

INDEX

Module-I

Extend Your Insight Office Excel 2010.....	02
Formulas.....	42

Module-II

Building macros in Excel.....	61
Programming Structure in Excel VBA.....	76
Excel VBA Controls' Properties and VBA Code.....	110

Extend Your Insight Office Excel 2010

For many people, Microsoft Office Excel 2010 is one of those programs you grow into. Unless their job responsibilities demand a detailed working knowledge of spreadsheets right from the start, their earliest experiences with Office Excel 2010 might have been creating simple financial documents—maybe in response to a request from a manager, a need in their own business, or a question from a financial advisor.

Over the years, Office Excel has evolved from a basic spreadsheet tools to a major business application that enables information workers to create, analyze, consolidate, report on. And share critical information that affects the entire business life cycle. This is not your grand-mothers calculator! Office Excel 2010 faces the same challenge that Microsoft* Office Word 2010 takes on: offering powerful, sophisticated tools in an accessible way that enables both experts and novices to find what they need easily and produce professional-quality, accurate, and insightful results.

The major changes in Office Excel 2010 offer exciting possibilities that can simplify and extend the way users create, work with, and share financial documents—spreadsheets, work-books, reports, and more. Specifically, the additions and enhancements in Office Excel 2010 help users create better spreadsheets faster, understand, visualize, and share important information more easily with others, and make data analysis easier and more effective, resulting in better insight for those decisions that affect the bottom line. This chapter introduces some of these significant changes.

The Design of Office Excel 2010

The new results-oriented look of Office Excel 2010 brings to your work area the tools you need-and only the tools you need-to complete what you're trying to create, analyze, or illustrate in your worksheet.

Consistent with other major Microsoft Office system applications, the user interface is designed to help you be more productive by offering a series of command tabs, as well as command sets and contextual commands related to specific objects in your worksheet. Here's a list of the command tabs you'll find in Office Excel 2010:

The Home tab includes the commands you need to work with the Clipboard; choose and change fonts; control the alignment of cell content; select number formats; choose cell style and format; and edit, sort, and search your data.

The Insert tab houses the commands for the objects you add to your worksheets; for example, tables, charts, illustrations, links, and various kinds of text items, such as column or row labels.

The Page Layout tab offers all things related to setting up the worksheet, including choosing themes, selecting page setup options, controlling the scaling of individual objects, selecting worksheet options, and arranging items on the sheet.

The Formulas tab includes the Function Wizard, the Function Library, the commands you need for creating and working with named cells, commands for formula auditing, and calculation options.

The Data tab offers commands for getting external data; managing the connections to external links; sorting and filtering your data; removing duplicates, validating and consolidating your data, and grouping and ungrouping cells.

The Review tab includes what you need to proof, comment on and share and protect the sheet.

The View tab provides commands for choosing different workbook \views. Hiding and redisplaying worksheet elements (gridlines, the ruler, the formula bar and more), magnifying or reducing the display, and working with the worksheet window.

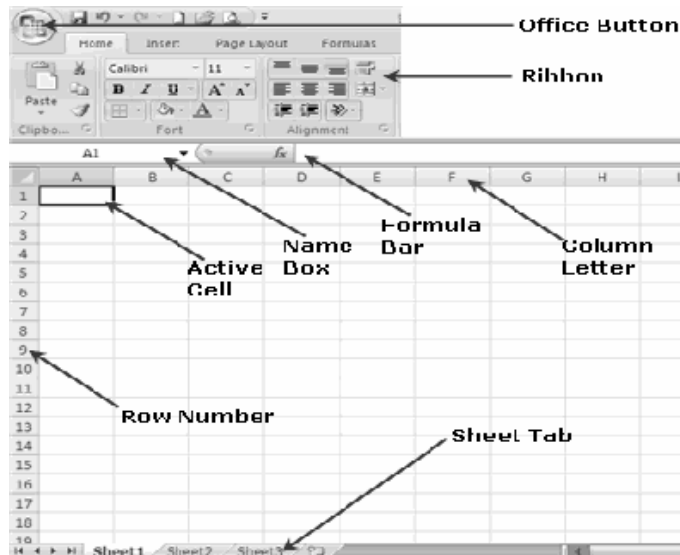
Tip: To find out more about the new elements in the Microsoft Office system user interface—including the new command sets, contextual tools live preview, galleries, and more—refer to Chapter 2 “A New Look.”

Page Layout: A New View for Better Printing

Printing worksheets—getting the margins just right and including everything you want on the page—can be a bit of a challenge (and time-consuming!) for both new and experienced Office Excel users. Office Excel 2010 includes a new view, called Page Layout new which gives you the ability to see how your worksheet is shaping up while you’re working on it. Display the worksheet in Page Layout view by clicking the View tab and selecting Page Layout View.

Page layout view makes printing easier by taking away the guesswork. Rulers along the top and left side of the work area enable you to plan for specific content areas on the page. Because everything in Page layout view is fully editable, you can make data changes, select new for-mattings options, and save and print in this view. You can also view multiple pages in Page Layout view by reducing the size of the pages using the Zoom control.

Tip: Make changes that relate to the appearance of the overall document—for example, adjusting margins, changing orientation, or filtering data—while you’re working in Page Layout view. This enables you to get the best “big picture” sense of how the changes will affect your worksheet when printed.



Page Layout view is also available in the view controls in the lower-right corner of the Office Excel 2010 window. The three views displayed in the control (to the left of the Zoom control) are Normal view, Page layout view and Page Break preview.

Tip: The Custom Views feature in Office Excel 2010 is now available in the Page Layout tab of the user interface. Set the display features the way you want them click Custom Views, and then click Add to create a view you can apply to other worksheets as well.

More Room, More Speed, More Choices

Some of the changes in Office Excel 2010 will be most important for power users who create, update, modify, and report on mega worksheets that demand the utmost in processing power and speed. Here’s an overview of some of the changes that affect the way Office Excel 2010 manages and supports large, complex worksheets.

If you were mixing out the row and column limits of Office Excel 2003. You’ll be glad to know that the Office Excel 2010 worksheet has been greatly enlarged According to David Gainer, Group Program Manager for Microsoft Office Excel. “Specifically, the Excel 12 grid will be 1,048,576 rows by 16,384 columns. That’s 1.500 percent more rows and 6, 30 percent more columns than in Excel 2003. And for those of you that are curious, columns now end at XFD instead of IV.”¹

And even on huge, formula-intensive worksheets you will experience faster calculations because Office Excel 2010 now supports dual processors and multithreaded chipsets, which enable Office Excel 2010 to take full advantage of your system's processing power and dramatically reduce your wait time. Another speed booster: The memory management in Office Excel 2010 has been increased from 1 GB to 2 GB, giving you more room to work with complex, sophisticated spreadsheets that previously mixed out your system resources. Additionally Office Excel 2010 moves the bar on some of the limits in the earlier version; for example, now you can set unlimited formatting in the same workbook (this used to be capped at 4,000 format types). The color palette of Office Excel 2010 has grown to support the full 16 million colors, and now you can create as many cell references to a single cell as your computer's memory will allow (this used to be limited to 8,000 references per cell).

Create Better Spreadsheets, Faster

Most professionally created worksheets today—if there are shared with an audience out of house—include at least a little bit of design. The titles are formatted to stand out; rules or shading might be used to call attention to special areas of interest. Worksheets, when they are included in financial reports, business plans, or presentations, often are given the same look and feel as the surrounding material, so the font, color, and arrangement of the information become important. Office Excel 2010 makes it easier to create better looking documents by providing a number of easy-to-apply formatting options. And if you're working in Page layout view, you can see how those design changes look when you print them.

