**University of Basra**

**College Computer Science and Information Technology**

**Computer Information Systems Dep.**



**Computer Applications**

**in Business**

**Computer Information System Dep.**

**Lecture**

**Assist Prof. Dr. Aliea S.Sabir**

**Last update :5/5/2021**

|  |
| --- |
| **Learning Outcomes** |

|  |
| --- |
| **Upon successful completion of this course students will be able to:**   * Plan, create, modify, and present spreadsheets * Organize, edit and enhance data in spreadsheets to achieve business standards * Use formulas and built-in functions appropriately and correctly to solve problems and critically assess the results * Design efficient spreadsheets capable of answering "what-if" questions to solve business problems * Plan, organize, create and present spreadsheet data in graphic form * Plan, create and present a custom Excel application using complex functions and macros * Use built-in financial, logical, and lookup functions appropriately and correctly to solve problems and critically assess the results |

|  |
| --- |
| **Topics** |
| * Principles of electronic spreadsheets using features available with currently popular spreadsheet software: * Worksheet creation and formatting; * entering of data, * formulas * functions * date and time functions * Math and trig functions * Statistical functions * Text functions * Logical functions * Database functions * Lookup and reference functions * editing and printing * basic charting * Advanced Tools * Conditional format * Tables * Arrays * Data validation * What-If Analysis * pivot tables * lookup tables |

|  |
| --- |
| **Textbook** |
| * Microsoft Excel 2010: Comprehensive By: Gary B. Shelly, Jeffrey J. Quasney, Steven M. Freund, Raymond E. Enger, and Mali B. Jones. 2012 Course Technology, Cengage Learning . * Excel 2007 for project managers : Comprehensive By: Kim Heldman ,William Heldman , Wiley Publishing, Inc.,2007 . * Excel® 2010 Formulas: Comprehensive By: John Walkenbach , Published by Wiley Publishing, Inc., Indianapolis, Indiana,2010 |

* **What is Excel?**

Software developed and manufactured by Microsoft Corporation that allows users to organize , format , and calculate data with formulas using a spreadsheet system broken up by rows and columns .

Microsoft Excel usually comes [bundled](http://www.businessdictionary.com/definition/bundle.html) with [Microsoft Office](http://www.businessdictionary.com/definition/Microsoft-Office.html) and is compatible with other [applications](http://www.businessdictionary.com/definition/application.html) offered in the suite of [products](http://www.businessdictionary.com/definition/product.html). The first software [program](http://www.businessdictionary.com/definition/program.html) similar to Excel was released in 1982 and was [called](http://www.businessdictionary.com/definition/call.html) Multiplan.

* **Principles of electronic spreadsheets .**
* As we sow in the following figure excel files named as book1, book2 ,… and so on .
* Every workbook contains from numbers of rows that intersection with numbers of columns which gives the cells that represent the sheet.
* Every workbook have three sheet at first and you can expended it to255 sheets by Press **shift +f11** or click insert worksheet button.
* Every sheet have 256 columns and 65536 rows ,
* Then every workbook file have(cells) = 256 columns 65536 rows \* 255 sheet .

**Title Bar**

**Menu Bar**

**Tool bar**

**Active cell**

**Name box**

**Office** **button**

**Quick Access**

**rows**

**Document area**

**Formula Bar**

**Insert worksheet**

**Tab worksheet**

zoom

Page layout style



**column**

**Ribbon bar**

* Press **Enter** button to move cross the column and Press **Tap** button to move cross the Row .
* Press **Alt +Enter** to write multiple line in same cell.
* As default menu bar contain seven submenu **(Home , Insert, Page layout, Formulas ,Data , Review and View )**.
* We can expended the menu bar sub menu by adding new submenu like **( Developer , load test and Team )** , which use in macro programming and many of advanced tool by click to

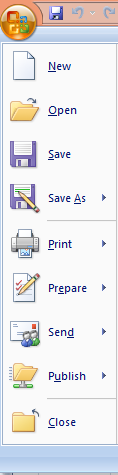
Office button 🡪 Excel Options 🡪 click 🗹 **Show Developer in The ribbon** under ***Top options for working with Excel***title .



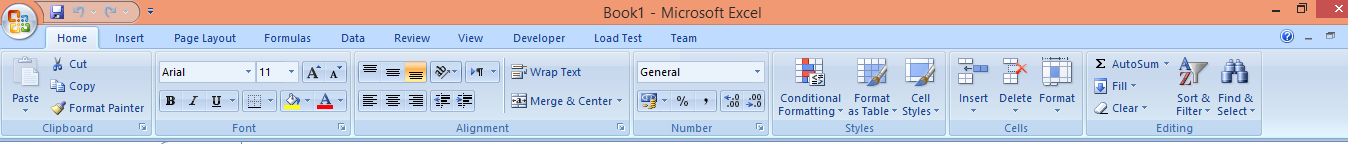
* In Excel options window we can change the most popular options in Excel as shown in the figure above.

**Note:** We will give brief information for every items in **menu bar** , the extended information we will talking about in the course Lab.

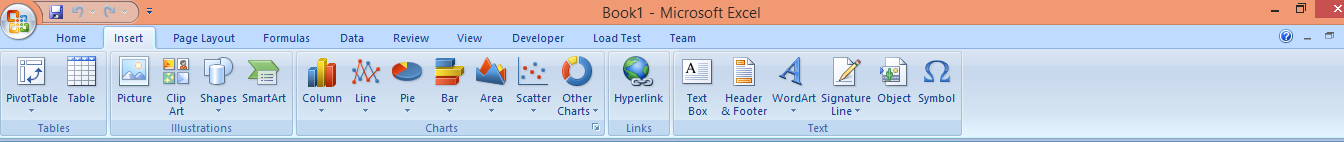
* **Office button :** this submenu deal with the operation with books file like , new , open , save , save as , and so on as shown in the next figure .

****

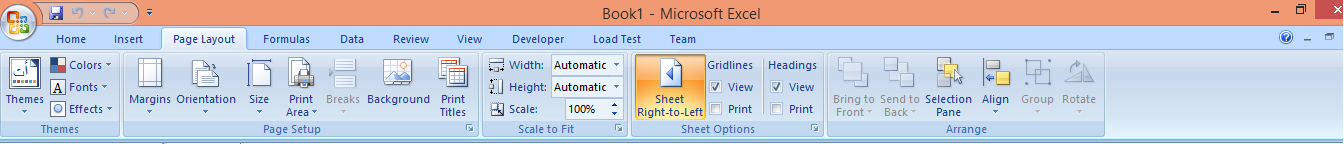
* **Home:** This submenu deals with basic sheet options like (**clipboard ,font ,alignment , number ,style ,cells and editing )** as shown in the next figure.

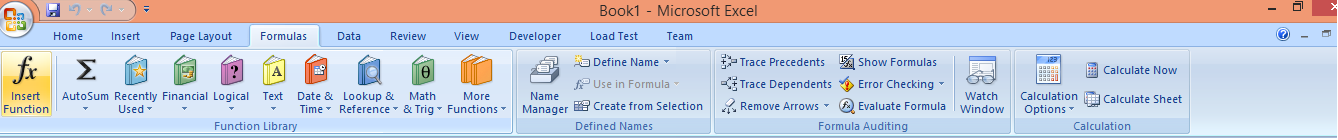
****

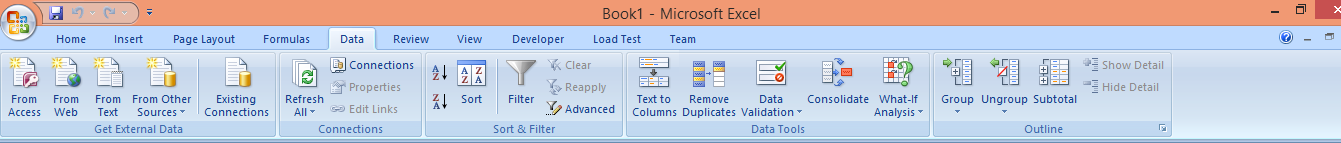
* **Insert :** This submenu deals with basic object that we can inserted inside the sheet like **(Pages , Tables , Illustrations ,Links , Header & Footer , Text and Symbols )** as shown in the next figure.

****

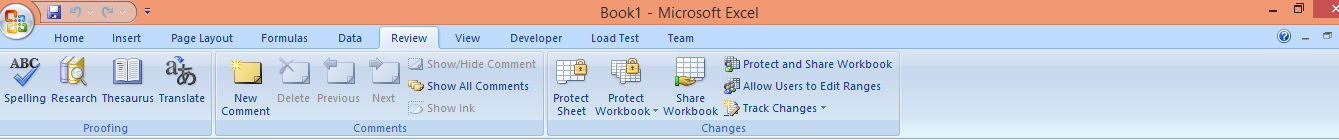
* **Page Layout:** This submenu deals with layout options of sheet like **(Themes , Page Setup, Scale to Fit , Sheet Options and Selection Pane and Align)** , as shown in figure above .

****

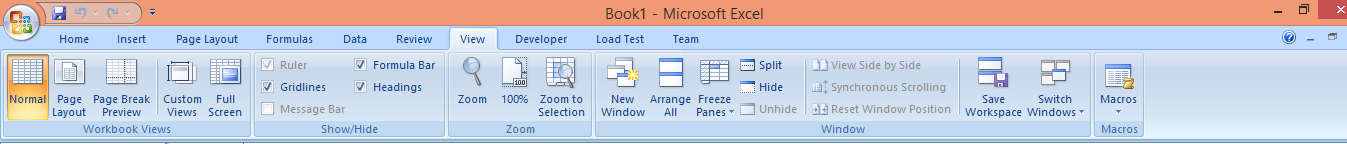
* **Formulas :** This submenu deals with formulas options and types that we can used in sheet like **(insert function , function Library , Defined Names , Formula Auditing and Calculation)** , as shown in figure bellow.
* **Data:** This submenu deals withsheetData options ,sources ,operation and tool like **(Get External Data , Connections , Sort & Filter , Data Tools and Outline ),** as shown in figure bellow.

****

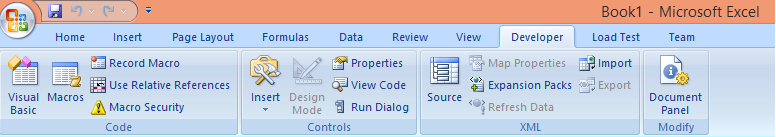
* **Review:** This submenu deals withthe basic sheet review like **(Proofing , Comments and Changes ),** as shown in figure bellow.



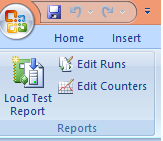
* **View:** This submenu deals withthe options of workbook view like **(Workbook Views , Show/Hide , Zoom , Window and Macros)** , as shown in figure bellow.

****

* **Developer:** This submenu its added to the menu bar when needed to work with advanced tools like **(Code , Controls , XML and Modify ) ,** as shown in figure bellow.

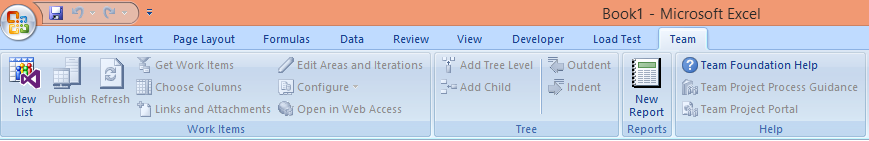
****

* **Load test:** This submenu deals with sheetload test options like **Report** , as shown in figure bellow.

****

* **Team:** This submenu deals with sheet Team option like **(Work item , Tree , new**

**Report and help ) ,** as shown in figure bellow.

****

* **Worksheet creation and formatting.**
* **General Guidelines**

While creating an Excel worksheet, you need to make several decisions that will determine the appearance and characteristics of the finished worksheet. As you create the worksheet you should follow these general guidelines:

**1. Select titles and subtitles for the worksheet.**

**2. Determine the contents for rows and columns.**

**3. Determine the calculations that are needed.**

**4. Determine where to save the workbook.**

**5. Identify how to format various elements of the worksheet.**

**6. Decide on the type of chart needed.**

**7. Establish where to position and how to format the chart.**

**8. Choose a name for the worksheet.**

**9. Determine the best method for distributing the workbook.**

* **Worksheet creation**

We can create new worksheet by click office button 🡪 new ,will open new worksheet .

* **Worksheet formatting**

we can change cell format by tools in **home** submenu or sheet format in **page Layout** submenu.

* **Entering of data.**

we can entering data in the sheet by select the cell or the range of cells that you wont to entering data in it , we can use **Enter** button to move over sheet column or press **TAP** button to move cross sheet Rows .

* **Absolute Cell Reference**

Absolute cell references can be very important when copying and pasting formulas. By default, when you copy and paste a formula it will use a relative cell reference. When the formula is pasted it will automatically change to a cell range, within the range of where the formula is being pasted. In some cases that is fine, but not when you want to reference the same cell range in the original formula.

Absolute cell references need to be used if you want to reference the same cell in a copied formula.

To reference an absolute cell put a dollar sign ($) in front of the reference that should be absolute. Cell references can be a mix of absolute and relative. Example of cell reference combinations, using cell C2:

**1.** **$C$2** – This is an example of an absolute cell reference that would always reference the value in cell C2.

**2.** **$C2** – This is an example using absolute and relative cell references. This cell reference would always reference column C, but the cell row would change.

**3.** **C$2** – This is an example using absolute and relative cell references. This cell reference would always reference row 2, but the column would change.

**4.** **C2** – This is an example of a relative cell reference that would change when the formula is copied. The cells would change in reference to where the formula is pasted.

**Note: The F4 key** can be used to toggle between the cell reference options, or the dollar sign can be typed in front of the cells that should be absolute cell references. To use the F4 option, click in the formula next to the cell reference and push the F4 key to toggle through the above cell reference options.

**Absolute Cell Reference Example**

***Example -1 :*Convert currency**

In a worksheet, currency has to be converted from dollars (column B) to euros (column C). The rate of exchange from dollars to euros is placed in cell C1; here we use 0.8021.

|  |  |
| --- | --- |
| **To convert currency:**  **1.** In a worksheet, enter your own data or the data shown in next Figure .  **2.** Select cells C4:C8.  **3.** Enter the following formula: **=B4\*$C$1**.  **4.** Press **<Ctrl+Enter>**.  **5.** From the Format menu, select **Cells**.  **6.** Select the Number tab and then select **Currency** from the Category list.  **7 .** Choose **Euro**.  **8.** Click **OK** |  |

**Note:** To convert euros back to dollars, use the following formula: **=C4/$C$1**.

***Example-2* : Calculate a price reduction**

All prices in a price list have to be reduced by a certain percentage. The amount of the price reduction is 15% and is entered in cell To reduce all prices by a certain percentage:

|  |  |
| --- | --- |
| **1.** In a worksheet, enter your own data or the data shown in the next Figure  **2.** Select cell C1 and type **-15%**.  **3.** Select cells C4:C8.  **4.** Enter the following formula: **=B4+(B4\*$C$1)**.  **5.** Press **<Ctrl+Enter>**. |  |

**Note:** Please note that the formula must have an absolute reference to cell C1. Also, columns B and C are formatted with the Currency style, which is available by clicking on the $ button on the Formatting toolbar.

* **Naming a Cell or Range of Cells**

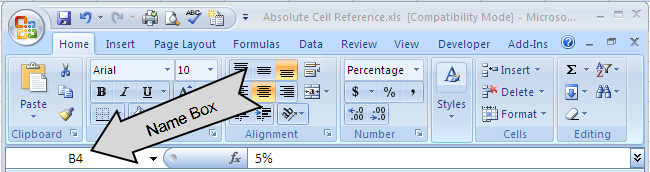
A named cell or range of cells can be used in formula references. Naming a cell or range of cells is a good time saver when writing formulas that are calculating between multiple sheets. Named cells or ranges can also be used in place of absolute cell references.

**1.** Select the cell or range of cells to be named

**2.** To the left of the formula bar is the name box, as shown in the picture to the right. By default, the name box will reference a cell that is selected.

**4.** Type in a name for the selected cell or range of cells

**5.** Press the enter key



**Note:** If a cell or range of cells has been named, the cell or range will need to be selected for the name to appear in the name box.

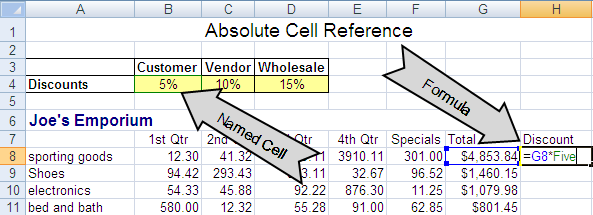
* **Using Names in Formulas**

Names can be used in formulas by referencing the name instead of the cell reference. For example: I want to use a formula that figures a discount of 5%. Cell B4 has the value of 5%. I have named the cell “Five”.

If I wanted to multiple a cell by the value in B4 (5%), I can simply reference the cell name no matter where I am in the workbook.

**Example of formula with cell name:** = G8\*Five, this would take the value in cell G8 and multiply itby the cell named “Five”, which is 5%.

**Note:** When using more advanced formulas it is much easier to reference a named cell or range of cells than having to select a whole table.

****

* **Finding a Named Cell or Range**

**By using the name box it is easy to find a named cell or range of cells.**

**1.** Click the drop-down arrow next to the name box

**Note:** The name box will reference the cell that is currently selected.

**2.** The drop-down menu will show a list of named cells or ranges

**3.** Select the name. The cursor will select that cell or range of cells



* **Editing a Named Cell or Range**

A named cell or range of cells can be changed by using the name manager.

**1.** Select the **formulas** tab

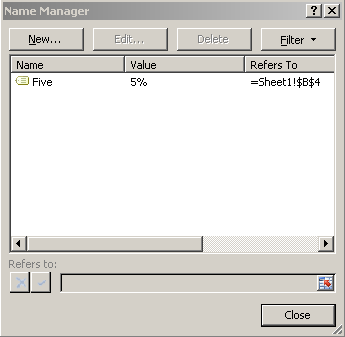
**2.** Under the **defined names** group, select the **name manager** button

**3.** Select the name to be changed

**4.** Click the edit button to change the name or cell range

**Note:** A new named range can also be created by using the New…button in the name manager window. The filter button can be used to search specific areas for defined names.

**5**. Click **close** when finished with the name manager dialog box



**Using Name in Formulas examples**

***Example-1 :*Calculate taxes**

In this exercise, tax on an item has to be calculated. We can also find the original price given the tax rate and the final price by using names.

|  |  |
| --- | --- |
| **To calculate the price with tax:**  **1.** Select cell A2 and type **8%** and type tax\_1 in name box then press **<Enter>** .  **2.** Select cell B2 and type **120** and type net\_1 in name box then press **<Enter>** .  **3.** Select cell C2 and type the following formula: **=net\_1+net\_1\*tax\_1**.  **To calculate the original price:**  **1.** Select cell A4 and type **8%** tax\_2 in name box then press **<Enter>** .  **2.** Select cell C4 and type **129.60** gross\_2 in name box then press **<Enter>** ..  **3.** Select cell B4 and type the following formula: **=gross\_2/(1+tax\_2)**. | **exp1_name.png** |

* **Formulas**
* **Formula Structure**
* To write complex formulas you will need to know basic formula structure. There are multiple ways to write formulas. Choose the best one that fits your needs for each scenario.
* Formulas are made up of three basic parts, the equals sign, a function or operator, and arguments. There are two main ways to write Excel formulas, with an operator or with a function. Functions can beused as a short cut to create commonly used formulas.
* Lastly are the arguments. Arguments are the numbers or cell references that will be used to figure a formula.
* **Example using operator: =** B1+B2+B3+B4+B5+B6+B7. The operator in this formula is the + sign.
* **Example using function:** =SUM(B1:B7). The function in this formula is SUM (which is for adding values).
* Formulas using functions have a different format than formulas using operators.
* Formulas that use function also start with an equals sign, but instead of a mixture of arguments and operators, the function is entered right after the equal sign with the arguments following in parentheses.
* **How to write a formula using a function**

1. **Individual Arguments**

Individual arguments can be used to reference specific cells within a spreadsheet. Commas can be used to separate individual arguments. **Example**

**= Function (Argument1; Argument 3)**

1. **Through Arguments**

A colon can be used to reference arguments that are in consecutive order. **Example**

**= Function(Argument 1:Argument 3).**

* **Calculating Across Worksheets**

Calculating across worksheets will allow you to pull values from multiple worksheets to create a desired formula. When calculating across worksheets you will need to reference sheet names.

