the Status Bar shows a count of visible records.



Sometimes it just says, "Filter Mode."



This can happen when your list has many formulas. There are articles in the Microsoft KnowledgeBase that explain:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q189479>

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;q213886>



LET's GRADUATE IN MS-EXCEL

The workaround is to go to [Options] dialogue box in the <Tools> and in the "Calculation" tab select 'Manual'. For a detailed explanation of the causes and solution to the problem, check out the links above.

14. Sum only visible rows in a filtered list.

To sum only visible rows which contain data, you can use the Subtotal function in a formula in the same row as your headings. For example,

=SUBTOTAL(9,D:D)		
C	D	E
Product	Total	3796
Bread	684	
Bread	782	
Bread	380	
Bread	120	
Bread	561	
Bread	635	
Bread	277	
Bread	357	

to sum the visible entries in column D which contain numbers, you could use this formula: **=SUBTOTAL(9, D:D)** . The number '9' in the first part of the formula, tells Excel to 'Sum' the numbers in column D.

15. Count visible rows in a filtered list?

Similarly, to count the visible rows in a filtered list, use the Subtotal formula with number '2' in the first part of the formula. For example,

=SUBTOTAL(2,D:D)		
C	D	E
Product	Total	8
Bread	684	
Bread	782	
Bread	380	
Bread	120	
Bread	561	
Bread	635	
Bread	277	
Bread	357	

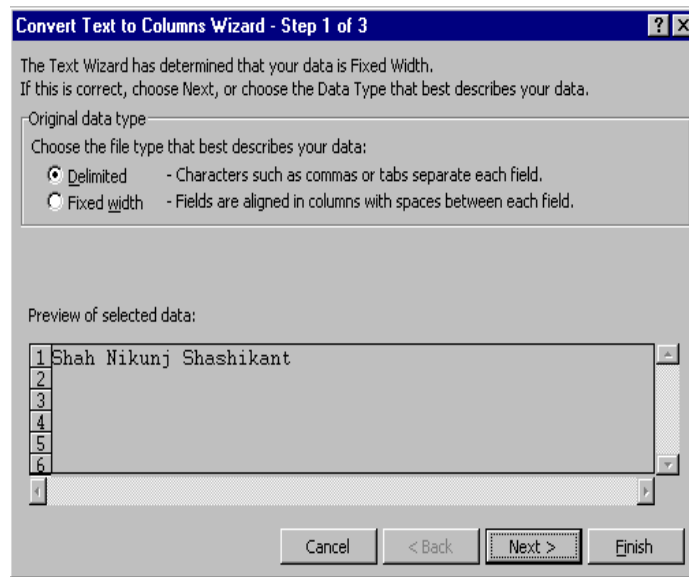
The formula used in the above screenshot counts the number of visible rows in column D.

16. Split Contents of Cells?

Suppose you have entered the full name of a person in a particular cell and you now want to have Surname, First Name and Middle Name in three different cells. To do this, simply select the cell that contains the Full Name of the person, Click **Data > Text to Columns...** This will activate the text to columns wizard (See Figure).



LET's GRADUATE IN MS-EXCEL



As a result a dialogue box captioned [**Convert Text to Columns Wizard - Step 1 of Three**] shall be displayed. In step No 1 select the First radio button captioned **<Delimited>**, then click the **<Next >** button. In step No. 2, from the various types of Delimiters, select the check Box [Space]. You now see how the Full Name shall be split into Surname, First Name and Middle Name. If you are fine with the display, click **<Next>**. In step No. 3, amongst other things, you are asked to provide the cell address from where you would like the split text to appear. That's it, on the main screen wizard of step Three click **<Finish>**. You now see the Full Name split into Surname, First Name and Middle Name in three different cells.

17. Monitor the value of a formula cell from any location?

If you have a spreadsheet model you may find it helpful to monitor the values in a few key cells as you change few input cells. To watch a cell, select the cell, right click and select add watch from the shortcut menu. Excel displays a floating window that shows the current value of that cell. See figure

	A	B	C	D
1	Parameters			
2	Present age	26		
3	Retirement age	65		
4	Life expectancy	80		
5	Present monthly expenses	25,000		
6	Inflation rate	7%		
7	Monthly savings earmarked towards retirement	20,000		
8	Annual increase in monthly savings	0%		
9	Income on retirement corpus (pre retirement)	8%		
10	Income on retirement corpus (post retirement)	6%		
11	How much corpus have you already built for your retirement?	100,000		

Book	Sheet	Name	Cell	Value
Retirement calculator ...	Frontend	Corpus	D72	72,022,191
Retirement calculator ...	Frontend	AdditionalSavings	D76	2,469



LET's GRADUATE IN MS-EXCEL

18. Join the Text of different rows in one row?

If you have a text in consecutive rows in a spreadsheet and wants to join them and present them in top select the text in all the rows you want to bring into a single row and some additional columns to accommodate the complete text then click **Edit > Fill > Justify...** who all the text of different rows are now in the top row. See figure below before and after justify:

Brfore				After				
1	Ram			1	Ram Asrey Verma			
2	Asrey			2				
3	Verma			3				
4				4				
5				5				
6				6				
7				7				
8				8				

This is not all, there are so many more tips which you can add to this library when you work more and more on Excel.