Easier Access to New Templates

When you start a new worksheet in Office Excel 2010 by choosing New from the File menu, the New Workbook window opens, immediately offering you a list of template categories from which you can choose. This greatly simplifies connecting to Microsoft Office Online (which Office Excel 2010 does automatically when you choose a template type) and searching for the template categories you might want to try. Click the template category you want to see (for example, Business), and if the category includes subcategories, click the one you want (such as Accounting). Office Excel 2010 connects to Microsoft Office Online and displays a collection of templates available in that category. The selected template appears in a preview pane on the right side of the window. If that's the template you want, click Download to download it to your system.

Three Things to Try

Mark Dodge and Craig Stinson authors of *Microsoft Office Excel 2010 Inside Out*, recommend these as their favorite new features:

1. Use Page Layout view to see how your worksheet will be distributed between pages, to switch quickly between portrait and landscape modes, and to enter headers and footers directly on the worksheet.
2. Use Table Styles in conjunction with Themes to give your workbooks a consistent, professional appearance.
3. Use the new conditional formatting features to highlight dates that meet dynamic conditions, such as yesterday, today, last week, next week, or next month.

Choosing Themes and Setting Cell Styles

The way in which you can find, try out and apply cell styles and themes in Office Excel 2010 is a huge improvement over the text-based, buried-in-the-menu options available in Office Excel 2003. Now you can highlight an area of your worksheet; click Themes in the Page layout tab of the user interface; and sample a gallery of theme styles that change the color, font, and spacing of the selected cells.

Tip: As you can see choosing a different theme changes a collection of format elements. Several characteristics are included in each theme: the colors used the fonts, and the effects. The three tools to the right of the Themes command enable you to set each one of these items separately, if you prefer Setting the format of a cell used to be something that required up-front thought and effort; now you can apply predesigned formats to cells and ranges by simply clicking the Cell Styles command and choosing the format from the gallery that appears. You'll find the Cell Styles command in the Home tab; just click the command to view and select an available format.

Tip: You can create formats for your own cell styles and add them to the gallery. Start by applying the format you want to a specific cell then click Cell Styles in the Sheet command tab and choose New Cell Style in the Cell Styles gallery. Review the information in the Style dialog box and click Format if you need to make any changes type a name for the style in the Style Name field and then click OK to save the style. The new style you created appears at the top of the gallery in the Custom category.

Click-and-Type Headers and Footers

If you ever had a problem trying to get headers and footers to print correctly on your work sheets in the past, you will appreciate the simplified way of adding and editing headers and footer in your Office Excel 2010 worksheets. Now you can simply click the Insert command tab and click Header & Footer in the Text command set. The worksheet is displayed in Page layout view, and the user interface changes to offer a collection of header and footer tools. A header box opens on the worksheet; you can simply click and

type your header and use the elements shown to add items you need; for example, the page number, date, time, and worksheet name.

Tip: The Auto Header and Auto Footer commands on the left side of the header and footer tools give you a collection of ready-made headers and footers you can apply to your worksheet. Simply click the command and click the selection to add it automatically to your worksheet.

The Benefits of Microsoft Office Open XML

The applications in the Microsoft Office system now use Microsoft Office Open XML format as the default file format. Open XML offers users several major benefits that relate directly to issues that Office Excel 2010 users care about:

- % Open XML enables you to save huge worksheet files by using just a fraction of the space required by the previous format required.

Open XML saves data independent of the format or schema, used to display the data in a particular way. This means the content can be preserved and used-independent of its particular form-in an almost unlimited number of ways. The information you create and share in your Office Excel 2010 worksheet, chart, or report can be incorporated in other worksheets or documents you create at a later time, saving you time and reducing the margin for error involved in rekeying important data.

Major Charting Enhancements

Charts provide you with a way to communicate-visually and quickly—the numeric story your worksheet is telling. Charts help others understand how you are interpreting your data, enabling you to show trends and comparisons quickly and colorfully. Office Excel 2010 includes a huge array of chart improvements with galleries of predesigned formats you can apply instantly, great new 3-D options, and a full set of contextual tools that you can use to communicate your message in just the right way. Begin the process by selecting the data range(s) you want to chart. Then click the insert tab and choose the chart type you want to create.

The chart appears on your worksheet, and the Chart Tools contextual tab offers three full sets of options for customizing your charts:

The Design tab gives you choices for selecting the chart type, data source and arrangement. Quick Styles and the Move Chart command.

The Layout tab in Chart Tools enables you to enter chart properties, choose Office Shapes, add or edit chart elements, and make choices related to 3-D charts.

The format lab provides you the means to select different chart elements, add styles to the chart shape, including 3-D edges, shadows, bevel and more.

New Office Shapes and WordArt

Office Shapes-the lines, rectangles, block arrows, and more-that used to be buried in the Drawing toolbar now have their own space on the Insert toolbar. To see the expanded collection of shapes, click the Shapes tools you'll find more choices and new categories, including new Equation Shapes. WordArt also has been improved and made more accessible, now housed in the Text command set of the Insert menu. When you choose the WordArt command, a gallery of styles appears. Click the one you want and the WordArt item is placed on your worksheet; click it to replace it with your own text

Tip: Office Excel 2010 includes the Smart Art option for those times when you want to add a sophisticated diagram to your worksheet. To create Smart Art graphics, choose the Insert tab. click Smart Art In the Illustrations command tab. select the diagram type you want to create choose the style you prefer, and click OK.

Conditional Formatting and Data Visualizations

Conditional formatting is a great feature that enables you to apply specific formatting to cells according to the value of a cell or the value of a formula. This feature helps you to easily point out certain values or trends in your worksheet-which enables others to grasp what you want them to see. Some conditional formatting was available in Office Excel 2003, but in Office Excel 2010 it's easier to find and use and you can set up conditional formatting without writing formulas at all.

Begin by selecting the range of cells to which you will apply the conditional formatting. Then in the Sheet tab, click Conditional formatting. The menu that appears offers two different sets of rules—Highlight Cell Rules and Top/Bottom Rules—that you can apply to your data simply by choosing the rule you want to apply. For example to find quickly the top 15 percent of the values in your selected range, choose Top/Bottom Rules and click Top N In the window that appears, increase the percentage to 15 and choose the condition you want to apply (the default is Red Fill With Dark Red Text. The change is previewed in the worksheet as soon as you make the selection; click OK to save the change In addition to these predesigned rules.

Office Excel 2010 includes new data visualizations that help you understand and illustrates trends and comparisons in your data here are the three new data visualizations:

Data bars show you immediately, in the form of value bars, how the values of selected cells compare to each other and to the whole.

Color scales apply coloring schemes you select to a specific range of values so that individual cells display in a particular color based on their value.

Icon sets provide you with a set of individual characters you can apply within the cell to show trends in a variety of ways.

Creating Conditional Formatting Rules

You can design your own rules to tailor the visualizations to your own worksheets. When you choose New Rule from the Conditional Formatting menu, the New Formatting Rule dialog box opens. (If you have applied a conditional formatting rule to the selected range previously, the settings for that rule are displayed by default.) Choose a rule type (for example, format Only Cells That Contain) and edit the rule description so that the format you want displays when the new rule is applied. Click OK to save and apply the new rule to the selected cells.

Services

Excel Services is a new feature in Office Excel 2010 that enables you to save your workbook to a server that is running Microsoft Office SharePoint Server 2010 (and also supports Office Excel Web Access).

Here are some of the biggest benefits of using Office Excel 2010 with Excel Services: •

- Share and manage collaborative worksheets more securely. Display only those worksheet areas users have permissions to see; hide confidential data, formulas, and macros

- Create, sort, filter, and modify PivotTable views using any Web browser.
- To set permissions for the worksheet to enable the worksheets or ranges you want to make available to the users **you** specify, open the File menu, point to Save As, and click Excel Services.