The symbol to reference a sheet is an exclamation point (**!**). The exclamation point will need to be added to the end of the sheet name.

* **Referencing Multiple Worksheets**

When referencing a different sheet than the one the formula will appear on, the formula will need to include an exclamation point after the sheet name. For example I have a worksheet with the name Library, to reference that sheet name I would type Library! Within the parenthesis of the argument area. The cell reference will follow the sheet name.

* **Sheet Separators**

If you will be referencing sheets that are in a consecutive row, use the symbol for through, which is a colon (:).

If you will be referencing only specific sheets use a comma ( , or ; ) between the sheet names.

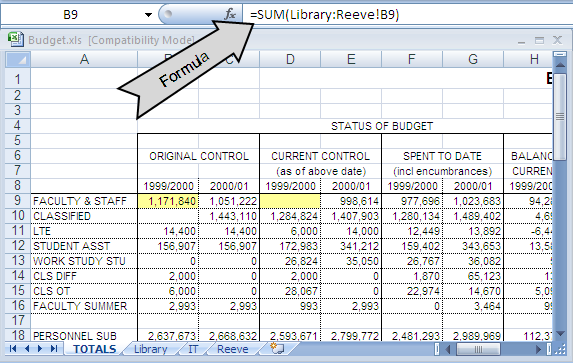
* **3-D Cell References**

Three-dimensional cell references are used when referencing the same cell across multiple worksheets. An example of a 3-dimentional reference is shown below.

**Using formulas for sheets in consecutive order:**

The example below shows how to create a formula with consecutive worksheets.

**=SUM(Library:Reeve!B9)**

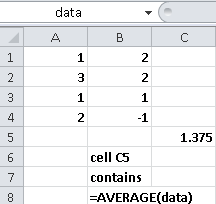


* **How Can I Create Named Ranges?**

There are three ways to create named ranges:

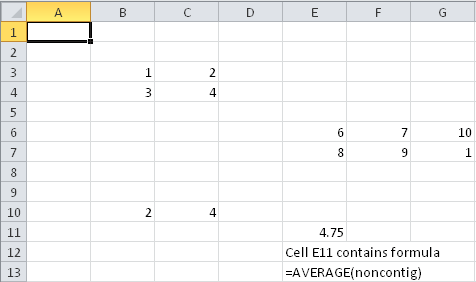
1. **Using the Name Box to Create a Range Name**

In previous sections we explain the methods of give name to single cell. **Now:** **Suppose you want to assign a rectangular range of cells** (such as A1:B4) the name *Data*. Simply select the cell range A1:B4, type **Data** in the Name box, and press Enter. Now a formula such as *=AVERAGE(Data)* would average the contents of cells A1:B4. As shown in the next figure .



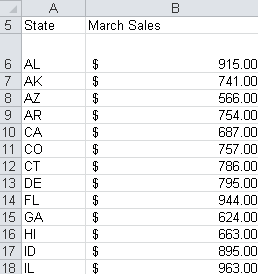
**Note**1 :Sometimes you want to name a range of cells made up of several noncontiguous rectangular ranges. For example, in the next Figure we might want to assign the name ***Noncontig*** to the range consisting of cells B3:C4, E6:G7, and B10:C10. To assign this name, select any one of the three rectangles making up the range (I chose B3:C4). ***Hold down the Ctrl key, and then select the other two ranges*** (E6:G7 and B10:C10).

Now release the Ctrl key, type the name **Noncontig** in the Name box, and press Enter. Using *Noncontig* in any formula will now refer to the contents of cells B3:C4, E6:G7, and B10:C10. For example, entering the formula *=AVERAGE(Noncontig)* in cell E10 yields 4.75 (because the 12 numbers in our range add up to 57 and 57/12=4.75).



1. **Creating Named Ranges by Using the Create From Selection Option**

The worksheet in the next figure contains sales during March for each of the 50 U. S. states. Figure 2-1 show a subset of this data. We would like to name each cell in the range B6:B55 with the correct state abbreviation. To do this, select the range A6:B55, and click Create From Selection in the Defined Names group on the Formulas tab (see Figure 2-2). Then select the Left Column check box, as indicated in Figure 2-3.

****

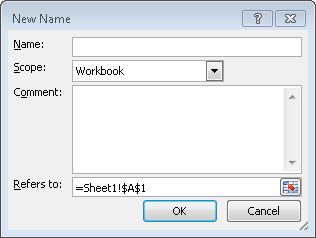
**FIGURE 2-1** By naming the cells that contain state sales with state abbreviations, you can use the abbreviation rather than the cell’s column letter and row number when you refer to the cell.

|  |  |
| --- | --- |
|  |  |
| **FIGURE 2-2** Select Create From Selection. | **FIGURE 2-3** Select the Left Column check box |

1. **Creating Range Names by Using the Name Manager Option**

If you click Name Manager on the Formulas tab and then choose New, the New Name dialog box

shown in Figure 2-4 opens.



**FIGURE 2-4** The New Name dialog box before creating any range names.

If you click the Scope arrow, you can select the option Workbook or any worksheet in your workbook. I’ll discuss this decision later, so for now just choose the default scope of Workbook. You can also add comments for any of your range names.

* **A Name’s Scope**

it’s important to understand that all names have a scope. A name’s ***scope*** defines where you can use the name. Names are scoped at either of two levels:

* **Workbook-level names:** Can be used in any worksheet. This is the , default type of range name . like = sale1 , where sale1 cell name.
* **Worksheet-level names:** Can be used only in the worksheet in which they are defined, unless they are preceded with the worksheet’s name. A workbook may contain multiple worksheet-level names that are identical.like , sheet1!sale1 .

**Note1:** If your worksheet name contains a space, enclose the worksheet name in single quotes when referring to a name defined on that sheet:

**=’My Sheet’!My\_Name**

**Note2:** Only the worksheet-level names on the current sheet appear in the Name box. Similarly, only worksheet-level names on the current sheet appear in the list under Formulas➜ Defined Names➜Use in Formulas.

* **Referencing names from another workbook**

The same rules apply when using names defined in another workbook.

**Note1:** For example, the following formula uses a range named *MonthlySales,* defined in a workbook named Budget.xlsx (which is assumed to be open):

**=AVERAGE(Budget.xlsx!MonthlySales)**

**Note2:** If the name *MonthlySales* is a worksheet-level name on Sheet1, the formula looks like this: **=AVERAGE([Budget.xlsx]Sheet1!MonthlySales)**

**Calculating Across Worksheets Examples**

***Example -1:*  find total summation Across consecutive orders**

Assume you enter the data of three student in three different courses in three consecutive sheet in the same worksheet (course1, course2, course3) as showen in the following figures.

|  |  |  |
| --- | --- | --- |
| 3d_1.png | 3d_2.png | 3d_3.png |
| **Course1-sheet** | **Course2-sheet** | **Course3-sheet** |
| **To calculate the total sum in total sheet :**   1. Select cells c2:c4 and type the formula :**=sum(course1:course3!c2)** 2. Press **<Ctrl +Enter>** | **3d_total.png** | |

* **Rules for naming names**

**we can samurais the rules for naming names as follows :**

* **Excel does not allow you to use the letters *r* and *c* as range names.**
* **If you use Create From Selection to create a range name and your name contains spaces, Excel inserts an underscore (\_) to fill in the spaces. For example, the name *Product 1* is created as *Product\_1*.**
* **Range names cannot begin with numbers or look like a cell reference. For example, *3Q* and *A4* are not allowed as range names. Because Excel 2010 has over 16,000 columns,**
* **a range name such as *cat1* is not permitted because there is a cell CAT1. If you try and name a cell *CAT1*, Excel tells you the name is invalid. Probably your best alternative is to name the cell cat1\_.**
* **The only symbols allowed in range names are periods (.) and underscores ( \_).**
* **Functions**
* **What Is a Function?**

A ***worksheet function*** is a built-in tool that you use in a formula. Worksheet functions allow you to perform calculations or operations that would otherwise be impossible. You’ll find functions useful because they

* **Simplify your formulas**
* **Permit formulas to perform otherwise impossible calculations**
* **Speed up some editing tasks**
* **Allow *conditional* execution of formulas — giving them rudimentary decision-making capability**
* **Function Argument Types**

. The information within the parentheses is the function’s *arguments.* Functions vary in how they use arguments. A function may use

* **No arguments**
* **One argument**
* **A fixed number of arguments**
* **An indeterminate number of arguments**
* **Optional arguments**
* **Names as arguments**

To calculate the sum of the values in A1:A20, you\ can use , **=SUM(A1:A20) ,** And, not surprisingly, if you’ve defined a name for A1:A20 (such as *Sales*), you can use the name in place of the reference: **=SUM(Sales)**

* **Full-column or full-row as arguments**

For example, the following formula sums all values in column B: =SUM(B:B) Using full-column and full-row references is particularly useful if the range that you’re summing changes. **If you do use an entire row or column, just make sure that the row or column doesn’t contain extraneous information that you don’t want to include in the sum.**

* **Literal values as arguments**

A ***literal argument*** refers to a value or text string that you enter directly. For example, the SQRT function, which calculates the square root of a number, takes one argument. In the following example, the formula uses a literal value for the function’s argument: **=SQRT(225)**

Using a literal argument with a simple function like this one usually defeats the purpose of using a formula. This formula always returns the same value, so you could just as easily replace it with the value 15. You may want to make an exception to this rule in the interest of clarity. For example, you may want to make it perfectly clear that you are computing the square root of 225

Using literal arguments makes more sense with formulas that use more than one argument. For example, the LEFT function (which takes two arguments) returns characters from the beginning of its first argument; the second argument specifies the number of characters. If cell A1 contains the text *Budget*, the following formula returns the first letter, or B: =LEFT(A1,1)

* **Expressions as arguments**

Excel also enables you to use expressions as arguments. Think of an *expression* as a formula within a formula (but without the leading equal sign). When Excel encounters an expression as a function’s argument, it evaluates the expression and then uses the result as the argument’s value. Here’s an example:

**=SQRT((A1^2)+(A2^2))**

This formula uses the SQRT function, and its single argument appears as the following expression: **(A1^2)+(A2^2)**

**Note: When** Excel evaluates the formula, it first evaluates the expression in the argument and then computes the square root of the result.

* **Other functions as arguments**

Because Excel can evaluate expressions as arguments, it shouldn’t surprise you that these expressions can include other functions. Writing formulas that have functions within functions is sometimes known as ***nesting functions.***

Excel starts by evaluating the most deeply nested expression and works its way out. Note this example of a nested function:  **=SIN(RADIANS(B9))**

* **Arrays as arguments**

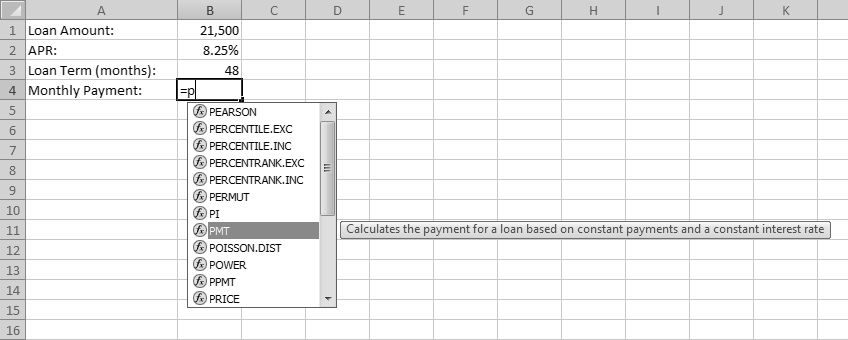
A function can also use an array as an argument. An ***array*** is a series of values separated by a comma and enclosed in brackets. The formula below uses the OR function with an array as an argument. The formula returns TRUE if cell A1 contains 1, 3, or 5. **=OR(A1={1,3,5})**

* **Ways to Enter a Function into a Formula**

You can enter a function into a formula by typing it manually, by using the Function Library commands, or by using the Insert Function dialog box.

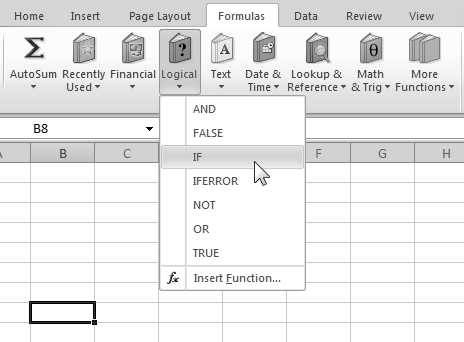
1. **Entering a function manually**

If you’re familiar with a particular function — you know its correct spelling and the types of arguments that it takes — you may choose to simply type the function and its arguments into your formula. Often, this method is the most efficient. As shown in the next figure



1. **Using the Fun ction Library commands**

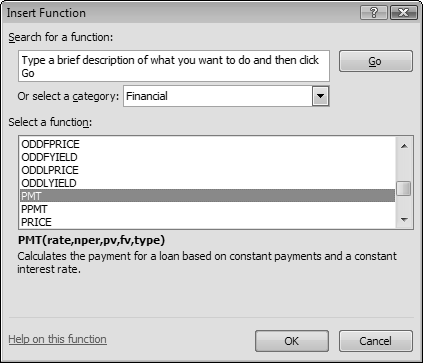
Another way to insert a function into a formula is to use the icons in the Formulas ➜ Function Library group. The next Figure shows these icons, each of which is a drop-down control.



1. **Using the Insert Function dialog box**

To insert a function, select the function from the Insert Function dialog box, as shown in the next Figure. You access this dialog box by Choosing

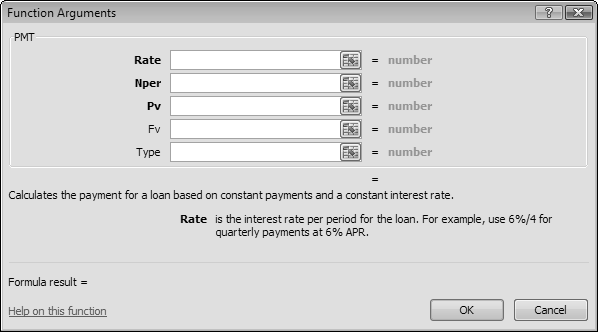
* **Formulas ➜Function**
* **Library➜Insert Function Choosing**
* **Formulas➜Function Library➜AutoSum, and then clicking More Functions in the drop-down list**
* **Clicking the *fx* icon to the left of the Formula bar**
* **Pressing Shift+F3**

****

**Figure:** The Insert Function dialog box**.**

When you locate the function that you want to use, click OK. Excel’s Function ***Arguments dialog box*** appears, as shown in the next Figure. Use the Function Arguments dialog box to specify the arguments for the function.

You can easily specify a ***range argument*** by clicking the Collapse Dialog button (the icon at the right edge of each argument field). Excel temporarily collapses the Function Arguments dialog box to a thin box, so that you can select a range in the worksheet.



**Figure :** The Function Arguments dialog box.

* **Editing and printing**
* **Formatting the Worksheet**

Although the worksheet contains the appropriate data, formulas, and functions, the text and numbers need to be formatted to improve their appearance and readability. a worksheet that is formatted not only is easier to read but also looks more professional. By formatting the contents of the worksheet, you can improve its overall appearance. When formatting a worksheet, consider the following formatting suggestions:

* **Increase the font size of cells**
* **Change the font color of cells**
* **Center the worksheet titles, subtitles, and column headings**
* **Modify column widths to best fit text in cells**
* **Change the font style of cells.**
* **Checking Spelling**
* Excel includes a **spell checker** in **Review Submenu** you can use to check a worksheet for spelling errors.
* The spell checker looks for spelling errors by comparing words on the worksheet against words contained in its standard dictionary. If you often use specialized terms that are not in the standard dictionary, you may want to add them to a custom dictionary using the Spelling dialog box.
* When the spell checker finds a word that is not in either dictionary, it displays word in the Spelling dialog box. You then can correct it if it is misspelled.
* **Preparing to Print the Worksheet**
* Excel allows for a great deal of customization in how a worksheet appears when printed. For example, the margins on the page can be adjusted. A header or footer can be added to each printed page as well.
* Excel also has the capability to work on the worksheet in Page Layout view.
* **Page Layout view** allows you to create or modify a worksheet while viewing how it will look in printed format.
* The default view that you have worked in up until this point in the book is called **Normal view**.

**🕭Tips*: Specify how the printed worksheet should appear.***

* + Before printing a worksheet, you should consider how the worksheet will appear when printed.
  + In order to fit as much information on the printed page as possible, the margins
  + of the worksheet should be set to a reasonably small width and height.
  + While the current version of a worksheet may print on one page, you may add more data in the future that causes the worksheet to extend to multiple pages. It is, therefore, a good idea to add a page header to the worksheet that prints in the top margin of each page.
  + A **header** is common content that prints on every page of a worksheet.