Sample Add-In in Microsoft Excel

1. Function for converting number into words

Function Towords(n As Double) As String

Dim units As Variant

'Dim teens As Variant

Dim tens As Variant

Dim r As Double

Dim r1 As Double

'Dim sand As String

'Dim s As String

units = Array(" ", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen")

tens = Array(" ", "Ten", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety")

n = n * 100

n1 = Val(Right(n, 2))

n = (n - n1) / 100

Select Case True

Case n > 99999999

r = n Mod 100000000

Select Case True

Case r = 0

Towords = Towords(n \ 100000000) & " Crores"

Case Else

Towords = Towords(n \ 100000000) & " Crores " & Towords(n Mod 100000000)

End Select

Case n > 99999

r = n Mod 100000

Select Case True

Case r = 0

Towords = Towords(n \ 100000) & " Lacs"

Case Else

Towords = Towords(n \ 100000) & " Lacs " & Towords(n Mod 100000)

End Select

Case n > 999

r = n Mod 1000

Select Case True

Case r = 0

Towords = Towords(n \ 1000) & " Thousand"

Case Else

Towords = Towords(n \ 1000) & " Thousand " & Towords(n Mod 1000)

End Select

Case n < 20

Towords = units(n)

Case n < 100



LET's GRADUATE IN MS-EXCEL

```
Towords = tens(n \ 10)
If (n Mod 10) <> 0 Then
    Towords = Towords & " " & units(n Mod 10)
End If
Case Else
    Towords = units(n \ 100) & " Hundred"
    If (n Mod 100) <> 0 Then
        Towords = Towords & " " & Towords(n Mod 100)
    End If
End Select
Select Case True
Case n1 <> 0
r1 = n1 Mod 100
    Select Case True
    Case r1 = 0
        Towords = Towords
    Case n1 < 20
        If Towords <> "" Then
            Towords = Towords & " and Paisa "
        Else
            Towords = Towords & " Paisa "
        End If
        Towords = Towords & units(n1)
    Case n1 < 100
        If Towords <> "" Then
            Towords = Towords & " and Paisa "
        Else
            Towords = Towords & " Paisa "
        End If
        Towords = Towords & tens(n1 \ 10)
        If (n1 Mod 10) <> 0 Then
            Towords = Towords & " " & units(n1 Mod 10)
        End If
    End Select
End Select
End Function
```

2. Function for calculating I/Tax for Individuals

```
Public Function Individual_tx08(Income As Double) As Double
    Select Case True
    Case Income > 250000
        'MsgBox "1" & i
        Individual_tx08 = Round((24000 + ((Income - 250000) * 0.3)), 0)
    Case Income > 150000
        'MsgBox "2" & i
        Individual_tx08 = Round((4000 + ((Income - 150000) * 0.2)), 0)
    Case Income > 110000
        'MsgBox "3" & i
        Individual_tx08 = Round(((Income - 110000) * 0.1), 0)
    Case Else
        'MsgBox "4" & i
```



LET's GRADUATE IN MS-EXCEL

```
Individual_tx08 = 0
End Select
End Function
```

3. Function for calculating I/Tax for Women

```
Public Function Women_twx08(Income As Double) As Double
    Select Case True
        Case Income > 250000
            'MsgBox "1" & Income
            Women_twx08 = Round((20500 + ((Income - 250000) * 0.3)), 0)
        Case Income > 150000
            'MsgBox "2" & Income
            Women_twx08 = Round((500 + ((Income - 150000) * 0.2)), 0)
        Case Income > 145000
            'MsgBox "3" & Income
            Women_twx08 = Round(((Income - 145000) * 0.1), 0)
        Case Else
            'MsgBox "4" & Income
            Women_twx08 = 0
    End Select
End Function
```

4. Function for calculating I/Tax for Senior Citizen

```
Public Function SeniorCitizen_txs08(Income As Double) As Double
    Select Case True
        Case Income > 250000
            'MsgBox "1" & Income
            SeniorCitizen_txs08 = Round((11000 + ((Income - 250000) * 0.3)), 0)
        Case Income > 195000
            'MsgBox "2" & Income
            SeniorCitizen_txs08 = Round(((Income - 195000) * 0.2), 0)
        Case Else
            'MsgBox "4" & Income
            SeniorCitizen_txs08 = 0
    End Select
End Function
```

5. Function for Extracting Hyperlink

```
Function Extract_Hyperlink(HyperlinkCell As Range)
    Extract_Hyperlink = Replace _
        (HyperlinkCell.Hyperlinks(1).Address, "mailto:", "")
End Function
```

6. Function for Checking existence of formula in a cell

```
Function Has_Formula(Check_Cell As Range)
    Has_Formula = Check_Cell.HasFormula
End Function
```



LET's GRADUATE IN MS-EXCEL

7. Function for Checking existence of Worksheet

```
Function Sheet_Exists(Sheet_Name As String) As Boolean
    Dim wsheet
    Application.Volatile
    On Error Resume Next
    Set wsheet = Sheets(Sheet_Name)
    Sheet_Exists = wsheet.Name = Sheet_Name
    On Error GoTo 0
End Function
```

8. Function for Checking existence of Workbook

```
Function Workbook_Exists(FilePath As String, Filename As String) As Boolean
    With Application.FileSearch
        .LookIn = FilePath
        '* represents wildcard characters
        .Filename = Filename
        Workbook_Exists = .Execute > 0
    End With
End Function
```

9. Function for Checking name of open Workbook

```
Function Workbook_Open(WbookName As String) As Boolean
    Dim wBookCheck As Workbook
    Application.Volatile
    On Error Resume Next
    Set wBookCheck = Workbooks(WbookName)
    Workbook_Open = Not wBookCheck Is Nothing
    On Error GoTo 0
End Function
```