The Save As dialog box provides an Excel Services Options button that enables you to control which sheets are visible in the user's browser. After **you** make your selections and click OK. Enter the name of the shared folder in the Save In field and click Save. The file is stored on Office SharePoint Server 2010 and your team members will be able to access the worksheets via the Web.

Quick Facts about Office SharePoint Server 2010 and Office Excel 2010

You can save sensitive or critical spreadsheets to Office SharePoint Server 2010 and indicate permissions for which elements (specific worksheets, tables, charts, ranges, and so on) will be available to other team members.

Office SharePoint Server 2010 includes the Report Center to help users create and work with “trusted” Data Connections so they can safely connect with external data sources. Use Office SharePoint Server 2010 too centrally store important spreadsheets to control the distribution of multiple versions and ensure that business members and partners are working with the most recent files.

Pivot Tables Views—More Support, Better Insight

If part of your work involves making sense of large amounts of data and displaying it in a format that is both flexible and understandable for others, you will be relieved that Office Excel 2010 makes PivotTables easier to understand and use. Sometimes the simple things make a great difference—such as adding an Undo function for most operations, using the familiar plus and minus indicators for drilling down, and allowing sorting and filtering of data.

More PivotTable Support Begin the process of creating a PivotTable by selecting the range of data you want to use in your report; then click PivotTable on the Insert command tab. The Create PivotTable dialog box will ask you to confirm the range you selected and choose whether to create the PivotTable report in a new worksheet or on the existing worksheet. Make your choice and click OK.

The PivotTable Field List is displayed in a panel along the right side of the work area, and the PivotTable Tools contextual tab appears above the user interface. The Design tab enables you to choose the layout of the PivotTable and add Quick Style features such as column and row headers, and choose a design from the PivotTable Quick Styles gallery.

Improved Pivot Charts

When you are happy with the PivotTable you created, you can put those numbers in a visual format by creating a PivotChart to illustrate your data. The expanded features for Pivot Charts parallel those found in the charting enhancements throughout Office Excel 2010; when you create a Pivot Chart both the Chart Tools and Pivot Chart Tools are at your disposal, giving you a huge range of choices for analyzing, displaying, sorting, editing, and saving your information.

The Building Blocks: Values, Cell References, & Operators. The building blocks of Excel formulas are values, cell references, and operators.

A **value** can take the form of a number (20, 100, .015), text (“The Fuqua School of Business”, “Duke basketball”), or a date (4/1/01, April 1, 2001).

A **cell reference** identifies a single cell or a range of cells on a worksheet and tells Excel where to look for the values or data you want to use in a formula.

With references, you can use data contained in different parts of a worksheet in a single formula or use the value from one cell in several formulas. You can also refer to cells on other sheets in the same workbook, to other workbooks, and to data in other programs. References to cells in other workbooks are called external references. References to data in other programs are called remote references. Excel has two cell reference styles: The A1 style and the R1C1 reference style. By default, Excel uses the A1 reference style. This style refers to columns with letters (A through IV, for a total of 256 columns) and to rows with numbers (1 through 65,536).

These letters and numbers are called row and column headings. To refer to a cell, enter the column letter followed by the row number. For example, D50 refers to the cell at the intersection of column D and row 50. To refer to a range of cells, enter the reference for the cell in the upper-left corner of the range, a colon (:), and then the reference to the cell in the lower-right corner of the range.

Examples of A1-style references:

To refer to Use

The range of cells in Column A and rows 10-20 A10:A20

The range of cells in Columns A-E and rows 1-20 A1:E20

All the cells in row 5 5:5

All the cells in rows 5 through 10 5:10

All the cells in Column A A:A

All the cells in Columns B through H B:H

To experiment with any reference types that may be unfamiliar to you, click the F5 key to open Excel's "Go To" dialog box and enter a reference in the "Reference" box to see the cell/range selection in your spreadsheet. With the R1C1 reference style, both the rows and the columns on a worksheet are numbered. The R1C1 style is particularly useful for computing row and column positions in macros. In the R1C1 style, Excel indicates the location of a cell with an "R" followed by a row number and a "C" followed by a column number. Toggle on or off the R1C1 reference style by choosing *Tools, Options* and selecting the "General" tab. 3 Excel's formula **operators** specify the type of calculation to perform on the elements of a formula. Excel has four different types of calculation operators: arithmetic, comparison, text, and reference.

Calculating With Date & Time Values

Excel stores dates as sequential numbers known as serial values. Excel stores times as decimal fractions because time is considered a portion of a day. Because Excel treats

dates and times as values they can be added, subtracted, and included in other calculations. For example, to determine the difference between two dates, subtract one date from the other. View a date as a serial value and a time as a decimal fraction by changing the format of the cell that contains the date or time to **General** (number) format. For example:

Date Number

4/10/01 36991

3:15 0.1354

For a date calculation, either refer in the formula to cells that contain dates, or -- to include actual dates in the formula – surround them with double quotes. For example, this formula:

=“4/10/01” – “3/10/01” yields the value **31**.

Excel supports two date systems: the 1900 and 1904 date systems. The default date system for Excel is 1900. If you have a need to the 1904 date system, choose *Tools, Options*, select the “Calculation” tab and toggle on the 1904 date system check box. The table below shows the first date and the last date for each date system and the serial value of each.

1900 January 1, 1900 (serial value 1)

December 31, 9999(serial value 2958465)

1904 January 2, 1904(serial value 1)

December 31, 9999(serial value 2957003)

Use the DATE function to manipulate a part of a date within a formula. Use the TIME function to manipulate a part of a time— such as the hour or minute— within a formula. See the “Date & Time” category of Excel’s built-in functions for more options.

By default, dates entered in a workbook are formatted to display two-digit years.

Use the Windows Control Panel to change the default date format if you want to display four-digit years instead of two-digit years.

Naming Cells & Ranges Used in Formulas

Excel will use names in place of cell references in formulas. Naming key cells can make your worksheet formulas easier to read. In the example below, the formula bar displays the formula in Cell E3 as =TotalSales*TaxRate. Without cell names, the formula would display as the less comprehensible =D3*B1. If you assign names to cells *after* having used those cells in calculations, Excel doesn’t automatically replace cell references in the calculations with the new cell names. However, you can over-ride this default and have new names used post formula- creation with the command *Insert, Name, Apply*.

Quickly name a cell or range by selecting it and entering the name in the “Name Box” at the left of the formula bar. The Name Box displays the current cell or range reference or the current cell or range name, if one is assigned. Name a cell or range to be able to go to it quickly using the F5 “Go To” key. Hitting F5 opens the *Go To* dialog box.

14 Range names for an Excel list (like the simple one below) can be used to simplify references and in calculations.

In this example, select the list (the range B3:C7 in the illustration below) and use *Insert, Name, Create* to open Excel's "Create Names" dialog. Excel guesses where the likely names are in relation to the data. In this case, the data is in C4:C7 and the names come from the top row and left column. With names established, you can use a "shorthand" reference to refer to the numeric cells. Cell C4 can also be called **January A** since it falls at the intersection of those row and column names. In like manner, Cell C5 can also be referred to as **January B**. Cell E5 contains the formula **=January A** and displays the value 10. Cell E6 contains the formula **= January A + January B** and displays the value 25. There's another way to name a range whose values you intend to use in a calculation: By using the special "Label Ranges" dialog. In this example, imagine you have a list of values like the one at right, with the label "Year99" at the top of the column. Select the range (including the label) and choose *Insert, Name, Label* to open the "Label Ranges" dialog. Excel fills in the label range.

Specify that you're using
"Column labels" and click *OK*.

Use this method to specify ranges that contain column and row labels on your worksheet.

Year99

10

30

50

50

15

You can now use the "label range name" in calculations.

For example, this formula:

=SUM(Year99) returns the value 140.

The formula **=SUM(Year99, 100)** returns the value 240.