Landscape orientation is a good choice for large worksheets because the printed worksheet’s width is greater than its length

***💣 Important notes about Formula***

* **Using constants in formulas:** A constant is a value that is not calculated. For example, the date 10/9/2008, the number 210, and the text "Quarterly Earnings" are all constants. An expression, or a value resulting from an expression, is not a constant. If you use constant values in the formula instead of references to the cells (for example, =30+70+110), the result changes only if you modify the formula yourself.
* **Types of operators:** There are four different types of calculation operators: arithmetic, comparison, text concatenation, and reference.
* **Arithmetic operators:** toperform basic mathematical operations such as addition, subtraction, or multiplication; combine numbers; and produce numeric results, use the following arithmetic operators.

|  |  |  |
| --- | --- | --- |
| **Arithmetic operator** | **Meaning** | **Example** |
| + (plus sign) | Addition | 3+3 |
| – (minus sign) | Subtraction  Negation | 3–1 –1 |
| \* (asterisk) | Multiplication | 3\*3 |
| / (forward slash) | Division | 3/3 |
| % (percent sign) | Percent | 20% |
| ^ (caret) | Exponentiation) | 3^2 |

* **Comparison operators:** You can compare two values with the following operators. When two values are compared by using these operators, the result is a logical value either TRUE or FALSE

|  |  |  |
| --- | --- | --- |
| **Comparison operator** | **Meaning** | **Example** |
| = (equal sign) | Equal to | A1=B1 |
| > (greater than sign) | Greater than | A1>B1 |
| < (less than sign) | Less than | A1<B1 |
| >= (greater than or equal to sign) | Greater than or equal to | A1>=B1 |
| <= (less than or equal to sign) | Less than or equal to | A1<=B1 |
| <> (not equal to sign) | Not equal to | A1<>B1 |

* **Text concatenation operator:** Use the ampersand (&) to join, or concatenate, one or more text strings to produce a single piece of text.

|  |  |  |
| --- | --- | --- |
| **Text operator** | **Meaning** | **Example** |
| & (ampersand) | Connects, or concatenates, two values to produce one continuous text value | "North"&"wind" |

* **Reference operators :** Combine ranges of cells for calculations with the following operators.

|  |  |  |
| --- | --- | --- |
| **Reference operator** | **Meaning** | **Example** |
| : (colon) | Range operator, which produces one reference to all the cells between two references, including the two references | B5:B15 |
| , (comma) or  ; (semicolon) | Union operator, which combines multiple references into one reference | SUM(B5:B15,D5:D15)  SUM(B5:B15;D5:D15) |
| (space) | Intersection operator, which produces on reference to cells common to the two references | B7:D7 C6:C8 |

* **Formulas error**
* **##### Error : This error occurs when a column is not wide enough, or a negative date or time is used**.
* **#DIV/0! Error: This error occurs when a number is divided by zero (0).**
* **#N/A error: This error occurs when a value is not available to a function or formula.**
* **#NAME? Error :** **This error occurs when Microsoft Office Excel doesn't recognize text in a formula**
* **#NULL! Error:** **This error occurs when you specify an intersection of two areas that do not intersect. The intersection operator is a space between references>**
* **#NUM! Error:** **This error occurs with invalid numeric values in a formula or function**.
* **#REF! Error:** **This error occurs when a** [**cell reference (cell reference: The set of coordinates that a cell occupies on a worksheet. For example, the reference of the cell that appears at the intersection of column B and row 3 is B3.)**](javascript:AppendPopup(this,'xldefCellReference_1')) **is not valid**
* **#VALUE! Error:** **This error occurs when the wrong type of** [**argument (argument: The values that a function uses to perform operations or calculations. The type of argument a function uses is specific to the function. Common arguments that are used within functions include numbers, text, cell references, and names.)**](javascript:AppendPopup(this,'IDH_xldefArgument_1')) **or** [**operand (operand: Items on either side of an operator in a formula. In Excel, operands can be values, cell references, names, labels, and functions.)**](javascript:AppendPopup(this,'IDH_xldefOperand_2')) **is used.**

**Different Examples for Formulas in Excel**

***Example -1:*** ***Convert from hours to minutes***

As a task, time in a timesheet has to be converted from hours minutes.

|  |  |
| --- | --- |
| **To convert time to minutes:**  **1.** In a worksheet, enter your own data or the data shown in next Figure  **2.** Select cells B4:B8.  **3.** Enter the following formula: **=A4\*24\*60**.  **4.** Press **<Ctrl+Enter>**.  **5.** Format cells B4:B8 as general by selecting the Format menu and choosing **Cells** and **General** then clicking **OK**. |  |

**Note:** To convert minutes to hours and minutes format, usethe formula =b4/24/60. Remember to format the cells with a time format, as shown in cell C4 in above Figure

***Example-4:* Incrementing row numbers**

Standard row numbering in Excel is often used, but you can also create your own numbering system in a table, such as incrementing by 10 as described below.

|  |  |
| --- | --- |
| **To increment row numbers by 10:**  **1.** Select cell A2 and type **0**.  **2.** Select cell A3 and type the following formula: **=A2+10**.  **3.** Select cells A3:A12.  **4.** From the Edit menu, select **Fill** and **Down**.  Note: If the value of cell A2 is changed, the values in all other cells change too. |  |

***Example-5:* Combine text and numbers**

In this example, we want to combine text and numbers. Use the & operator to accomplish this.

|  |  |
| --- | --- |
| **To combine cells containing text and numbers:**  **1.** Select cell B1 and type **computers**.  **2.** Select cell B2 and type **5**.  **3.** Select cell B4 and type the following formula:  **="You** **ordered"&B2&""&B1&"today!"**. |  |

**Note:** Each cell reference must be placed between **&** operators and additional text must be surrounded by quotation marks.

* **Function Categories**

I list and briefly describe Excel’s function categories in the following sections.

|  |  |
| --- | --- |
| **Function Categories** | **Short Description** |
| **Financial functions** | The financial functions enable you to perform common business calculations that deal with money. For example, you can use the PMT function to calculate the monthly payment for a car loan. (You need to provide the loan amount, interest rate, and loan term as arguments.) |
| **Date and time functions** | The functions in this category enable you to analyze and work with date and time values in formulas. For example, the TODAY function returns the current date (as stored in the system clock). |
| **Math and trig functions** | This category contains a wide variety of functions that perform mathematical and trigonometric calculations. |
| **Statistical functions** | The functions in this category perform statistical analysis on ranges of data. For example, you can calculate statistics such as mean, mode, standard deviation, and variance. Excel 2010 includes many new functions in this category |
| **Lookup and reference functions** | Functions in this category are used to find (look up) values in lists or tables. A common example is a tax table. You can use the VLOOKUP function to determine a tax rate for a particular income level. |
| **Database functions** | Functions in this category are useful when you need to summarize data in a list (also known as a worksheet database) that meets specific criteria. For example, assume you have a list that containsmonthly sales information. You can use the DCOUNT function to count the number of records that describe sales in the Northern region with a value greater than 10,000. |
| **Text functions** | The text functions enable you to manipulate text strings in formulas. For example, you can use the MID function to extract any number of characters beginning at any character position. Other functions enable you to change the case of text (convert to uppercase, for example). |
| **Logical functions** | This category consists of only seven functions that enable you to test a condition (for logical TRUE or FALSE). You will find the IF function very useful because it gives your formulas simple decision-making capabilities. |
| **Information functions** | The functions in this category help you determine the type of data stored within a cell. For example, the ISTEXT function returns TRUE if a cell reference contains text. Or you can use the ISBLANK  function to determine whether a cell is empty. The CELL function returns lots of potentially useful information about a particular cell. |
| **User-defined functions** | Functions that appear in this category are custom worksheet functions created by using VBA. These functions can operate just like Excel’s built-in functions. One difference, however, is that  custom functions do not always display a description of each argument in the Paste Function dialog box. It’s up to the programmer to provide these descriptions. Also, user-defined functions do not convert to uppercase when you enter them. |
| **Engineering functions** | The functions in this category can prove useful for engineering applications. They enable you to work with complex numbers and to perform conversions between various numbering and measurement systems. |
| **Cube functions** | The functions in this category allow you to manipulate data that is part of an OLAP data cube. |
| **Compatibility functions** | The Compatibility category is new to Excel 2010. Functions in this category are statistical functions that have been replaced with more accurate functions. However, they are still available for situations in which you need to share your workbook with those who don’t have Excel 2010. |
| **Other function categories** | In addition to the function categories described previously, Excel includes four other categories that may not appear in the Paste Function dialog box: Commands, Customizing, Macro Control, and DDE/External. These categories appear to be holdovers from older versions of Excel. If you create a custom function, you can assign it to one of these categories. In addition, you may see other function categories created by macros. |

**Note : The brief Description and Examples of each of Function Categories listed in Lab Sheet**

.

**💣Note: the Examples about each Categories will listed in later**

* **logical functions**

Logical functions use to compare between values and give logical result only **(True, False)**,the following table describe these Category functions, which list ***all* t**hese function:

|  |  |
| --- | --- |
| **Function** | **Description** |
| [**AND**](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP05208986.htm) | **Returns TRUE if all its arguments are TRUE; returns FALSE if one or more argument is FALSE.**  **Syntax : AND(logical1 ; logical2 ; ...)**  **Logical1, logical2, ...** are 1 to 255 conditions you want to test that can be either TRUE or FALSE.  **Remarks**   * The arguments must evaluate to logical values such as TRUE or FALSE, or the arguments must be [arrays (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)](javascript:AppendPopup(this,'xldefArray_1')) or references that contain logical values. * If an array or reference argument contains text or empty cells, those values are ignored. * If the specified range contains no logical values, AND returns the #VALUE! error value. |
| [**FALSE**](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP05209086.htm) | **Returns the logical value FALSE**.  **Syntax :**   **FALSE**( )  **Remark** : You can also type the word FALSE directly onto the worksheet or into the formula, and Microsoft Excel interprets it as the logical value FALSE. |
| [**IF**](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP05209118.htm) | **Returns one value if a condition you specify evaluates to TRUE and another value if it evaluates to FALSE. Use IF to conduct conditional tests on values and formulas.**  **Syntax :** **IF**(**logical\_test ; value\_if\_true ; value\_if\_false)**  **Logical\_test** :   is any value or expression that can be evaluated to TRUE or FALSE  **Value\_if\_true** :  is the value that is returned if logical\_test is TRUE.  **Value\_if\_false:**   is the value that is returned if logical\_test is FALSE  **Remarks**   * Up to 64 IF functions can be nested as value\_if\_true and value\_if\_false arguments to construct more elaborate tests. * When the value\_if\_true and value\_if\_false arguments are evaluated, IF returns the value returned by those statements. * If any of the arguments to IF are [**arrays (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)**](javascript:AppendPopup(this,'xldefArray_1'))**,** every element of the array is evaluated when the IF statement is carried out. * Microsoft Excel provides additional functions that can be used to analyze your data based on a condition. For example,[COUNTIF](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP10069840.htm) , [COUNTIFS](ms-help://MS.EXCEL.12.1033/EXCEL/content/HA10047494.htm) , [SUMIF](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP10062465.htm) and [SUMIFS](ms-help://MS.EXCEL.12.1033/EXCEL/content/HA10047504.htm) functions. |
| [**NOT**](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP05209196.htm) | **Reverses the value of its argument. Use NOT when you want to make sure a value is not equal to one particular value**.  **Syntax** : **NOT**(**logical**)  **Logical**   is a value or expression that can be evaluated to TRUE or FALSE.  **Remark** :If logical is FALSE, NOT returns TRUE; if logical is TRUE, NOT returns FALSE. |
| [**OR**](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP05209209.htm) | Returns TRUE if any argument is TRUE; returns FALSE if all arguments are FALSE.  **Syntax** : **OR**(**logical1** ; logical2 ; ...)  **Logical1, logical2,...**   are 1 to 255 conditions you want to test that can be either TRUE or FALSE.  **Remarks** :The arguments must evaluate to logical values such as TRUE or FALSE, or in [arrays (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)](javascript:AppendPopup(this,'xldefArray_1')) or references that contain logical values.   * If an array or reference argument contains text or empty cells, those values are ignored. * If the specified range contains no logical values, OR returns the #VALUE! error value. * You can use an OR array formula to see if a value occurs in an array. To enter an array formula, press CTRL+SHIFT+ENTER. |
| [**TRUE**](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP05209323.htm) | **Returns the logical value TRUE.**  **Syntax** : **TRUE**( )  **Remark** : You can enter the value TRUE directly into cells and formulas without using this function. The TRUE function is provided primarily for compatibility with other spreadsheet programs |

***Example-1:* Use the AND function to compare two columns**

Two columns in a worksheet have to be evaluated. If the value in column A is greater than 20 and the value in column B is greater than 25, both values are valid

|  |  |
| --- | --- |
| **To compare two columns:**  **1.** In cells A2:A10, enter values from 1 to 100.  **2.** In cells B2:B10, enter values from 1 to 100.  **3.** Select cells C2:C10 and type the following formula: **=AND(A2>20 ; B2>25)**.  4. Press **<Ctrl+Enter>**. |  |

**Note: If both criteria are valid, Excel shows the value as TRUE; otherwise it is FALSE.**

***Example-2:* Use the AND function to show sales for a specific period of time**

This example checks all rows for a specific time period using the AND function. The function returns TRUE if the arguments are TRUE and FALSE if one or more arguments are FALSE.

**Note: Up to 30 conditions can be used in one formula.**

|  |  |
| --- | --- |
| **To show sales in a period of time:**  **1.** Select cell B1 and ent er the start date.  **2.** Select cell B2 and enter the end date.  **3.** The range A5:A16 contains dates from 09/13/04 to 09/21/04.  **4.** The range B5:B16 contains sales amounts.  **5.** Select cells C5:C16 and type the following =**AND(A5>=$B$1 ; A5<=$B$2)**.  **6.** Press **<Ctrl+Enter>**. |  |

***Example-3:* Use the OR function to check cells for text**

A worksheet contains several words in column A. Each row has to be checked for the words “new” or “actual” in column A. The OR function is used for this task. The function returns TRUE if either argument is true and FALSE if the arguments are not true.

**Note:** Up **to 30 conditions can be used in one formula.**

|  |  |
| --- | --- |
| **To use the OR function to check for two or more criteria:**  **1.** Enter in range A2:A11 words like “new,” “actual,” and “old.”  **2.** Select cells B2:B11 and type the following formula:  **=OR(A2="New" ; A2="actual")**.  **3.** Press **<Ctrl+Enter>**. |  |

***Example-4:* Use the OR function to check cells for numbers**

A worksheet contains several values in column A. Each row has to be evaluated based on certain criteria in column A. The OR function is used for this task. The function returns TRUE if any argument is TRUE and FALSE if all arguments are FALSE.

|  |  |
| --- | --- |
| **To check for two or more criteria:**  **1.** Enter in range A2:A12 values from –43 to 100.  **2.** Select cells B2:B12 and type the following formula: **=OR(A2=1 ; A2>=99,A2<0)**.  **3.** Press **<Ctrl + Enter>**. |  |

***Example-5:* Use the IF function to compare columns and return a specific result**

As shown in earlier examples, Excel returns the value TRUE or FALSE when using the OR and AND functions. The IF function can also be used to conduct conditional tests on values and formulas. This example compares two columns and shows the result in column C. This example compares two columns and shows the result in column C.

|  |  |
| --- | --- |
| ***To return specific text after comparing values:***  **1.** Enter in range A2:A12 values from 1 to 1000.  **2.** Enter in range B2:B12 values from 1 to 1000.  **3.** Select cells C2:C12 and type the following formula:  **=IF(A2>=B2 ; "Column A is greater or equal" ; "Column** **B is greater")**.  **4.** Press **<Ctrl+Enter>**. |  |

***Example-6:*  Use the IF function to check for larger, equivalent, or smaller values**

In the previous example, two different messages were used as the result for comparing values. To check for three conditions in column A and present the result as “Column A is larger,” “equal,” or “Column A is smaller,” perform the following steps.

|  |  |
| --- | --- |
| ***To compare columns and show the result*:**  **1.** enter the your data .  **2.** Select cells C2:C12 and type the following formula:  **=IF(A2>B2; "Column A is larger "; IF(A2=B2; "equal" ; "Column A is smaller"))**.  **3.** Press **<Ctrl + Enter>**.  ***Note: Up to seven IF functions can be combined in one cell*.** |  |

***Example-7:* Use the IF function to determine the quarter of a year**

After entering an initial value, Excel can automatically fill worksheet cells with the names of weekdays or months. Open a new worksheet and type the word “January” in cell A2. Then drag the lower-right point of this cell down to A13 to let Excel create a list containing the months of the year. In this example, we want to indicate which months fall into which quarter.

|  |
| --- |
| **To determine the quarter of a year in which a particular**   1. Select cells B2:B13 and type the following formula:   **=IF(OR(A2="January";A2="February";A2="March"),"1stquarter"; IF(OR(A2="April";A2="May";A2="June");"2ndquarter";IF(OR(A2="July";A2="August"; A2="September");"3rdquarter"; "4th quarter")))**.  **2.**  Press **<Ctrl+Enter>**. |
|  |

***Example-8:***  **Use the IF function to calculate with different tax rates**

If two or more different tax rates have to be handled, you can use the IF function to calculate each one individually. Simply combine several IF functions, depending on the calculation.

|  |  |
| --- | --- |
| **To calculate the price after tax:**  **1.** In column A, enter some prices.  **2.** In column B, enter different tax percentages (0, 8, or 10 for this example).  **3.**Select cells C2:C10 and type the following formula:  **=IF(B2=8;A2/100\*8;IF(B2=10;A2/100\*10;A2/100\*0))**.  **4.** Press **<Ctrl + Enter>**.  **5.** Select cells D2:D10 and type the formula **=A2+C2**.  **6.** Press **<Ctrl +Enter>**. |  |

***Example-9:* Use the IF function to calculate the commissions for individual sales**

A company has a policy for individual commissions depending on sales, as shown below:

**Sale < $100 3%**

**Sale => $100 and < $500 5%**

**Sale >= $500 8%**

|  |  |
| --- | --- |
| **To calculate the commissions:**   1. Enter different possible sales amounts in column A.   **2.** Select cells B2:B12 and type the following formula:  **=A2\*IF(A2>=500;0.08;IF(A2>=100;0.05,0.03))**.   1. Press **<Ctrl + Enter>**. |  |

***Example-10* Use the IF function to compare two cells**

The following tip is a solution for comparing two cells line by line. Prepare a new worksheet, filling the first two columns with the values 0 and 1 as shown in the next Figure

|  |  |
| --- | --- |
| 1. Select cell 2. s C2:C11 and type the following formula:   **=IF(A2&B2="11";"OK",IF(A2&B2="10";"First Value is OK"; IF(A2&B2="01";"Second Value is OK"; ”Both Values are FALSE”)))**.   1. Press **<Ctrl+ Enter>**. |  |

***Example-11* Use the INT function with the IF function**

To see if one value is a whole number and can be divided by another value, use the IF function in combination with the INT function.

|  |  |
| --- | --- |
| **To see if a whole number can be divided by 4:**   1. Select cells B2:B10 and type the following formula:   **=IF(INT(A2/4)=A2/4;"whole number divisible by 4";FALSE)**.   1. Press **<Ctrl + Enter>**.   **Or**   1. Select cells C2:C10 and type the following formula:   **=IF(A2/4-INT(A2/4)=0;"whole number divisible by 4"; FALSE)**.   1. Press **<Ctrl+ Enter>**. |  |

***Example-12* Use the IF function to check whether a date is in the past or future**

In this example we want to check whether a particular date is in the past or the future. To do so, the TODAY() function is used with IF to compare dates with the actual date and show its result.

|  |
| --- |
| **To compare dates — variant A:**  **1.** Select cell B2:B11 and type the following formula:  **=IF(NOT(A2>TODAY());"past" ;"future")**.  **2.** Press **<Ctrl+ Enter>**.  **To compare dates — variant B:**  **1.** Select cell B2:B11 and type the following formula:  **=IF(A2>=TODAY() ; IF(A2=TODAY(); "Today"; "Future"); "Past")**.  **2.** Press **<Ctrl+ Enter>**. |
|  |