As a quicker alternative to using the *Insert, Name, Label* command, you can select a range (not including its header) and use the Name Box to assign a range name. Then use that range name in calculations, as above. A named cell (or range) is treated as an absolute reference in a formula. If you assign a cell or range a name and then want to change it, you can. But Excel offers no simple method. The only way to change a range name is to use *Insert, Name, Define*, add the new name, and then delete the old name.

Understanding Relative, Absolute, & Mixed Addressing

You may often want to use the same formula in one or more places in a worksheet. It's most efficient to build the formula one time and then copy it to the other locations

where you need it. If you copy a formula, you must understand Excel's relative and absolute addressing. When you copy a formula Excel uses relative addressing as the default. That is, when you copy the formula to a new location Excel adjusts the cell references in the copied formula relative to its new location. In the illustration at left below, the formula in Cell D4 is used to calculate the total sales in January and February for Dept 1: **=(B4+C4)**. We need the same basic formula in Cells D5 and D6 for Departments 2 and 3. Although we could enter two new formulas, it's easier to copy the formula in D4 to these locations. Point with the mouse to the "fill box" at the lower righthand corner of Cell D4 and drag down through Cells D5 and D6 to copy the formula. For each copied formula, Excel automatically adjusted the cell references to refer to the data in the correct row. This is Excel's default mode when copying formulas. It's **relative addressing**. 20

There may be occasions, however, when you want to override Excel's relative addressing default. In the example below, the formula in Cell E4 (in the Tax column) multiplies the Dept 1 Total Sales value (Cell D4) and the 5% value (Cell B1). The result is correct for this row. However, if we copy the formula in E4 down to E5 and E6 for Depts 2 and 3, the result for those rows is *not* correct. In the absence of any other instruction, Excel has again used relative addressing in the formula copy. The formulas in Cells E5 and E6 read as follows:

=D5*B2

=D6*B3

The first reference in each of these formulas is OK. We *do* want the copied formulas to refer to the Total Sales values for their respective rows (Cells D5 and D6). The problem is with the reference to Tax in Cell B1. The master formula in Cell E4 correctly reads **=D4*B1**. But the copied formula in E5 refers to Cell B2 (an empty cell), resulting in a calculation of \$0.00. And the copied formula in E6 refers to Cell B3 (which holds the text value "Jan"), resulting in a #VALUE error statement. To make sure the copied formulas in this case refer to the correct Tax cell, use **absolute addressing** for the reference to Tax. Absolute addressing in a formula fixes the reference to a cell no matter where the formula is copied. Absolute addressing can apply to a single cell, a range of cells, or to one, several, all, or none of the cells referred to in a formula.

Specify absolute addressing with a dollar sign (an arbitrarily selected symbol) before the row reference and the column reference. For example, a relative reference in a formula to the Tax value in Cell B1 is written as **B1**. An absolute reference in a formula to the Tax value is written as **\$B\$1**. (It's also possible to "fix" just the row but not the column by writing **B\$1** or just the column and not the row by writing **\$B1**. This is called **mixed addressing**.)

Here we specified absolute addressing to refer to the Tax cell in the "master" formula in

E4. Copying the master formula to E5 and E6 results in correct values for those formulas.

Tip: When positioned over a cell or range reference on the formula bar, the F4 key cycles through the four addressing options: completely relative, completely absolute, absolute row but not column, and absolute column but not row. That is:

B1, \$B\$1, B\$1, and \$B1.

Finding Formulas

Excel's *Go To* feature provides a quick way to identify all the calculations in your worksheet. Click *Edit*, *Go To* or hit F5 to open the *Go To* dialog. Click the *Special* button on the *Go To* dialog to open the "Go To Special" dialog. In the "Go To Special" dialog, choose "Formulas" and the type of data you want to select. Then click *OK*. Excel highlights all the formulas of the type you selected. Using the *Go To, Special, "Errors"* option to find incorrect formulas can be useful when debugging a spreadsheet.

Data Form

When a spreadsheet is designed for data the menu item Data: Data Form can be used for data entry. Select Data Form from the Data menu. Click on the New button in the upper right corner. Enter the data shown below. Use the Tab key to move to the next field blank. Do NOT use the down arrow key: it will cause the Data form to "go blank." The reason for this is because the down arrow generates a next New record.

Data	
Course:	MS 152
Sect:	M15
Last:	Seldon
First:	Harry
Sx:	M
St:	Y
HS:	YHS
Elem:	Tamil Elem
Lang:	Yapese
T1:	79
T2:	81
T3:	92

New Record

New

Delete

Restore

Find Prev

Find Next

Criteria

Close

Use of the tab key to move from one field to the next has been a standard in the data entry industry for over 30 years. The Tab key will work in Excel, Microsoft Access, and fill-in-the-blank fields found on the Internet and all other data entry applications.

The Choose function

The choose function selects the Nth item from a list. The syntax for choose is =choose(n,firstItem,secondItem,thirdItem,fourthItem,...) where n must be a cell address that contains an integer. The Nth item is chosen by the integer. Click in the cell P2. Enter =CHOOSE(O2+1,"F","D","C","B","A")

Note the O2+1: there is no zeroth item allowed in choose. The O2+1 converts a GP of 0 to 1 (there is no zeroth element of a list in Excel. Special note for users of Microsoft Works only: In a single encounter with Microsoft Works 4.0 there appears to be a zeroth element in the choose function.)

Fill down the grade column. The worksheet should now look something like:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Course	Section	Last	First	Sx	St	HS	Elem	Lang	T1	T2	T3	Avg	Pass	GP	Grade
2	MS101	M08	Albert	Abena	F	K	KHS	Lelu	Kosraen	83	82	81	82	Promote	3	B
3	MS101	M08	Aldis	Adjoa	F	P	PICS	PCS	Pohnpeian	77	77	77	77	Promote	2	C
4	MS101	M08	Elidok	Ama	F	C	CHS	Puluwat	Puluwatese	94	92	90	92	Promote	4	A
5	MS101	M08	Georgie	Kweku	M	K	KHS	Malem	Kosraen	58	59	60	59	Repeat	0	F

Charting

Note that Office 97 does things a little differently in a slightly different order with dialog boxes that, unfortunately, look very different. Design for charting is like designing for data. Include a field name row at the top of the data. Make the leftmost ("first") column a column of labels. The **Dist** worksheet is set up this way already.

Grade	GP	Count
A	4	5
B	3	7

C	2	8
D	1	6
F	0	3

Column chart

What many call a "bar chart" Excel calls a "Column chart" or column graph.

To make a column chart, select the cells A1 to A6. Hold down the Control (Ctrl) key and select the cell C1 to C6. The control key allows the selection of non-adjacent data.

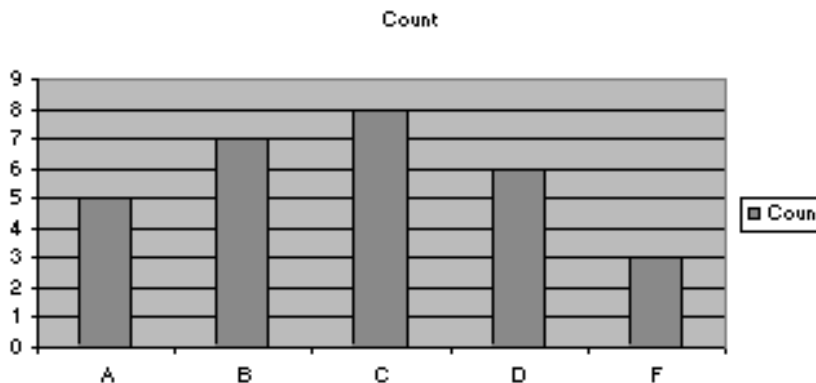
Always include all label rows and columns in the selection, this will cause automatic labeling of the chart axes.

Click on Next to leave step 1.

Click on Column chart option in the middle of the top row of step 2 and then on Next.

Click on format option 1 in the upper left corner of step 3 for a basic column chart and then on Next.

The dialog box for step 4 is the first of two options dialog boxes that are important. This is where one can specify whether the data is in rows or columns. If one has more than one label row or column, this is the place to specify multiple label rows or columns. We have only one label column if we used the Control key properly. Excel should have chosen the following set-up:



In step 5 options dialog box select No under "Add a legend?" For a single data set a legend is unnecessary. If desired, type in a chart title, x-axis and y-axis label. Click on finish

Other types of charts: Population Pyramid

Age	Female	Male
0-4	-7117	7545
5-9	-7158	7932

10-14	-7288	7656
15-19	-5893	6532
20-24	-4703	4489
25-29	-3528	3514
30-34	-3407	3393
35-39	-2949	3049
40-44	-2456	2975
45-49	-1660	1941
50-54	-1160	1111
55-59	-1091	998
60-64	-965	1013
65-69	-669	639
70-74	-627	542
75-79	-281	263
80-84	-176	137
85-89	-58	41
90-94	-30	26
95-99	-8	4

There are many types of charts available. The math teacher is likely to find the x-y scatter diagrams useful for graphing functions. Put the x values in the first column and the y-values or a function in the second column.

Excel will, with a little thought, make specialty graphs that are not immediately evident from an inspection of the chart wizard. In attempt to stretch thinking, the following will produce a population pyramid type chart.

Select the data including the field names row and the labels column. Note the use of negative values for the first column.

Click on the Chart wizard button in the tool bar.

Drag the mouse to set the location of the chart on the worksheet.

Click on Next.

Click on Bar chart option and then on Next.

Click on option 8 for then on Next.

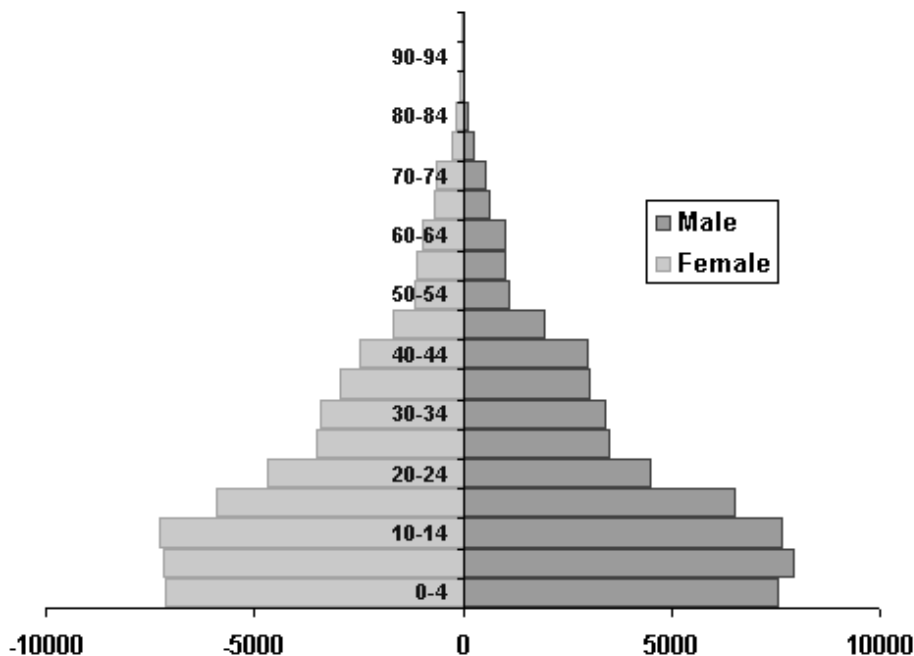
This first of two options dialog boxes is important. This is where one can specify whether the data is in rows or columns. If one has more than one label row, this is the place to specify multiple label rows. It is likely that no adjustments need to be done for this graph: Excel 95 will likely "guess" correctly based on the existence of a field name row and the label column. Click on Next when done.

In this second options dialog box select Yes under "Add a legend?" The wizard may already have selected yes. For two or more data columns a legend is informative. A field name row is necessary for Excel to correctly set up the legend. Type in a chart titled, x-axis, and y-axis labels. Click on finish.

Click on the Options button in the Chart Type dialog box.

Click on the Options tab at the top of the Format Bar Group dialog box.

Set the Overlap to 100 and the gap width to 0. Click on OK. Click outside the area of the graph to turn off the hash mark border. This also takes Excel out of the graphing mode and into the regular mode.



Intermediate Excel: Subtotals

Using Excel to generate subtotals by course and section

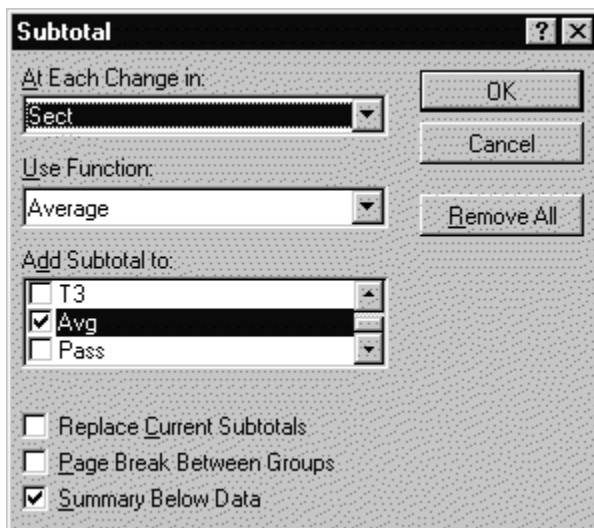
Suppose we wanted to know whether our course averages differed from one section to the next or wanted to know the relative performance of the students by state or

gender. Excel can sort and summarize such data using subtotals. In order for Excel to generate subtotals the data must be sorted by the criterion to be summarized.

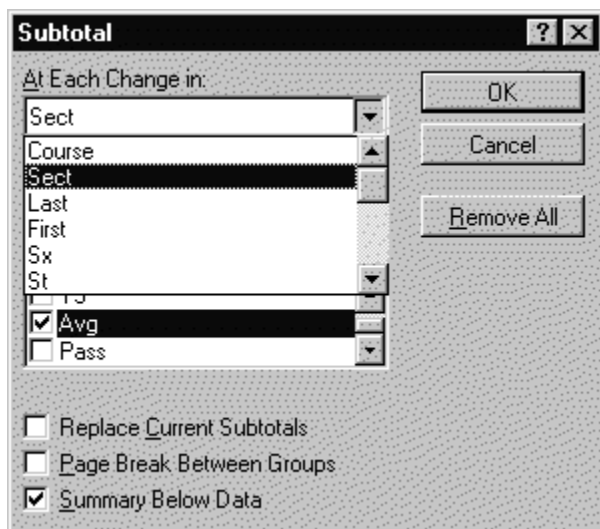
Click in cell A1 of the Grades sheet of the Gradebook workbook. Do not select any cells: sort will make the presumption that only the selected cells are to be sorted and will mix up the data. Choose Sort from the Data menu. Excel should have detected the field names and the radio button "My List has... Header Row" should already have been selected by Excel. Good initial design for data is important to proper sorting and subtotaling. Sort by Course and Then By Sect and Then By Last (name). Click on OK.

Set up the subtotals

Choose Subtotals from the Data menu in Excel.



Set at "At Each Change in" to **Sect** by clicking on the downward pointing triangle on the right side of the "At Each Change in" list box.



Set the "Use Function" list box to Average.

In the "Add Subtotal to:" list click on the check boxes next to T1, T2, T3, Avg, and GP. Scroll the list box to see T1, T2, T3, Avg, and GP. Click on OK.