* **Statistical functions**

The functions in this category perform statistical analysis on ranges of data. For example, you can calculate statistics such as mean, mode, standard deviation, and variance. the following table describe these Category functions, which list ***some*  of** these function:

|  |  |
| --- | --- |
| **Fu­­­nction** | **Description** |
| **AVERAGE** | ***Returns the average (arithmetic mean) of the arguments.***  **Syntax :** **AVERAGE**(**number1**,number2,...)  **Number1, number2, ...**   are 1 to 255 numeric arguments for which you want the average.  **Remarks**   * Arguments can either be numbers or names, arrays, or references that contain numbers. * Logical values and text representations of numbers that you type directly into the list of arguments are not counted. * If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value zero are included. * Arguments that are error values or text that cannot be translated into numbers cause errors. * If you want to include logical values and text representations of numbers in a reference as part of the calculation, use **the AVERAGEA** function. |
| **AVERAGEA** | ***Calculates the average (arithmetic mean) of the values in the list of arguments***.  **Syntax : AVERAGEA(value1,value2,...)**  **Value1, value2, ...**   are 1 to 255 cells, ranges of cells, or values for which you want the average.  **Remarks**   * Arguments can be the following: numbers; names, arrays, or references that contain numbers; text representations of numbers; or logical values, such as TRUE and FALSE, in a reference. * Logical values and text representations of numbers that you type directly into the list of arguments are counted. * Arguments that contain TRUE evaluate as 1; arguments that contain FALSE evaluate as 0 (zero). * Array or reference arguments that contain text evaluate as 0 (zero). Empty text ("") evaluates as 0 (zero). * If an argument is an array or reference, only values in that array or reference are used. Empty cells and text values in the array or reference are ignored. * Arguments that are error values or text that cannot be translated into numbers cause errors. * If you do not want to include logical values and text representations of numbers in a reference as part of the calculation, use the **AVERAGE** function * ***Example :***  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | **1** | | **2** | | **3** | | **4** | | **5** | | **6** | | **7** | | |  | | --- | | **A** | | **Data** | | **10** | | **7** | | **9** | | **2** | | **Not available** | |  | | Formula | Description (Result) | | **=AVERAGEA(A2:A6)** | Average of the numbers above, and the text "Not Available". The cell with the text "Not available" is used in the calculation. (5.6) | | **=AVERAGEA(A2:A5,A7)** | Average of the numbers above, and the empty cell. (7) | | |
| **AVERAGEIF** | ***Returns the average (arithmetic mean) of all the cells in a range that meet a given criteria.***  **Syntax** :**AVERAGEIF**(**range**,**criteria**,[average\_range])  **Range**  is one or more cells to average, including numbers or names, arrays, or references that contain numbers.  **Criteria**  is the criteria in the form of a number, expression, cell reference, or text that defines which cells are averaged. For example, criteria can be expressed as 32, "32", ">32", "apples", or B4.  **Average\_range** Optional  is the actual set of cells to average. If omitted, range is used.  **Remarks**   * Cells in range that contain **TRUE** or **FALSE** are ignored. * If a cell in average\_range is an empty cell, **AVERAGEIF** ignores it. * If range is a blank or text value, **AVERAGEIF** returns the #DIV0! error value. * If a cell in criteria is empty, **AVERAGEIF** treats it as a 0 value. * If no cells in the range meet the criteria, **AVERAGEIF** returns the #DIV/0! error value. * You can use the wildcard characters, question mark (**?**) and asterisk (**\***), in criteria. A question mark matches any single character; an asterisk matches any sequence of characters. If you want to find an actual question mark or asterisk, type a tilde (**~**) before the character. * **Average\_range** does not have to be the same size and shape as range. The actual cells that are averaged are determined by using the top, left cell in average\_range as the beginning cell, and then including cells that correspond in size and shape to range. For example  |  |  |  | | --- | --- | --- | | **If range is** | **And average\_range is** | **Then the actual cells evaluated are** | | **A1:A5** | **B1:B5** | **B1:B5** | | **A1:A5** | **B1:B3** | **B1:B5** | | **A1:B4** | **C1:D4** | **C1:D4** | | **A1:B4** | **C1:C2** | **C1:D4** | |
| **AVERAGEIFS** | ***Home work*** |
| **COUNT** | ***Counts the number of cells that contain numbers and counts numbers within the list of arguments. Use COUNT to get the number of entries in a number field that is in a range or array of numbers.***  **Syntax : COUNT**(**value1**,value2,...)  **Value1, value2, ...**   are 1 to 255 arguments that can contain or refer to a variety of different types of data, but only numbers are counted.  **Remarks**   * Arguments that are numbers, dates, or text representation of numbers are counted. * Logical values and text representations of numbers that you type directly into the list of arguments are not counted. * Arguments that are error values or text that cannot be translated into numbers are ignored. * If an argument is an array or reference, only numbers in that array or reference are counted. Empty cells, logical values, text, or error values in the array or reference are ignored. * If you want to count logical values, text, or error values, use the COUNTA function. |
| **COUNTA** | ***Counts the number of cells that are not empty and the values within the list of arguments. Use COUNTA to count the number of cells that contain data in a range or array.***  **Syntax :** **COUNTA**(**value1**,value2,...)  **Value1, value2, ...**   are 1 to 255 arguments representing the values you want to count.  **Remarks**   * A value is any type of information, including error values and empty text **("")**. A value does not include empty cells. * If an argument is an array or reference, only values in that array or reference are used. Empty cells and text values in the array or reference are ignored. * If you do not need to count logical values, text, or error values, use the **COUNT** function. |
| **COUNTBLANK** | ***Counts empty cells in a specified range of cells.***  **Syntax :COUNTBLANK**(**range**)  **Range**   is the range from which you want to count the blank cells.  **Remark :** Cells with formulas that return "" (empty text) are also counted. Cells with zero values are not counted |
| **COUNTIF** | ***Counts the number of cells within a range that meet the given criteria.***  **Syntax :COUNTIF**(**range**,**criteria**)  **Range**  is one or more cells to count, including numbers or names, arrays, or references that contain numbers. Blank and text values are ignored.  **Criteria**  is the criteria in the form of a number, expression, cell reference, or text that defines which cells will be counted. For example, criteria can be expressed as 32, "32", ">32", "apples", or B4.  **Remark :** You can use the wildcard characters, question mark **(?)** and asterisk **(\*)**, in criteria. A question mark matches any single character; an asterisk matches any sequence of characters. If you want to find an actual question mark or asterisk, type a tilde **(~)** before the character.  **Example 1: Common COUNTIF formulas**  **Example 1: Common COUNTIF formulas**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | |  | | **1** | | **2** | | **3** | | **4** | | **5** | | |  |  | | --- | --- | | **A** | **B** | | **Data** | **Data** | | Apples | 32 | | Oranges | 54 | | Peaches | 75 | | Apples | 86 | | **Formula** | **Description (result)** | | =COUNTIF(A2:A5,"apples") | Number of cells with apples in the first column above (2) | | =COUNTIF(A2:A5,A4) | Number of cells with peaches in the first column above (1) | | =COUNTIF(A2:A5,A3)+COUNTIF(A2:A5,A2) | Number of cells with oranges and apples in the first column above (3) | | =COUNTIF(B2:B5,">55") | Number of cells with a value greater than 55 in the second column above (2) | | =COUNTIF(B2:B5,"<>"&B4) | Number of cells with a value not equal to 75 in the second column above (3) | | =COUNTIF(B2:B5,">=32")-COUNTIF(B2:B5,">85") | Number of cells with a value greater than or equal to 32 and less than or equal to 85 in the second column above (3) | |   **Example 2: COUNTIF formulas using wildcard characters and handling blank values**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | |  | | **1** | | **2** | | **3** | | **4** | | **5** | | **6** | | **7** | | |  |  | | --- | --- | | **A** | **B** | | **Data** | **Data** | | Apples | Yes | |  |  | | Oranges | NO | | Peaches | No | |  |  | | Apples | YeS | | **Formula** | **Description (result)** | | =COUNTIF(A2:A7,"\*es") | Number of cells ending with the letters "es" in the first column above (4) | | =COUNTIF(A2:A7,"?????es") | Number of cells ending with the letters "les" and having exactly 7 letters in the first column above (2) | | =COUNTIF(A2:A7,"\*") | Number of cells containing text in the first column above (4) | | =COUNTIF(A2:A7,"<>"&"\*") | Number of cells not containing text in the first column above (2) | | =COUNTIF(B2:B7,"No") / ROWS(B2:B7) | The average number of No votes including blank cells in the second column above formatted as a percentage with no decimal places (33%) | | =COUNTIF(B2:B7,"Yes") / (ROWS(B2:B7) -COUNTIF(B2:B7, "<>"&"\*")) | The average number of Yes votes excluding blank cells in the second column above formatted as a percentage with no decimal places (50%) | | |
| **COUNTIFS** | ***Home work*** |
| **MAX** | ***Returns the largest value in a set of values.***  **Syntax :** **MAX**(**number1** ; number2,...)  **Number1, number2, ...**   are 1 to 255 numbers for which you want to find the maximum value.  **Remarks**   * Arguments can either be numbers or names, arrays, or references that contain numbers. * Logical values and text representations of numbers that you type directly into the list of arguments are counted. * If an argument is an array or reference, only numbers in that array or reference are used. Empty cells, logical values, or text in the array or reference are ignored. * If the arguments contain no numbers, MAX returns 0 (zero). * Arguments that are error values or text that cannot be translated into numbers cause errors. * If you want to include logical values and text representations of numbers in a reference as part of the calculation, use the **MAXA** function. |
| **MAXA** | ***Returns the largest value in a list of arguments. MAXA is similar to MINA. For more information, see the examples for MINA.***  **Syntax**  : **MAXA**(**value1**; value2,...)  **Value1, value2, ...**   are 1 to 255 values for which you want to find the largest value.  **Remarks**   * Arguments can be the following: numbers; names, arrays, or references that contain numbers; text representations of numbers; or logical values, such as TRUE and FALSE, in a reference. * Logical values and text representations of numbers that you type directly into the list of arguments are counted. * If an argument is an array or reference, only values in that array or reference are used. Empty cells and text values in the array or reference are ignored. * Arguments that are error values or text that cannot be translated into numbers cause errors. * Arguments that contain TRUE evaluate as 1; arguments that contain text or FALSE evaluate as 0 (zero). * If the arguments contain no values, MAXA returns 0 (zero). * If you do not want to include logical values and text representations of numbers in a reference as part of the calculation, use the **MAX** function. |
| **MIN** | ***Returns the smallest number in a set of values.***  **Syntax :MIN**(**number1**; number2,...)  **Number1, number2, ...**   are 1 to 255 numbers for which you want to find the minimum value.  **Remarks**   * Arguments can either be numbers or names, arrays, or references that contain numbers. * Logical values and text representations of numbers that you type directly into the list of arguments are counted. * If an argument is an array or reference, only numbers in that array or reference are used. Empty cells, logical values, or text in the array or reference are ignored. * If the arguments contain no numbers, MIN returns 0. * Arguments that are error values or text that cannot be translated into numbers cause errors. * If you want to include logical values and text representations of numbers in a reference as part of the calculation, use **the MINA** function |
| **MINA** | ***Home work*** |
| **LARGE** | ***Returns the k-th largest value in a data set. You can use this function to select a value based on its relative standing. For example, you can use LARGE to return the highest, runner-up, or third-place score.***  ***Syntax :*** **LARGE**(**array** ; **k**)  **Array**   is the array or range of data for which you want to determine the k-th largest value.  **K**   is the position (from the largest) in the array or cell range of data to return.  **Remarks**   * If array is empty, LARGE returns the #NUM! error value. * If k ≤ 0 or if k is greater than the number of data points, LARGE returns the #NUM! error value.   If n is the number of data points in a range, then LARGE(array,1) returns the largest value, and LARGE(array,n) returns the smallest value. |
| **SMALL** | ***Returns the k-th smallest value in a data set. Use this function to return values with a particular relative standing in a data set.***  ***Syntax* : SMALL**(**array** ; **k**)  **Array**   is an array or range of numerical data for which you want to determine the k-th smallest value.  **K**   is the position (from the smallest) in the array or range of data to return.  ***Remarks***   * If array is empty, SMALL returns the #NUM! error value. * If k ≤ 0 or if k exceeds the number of data points, SMALL returns the #NUM! error value. * If n is the number of data points in array, SMALL(array,1) equals the smallest value, and SMALL(array,n) equals the largest value. |

***Example-1:*Use the Min function to determine the smallest value in a range**

|  |  |
| --- | --- |
| **To determine the lowest monthly sales:**  **1.** In a worksheet, copy the range A1:E10 shown in next figure  **2.** Select cells B12:E12 and type the following formula: **=MIN(B2:B10)**.  **3.** Press **<Ctrl+ Enter>**. |  |

***Home work: use Max function to determine the largest value in the range****.*

***Example-2 :* Use the MIN function to detect the smallest value in a column**

|  |  |
| --- | --- |
| **To determine the smallest value in a column:**  **1.** In column A, type any values down to cell A10.  **2.** Select cell B1 and type the following formula: **=MIN(A:A)**.  **3.** Press **<Enter>**. |  |

***Note1:*** To determine the smallest value in a row, such as the smallest value in the first row, use the formula **=MIN(1:1).** To get the smallest value of the first three rows, use the following function: =**MIN(1:3).**

***Note2:***in the same way , we can use **MAX** function to fined the maximum value.

***Example-3 :* Use the SMALL function to find the smallest values in a list**

|  |  |
| --- | --- |
| **To determine the three smallest values of a range:**  **1.** In cells A1:A10 enter any values from 100 to 999.  **2.** Select cell C1 and type the following formula **=SMALL($A$1:$A$10;1)** to get the smallest value.  **3.** In cell C2 type the formula **=SMALL($A$1:$A$10;2)** to get the second smallest value.  **4.** In cell C3 type the formula **=SMALL($A$1:$A$10;3)** to get the third smallest value. |  |

***Note:***in the same way , we can use **large** function to fined the largest value.

***Example - 4 :* Use the AVERAGE function to calculate the average output**

|  |
| --- |
| **To calculate the average of the three highest capacities of each production line:**  **1.** In cells B2:D10 type the output of each machine.  **2.** Select cells B13:D13 and type the following formula:  **=AVERAGE(LARGE(B$2:B$10 ;1);LARGE(B$2:B$10 ;2);LARGE(B$2:B$10;3))**.  **3.**Press **<Ctrl+ Enter>**. |
|  |

***Example – 5* Use the COUNT function to count cells containing numeric data**

|  |  |
| --- | --- |
| **To count the number of cells that contain numbers:**  **1.** In cells A1:A10 type data (numeric and text).  **2.** Select cell C1 and type the following formula: **=COUNT(A1:A10)**.  **3.** Press **<Enter>**. |  |

**Note:** Arguments that are date and time values are counted as numeric too

***Example -6* Use the COUNTA function to count cells containing data**

|  |  |
| --- | --- |
| **To count all cells containing data:**  **1.** In cells A1:A10 type any kind of data (numeric and text).  **2.** Select cell C1 and type the following formula: **=COUNTA(A1:A10)**.  **3.** Press **<Enter>**. |  |

***Note1:*** The **COUNTA** function does not count empty cells.

***Note2:*** To count all cells that contain text data, use a combination of functions in one formula. The number of cells with any kind of data is counted with the **COUNTA** function. All numeric cells are counted with the **COUNT** function. Just subtract the results of the **COUNT** function from the results of the **COUNTA** function, using the same range, to get all cells containing text. By using the flowing formula: =**COUNTA(A1:A10)-COUNT(A1:A10)**.

***Example -7* Use the COUNTBLANK function to count empty cells**

|  |  |
| --- | --- |
| **To count all empty cells in a specified range:**   1. In cells A1:A10 type data (numeric and text). Be sure to leave a few cells empty. 2. Select cell C1 and type the following formula: **=COUNTBLANK(A1:A10).** 3. *Press <Enter>.* |  |

***Example -8* Use the COUNTIF function to count selected cells**

|  |  |
| --- | --- |
| **To count all selected cells in a specified range:**   1. To find number of **Cs** student Select cell **C8** then type the following formula:   **=COUNTIF(B1:B6;"cs")**   1. To find number of **is**  student Select cell **C9** then type the following formula:   **=COUNTIF(B1:B6;"is")**   1. To find number of number of **failed**  student Select cell **C10** then type the following formula: **=COUNTIF(C1:C6;"<50")** 2. To find number of number of **failed** student Select cell **C11** then type the following formula: **=COUNTIF(C1:C6;">=50")** |  |

* **Math and trig functions**

This category contains a wide variety of functions that perform mathematical and trigonometric calculations.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SUM** | ***Adds all the numbers in a range of cells.***  **Syntax** : **SUM** (**number1**,number2, ...)  **Number1, number2, ...**   are 1 to 255 arguments for which you want the total value or sum.  **Remarks**   * Numbers, logical values, and text representations of numbers that you type directly into the list of arguments are counted. See the first and second examples following. * If an argument is an array or reference, only numbers in that array or reference are counted. Empty cells, logical values, or text in the array or reference are ignored. See the third example following. * Arguments that are error values or text that cannot be translated into numbers cause errors.   **Example :**  **sum1.png**     |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Formula** | **Description (Result)** | | **=SUM(3, 2)** | **Adds 3 and 2 (5)** | | **=SUM("5", 15, TRUE)** | **Adds 5, 15 and 1, because the text values are translated into numbers, and the logical value TRUE is translated into the number 1 (21)** | | **=SUM(A2:A4)** | **Adds the first three numbers in the column above (40)** | | **=SUM(A2:A4, 15)** | **Adds the first three numbers in the column above, and 15 (55)** | | **=SUM(A5,A6, 2)** | **Adds the values in the last two rows above, and 2. Because nonnumeric values in references are not translated, the values in the column above are ignored (2)** | | |
| **SUMIF** | ***Adds the cells specified by a given criteria.***  **Syntax**  **SUMIF**(range,criteria,sum\_range)  **Range**  is the range of cells that you want evaluated by criteria. Cells in each range must be numbers or names, arrays, or references that contain numbers. Blank and text values are ignored.  **Criteria**  is the criteria in the form of a number, expression, or text that defines which cells will be added. For example, criteria can be expressed as 32, "32", ">32", or "apples".  **Sum\_range**   are the actual cells to add if their corresponding cells in range match criteria. If sum\_range is omitted, the cells in range are both evaluated by criteria and added if they match criteria.  **Remarks**   * Sum\_range does not have to be the same size and shape as range. The actual cells that are added are determined by using the top, left cell in sum\_range as the beginning cell, and then including cells that correspond in size and shape to range. * You can use the wildcard characters, question mark (?) and asterisk (\*), in criteria. A question mark matches any single character; an asterisk matches any sequence of characters. If you want to find an actual question mark or asterisk, type a tilde (~) preceding the character.   **Example**  sumif.png   |  |  | | --- | --- | | **Formula** | **Description (Result)** | | **=SUMIF(A2:A5,">160000",B2:B5)** | **Sum of the commissions for property values over 160,000 (63,000)** | | **=SUMIF(A2:A5,">160000")** | **Sum of the property values over 160,000 (900,000)** | | **=SUMIF(A2:A5,"=300000",B2:B3)** | **Sum of the commissions for property values equal to 300,000 (21,000)** | |
| **SUMIFS** | ***Home work*** |
| **PRODUCT** | ***Multiplies all the numbers given as arguments and returns the product.***  **Syntax PRODUCT(number1,number2,...)**  **Number1, number2**, ...   are 1 to 255 numbers that you want to multiply.  **Remarks**   * Arguments that are numbers, logical values, or text representations of numbers are counted; arguments that are error values or text that cannot be translated into numbers cause errors. * If an argument is an array or reference, only numbers in the array or reference are counted. Empty cells, logical values, text, or error values in the array or reference are ignored.   pro.png   |  |  | | --- | --- | | **Formula** | **Description (Result)** | | **=PRODUCT(A2:A4)** | **Multiplies t he numbers above (2250)** | | **=PRODUCT(A2:A4, 2)** | **Multiplies the numbers above and 2 (4500)** | |
| **SUMPRODUCT** | ***Home work*** |
| **RANDBETWEEN** | ***Returns a random integer number between the numbers you specify. A new random integer number is returned every time the worksheet is calculated.***  **Syntax** **RANDBETWEEN**(**bottom**,**top**)  **Bottom**   is the smallest integer RANDBETWEEN will return.  **Top**   is the largest integer RANDBETWEEN will return.  **Remark :**  **Press <F9> to recalculate and generate new randomized numbers for the range**  **Example :**   |  |  | | --- | --- | | **A** | **B** | | **Formula** | **Description (Result)** | | **=RANDBETWEEN(1,100)** | **Random number between 1 and 100 (varies)** | | **=RANDBETWEEN(-1,1)** | **Random number between -1 and 1 (varies)** | |
| **ROUND** | ***Rounds a number to a specified number of digits.***  **Syntax** **ROUND**(**number**,**num\_digits**)  **Number**   is the number you want to round.  **Num\_digits**   specifies the number of digits to which you want to round number.  **Remarks**   * If num\_digits is greater than 0 (zero), then number is rounded to the specified number of decimal places. * If num\_digits is 0, then number is rounded to the nearest integer. * If num\_digits is less than 0, then number is rounded to the left of the decimal point.   **Example**   |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **A** | **B** | | **Formula** | **Description (Result)** | | **=ROUND(2.15, 1)** | **Rounds 2.15 to one decimal place (2.2)** | | **=ROUND(2.149, 1)** | **Rounds 2.149 to one decimal place (2.1)** | | **=ROUND(-1.475, 2)** | **Rounds -1.475 to two decimal places (-1.48)** | | **=ROUND(21.5, -1)** | **Rounds 21.5 to one decimal place to the left of the decimal point (20)** | | |
| **ROUNDDOWN** | ***Rounds a number down, toward zero.***  **Syntax** **ROUNDDOWN**(**number**,**num\_digits**)  **Number**   is any real number that you want rounded down.  **Num\_digits**   is the number of digits to which you want to round number.  **Remarks**   * ROUNDDOWN behaves like ROUND, except that it always rounds a number down. * If num\_digits is greater than 0 (zero), then number is rounded down to the specified number of decimal places. * If num\_digits is 0, then number is rounded down to the nearest integer. * If num\_digits is less than 0, then number is rounded down to the left of the decimal point  |  |  | | --- | --- | | **A** | **B** | | **Formula** | **Description (Result)** | | **=ROUNDDOWN(3.2, 0)** | **Rounds 3.2 down to zero decimal places (3)** | | **=ROUNDDOWN(76.9,0)** | **Rounds 76.9 down to zero decimal places (76)** | | **=ROUNDDOWN(3.14159, 3)** | **Rounds 3.14159 down to three decimal places (3.141)** | | **=ROUNDDOWN(-3.14159, 1)** | **Rounds -3.14159 down to one decimal place (-3.1)** | | **=ROUNDDOWN(31415.92654, -2)** | **Rounds 31415.92654 down to 2 decimal places to the left of the decimal (31400)** | |
| **ROUNDUP** | ***Home work*** |
| **MOD** | ***Returns the remainder after number is divided by divisor. The result has the same sign as divisor.***  **Syntax** **MOD**(**number**,**divisor**)  **Number**   is the number for which you want to find the remainder.  **Divisor**   is the number by which you want to divide number.  ***Remarks***   * If divisor is 0, MOD returns the #DIV/0! error value. * The MOD function can be expressed in terms of the INT function:   MOD(n, d) = n - d\*INT(n/d)  ***Example:***   |  |  | | --- | --- | | **A** | **B** | | **Formula** | **Description (Result)** | | **=MOD(3, 2)** | **Remainder of 3/2 (1)** | | **=MOD(-3, 2)** | **Remainder of -3/2. The sign is the same as divisor (1)** | | **=MOD(3, -2)** | **Remainder of 3/-2. The sign is the same as divisor (-1)** | | **=MOD(-3, -2)** | **Remainder of -3/-2. The sign is the same as divisor (-1)** | |
| **INT** | ***Rounds a number down to the nearest integer.***  **Syntax** **INT**(**number**)  **Number**   is the real number you want to round down to an integer.  **Example**   |  | | --- | | **A** | | **Data** | | **19.5** | | **Formula** | **Description (Result)** | | **=INT(8.9)** | **Rounds 8.9 down (8)** | | **=INT(-8.9)** | **Rounds -8.9 down (-9)** | | **=A2-INT(A2)** | **Returns the decimal part of a positive real number in cell A2 (0.5** | |
| **FACT** | ***Home work*** |
| **FLOOR** | ***Home work*** |
| **CEILING** | ***Home work*** |
| **EVEN** | ***Returns number rounded up to the nearest even integer. You can use this function for processing items that come in twos. For example, a packing crate accepts rows of one or two items. The crate is full when the number of items, rounded up to the nearest two, matches the crate's capacity.***  **Syntax** **EVEN**(**number**)  **Number**   is the value to round.  **Remarks**   * If number is nonnumeric, EVEN returns the #VALUE! error value. * Regardless of the sign of number, a value is rounded up when adjusted away from zero. If number is an even integer, no rounding occurs.   **Example**   |  |  | | --- | --- | | **A** | **B** | | **Formula** | **Description (Result)** | | **=EVEN(1.5)** | **Rounds 1.5 up to the nearest even integer (2)** | | **=EVEN(3)** | **Rounds 3 up to the nearest even integer (4)** | | **=EVEN(2)** | **Rounds 2 up to the nearest even integer (2)** | | **=EVEN(-1)** | **Rounds -1 up to the nearest even integer (-2)** | |
| **ISEVEN** | ***Home work*** |
| **ODD** | ***Returns number rounded up to the nearest odd integer.***  **Syntax** **ODD**(**number**)  **Number**   is the value to round.  **Remarks**   * If number is nonnumeric, ODD returns the #VALUE! error value. * Regardless of the sign of number, a value is rounded up when adjusted away from zero. If number is an odd integer, no rounding occurs.   **Example**   |  |  | | --- | --- | | **A** | **B** | | **Formula** | **Description (Result)** | | **=ODD(1.5)** | **Rounds 1.5 up to the nearest odd integer (3)** | | **=ODD(3)** | **Rounds 3 up to the nearest odd integer (3)** | | **=ODD(2)** | **Rounds 2 up to the nearest odd integer (3)** | | **=ODD(-1)** | **Rounds -1 up to the nearest odd integer (-1)** | | **=ODD(-2)** | **Rounds -2 up to the nearest odd integer (-3)** | |
| **ISODD** | ***Home work*** |

***Example – 1*  Use the SUM function to sum several ranges**

|  |  |
| --- | --- |
| **To sum several ranges:**  **1.** In cells A2:A10 enter prices from $1 to $100.  **2.** Select cells B2:B10 and type the formula **=A2\*8%** to calculate the tax amount.  **3.** Press **<Ctrl+Enter>**.  **4.** In cells D2:D10 type some discount values from –1 to –3.  **5.** In cell B12, sum all three columns with the following function:  **=SUM(A2:A10 ; B2:B10 ; D2:D10)**.  **6.** Press **<Enter>**. |  |

***Example – 2 :*** Use the SUMIF function to determine sales of a team

|  |  |
| --- | --- |
| **To sum specified data:**  **1.** In cells A2:A10 enter a team number from 1 to 3.  **2.** List all team members in cells B2:B10.  **3.** In cells C2:C10 enter the daily sales of each employee.  **4.** List the numbers 1, 2, 3 for each team in cells E2:E4.  **5.** Select cells F2:F4 and type the following formula:  **=SUMIF($A$2:$A$10,E2,$C$2:$C$10)**.  **6.** Press **<Ctrl+ Enter>**. |  |

***Example – 3 :* Use the PRODUCT function to multiply conditional values**

|  |  |
| --- | --- |
| **To multiply conditional values:**  **1.** In cells A2:A10 enter some numbers for value 1.  **2.** In cells B2:B10 enter some numbers for value 2.  **3.** Select cells C2:C10 and type the following formula:  **=IF(OR(A2="",B2=""),"",PRODUCT(A2,B2))**.  **4.** Press **<Ctrl +Enter>**. |  |

***Example – 3 :* Use the ROUND function to round numbers**

|  |  |
| --- | --- |
| **To round numbers:**  **1.** In cells A2:A10 enter numbers with a decimal point.  **2.** In cells B2:B10 enter the number of decimal places the number should be rounded to.  **3.** Select cells C2:C10 and type the following formula: **=ROUND($A2,$B2)**.  **4.** Press **<Ctrl +Enter>**. |  |

***Example – 4:* Use the ROW function to mark every other row**

|  |  |
| --- | --- |
| **To mark every other row:**  **1.** Select cells A1:A10 and type the following formula: **=IF(MOD(ROW(),2),"XXX"," ")**.  **2.** Press **<Ctrl + Enter>**. |  |

**Note:** If every other column has to be marked, use the following formula:

**=IF(MOD(COLUMN(),2),"XXX"," ").**

***Example – 5 :* Use the EVEN and ODD functions to determine the nearest even/odd value**

|  |  |
| --- | --- |
| **To determine the nearest even/odd value:**  **1.** In cells A2:A10 list some valid numbers with decimal points.  **2.** Select cells B2:B10 and enter the following function: **=EVEN(A2)**.  **3.** Press **<Ctrl+Enter>**.  **4.** Select cells C2:C10 and enter the following function: **=ODD(A2)**.  **5.** Press **<Ctrl+Enter>**. |  |

***Example – 6* Use the ISEVEN and ISODD functions to check if a number is even or odd**

|  |  |
| --- | --- |
| **To check if a number is even or odd:**  **1.** In cells A2:A10 enter some numbers.  **2.** Select cells B2:B10 and type the following formula: **=IF(ISEVEN(A2),"X","")**.  **3.** Press **<Ctrl +Enter>**.  **4.** Select cell C2:C10 and type the following formula: **=IF(ISODD(A2),"X","")**.  **5.** Press **<Ctrl +Enter>**. |  |

***Example – 7 :* Use the ISODD and ROW functions to determine odd rows**

|  |  |
| --- | --- |
| **To determine odd rows and mark them:**   1. Select cells A1:E11 and type the following formula: 2. **=IF(ISODD(ROW()),"X","")**.   **2.** Press **<Ctrl+ Enter>**. |  |

**Note: To mark all even rows, use the following formula: =IF(ISEVEN(ROW()),"X","").**

***Example – 8 :*Use the ISODD and COLUMN functions to determine odd columns**

|  |  |
| --- | --- |
| **To determine odd columns:**  **1.** Select cells A1:E11 and type the following formula: **=IF(ISODD(COLUMN()),"X ","").**  **2.** Press **<Ctrl+Enter>**. |  |

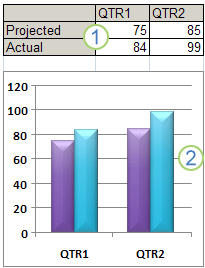
**Note: To mark even columns, type the following formula =IF(ISEVEN(COLUMN()),"X","").**

* **Basic charting**
* **Overview of charting**

In Microsoft Office Excel, it is easy to create professional looking charts. Simply by choosing a **chart type**, **a chart layout**, and a **chart style**—all of which are within easy reach on the new Office Excel Ribbon—you will have instant professional results every time that you create a chart. You can make it even easier by saving your favorite charts as a chart template that you can then quickly apply whenever you create a new chart.

* **Creating charts in Excel**

To create a basic chart in Excel that you can modify and format later, you start by entering the data for the chart on a worksheet. Then you simply select that data and choose the chart type that you want to use on the Ribbon (**Insert** tab, **Charts** group).



Callout 1**Worksheet data**

**Callout 2Chart created from worksheet data**

* **Available chart types**

Microsoft Office Excel supports numerous types of charts to help you display data in ways that are meaningful to your audience. When you want to create a chart or change an existing chart, you can choose from a wide range of chart subtypes available for each of the following chart types.

|  |  |  |
| --- | --- | --- |
| **Chart types** | **Sh**  **ort description** | **Chart subtypes and shapes** |
| **Column ch arts** | **Data that is arranged in columns or rows on a worksheet can be plotted in a column chart. Column charts are useful for showing data changes over a period of time or for illustrating comparisons among items. In column charts, categories are typically organized along the horizontal axis and values along the vertical axis.** | 1. Clustered column and clustered column in 3-D. 2. Stacked column and stacked column in 3-D . 3. 100% stacked column and 100% stacked column in 3-D . 4. 3-D column  . 5. Cylinder, cone, and pyramid  .   col.png |
| **Line charts** | Line charts can display continuous data over time, set against a common scale, and are therefore ideal for showing trends in data at equal intervals. In a line chart, category data is distributed evenly along the horizontal axis, and all value data is distributed evenly along the vertical axis. | 1. Line and line with markers . 2. Stacked line and stacked line with markers. 3. 100% stacked line and 100% stacked line with markers 4. line.png3-D line |
| **Pie charts** | Data that is arranged in one column or row only on a worksheet can be plotted in a pie chart. Pie charts show the size of items in one [data series (data series: Related data points that are plotted in a chart. Each data series in a chart has a unique color or pattern and is represented in the chart legend. You can plot one or more data series in a chart. Pie charts have only one data series.)](javascript:AppendPopup(this,'xldefDataSeries_2')), proportional to the sum of the items. The [data points (data points: Individual values plotted in a chart and represented by bars, columns, lines, pie or doughnut slices, dots, and various other shapes called data markers. Data markers of the same color constitute a data series.)](javascript:AppendPopup(this,'grdefDataPoints_3')) in a pie chart are displayed as a percentage of the whole pie.  ***Consider using a pie chart when:***   * **You only have one data series that you want to plot.** * **None of the values that you want to plot are negative.** * **Almost none of the values that you want to plot are zero values.** * **You don't have more than seven categories.** * **The categories represent parts of the whole pie.** | 1. Pie and pie in 3-D . 2. Pie of pie and bar of pie . 3. pie.pngExploded pie and exploded pie in 3-D . |
| **Bar charts** | Data that is arranged in columns or rows on a worksheet can be plotted in a bar chart. Bar charts illustrate comparisons among individual items.  ***Consider using a bar chart when:***   * **The axis labels are long.** * **The values that are shown are durations** | 1. Clustered bar and clustered bar in 3-D. 2. Stacked bar and stacked bar in 3-D . 3. 100% stacked bar and 100% stacked bar in 3-D . 4. Horizontal cylinder, cone, and pyramid  .   bar.png |
| **Area charts** | Data that is arranged in columns or rows on a worksheet can be plotted in an area chart. Area charts emphasize the magnitude of change over time, and can be used to draw attention to the total value across a trend. For example, data that represents profit over time can be plotted in an area chart to emphasize the total profit.  By displaying the sum of the plotted values, an area chart also shows the relationship of parts to a whole. | **1-** Area and area in 3-D .  2-Stacked area and stacked area in 3-D.  3-100% stacked area and 100% stacked area in 3-D  4-3-D area  stack.png |

**Note : There are many other chart types :**

* + 1. **XY (scatter) charts .**
    2. **Stock charts**
    3. **Surface charts**
    4. **Doughnut charts**
    5. **Bubble charts**
    6. **Radar charts**
* **Lookup and reference Functions**

Functions in this category are used to find (look up) values in lists or tables. A common example is a tax table. You can use the VLOOKUP function to determine a tax rate for a particular income level.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ROW** | ***Returns the row number of a reference.***  **Syntax** **ROW**([reference])  **Reference**   is the cell or range of cells for which you want the row number.  **Remarks:**   * If reference is omitted, it is assumed to be the reference of the cell in which the ROW function appears. * If reference is a range of cells, and if ROW is entered as a vertical [array (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)](javascript:AppendPopup(this,'xldefArray_1')), ROW returns the row numbers of reference as a vertical array. * Reference cannot refer to multiple areas.   **Example1:**   |  |  | | --- | --- | | **Formula** | **Description (Result)** | | **=ROW()** | **Row in which the formula appears (2)** | | **=ROW(C10)** | **Row of the reference (10)** |   **Example2:**   |  |  | | --- | --- | | **A** | **B** | | **Formula** | **Description (Result)** | | **=ROW(C4:D6)** | **First row in the reference (4)** | | **Second row in the reference (5)** | | **Third row in the reference (6)** | |
| **ROWS** | ***Home work*** |
| **COLUMN** | ***Returns the column number of the given reference.***  **Syntax** **COLUMN**(reference)  **Reference**   is the cell or range of cells for which you want the column number.   * If reference is omitted, it is assumed to be the reference of the cell in which the COLUMN function appears. * If reference is a range of cells, and if COLUMN is entered as a horizontal [array (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)](javascript:AppendPopup(this,'xldefArray_1')), COLUMN returns the column numbers of reference as a horizontal array. * Reference cannot refer to multiple areas.   **Example:**   |  |  | | --- | --- | | **A** | **B** | | **Formula** | **Description (Result)** | | **=COLUMN()** | **Column in which the formula appears (1)** | | **=COLUMN(A10)** | **Column of the reference (1)** | |
| **COLUMNS** | ***Home work*** |
| **ADDRESS** | ***Creates a cell address as text, given specified row and column numbers.***  **Syntax** **ADDRESS**(**row\_num**,**column\_num,[abs\_num],[a1],[sheet\_text])**  **Row\_num**   is the row number to use in the cell reference.  **Column\_num**   is the column number to use in the cell reference.  **Abs\_num**   specifies the type of reference to return.   |  |  | | --- | --- | | **Abs\_num** | **Returns this type of reference** | | **1 or omitted** | **Absolute** | | **2** | **Absolute row; relative column** | | **3** | **Relative row; absolute column** | | **4** | **Relative** |   **A1**   is a logical value that specifies the A1 or R1C1 reference style. If a1 is TRUE or omitted, ADDRESS returns an A1-style reference; if FALSE, ADDRESS returns an R1C1-style reference.  **Sheet\_text**   is text specifying the name of the worksheet to be used as the external reference. If sheet\_text is omitted, no sheet name is used.  **Example**   |  |  | | --- | --- | | **A** | **B** | | **Formula** | **Description (Result)** | | **=ADDRESS(2,3)** | **Absolute reference ($C$2)** | | **=ADDRESS(2,3,2)** | **Absolute row; relative column (C$2)** | | **=ADDRESS(2,3,2,FALSE)** | **Absolute row; relative column in R1C1 reference style (R2C[3])** | | **=ADDRESS(2,3,1,FALSE,"[Book1]Sheet1")** | **Absolute reference to another workbook and worksheet ([Book1]Sheet1!R2C3)** | | **=ADDRESS(2,3,1,FALSE,"EXCEL SHEET")** | **Absolute reference to another worksheet ('EXCEL SHEET'!R2C3** | |
| **MATCH** | ***Returns the relative position of an item in an*** [***array (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)***](javascript:AppendPopup(this,'xldefArray_1')) ***that matches a specified value in a specified order. Use MATCH instead of one of the LOOKUP functions when you need the position of an item in a range instead of the item itself.***  **Syntax** **MATCH**(**lookup\_value**,**lookup\_array**,match\_type)  **Lookup\_value**   is the value you use to find the value you want in a table.   * Lookup\_value is the value you want to match in lookup\_array. For example, when you look up someone's number in a telephone book, you are using the person's name as the lookup value, but the telephone number is the value you want. * Lookup\_value can be a value (number, text, or logical value) or a cell reference to a number, text, or logical value.   **Lookup\_array**   is a contiguous range of cells containing possible lookup values. Lookup\_array must  be an array or an array reference.  **Match\_type**   is the number -1, 0, or 1. Match\_type specifies how Microsoft Excel matches lookup\_value with values in lookup\_array.   * **If match\_type is 1**, MATCH finds the ls smalliestn value that is less than or equal to lookup\_value. Lookup\_array must be placed in ascending order: ...-2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE. * **If match\_type is 0**, MATCH finds the first value that is exactly equal to lookup\_value. Lookup\_array can be in any order. * If **match\_type is -1**, MATCH finds the largest that is greater than or equal to lookup\_value. Lookup\_array must be placed in descending order: TRUE, FALSE, Z-A, ...2, 1, 0, -1, -2, ..., and so on. * If match\_type is omitted, it is assumed to be 1.   **Remarks**   * MATCH returns the position of the matched value within lookup\_array, not the value itself. For example, MATCH("b",{"a","b","c"},0) returns 2, the relative position of "b" within the array {"a","b","c"}. * MATCH does not distinguish between uppercase and lowercase letters when matching text values. * If MATCH is unsuccessful in finding a match, it returns the #N/A error value. * If match\_type is 0 and lookup\_value is text, you can use the wildcard characters, question mark (?) and asterisk (\*), in lookup\_value. A question mark matches any single character; an asterisk matches any sequence of characters. If you want to find an actual question mark or asterisk, type a tilde (~) before the character.   **Example**   |  |  | | --- | --- | | **A** | **B** | | **Product** | **Count** | | **Bananas** | **25** | | **Oranges** | **38** | | **Apples** | **40** | | **Pears** | **41** | | **Formula** | **Description (Result)** | | **=MATCH(39,B2:B5,1)** | **Because there is not an exact match, the position of the next lowest value (38) in the range B2:B5 is returned. (2)** | | **=MATCH(41,B2:B5,0)** | **The position of 41 in the range B2:B5. (4)** | | **=MATCH(40,B2:B5,-1)** | **Returns an error because the range B2:B5 is not in descending order. (#N/A)** | |
| **INDEX** | ***Returns a value or the reference to a value from within a table or range. There are two forms of the INDEX function: the*** [***array (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)***](javascript:AppendPopup(this,'xldefArray_1')) ***form and the* reference form.**   * 1. **Array form : Returns the value of an element in a table or an** [**array (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)**](javascript:AppendPopup(this,'xldefArray_2'))**, selected by the row and column number indexes. Use the array form if the first argument to INDEX is an array constant.**   **Syntax** : **INDEX ( array ; row\_num ; [column\_num])**  **Array  :**   is a range of cells or an array constant .  **Row\_num :**  selects the row in array from which to return a value. If row\_num is omitted, column\_num is required.  **Column\_num:** selects the column in array from which to return a value. If column\_num is omitted, row\_num is required.  Example 1   |  |  | | --- | --- | | A | B | | Data | Data | | Apples | Lemons | | Bananas | Pears | | Formula | Description (Result) | | =INDEX(A2:B3,2,2) | Value at the intersection of the second row and second column in the range (Pears) | | =INDEX(A2:B3,2,1) | Value at the intersection of the second row and first column in the range (Bananas |  * 1. **Reference form**   ***Returns the reference of the cell at the intersection of a particular row and column. If the reference is made up of nonadjacent selections, you can pick the selection to look in.***  **Syntax : INDEX**(**reference**,row\_num,column\_num,area\_num)  **Reference**   is a reference to one or more cell ranges   * If you are entering a nonadjacent range for the reference, enclose reference in parentheses. * If each area in reference contains only one row or column, the row\_num or column\_num argument, respectively, is optional. For example, for a single row reference, use INDEX(reference,,column\_num).   **Row\_num**   is the number of the row in reference from which to return a reference.  **Column\_num**   is the number of the column in reference from which to return a reference.  **Area\_num**   selects a range in reference from which to return the intersection of row\_num and Acolumn\_num. The first area selected or entered is numbered 1, the second is 2, and so on. If area\_num is omitted, INDEX uses area 1.   |  |  |  |  | | --- | --- | --- | --- | |  | **A** | **B** | **C** | | **1** | **Fruit** | **Price** | **Count** | | 2 | Apples | 0.69 | 40 | | 3 | Bananas | 0.34 | 38 | | 4 | Lemons | 0.55 | 15 | | 5 | Oranges | 0.25 | 25 | | 6 | Pears | 0.59 | 40 | | 7 | Almonds | 2.80 | 10 | | 8 | Cashews | 3.55 | 16 | | 9 | Peanuts | 1.25 | 20 | | 10 | Walnuts | 1.75 | 12 |  * For example, if reference describes the cells (A1:B4,D1:E4,G1:H4), then area\_num 1 is the range A1:B4, area\_num 2 is the range D1:E4, and area\_num 3 is the range G1:H4.  |  |  | | --- | --- | | **Formula** | **Description (Result)** | | =INDEX(A2:C6,2,3) | The intersection of the second row and third column in the range A2:C6, which is the content of cell C3. (38) | | =INDEX((A1:C6,A8:C11),2,2,2) | The intersection of the second row and second column in the second area of A8:C11, which is the content of cell B9. (3.55) | | =SUM(INDEX(A1:C11,0,3,1)) | The sum of the third column in the first area of the range A1:C11, which is the sum of C1:C6. (216) | | =SUM(B2:INDEX(A2:C6,5,2)) | The sum of the range starting at B2, and ending at the intersection of the fifth row and the second column of the range A2:A6, which is the sum of B2:B6. (2.42) | |
| **VLOOKUP** | ***Searches for a value in the first column of a table array and returns a value in the same row from another column in the table array.***  The V in VLOOKUP stands for vertical. Use VLOOKUP instead of HLOOKUP when your comparison values are located in a column to the left of the data that you want to find.  **Syntax : VLOOKUP**(**lookup\_value**,**table\_array**,**col\_index\_num,range\_lookup**)  **Lookup\_value**  The value to search in the first column of the table [array (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)](javascript:AppendPopup(this,'xldefArray_1')). Lookup\_value can be a value or a reference. If lookup\_value is smaller than the smallest value in the first column of table\_array, VLOOKUP returns the #N/A error value.  **Table\_array**  Two or more columns of data. Use a reference to a range or a range name. The values in the first column of table\_array are the values searched by lookup\_value. These values can be text, numbers, or logical values. Uppercase and lowercase text are equivalent.  **Col\_index\_num**  The column number in table\_array from which the matching value must be returned. A col\_index\_num of 1 returns the value in the first column in table\_array; a col\_index\_num of 2 returns the value in the second column in table\_array, and so on. If col\_index\_num is:   * Less than 1, VLOOKUP returns the #VALUE! error value. * Greater than the number of columns in table\_array, VLOOKUP returns the #REF! error value.   **Range\_lookup**  A logical value that specifies whether you want VLOOKUP to find an exact match or an approximate match:   * If TRUE or omitted, an exact or approximate match is returned. If an exact match is not found, the next largest value that is less than lookup\_value is returned.   The values in the first column of table\_array must be placed in ascending sort order; otherwise, VLOOKUP may not give the correct value. For more information, see [Sort data](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP10073947.htm).   * If FALSE, VLOOKUP will only find an exact match. In this case, the values in the first column of table\_array do not need to be sorted. If there are two or more values in the first column of table\_array that match the lookup\_value, the first value found is used. If an exact match is not found, the error value #N/A is returned.   **Remarks**   * When searching text values in the first column of table\_array, ensure that the data in the first column of table\_array does not have leading spaces, trailing spaces, inconsistent use of straight ( ' or " ) and curly ( ‘ or “) quotation marks, or nonprinting characters. In these cases, VLOOKUP may give an incorrect or unexpected value. For more information, see [CLEAN](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP10062560.htm) and [TRIM](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP10062581.htm). * When searching number or date values, ensure that the data in the first column of table\_array is not stored as text values. In this case, VLOOKUP may give an incorrect or unexpected value. For more information, see [Convert numbers stored as text to numbers](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP03055900.htm) |
| **HLOOKUP** | Searches for a value in the top row of a table or an [array (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)](javascript:AppendPopup(this,'xldefArray_1')) of values, and then returns a value in the same column from a row you specify in the table or array. Use HLOOKUP when your comparison values are located in a row across the top of a table of data, and you want to look down a specified number of rows. Use VLOOKUP when your comparison values are located in a column to the left of the data you want to find.  The H in HLOOKUP stands for "Horizontal."  **Syntax** : **HLOOKUP**(**lookup\_value**,**table\_array**,**row\_index\_num**,range\_lookup)  **Lookup\_value**   is the value to be found in the first row of the table. Lookup\_value can be a value, a reference, or a text string.  **Table\_array**   is a table of information in which data is looked up. Use a reference to a range or a range name.   * The values in the first row of table\_array can be text, numbers, or logical values. * If range\_lookup is TRUE, the values in the first row of table\_array must be placed in ascending order: ...-2, -1, 0, 1, 2,... , A-Z, FALSE, TRUE; otherwise, HLOOKUP may not give the correct value. If range\_lookup is FALSE, table\_array does not need to be sorted. * Uppercase and lowercase text are equivalent. * Sort the values in ascending order, left to right. For more information, see [Sort data](ms-help://MS.EXCEL.12.1033/EXCEL/content/HP10073947.htm).   **Row\_index\_num**   is the row number in table\_array from which the matching value will be returned. A row\_index\_num of 1 returns the first row value in table\_array, a row\_index\_num of 2 returns the second row value in table\_array, and so on. If row\_index\_num is less than 1, HLOOKUP returns the #VALUE! error value; if row\_index\_num is greater than the number of rows on table\_array, HLOOKUP returns the #REF! error value.  **Range\_lookup**   is a logical value that specifies whether you want HLOOKUP to find an exact match or an approximate match. If TRUE or omitted, an approximate match is returned. In other words, if an exact match is not found, the next largest value that is less than lookup\_value is returned. If FALSE, HLOOKUP will find an exact match. If one is not found, the error value #N/A is returned.  **Remarks**   * If HLOOKUP can't find lookup\_value, and range\_lookup is TRUE, it uses the largest value that is less than lookup\_value. * If lookup\_value is smaller than the smallest value in the first row of table\_array, HLOOKUP returns the #N/A error value. * If range\_lookup is FALSE and lookup\_value is text, you can use the wildcard characters, question mark (?) and asterisk (\*), in lookup\_value. A question mark matches any single character; an asterisk matches any sequence of characters. If you want to find an actual question mark or asterisk, type a tilde (~) before the character. |