Note the change in the worksheet: a new panel has appeared on the left, the subtotals control panel. There are three levels shown at the top by three numbered buttons. Click on button number 1 to see only the Grand Average.

A1			Course		
1	2	3	A	B	C
1	Course	Sect	Last		
2	MS 101	M08	Albert		
3	MS 101	M08	Aldis		
4	MS 101	M08	Flirnk		

Click on button number 2 to see the individual section averages. Note the buttons with the plus signs that appear. Clicking on a plus sign button opens up the details for a single section. After clicking on a plus button, a minus button appears. Click on the minus button to collapse the section.

Click on button number 3 to see the list of all students.

Suppose we now wanted to see the overall student averages by state. Changing a subtotals view is a three phase process.

1. Remove the existing subtotals
2. Resort by state (the new criteria)
3. Reapply the subtotals.

The key concept here is that the sort order must mirror the "subtotal at each change in" choice. Subtotals subtotal at each change as Excel moves down the list. If the states are not in alphabetic order, then at each and every change of state from row to row Excel will insert a subtotal. Subtotals on one field cannot be resorted on another without removing all existing subtotals first, hence the first step of removing the subtotals.

- Phase one: Choose Subtotals from the Data menu and then click on the Remove All button.
- Phase two: Choose Sort from the Data menu.
Sort by State and Then By Sect and Then By Last
Click on OK.
- Phase three: Set up the subtotals.
Choose Subtotals from the Data menu in Excel.
Set at "Each Change In" to State by clicking on the downward pointing triangle on the right side of the Each Change In list box. Set the Use Function list box to Average.

In the "Add Subtotal to:" list the check boxes may still be checked from the earlier exercise. If they are not checked, then click on the check boxes next to T1, T2, T3, Avg, and GP. Scroll the list box to see T1, T2, T3, Avg, and GP. Click on OK.

Other subtotal functions

Subtotals functions include sum (the additive total), count (how many items), the average, the maximum value in the subset, the minimum value in the subset, and standard deviations among other specialized functions. The sum and average functions are the most commonly needed functions. On the worksheet tab labeled Inventory is an example of an inventory spreadsheet demonstrating the use of the Sum function to tally up the value of property in the A204 laboratory. Click on the Subtotal control panel numbered buttons to explore the different levels of this spreadsheet.

Filtering

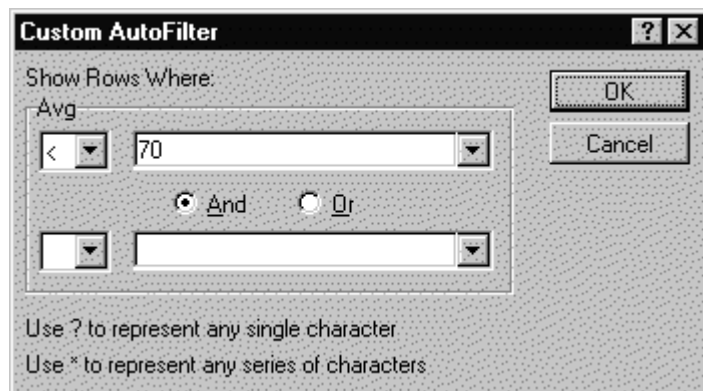
Filtering allows one to display subsets of the data based on specific condition. Suppose we want to display only the students who will be on the deficiency list, those with an average less than 70. Start filtering by removing the subtotals. Choose Subtotals on the Data menu and then click on the "Remove All" button.

Choose Sort on the Data menu and sort by last name and then by first name. Click on OK.

Scroll to the top of the worksheet. From the Data menu choose Filter. On the submenu that pops up choose Autofilter. Small grey buttons with triangles in them should appear in the top row of your spreadsheet.

	M1				Avg			
	I	J	K	L	M	N	O	P
1	Lang				Avg	Pas	C	Grat
2	Kosraen	83	82	81	82	Pass	3	B
3	Pohnpeian	77	77	77	77	Pass	2	C
4	Puluwatese	94	92	90	92	Pass	4	A
5	Kosraen	58	59	60	59	Repeat	0	F
6	Mortlockese	65	63	61	63	Repeat	1	D
7	Kosraen	58	57	56	57	Repeat	0	F

Click on the grey button in the M (Avg) column to see a "drop-down" list of options. This list includes preset values one can select to filter the data. Choose (Custom...), the third item down the list. The following dialog box should appear:



Use the little "down triangle" button to the right of the equals sign in the first blank under the word "Avg" to choose a less than < sign.

Either select 70 from the drop down list in the next box or type 70 in the next blank box as seen above. Click on OK.

The result is a list of students who are deficient (below 70). If one chooses print with a filter on then only the names shown will print.

Getting back all of your data: Choose Filter again from the Data menu and this time select "Show All" on the submenu.

Filters are another powerful use of Excel provided that your spreadsheet is set up in database format (field names in row one and no blank rows, one physical item per row in rows two and higher). Assets larger or smaller than a given number could be displayed, or students with a TOEFLs above or below a specific point could be shown. The "Top ten" filter option can display just that, the "top ten" in a category.

Pivot tables

Using Excel to set up a pivot table to study course versus gender grade averages. This section presumes that the worksheet named **Dist** has already been inserted in the workbook during work on the frequency function.

Pivot tables are easiest to set up when the spreadsheet is designed for data. Field names should be in row one of the table, with data below. There should be no blank rows within the data.

Pivot tables (known in Microsoft Access as a cross-tab table) groups data by two categories, producing summary information such as average, sum, or count according to two or more categorizations. As a part of this process, pivot tables have the ability to take categories in data rows and turn them into field names. The result feels like a "rotation" of the data, hence the name Pivot table. As an example, the data on the left below is pivoted to produce the result on the right.

	A	B	C	D	E	F	G	H
1	State	Sex	T3		Average of T3	Sex		
2	Pohnpei	F	80		State	F	M	Grand Total
3	Pohnpei	M	65		Chuuk	82.5	67.5	75
4	Pohnpei	F	70		Pohnpei	75	70	72.5
5	Pohnpei	M	75		Grand Total	78.75	68.75	73.75
6	Chuuk	F	95					
7	Chuuk	M	60					
8	Chuuk	F	70					
9	Chuuk	M	75					

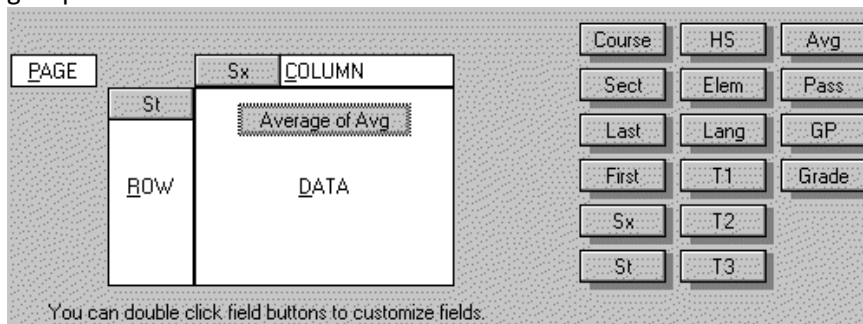
The result are averages based on state and sex. The data in the Sex column, F and M, has become field names in a new row one of a table. The data has, in a sense, been "pivoted" or "rotated" up out of a column and tabulated across as field names in a new table (hence the use of term "cross-tab" by Microsoft Access).

To start a pivot table in the Gradebook workbook, click anywhere inside the field row or the data rows of the Grades worksheet. Do not select a cell, just have the cursor in a cell inside the data to be pivot tabled. Choose Pivot Table from the Data menu.

Click on Next in the first dialog box. The default Microsoft Excel List or Database is usually selected and is the correct selection.