***Example –1:* Use the ADDRESS, MATCH, and MAX functions to find the largest number**

|  |  |
| --- | --- |
| **To search for the cell reference of the greatest number:**  **1.** In cells A2:A10 enter some numbers.  **2.** Select cell C2 and type the following formula:  **=ADDRESS(MATCH(MAX(A1:A10) ;A1:A10) ;1; 4)**.  Press **<Enter>**. |  |

***Example –2:* *Use the INDEX, MATCH, and LARGE functions to determine and locate the best salesperson***

|  |  |
| --- | --- |
| **To determine and locate the best employee:**  1. In cells B2:B10 type the daily sales of the employees.  2. Select cell D3 and type the following formula:  **=INDEX($A$2:$A$10;MATCH(LARGE($B$2:$B$10;1),$B$2:$B$10;0);1)**.  3. Press **<Enter>**.  4. Select cell D6 and type the following formula: **=LARGE($B$2:$B$10;1)-LARGE($B$2:$B$10;2)**.  5. Press **<Enter>**. |  |

***Example –3:* Use the VLOOKUP function to look up and extract data from a database**

|  |  |
| --- | --- |
| **To look up and extract data from a list:**  **1.** In cell B2 type the following formula: **=VLOOKUP($B$1 ;$A$7:$D$16 ;2; FALSE)**.  **2.** In cell B3 type the following formula: **=VLOOKUP($B$1;$A$7:$D$16;3;FALSE)**.  **3.** In cell B4 type the following formula: **=VLOOKUP($B$1;$A$7:$D$16;4;FALSE)**.  **4.** Press **<Enter>**. |  |

***Example –4: Use the HLOOKUP function to determine sales and costs of a team***

|  |  |
| --- | --- |
| **To determine sales and cost for a team:**  1. In a worksheet, copy the information in cells A1:E3  2. In cell A7 enter a valid team name.  3. In cell B7 type the following formula:  **=HLOOKUP($A$7,$B$1:$E$3,2,FALSE)**.  4. Press **<Enter>**.  5. Select cell C7 and type the following formula:  **=HLOOKUP($A$7,$B$1:$E$3,3,FALSE)**.  6. Press **<Enter>**. |  |

* **Text functions**

|  |  |
| --- | --- |
| **LEFT** | ***LEFT returns the first character or characters in a text string, based on the number of characters you specify***  **Syntax** **LEFT**(**text** ; num\_chars)  **Text**   is the text string that contains the characters you want to extract.  **Num\_chars**   specifies the number of characters you want LEFT to extract.   * Num\_chars must be greater than or equal to zero. * If num\_chars is greater than the length of text, LEFT returns all of text. * If num\_chars is omitted, it is assumed to be |
| **Right** | **homework** |
| **LEN** | **homework** |
| **SEARCH** | ***SEARCH and SEARCHB locate one text string within a second text string, and return the number of the starting position of the first text string from the first character of the second text string.***  **Syntax** : **SEARCH**(**find\_text**,**within\_text**,start\_num)  **Find\_text**   is the text you want to find.  **Within\_text**   is the text in which you want to search for find\_text.  **Start\_num**   is the character number in within\_text at which you want to start searching.  **Remarks**   * Use SEARCH to determine the location of a character or text string within another text string so that you can use the MID or REPLACE functions to change the text. * SEARCH are **not case sensitive**. If you want to do a case sensitive search, you can use Find function. |
| **MID** | MID returns a specific number of characters from a text string, starting at the position you specify, based on the number of characters you specify.  **Syntax** : **MID**(**text**,**start\_num**,**num\_chars**)  **Text**   is the text string containing the characters you want to extract.  **Start\_num**   is the position of the first character you want to extract in text. The first character in text has start\_num 1, and so on.  **Num\_chars**   specifies the number of characters you want MID to return from text.  **Remarks**   * If start\_num is greater than the length of text, MID returns "" (empty text). * If start\_num is less than the length of text, but start\_num plus num\_chars exceeds the length of text, MID returns the characters up to the end of text. * If start\_num is less than 1, MID returns the #VALUE! error value. * If num\_chars is negative, MID returns the #VALUE! error value. * If num\_bytes is negative, MIDB returns the #VALUE! error value. |
| **EXACT** | Compares two text strings and returns TRUE if they are exactly the same, FALSE otherwise. EXACT is **case-sensitive** but ignores formatting differences. Use EXACT to test text being entered into a document.  **Syntax** : **EXACT**(**text1**,**text2**)  **Text1**   is the first text string.  **Text2**   is the second text string. |
| **SUBSTITUTE** | ***Substitutes new\_text for old\_text in a text string.*** Use SUBSTITUTE when you want to replace specific text in a text string; use REPLACE when you want to replace any text that occurs in a specific location in a text string.  **Syntax** : **SUBSTITUTE**(**text**,**old\_text**,**new\_text**,[instance\_num])  **Text**   is the text or the reference to a cell containing text for which you want to substitute characters.  **Old\_text**   is the text you want to replace.  **New\_text**   is the text you want to replace old\_text with.  **Instance\_num**   specifies which occurrence of old\_text you want to replace with new\_text. If you specify instance\_num, only that instance of old\_text is replaced. Otherwise, every occurrence of old\_text in text is changed to new\_text. |
| **REPLACE** | ***REPLACE replaces part of a text string, based on the number of characters you specify, with a different text string.***  **Syntax** : **REPLACE**(**old\_text**,**start\_num**,**num\_chars**,**new\_text**)  **Old\_text**   is text in which you want to replace some characters.  **Start\_num**   is the position of the character in old\_text that you want to replace with new\_text.  **Num\_chars**   is the number of characters in old\_text that you want REPLACE to replace with new\_text.  **New\_text**   is the text that will replace characters in old\_text. |
| **FIND** | ***FIND locate one text string within a second text string, and return the number of the starting position of the first text string from the first character of the second text string.***  **Syntax** : **FIND**(**find\_text**,**within\_text**,start\_num)  **Find\_text**   is the text you want to find.  **Within\_text**   is the text containing the text you want to find.  **Start\_num**   specifies the character at which to start the search. The first character in within\_text is character number 1. If you omit start\_num, it is assumed to be 1.  **Remarks**   * FIND are **case sensitive** and don't allow wildcard characters. If you don't want to do a case sensitive search or use wildcard characters, you can use SEARCH . * If find\_text is "" (empty text), FIND matches the first character in the search string (that is, the character numbered start\_num or 1). * Find\_text cannot contain any wildcard characters. * If find\_text does not appear in within\_text, FIND return the #VALUE! error value. * If start\_num is not greater than zero, FIND return the #VALUE! error value. * If start\_num is greater than the length of within\_text, FIND and FINDB return the #VALUE! error value. * Use start\_num to skip a specified number of characters. Using FIND as an example, suppose you are working with the text string "AYF0093.YoungMensApparel". To find the number of the first "Y" in the descriptive part of the text string, set start\_num equal to 8 so that the serial-number portion of the text is not searched. FIND begins with character 8, finds find\_text at the next character, and returns the number 9. FIND always returns the number of characters from the start of within\_text, counting the characters you skip if start\_num is greater than 1. |
| **TRIM** | ***Removes all spaces from text except for single spaces between words. Use TRIM on text that you have received from another application that may have irregular spacing.***  **Syntax** : **TRIM**(**text**)  **Text**   is the text from which you want spaces removed. |
| **REPT** | ***Repeats text a given number of times. Use REPT to fill a cell with a number of instances of a text string.***  **Syntax** : **REPT**(**text**,**number\_times**)  **Text**   is the text you want to repeat.  **Number\_times**   is a positive number specifying the number of times to repeat text.  **Remarks**   * If number\_times is 0 (zero), REPT returns "" (empty text). * If number\_times is not an integer, it is truncated. * The result of the REPT function cannot be longer than 32,767 characters, or REPT returns #VALUE!. |

***Example –1:* Use the LEFT function to convert invalid numbers to valid numbers**

|  |  |
| --- | --- |
| **To cut off the last digit and display a negative value:**   1. In a worksheet, enter a series of numbers in cells A2:A10 that have a minus sign at   the end.   1. Select cells B2:B10 and type the following formula: **=-LEFT(A2;LEN(A2) ;1)**. 2. Press **<Ctrl+Enter>**. |  |

***Example –2:*Use the SEARCH function to separate first name from last name**

|  |  |
| --- | --- |
| **To separate the first and last names:**   * 1. In a worksheet, enter a series of full names in cells A2:A11.   2. Select cells B2:B11 and type the following formula:   **=LEFT(A2; SEARCH(" ";A2)-1)**.   * 1. Press **<Ctrl+Enter>**. |  |

***Example –3:* Use the MID function to separate last name from first name**

|  |  |
| --- | --- |
| **To separate the last name from the first name:**  **1.** In a worksheet, enter a series of full names in cells A2:A11.  **2**. Select cells B2:B11 and type the following formula: **=MID(A2;SEARCH(" "; A2)+1;100)**.  **3**. Press **<Ctrl+ Enter>**. |  |

***Example –4:***  **Use the EXACT function to compare two columns**

|  |  |
| --- | --- |
| **To compare two columns:**  **1.** Select cells C2:C10 and type the following formula:  **=EXACT(A2; B2)**.  **2.** Press **<Ctrl+ Enter>**.  **3**. Select cells D2:D10 and type the following formula:  **=IF(A2=B2 ; TRUE ; FALSE)**.  **4.** P ress **<Ctrl+ Enter>**. |  |

***Example –5:***  **Use the SUBSTITUTE function to substitute characters**

|  |  |
| --- | --- |
| **To use SUBSTITUTE and force Excel to calculate:**  **1.** Format column A as text.  **2.** Enter a series of numbers in cells A2:A10. Notice that Excel tags them with green triangles in the upper-left corner to indicate the numbers have been entered as text.  **3.** Select cells B2:B10 and type the following formula:  **=VALUE(SUBSTITUTE(A2;"'";""))**.  **4.** Press **<Ctrl +Enter>**.  **5.** Select cell A12, type the following formula: **=SUM(A2:A10)**, and press **<Enter>**.  **6.** Select cell B12, type the following formula: **=SUM(B2:B10)**, and press **<Enter>**.  ***Note:*** If you want to substitute the second occurrence of this character, use the following formula: = **SUBSTITUTE (A2,"-","",2).** |  |

***Example –8:* Use the REPLACE function to replace and calculate**

|  |  |
| --- | --- |
| **To replace periods with colons and calculate:**  1.Select cells D2:D10 and type the following formula:    **=(REPLACE(C2 ;SEARCH(".";C2),1;":")-REPLACE (B2,SEARCH(".";B2);1 ;":"))**.  **2**. Press **<Ctrl+ enter >** |  |

***Example –8:* Use the FIND function to combine text and date**

|  |  |
| --- | --- |
| **To combine and format data at the same time:**  1. In a worksheet, copy the data shown in Figure  2. Select cells C2:C6 and type the following formula:  **=REPLACE(A2,FIND("XXX";A2;1);3;TEXT (B2;"MM-DD-YYYY"))**.  3. Press **<Ctrl +Enter>**. |  |

***Example –9:* Use the TRIM function to convert “text-numbers” to real numbers**

|  |  |
| --- | --- |
| **To convert text that represents a number to a value:**  **1.** Format column A as text.  **2.** In cells A2:A10, type a series of numbers with leading spaces.  **3.** Select cells B2:B10 and type the following formula: **=VALUE (TRIM(A2))**.  **4.** Press **<Ctrl +Enter>**. |  |

***Example –10 :* Use the REPT function to show data in graphic mode**

|  |  |
| --- | --- |
| **To show data in a simple chart:**  **1.** In cells A2:A10, type numbers from 1 to 10.  **2.** Select cells B2:B10 and type the following formula: **=REPT("n"; A2)**.  **3.** Press **<Ctrl+Enter>**.  **4**. From the Format menu, select **Cells**.  **5.** Select the Font tab.  **6.** Select **Wingdings** from the Font list and click **OK**. |  |