Click on Next in the second dialog box. If the spreadsheet is designed properly for data then Excel will have correctly detected the data range.

Dialog box three presents the screen where choices are made as to row and column groups.

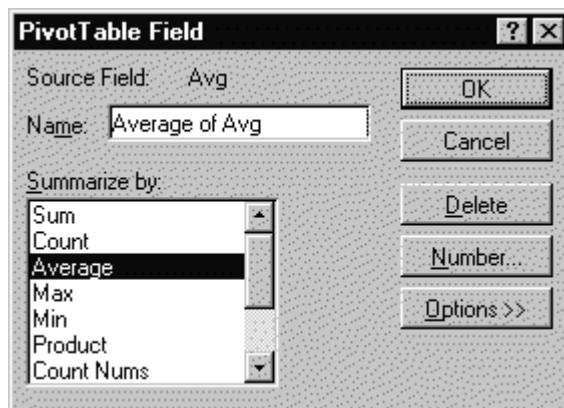


Drag and drop the button marked State (St) from the right hand area of the dialog box to the area marked Row.

Drag and drop the button marked Sex (Sx) from the right hand side to the Column area.

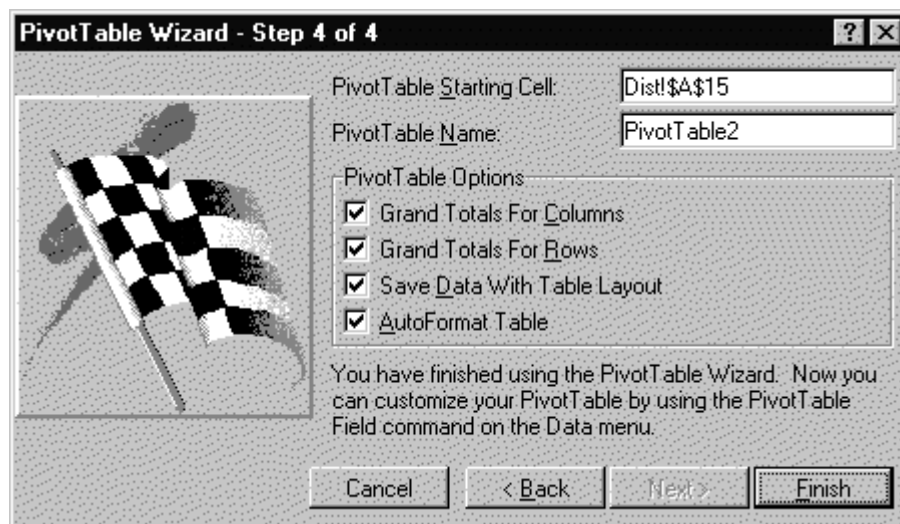
Drag and drop the button marked Avg from the area on the right to the Data area.

Double click on the Avg button in the Data layout area at the center of the screen. From the PivotTable Field pop-up dialog box choose the function average. Note that all of the basic functions are available including sum, average, and count among others. These are the same functions we encountered on the first day. Click on OK.

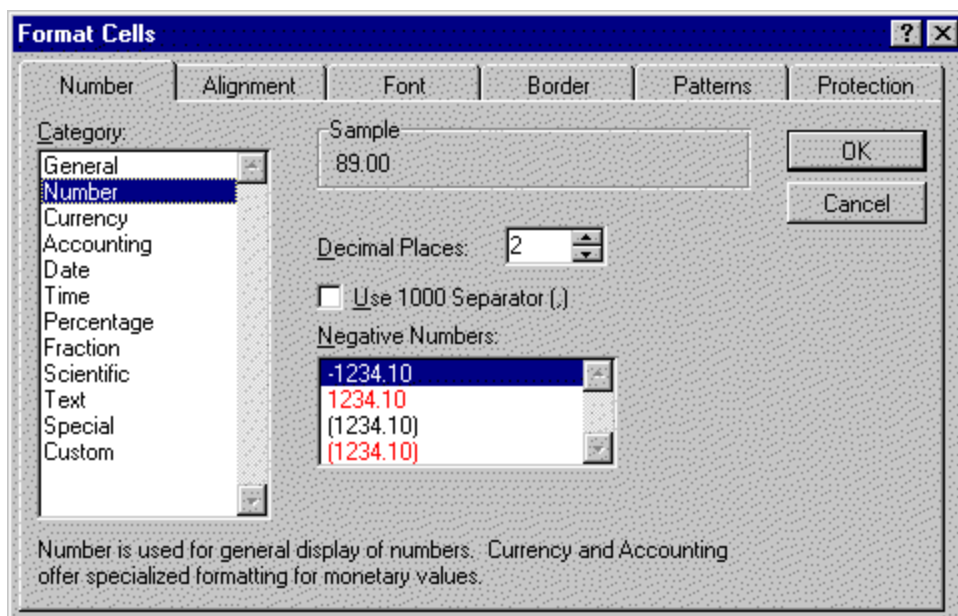


Click on Next.

The next dialog requests a location for the pivot table. Click on the **Dist** worksheet tab (created during the frequency function lesson). Click in A15 (below the chart if one exists in the grade book). Click on Finish.



The data in the resulting PivotTable summarizes averages by state and sex for the fictional College of Micronesia-FSM spreadsheet. To make the data more presentable: Select the data in the pivot table by dragging the mouse across the number data. Choose Cells... on the Format menu.



The dialog box should open to the Number tab. If not, click on the tab marked Number. Click on the item Number in the Category list box.

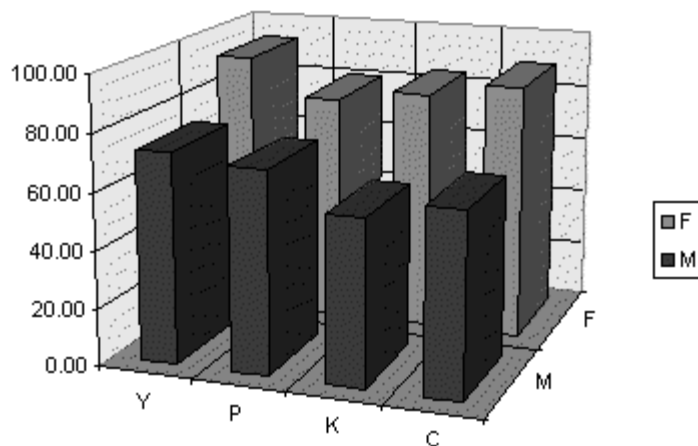
Note the default number of decimal places is preset to 2. This can be changed. For now, click on OK and the numbers in the table will be displayed to two decimal places.

Graphing Pivot Tables

The data in a pivot table can be graphed to produce charts as was covered in the second session. Be careful to select only the state and sex breakdown data and not the grand total data. Look carefully at the diagram below and the note following the diagram.

	C20		73.222222222222	
	A	B	C	D
14				
15	Average of Avg	Sx		
16	St	F	M	Grand Total
17	C	89.00	63.50	80.50
18	K	84.00	57.75	69.00
19	P	80.00	70.21	72.88
20	Y	93.00	73.22	84.52
21	Grand Total	87.14	67.02	76.11

Making the selection shown must be done by dragging from C20 to A16. Dragging from A16 to C20 is not possible as A16 is a button that activates when clicked



Note that in the particular chart shown has been adjusted by rotating the chart 180° from the initial position. This was done by double-clicking on the chart and then choosing 3-D View from the Format menu. In the 3-D View dialog box the rotation was set to 200° (initial rotation was 20°).