* **Date and time functions**

|  |  |  |
| --- | --- | --- |
|  | **Function** | **Description** |
|  | **DATE** | Returns the serial number of a particular date, **The syntax is:**  **DATE(*year*, *month*, *day*) , where**  ***year*:** This argument can be from one to four digits.  ***month*:** A number representing the month of the year (1 to 12).  ***day*:** A number representing the day of the month (1 to 31). |
|  | **DATEVALUE** | Converts a date in the form of text to a serial number , **The syntax is:**  **DATEVALUE(date\_text) , where**  the date\_text argument must represent a date between January 1, 1904 and December 31, 9999. The DATEVALUE function returns the #VALUE! error value if the value of the date\_text argument falls outside of this range. |
|  | **DAY** | Converts a serial number to a day of the month , **The syntax is:**  **DAY(serial\_number), where**  Serial\_number  Required. The date of the day you are trying to find. Dates should be entered by using the DATE function, or as results of other formulas or functions |
|  | **EDATE** | Returns the serial number of the date that is the indicated number of months  before or after the start date, **The syntax is:**  **EDATE(start\_date, months) , where**  **Start\_date**  Required. A date that represents the start date. Dates should be entered by using the DATE function, or as results of other formulas or functions.  **Months**  Required. The number of months before or after start\_date. A positive value for months yields a future date; a negative value yields a past date. |
|  | **EOMONTH** | Returns the serial number of the last day of the month before or after a  specified number of months(offset\_month) , **The syntax is:**  **EOMONTH(start\_date, months), where**  **Start\_date  Required**. A date that represents the starting date. Dates should be entered by using the DATE function, or as results of other formulas or functions.  **Months  Required**. The number of months before or after start\_date. A positive value for months yields a future date; a negative value yields a past date.  **Note**   * If months is not an integer, it is truncated * Problems can occur if [dates are entered as text](ms-help://MS.EXCEL.14.1033/EXCEL/content/HP10054141.htm). |
|  | **HOUR** | Converts a serial number to an hour , **the syntax is:**  **HOUR(serial\_number) ,where**  **Serial\_number**  Required. The time that contains the hour you want to find. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008. Problems can occur if [dates are entered as text](ms-help://MS.EXCEL.14.1033/EXCEL/content/HP10054141.htm). |
|  | **MINUTE** | Converts a serial number to a minute The minute is given as an integer, ranging from 0 to 59. , **the syntax is:**  **MINUTE(serial\_number) , where**    **Serial\_number  Required**. The time that contains the minute you want to find. Times may be entered as text strings within quotation marks (for example, "6:45 PM"), as decimal numbers (for example, 0.78125, which represents 6:45 PM), or as results of other formulas or functions (for example, TIMEVALUE("6:45 PM")). |
|  | **MONTH** | Converts a serial number to a month , The month is given as an integer, ranging from 1 (January) to 12 (December) , **the syntax is:**  **MONTH(serial\_number) , where**  **Serial\_number  Required**. The date of the month you are trying to find. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008. Problems can occur if [dates are entered as text](ms-help://MS.EXCEL.14.1033/EXCEL/content/HP10054141.htm). |
|  | **NETWORKDAYS** | Returns the number of whole working days between start\_date and end\_date. Working days exclude weekends and any dates identified in holidays. Use NETWORKDAYS to calculate employee benefits that accrue based on the number of days worked during a specific term.  **Tip**  To calculate whole workdays between two dates by using parameters to indicate which and how many days are weekend days, use the [NETWORKDAYS.INTL function](ms-help://MS.EXCEL.14.1033/EXCEL/content/HA10354379.htm)  **NETWORKDAYS(start\_date, end\_date, [holidays]) , where**   (argument: A value that provides information to an action, an event, a method, a property, a function, or a procedure.)**Start\_date  Required**. A date that represents the start date.  **End\_date  Required.** A date that represents the end date.  **Holidays  Optional.** An optional range of one or more dates to exclude from the working calendar, such as state and federal holidays and floating holidays |
|  | **NETWORKDAYS.INTL** | Returns the number of whole workdays between two dates using parameters to indicate which and how many days are weekend days. Weekend days and any days that are specified as holidays are not considered as workdays. **the syntax is:**  **NETWORKDAYS.INTL(start\_date, end\_date, [weekend], [holidays])**  **Start\_date and end\_date**  Required. The dates for which the difference is to be computed. The start\_date can be earlier than, the same as, or later than the end\_date.  **Weekend**  Optional. Indicates the days of the week that are weekend days and are not included in the number of whole working days between start\_date and end\_date. Weekend is a weekend number or string that specifies when weekends occur.  **Weekend number values indicate the following weekend days**   |  |  |  |  | | --- | --- | --- | --- | | **Weekend number** | **Weekend days** | **Weekend number** | **Weekend days** | | **1 or omitted** | **Saturday, Sunday** | **11** | **Sunday only** | | **2** | **Sunday, Monday** | **12** | **Monday only** | | **3** | **Monday, Tuesday** | **13** | **Tuesday only** | | **4** | **Tuesday, Wednesday** | **14** | **Wednesday only** | | **5** | **Wednesday, Thursday** | **15** | **Thursday only** | | **6** | **Thursday, Friday** | **16** | **Friday only** | | **7** | **Friday, Saturday** | **17** | **Saturday only** |   Weekend string values are seven characters long and each character in the string represents a day of the week, starting with Monday. 1 represents a non-workday and 0 represents a workday. Only the characters 1 and 0 are permitted in the string. Using 1111111 will always return 0.  For example, 0000011 would result in a weekend that is Saturday and Sunday.  **Holidays  Optional.** An optional set of one or more dates that are to be excluded from the working day calendar. holidays shall be a range of cells that contain the dates, or an array constant of the serial values that represent those dates. The ordering of dates or serial values in holidays can be arbitrary. |
|  | **NOW** | Returns the serial number of the current date and time. **the syntax** :  **NOW() , where**  The NOW function syntax has no [arguments (argument: A value that provides information to an action, an event, a method, a property, a function, or a procedure.)](javascript:AppendPopup(this,'ofArgument_2_2')).  Note    If the NOW function does not update cell values when you expect it to, you might need to change settings that control when the workbook or worksheet recalculates. |
|  | **SECOND** | Converts a serial number to a second, The second is given as an integer in the range 0 (zero) to 59. **the syntax :**  **SECOND(serial\_number) , where**  **Serial\_number**  Required. The time that contains the seconds you want to find. Times may be entered as text strings within quotation marks (for example, "6:45 PM"), as decimal numbers (for example, 0.78125, which represents 6:45 PM), or as results of other formulas or functions (for example, TIMEVALUE("6:45 PM")). |
|  | **TIME** | Returns the decimal number for a particular time. If the cell format was **General** before the function was entered, the result is formatted as a date , **the syntax :**  **TIME(hour, minute, second) , where**  **Hour**  Required. A number from 0 (zero) to 32767 representing the hour. Any value greater than 23 will be divided by 24 and the remainder will be treated as the hour value. For example, TIME(27,0,0) = TIME(3,0,0) = .125 or 3:00 AM.  **Minute**  Required. A number from 0 to 32767 representing the minute. Any value greater than 59 will be converted to hours and minutes. For example, TIME(0,750,0) = TIME(12,30,0) = .520833 or 12:30 PM.  **Second**  Required. A number from 0 to 32767 representing the second. Any value greater than 59 will be converted to hours, minutes, and seconds. For example, TIME(0,0,2000) = TIME(0,33,22) = .023148 or 12:33:20 AM |
|  | **TIMEVALUE** | Converts a time in the form of text to a serial number , in other word`, Returns the decimal number of the time represented by a text string. The decimal number is a value ranging from 0 (zero) to 0.99999999, representing the times from 0:00:00 (12:00:00 AM) to 23:59:59 (11:59:59 P.M.) **, the syntax:**  **TIMEVALUE(time\_text) , where**  **Time\_text**  Required. A text string that represents a time in any one of the Microsoft Excel time formats; for example, "6:45 PM" and "18:45" text strings within quotation marks that represent time. |
|  | **TODAY** | Returns the **serial number** of today's date ,  **The serial number is** the date-time code used by Excel for date and time calculations. If the cell format was **General** before the function was entered, Excel changes the cell format to **Date**. If you want to view the serial number, you must change the cell format to **General** or **Number**. **The syntax:**    **TODAY() , where**  The TODAY function syntax has no [arguments (argument: A value that provides information to an action, an event, a method, a property, a function, or a procedure.)](javascript:AppendPopup(this,'ofArgument_2_2')).  Note: The **TODAY** function is useful when you need to have the current date displayed on a worksheet, regardless of when you open the workbook. It is also useful for calculating intervals. For example, if you know that someone was born in 1963, you might use the following formula to find that person's age as of this year's birthday:  **=YEAR(TODAY())-1963**  This formula uses the **TODAY** function as an argument for the **YEAR** function to obtain the current year, and then subtracts 1963, returning the person's age. |
|  | **WEEKDAY** | Returns the day of the week corresponding to a date. The day is given as an integer, ranging from 1 (Sunday) to 7 (Saturday), by default. **The syntax:**  **WEEKDAY(serial\_number,[return\_type]), where**  **Serial\_number**  Required. A sequential number that represents the date of the day you are trying to find. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008. Problems can occur if dates are entered as text.  **Return\_type**  Optional. A number that determines the type of return value   |  |  | | --- | --- | | **Return\_type** | **Number returned** | | 1 or omitted | Numbers 1 (Sunday) through 7 (Saturday). Behaves like previous versions of Microsoft Excel. | | 2 | Numbers 1 (Monday) through 7 (Sunday). | | 3 | Numbers 0 (Monday) through 6 (Sunday). | | 11 | Numbers 1 (Monday) through 7 (Sunday). | | 12 | Numbers 1 (Tuesday) through 7 (Monday). | | 13 | Numbers 1 (Wednesday) through 7 (Tuesday). | | 14 | Numbers 1 (Thursday) through 7 (Wednesday). | | 15 | Numbers 1 (Friday) through 7 (Thursday). | | 16 | Numbers 1 (Saturday) through 7 (Friday). | | 17 | Numbers 1 (Sunday) through 7 (Saturday). | |
|  | **WEEKNUM** | Returns the week number of a specific date. For example, the week containing January 1 is the first week of the year, and is numbered week 1.  There are two systems used for this function:   * **System 1**  The week containing January 1 is the first week of the year, and is numbered week 1. * **System 2**  The week containing the first Thursday , **The syntax :**   **WEEKNUM(serial\_number,[return\_type] , where**  **Serial\_number**   Required. A date within the week. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008. Problems can occur if dates are entered as text.  **Return\_type**   Optional. A number that determines on which day the week begins. The default is 1.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Return\_type** | **Week begins on** | **System** | **Return\_type** | **Week begins on** | **System** | | **1 or omitted** | **Sunday** | **1** | **14** | **Thursday** | **1** | | **2** | **Monday** | **1** | **15** | **Friday** | **1** | | **11** | **Monday** | **1** | **16** | **Saturday** | **1** | | **12** | **Tuesday** | **1** | **17** | **Sunday** | **1** | | **13** | **Wednesday** | **1** | **21** | **Monday** | **2** | |
|  | **WORKDAY** | **Returns a number that represents a date that is the indicated number of working days before or after a date (the starting date). Working days exclude weekends and any dates identified as holidays. Use WORKDAY to exclude weekends or holidays when you calculate invoice due dates, expected delivery times, or the number of days of work performed.**  **The syntax:**  **WORKDAY(start\_date, days, [holidays]), where**  **Start\_date**  Required. A date that represents the start date.  **Days**  Required. The number of nonweekend and nonholiday days before or after start\_date. A positive value for days yields a future date; a negative value yields a past date.  **Holidays**  Optional. An optional list of one or more dates to exclude from the working calendar, such as state and federal holidays and floating holidays. The list can be either a range of cells that contain the dates or an [array constant (array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.)](javascript:AppendPopup(this,'xldefArray_3_3')) of the serial numbers that represent the dates. |
|  | **YEAR** | **Returns the year corresponding to a date. The year is returned as an integer in the range 1900-9999. The syntax:**  **YEAR(serial\_number), where :**  **Serial\_number**  Required. The date of the year you want to find. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008. Problems can occur if [dates are entered as text](ms-help://MS.EXCEL.14.1033/EXCEL/content/HP10054141.htm). |
|  | **YEARFRAC** | **Calculates the fraction of the year represented by the number of whole days between two dates (the start\_date and the end\_date). The syntax** :  **YEARFRAC(start\_date, end\_date, [basis]), where**  **Start\_date**  Required. A date that represents the start date.  **End\_date**  Required. A date that represents the end date.  **Basis**  Optional. The type of day count basis to use.   |  |  | | --- | --- | | **Basis** | **Day count basis** | | 0 or omitted | US (NASD) 30/360 | | 1 | Actual/actual | | 2 | Actual/360 | | 3 | Actual/365 | | 4 | European 30/360 | |
|  | **DATEDIF** | **This function calculates the exact number of years, months, and days between two dates , this function undocumented you can use function from the Analysis ToolPak add-in.**  **The syntax is:**  **DATEDIF(start\_date, end\_date, format), where**  **start\_date**: The start date.  **end\_date:** The end date.  **format:** Indicates the format to use.  **“y”** gives the difference in years;  **“m”** in months;  “d” in days;  **“ym”** the difference in months, ignoring the year;  **“yd”** in days, ignoring the year; and  **“md”** in days, ignoring the month and year. |

|  |  |
| --- | --- |
| **Example1:**    **Use custom formatting to display the day of the week**  To display weekdays using customized formatting:  1. Select cells B2:B10 and type the formula **=A2**.  2. Press **<Ctrl+Enter>**.  3. From the Format menu, select **Cells**.  4. Select the Number tab and click on **Custom** below  Category.  5. In the Type box, change the number format to **dddd**.  6. Press **OK**. |  |
| **Example2:**  **Use the WEEKDAY function to determine the weekend**  To determine the weekend:  **1.** Using the worksheet from the previous example, select cells C2:C10 and type the following formula:  **=IF(OR(WEEKDAY(A2)=7,WEEKDAY(A2)=1),"weekend","")**.  **2.** Press **<Ctrl+Enter>**. |  |
| **Example3:**  **Use the TODAY function to check for future date To check for future dates:**  **1.** In cell D1, type the formula **=TODAY()** to show the current date.  **2.** Select cells B2:B10 and type the following formula:  **=IF(A2<=TODAY(),"n","y")**.  **3.**Press **<Ctrl+Enter>**. |  |
| **Example4:**  **Use the TEXT function to calculate with the TODAY function**  To calculate with the TODAY function:  **1.** In cell B1 type the formula **=TODAY()**.  **2.** In cell B2 type the formula **=TODAY()+10** to add ten days to the current date.  **3.** Select cell A4 and type the following formula: **="The** **project starts on " & TEXT(B1,"MM/DD/YYYY") & " and ends on " & TEXT(B2,"MM/DD/YYYY")**.  **4.** Press **<Enter>**. |  |
| **Example5:**  **Use the NOW function to calculate time**  To calculate with time:  **1.** In cell B1 type the formula **=NOW()**.  **2.** In cell B2 type the formula **=B1+0.25** to add six hours to the current time in cell B1.  **3.** Type the following formula in cell C1:  **= "The meeting** **starts at " & TEXT(B1,"hh:mm") &**  **" and ends at " &** **TEXT(B2,"hh:mm")**.  **4.** Press **<Enter>**. |  |
| **Example6:**  **Use the DATE function to combine columns with date parts**  **To combine values of cells into one date:**  **1.** Select cells D2:D10 and type the following formula: **=DATE(A2,B2,C2)**.  **2.** Press **<Ctrl+Enter>**. |  |
| **Example7:**  Use the LEFT, MID, and RIGHT functions to extract date parts  To extract, combine, and display the correct format:  **1.** Select cells B2:B10 and type the following formula:  **=DATE(LEFT(A2,4),MID(A2,FIND(".",A2,1)+1,2),**  **RIGHT(A2,2))**.  **2.** Press **<Ctrl+Enter>**. |  |
| **Example8:**  **To determine the last day of a month:**  **1.** In cells A2:A10 enter some dates.  **2.** In cells B2:B10 enter the desired offset from the start date (positive or negative values).  **3.** Select cells C2:C10 and type the following formula:  **=EOMONTH(A2,B2)**.  **4.** Press **<Ctrl+Enter>**. |  |
| **Example9:**  **Use the WEEKDAY function to calculate with different hourly pay rates :**  To calculate with different hourly pay rates:  **1**. In a worksheet, enter the data shown in columns A, B, and C  **2**. Select cell F2 and enter **12.50** (hourly rate for Monday through Friday).  **3**. Select cell F5 and enter **18.50** (hourly rate for Saturday and Sunday).  **4**. Select cells D2:D10 and type the following formula:  **=IF(OR(WEEKDAY(A2)=1,WEEKDAY(A2)=7), C2\*$F$5,C2\*$F$2)**.  **5**. Press **<Ctrl+Enter>**. |  |
| **Example10:**  **Use the WEEKNUM function to determine the week number**  To determine the week number:  1. Type different dates of the year in cells A2:A10.  2. Select cells B2:B10 and type the following formula:  **=WEEKNUM(A2)**.  3. Press **<Ctrl+Enter>**. |  |
| **Example11:**  **Use the WORKDAY function to calculate workdays**  **To determine the end date of a project**:  **1**. In cell C2, enter the start date of the project.  **2**. In column B enter the estimated days to finish each step.  **3.** In cell D2 type the following formula:  **=WORKDAY(C3,B3,$F$2:$F$8)**.  **4.** In cells F1:F8 additional holidays can be listed individually.  **5.** In cell C3 type the formula **=D2+1**.  **6.** Fill cells C3 and D2 down to C6 and D6. |  |
| **Example12:**  **Use the NETWORKDAYS function to determine the number of workdays**  To determine the number of workdays:  **1.** In column B type the start date of each step.  **2.** In column C type the end date of each step.  **3.** List additional holidays in cells F2:F6.  **4.** Select cells D2:D6 and type the following formula:  **=NETWORKDAYS(B2,C2,$F$2:$F$6)**.  **5.** Press **<Ctrl+Enter>**. |  |
| **Example13:**  **Use the YEARFRAC function to calculate ages of employees**  To calculate the age of employees based on the current date:  **1.** In column A list the names of employees.  **2.** In column B enter their birthdays.  **3.** Select cells C2:C10 and type the formula **TODAY()**.  **4.** Press **<Ctrl+Enter>**.  **5.** Select cells D2:D10 and type the following formula:  **=YEARFRAC(B2,C2,0)**.  **6.** Press **<Ctrl+Enter>**. |  |
| **Example14:**  **Use the DATEDIF function to calculate the ages of employees**  **To calculate the ages of employees:**  **1.** In column A list the names of employees.  **2.** In column B enter their birthdays.  **3.** Select cells C2:C10 and type the formula **TODAY()**  **4.** Press **<Ctrl+Enter>**.  **5.** Select cells D2:D10 and type the following formula:  **=DATEDIF(B2,C2,"Y") & " years and " &**  **DATEDIF(B2,C2,"YM") & " months"**.  **6.** Press **<Ctrl+Enter>**. |  |

* **Quick preview of advance tools**
* **Conditional format**

**What is Conditional Formatting?**

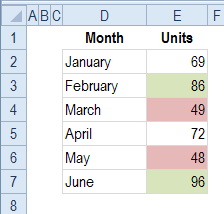
With conditional formatting, you can select one or more cells, and create rules (conditions) for when and how those cells are formatted. The conditions can be, based on the selected cell's contents, or based on the contents of another cell.

**You can control the following formats:**

* **Number format**
* **Font, font style, and font colour (but not font size)**
* **Fill colo arnd fill pattern**
* **Border colour and border style (but not border thickness)**

***Note :*** If the rules (conditions) that you specified are met, then the formatting is applied.

***Example -1*** , you can set conditional formatting so that a cell turns red if its value is low, and turns green if its value is high.

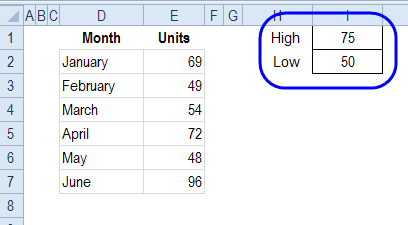


**In this example, you'll set conditional formats so that a cell**:

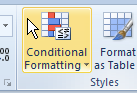
* turns green if it contains a value higher than 75 and
* turns red if it contains a value lower than 50.

Follow these steps to apply conditional formatting to cells:

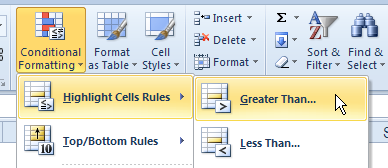
1. In cell I1, type the high value -- 75
2. In cell I2, type the low value -- 50



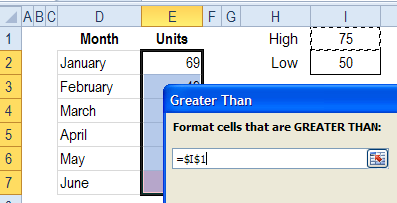
1. Select the cells to be formatted. In this example, cells E2:E7 are selected.
2. On the Ribbon's Home tab, click Conditional Formatting



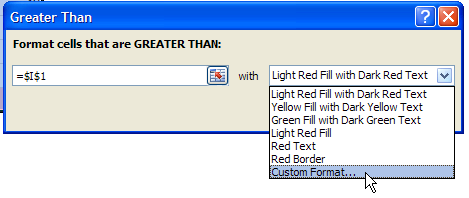
1. To format the high values, click Highlight Cell Rules, then click Greater Than...



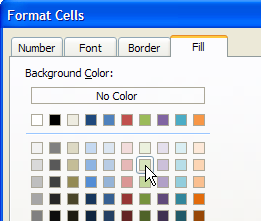
1. In the Greater Than window, delete the value that appears, and click on cell I1, where the High value is entered.



1. Click the drop down list for formats, and click Custom Format.

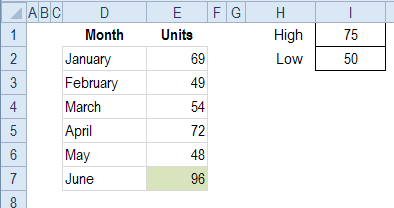


1. In the Format Cells window, click the Fill tab, and click on the green fill colour that you want.



1. Click OK to close the Format Cells window, and click OK to close the Greater Than window.

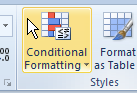
The cells with values greater than 75 are now coloured green.



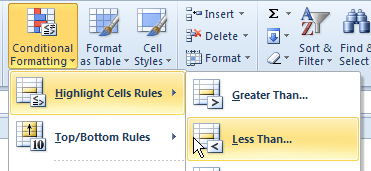
Example – 2: Apply 2nd Conditional Format

To color the low values in red fill, you can apply a second conditional formatting rule to the cells.

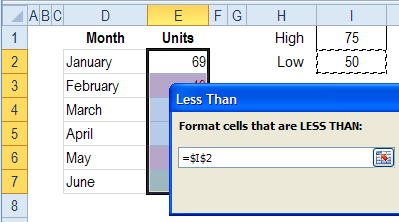
1. Select the cells to be formatted. In this example, cells E2:E7 are selected.
2. On the Ribbon's Home tab, click Conditional Formatting



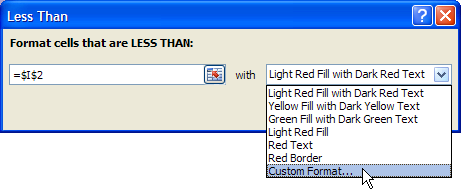
1. To format the high values, click Highlight Cell Rules, then click Less Than...



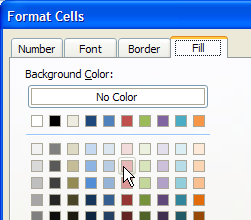
1. In the Less Than window, delete the value that appears, and click on cell I2, where the Low value is entered.



1. Click the drop down list for formats, and click Custom Format.

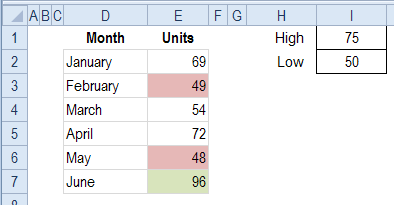


1. In the Format Cells window, click the Fill tab, and click on the green fill colour that you want.



1. Click OK to close the Format Cells window, and click OK to close the Less Than window.

The cells with values greater than 75 are now coloured green, and cells less than 50 are red.



* **pivot tables**
* **What is a Pivot Table?**

Excel’s Pivot Table is probably the most useful and time-saving tool for analyzing data that’s in table format. In the simplest Pivot Table, one identifies a row value, a column value, and a data value. The data value (usually a numeric value) in this simple Pivot Table is automatically summarized at each row and column intersection.

**Note :** Create a pivot table that is based on a **dynamic data source** -- that is a range that adjusts automatically, if data is added or removed**.**

* **Source Data Requirements**

The most basic of Pivot Tables is created from source data that’s in a table or range in

an Excel workbook. Data suitable for use in a Pivot Table must have these characteristics:

**1.** The top row of data contains column headers.

**2.** Each row of data is a record about a particular entity or transaction.

**3.** Each column of data holds the same kind of information.

**4.** There are no entirely blank rows in the data.

**5.** There are no entirely blank columns in the data.

**6.** If a column contains numbers, use a zero instead of a blank cell when you don’t have

a value.

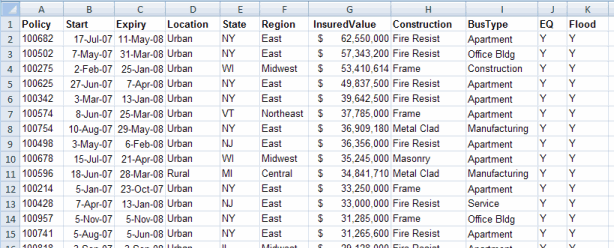
# Create an Excel Pivot Table

Step-by-step instructions for creating a pivot table in Excel 2010 or Excel 2007.

## Preparing Your Pivot Table Data

Before you create a pivot table, make sure your data is organized correctly. There are instructions on the following pages, for setting up your source data in a table, organized into rows and columns.

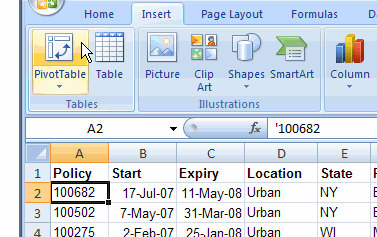
In the fallowing example the source data contains information about property insurance policies. Each row has the details about one insurance policy, such as the region, state, construction type and the value of the insured property.



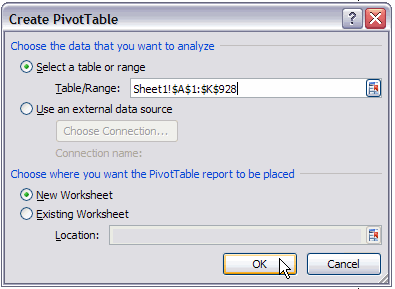
## Creating a Simple Pivot Table

After your source data is prepared, you can create a pivot table. We'll create a pivot table that shows the total insured value in each of the four regions where we sell insurance.

1. Select any cell in the source data table.
2. On the Ribbon, click the Insert tab.
3. In the Tables group, click PivotTable.



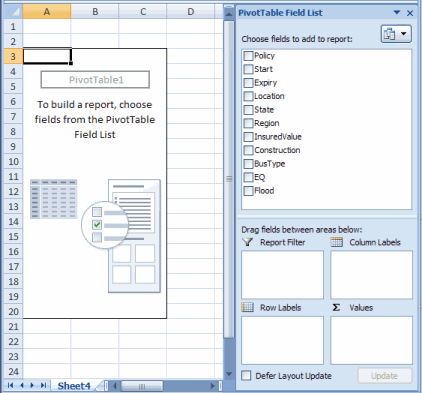
1. In the Create PivotTable dialog box, the address of your source data table should be automatically entered in the Table/Range box. If not, click on the worksheet, and select the range manually.



1. Next, select New Worksheet or Existing Worksheet as the location for your pivot table, then click OK.

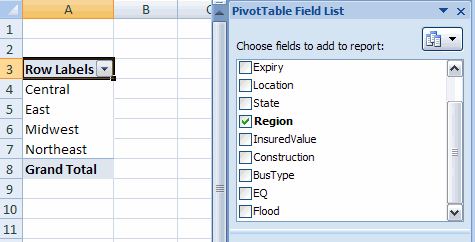
## Adding Fields to the Pivot Table

An empty pivot table is created in your workbook, either on a new sheet, or the existing sheet that you selected. When you select a cell within the pivot table, a PivotTable Field List appears, at the right of the worksheet.

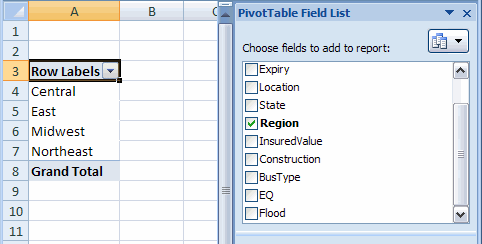


We want to see the total insured value in each of the four regions, so we'll add the Region and InsuredValue fields to the pivot table.

1. In the PivotTable Field List, add a check mark to the Region field. The Region field is automatically added to the pivot table, in the Row Labels area.



1. Add a check mark to the InsuredValue field, and it will be automatically added to the Values area. You can now see the total insured value in each region.



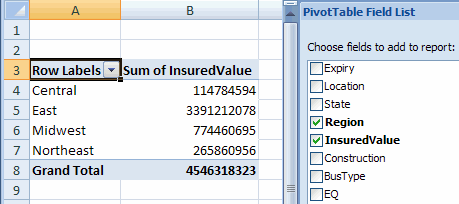
## 

## Modifying the Pivot Table

After you've created a pivot table, you can add more fields, remove fields, or move the fields to a different location in the pivot table layout. We'll remove the Region field, and add the Location field, to see the value of Rural policies compared to Urban.

1. To remove the Region field, click on its check box, to remove the check mark.
2. To add the Location field, click on its check box, to add a check mark.

The pivot table now shows the totals for Rural and Urban locations.



## Group A Pivot Table

When analyzing data in an Excel Pivot Table, it is often useful to group the Pivot Table data into categories.

For example, instead of displaying total sales values for each day, you might prefer to group the days into months and display the total sales values for each month.

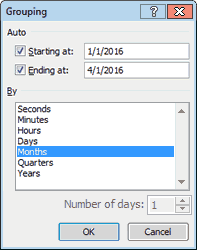
Excel can automatically group numeric values (including dates & times) in pivot tables. This is shown in the following examples.

## Pivot table showing sum of sales figures per dayExample 1: Group a Pivot Table by Date

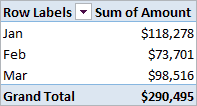
|  |
| --- |
|  |

Imagine you have created the pivot table on the left, which shows sales figures for each date of the first quarter of 2016.

If you want to group the sales figures by month, you can do this as follows:

* Right click on the left column of the pivot table (containing the dates) and select the option **Group...**;
* You will be presented with the 'Grouping' dialog box for dates;
* Select the value **Months** and click **OK**.

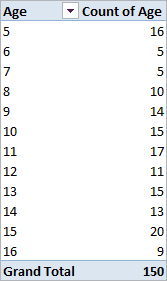
This will group the figures by month, as shown in the pivot table below.



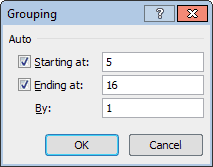
## Example 2: Group a Pivot Table by Range

Imagine you have created the pivot table on the left, which groups 150 children according to age. The ages range from 5 to 16 years.

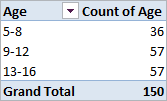
If you want to group the ages into the age ranges 5-8 years, 9-12 years and 13-16 years, you can this as follows:



* Right click on left column of the pivot table (containing the ages) and select the option **Group...**;
* You will be presented with the 'Grouping' dialog box for numbers;
* Excel will automatically enter the minimum and maximum values from the data, (which in this case are of 5 and 16);
* We want to group the ages into 4-year periods, so we change the value in the lower box (labelled **By:**) to a 4;
* Click **OK**



This will group the ages into categories, beginning with age 5-8 and increasing by 4 years each time. The resulting pivot table is shown below:



## Ungrouping a Pivot Table

**To ungroup the values in a pivot table, simply:**

* Right click on left column of the pivot table (containing the grouped values);
* Select the option Ungroup...
* **Data Validation**

## What is Data Validation?

In Excel, the data validation feature helps you control what can be entered in your worksheet. For example, you can:

* create a drop down list of items in a cell
* [restrict entries](http://www.contextures.com/xlDataVal06.html), such as a date range or whole numbers only
* create [custom rules](http://www.contextures.com/xlDataVal07.html) for what can be entered

## How to Create a Drop Down List

With Data Validation, you can create a dropdown list of options in a cell. There are 3 easy steps:

[1. Create a Table of Items](http://www.contextures.com/xlDataVal01.html#table) OR [Create a List](http://www.contextures.com/xlDataVal01.html#differentsheet)

[2. Name the List](http://www.contextures.com/xlDataVal01.html#name)

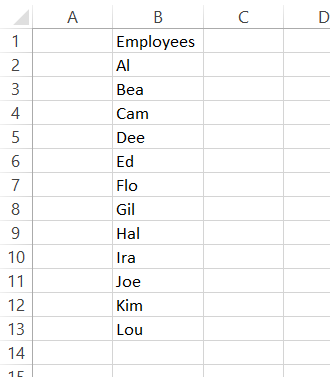
[3. Create the Drop Down](http://www.contextures.com/xlDataVal01.html#create)

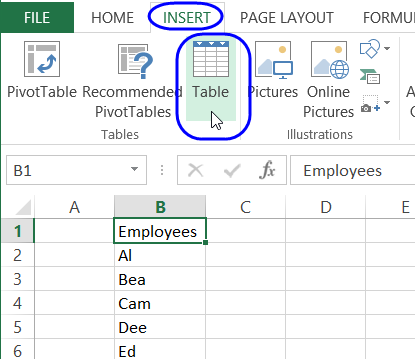
**Note**: Data validation is not foolproof. It can be circumvented by pasting data into the cell, or by choosing Clear > Clear All, on the Ribbon's Home tab.

#### 1. Create a Table of Items

The easiest way to create and maintain the list of options, is to type them on a worksheet. You can do this on the sheet that will contain the drop down lists, or on a different sheet. In this example, the list will be stored on a sheet named *Lists*.

NOTE: A data validation list can show up to show 32,767 items from a list on the worksheet.

1. Type a heading for the list -- Employees in this example
2. Immediately below the heading cell, in single column, type the entries you want to see in the drop down list. Do not leave blank cells between the entries.
3. Select a cell in the list, and on the Ribbon's Insert tab, click Table



1. Add a check mark in "My table has Headers" and click OK



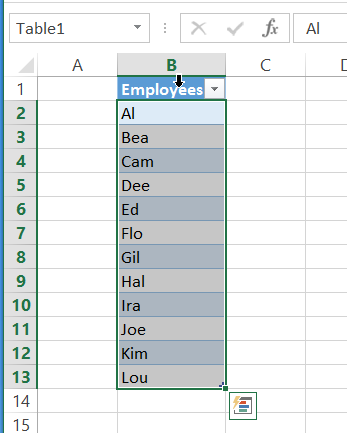
The table is now a Named Excel Table.

#### 

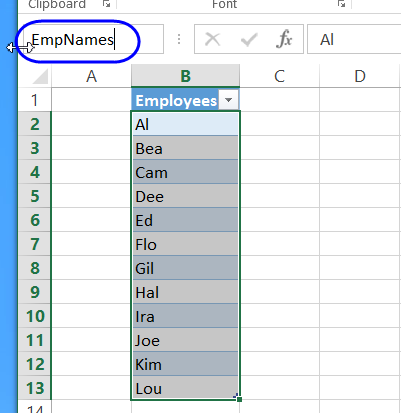
#### 2. Name the List

Next, you will create a named range that does not include the heading cell in the table. This named range will be dynamic -- it will adjust automatically if items are added to or deleted from the list.

1. Click at the top of the heading cell, to select all the cells in the list (the heading will not be selected).



1. Click in the Name box, to the left of the formula bar
2. Type a one-word name for the list, e.g. EmpNames.

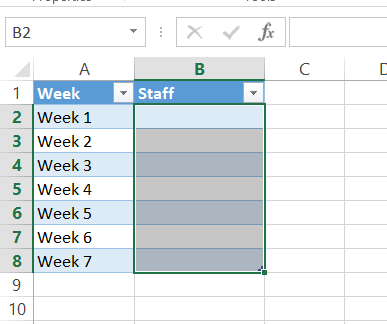


1. Press the Enter key, to complete the naming process. (After you press Enter, the name will disappear, and the Table name will appear in the name box.)

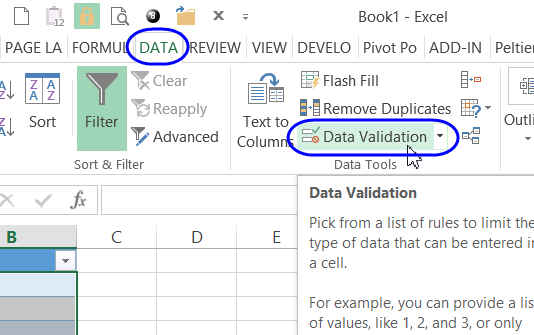
#### 3. Create the Drop Down List

Now that you have created a named range, you can use that to create a drop down list in one or more cells

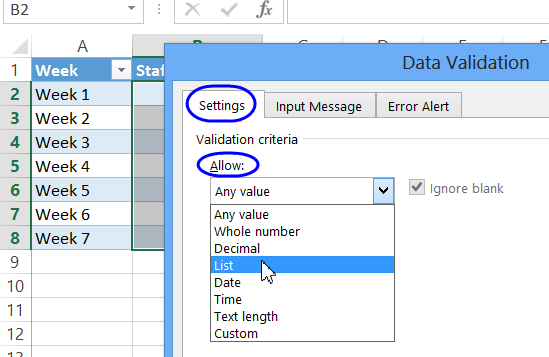
1. Select the cells in which you want the drop down list



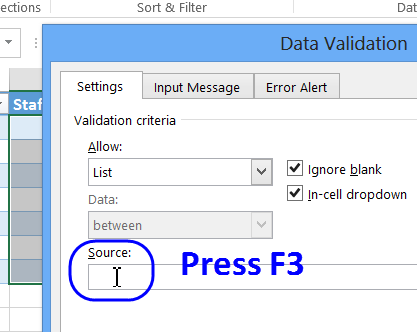
1. On the Ribbon's Data tab, click Data Validation.



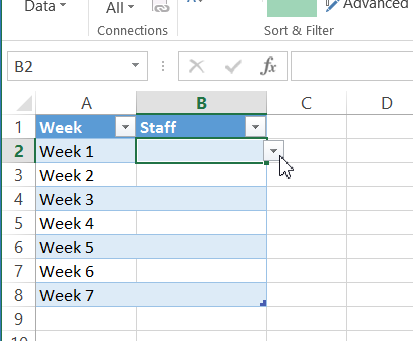
1. From the Allow drop-down list, choose List



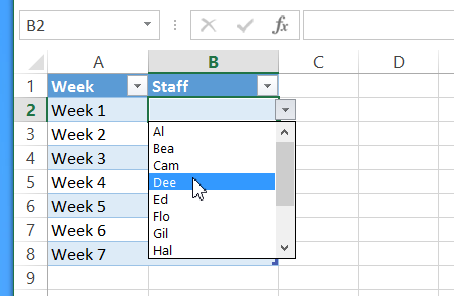
1. Click in the Source box, and type an equal sign, and the list name, for example:  
   =EmpNames  
   OR, press the F3 key, to see a list of names, click on a name, and click OK



1. Click OK to close the Data Validation dialog box.
2. Click on one of the cells, and click the drop down arrow



1. Click on an item in the drop down list, to enter it into the cell.

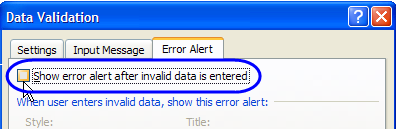


## List on Different Sheet

If you prefer not to create a named Excel table, you can create a named range, and use that as the source for a drop down list. The drop down lists can be on the same sheet as the source list, or on a different sheet.

## Allow Entries Not in Drop Down List

To allow users to also type items that are not in the list, remove the check mark to turn off the [Error Alert](http://www.contextures.com/xlDataVal04.html#Error).



* **Function shortcut keys**

|  |  |
| --- | --- |
|  |  |
|  |  |