Pivot tables are best produced by good data designs. In turn, good data designs are produced by considering the pivot table implications. Good design requires that each data row be a single object or instance, not an aggregate of data. Gradebooks almost naturally generate good data designs: a single student in a single class per row (a "student-seat"). In other applications forethought may be necessary. In a study of lizards, for example, proper design of a good table is likely to involve listing each individual lizard in its own row. This would be as opposed to a table where each row was a location and the data was the number of lizards in that location. The following fictitious table is not well thought out from a pivot table perspective. For example, although the original tallies might have indicated the number of females with tails, the resulting table has lost that information. The design is also prone to typographic errors that result in internally inconsistent data: examine the sum of the number of males and females on trees.

Location	Number of lizards	Number of Females	Number of Males	Num w/ tail
Beach	2	1	1	0
Pond	6	3	3	5
River	8	5	3	4
Rock	7	5	2	7
Tree	10	6	5	9

A better design would have been:

Location	Sex	With Tail
Tree	F	1
Tree	F	1
Tree	F	1
Tree	F	1
Tree	F	1
Tree	M	0
Tree	M	1
Tree	M	1
Tree	M	1
Tree	M	1
Tree	M	1
Pond	F	0
Pond	F	1
Pond	F	1
Pond	M	1
Pond	M	1
Pond	M	1

... and so forth. The resulting table can be subtotaled or pivoted to obtain accurate summary information. Although the original data table will be lengthy, most reports will use the results of subtotal calculations and pivot tables. The table may have typographic errors, but the totals will at least be consistent with the data, there will not be any internally conflicting data. Typos will also not likely affect counts the way they can in the earlier table.

VLookup Function

In Excel, the **VLookup** function searches for value in the left-most column of *table_array* and returns the value in the same row based on the *index_number*.

The syntax for the **VLookup** function is:

VLookup(value, table_array, index_number, not_exact_match)

value is the value to search for in the first column of the *table_array*.

table_array is two or more columns of data that is sorted in ascending order.

index_number is the column number in *table_array* from which the matching value must be returned. The first column is 1.

not_exact_match determines if you are looking for an exact match based on *value*. Enter FALSE to find an exact match. Enter TRUE to find an approximate match, which means that if an exact match is not found, then the VLookup function will look for the next largest value that is less than *value*.

Simple Vlookup Tutorial

Let's assume we are given the following sales table:

	A	B	C	
1	Salesperson	January Sales	February Sales	
2	Jake	45	54	
3	Dan	32	36	
4	Samantha	24	56	
5				
6				
7				

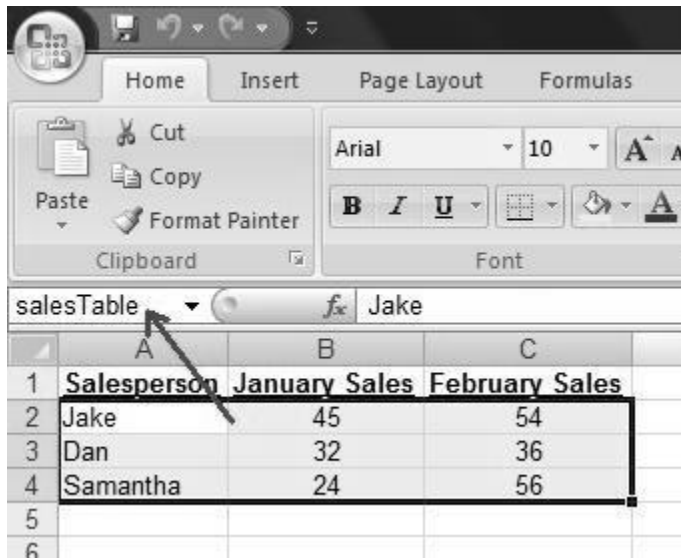
If you wish to retrieve Dan's January sales (hence cell B3), the vlookup function will be:
=vlookup("Dan" , a2:c4 , 2 , false)

Let's go over the different parts of the function in plain English (note the highlighted parts in the following sentence, taken out directly from the formula):

The function will look for the word "Dan" at the first (left) column of the range a2:c4. It will retrieve the value adjacent to it on the 2nd column (thus the "January" column). The word "False" tells the function to find Exactly the word "Dan", and not something close or similar to it.

A way to make the function more readable is to name the table, and use this name in the vlookup function:

Select the table (no need to include the headers row), and write a name inside the name box (its located at the top left of the screen). In our vlookup example, we used the name "salesTable" (Try to avoid using spaces).



Thus, the function can now look like this:

=vlookup("Dan",salesTable,2,false).

And if you don't want to retrieve always Dan's January sales, but to retrieve the name that will be written in cell A6?

=vlookup(A6,salesTable,2,false)

And if you wish the function to retrieve February's sales (instead of January)?

=vlookup(A6,salesTable,3,false)

When and how should you use the Vlookup function?

When you have a table with data, and you wish to retrieve specific information from it. Actually whenever you are looking up a name in a phone book, you are performing a "real" vlookup action: you look up the name of your recipient on the first (most left) column, and when finding it, you retrieve the number appearing next to that name. An implementation of the vlookup in Excel could be: You have an Excel table with student names and their grades. You wish that you could somewhere in the sheet type a student name, and immediately retrieve his grade (based on the data in the table). To achieve this, you can use "Vlookup": the function will look for the student's name in the first column in the table, and will retrieve the information that is next to is name in the second column (which is his grade). Check the video links above to learn how exactly the vlookup function is written and implemented inside the worksheet. Another vlookup example:

You have a big table consisting thousands of bank accounts.

You wish to retrieve in another worksheet information regarding some specific accounts. To achieve this, you can type these specific account numbers, and put a

Vlookup function next each one of them. The function will look for the account numbers in the big table, and retrieve relevant information from it.

The difference between “Exact match” and “Closest match”:

When you use the Vlookup Excel function to retrieve information based on a student name or a bank account number, you cannot allow it to find something close or similar to “Jake”, or close to the account number “3647463”, but rather it has to find them exactly.

But sometimes you have a table that defines ranges, for example:

\$5,000 – “Small deposit”

\$20,000 – “Medium deposit”

\$100,000 – “Big deposit”

\$500,000 – “Huge deposit”

If you want the Vlookup to find the description for a deposit of \$23,000 (which should retrieve “Medium deposit”), you will ask it to find a close match, and it will find \$20,000.

This is very useful when dealing with dates. Look at the following table:

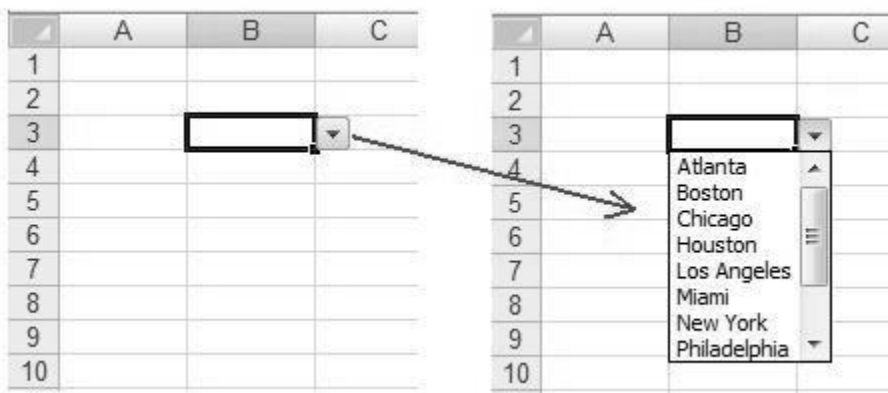
4/1/2008 – Payment on time.

6/1/2008 – Late with payment (small fine).

8/1/2008 – Very late with payment (big fine).

If you would like to find what happens with a payment made on 7/14/2008, the function will relate it to the date 6/1/2008 and retrieve us “Late with payment (small fine)”.

Please note – the function will always retrieve the smaller closest match (in case it doesn’t find an exact match). A drop down list enables you to choose a value from a list, instead of typing it.

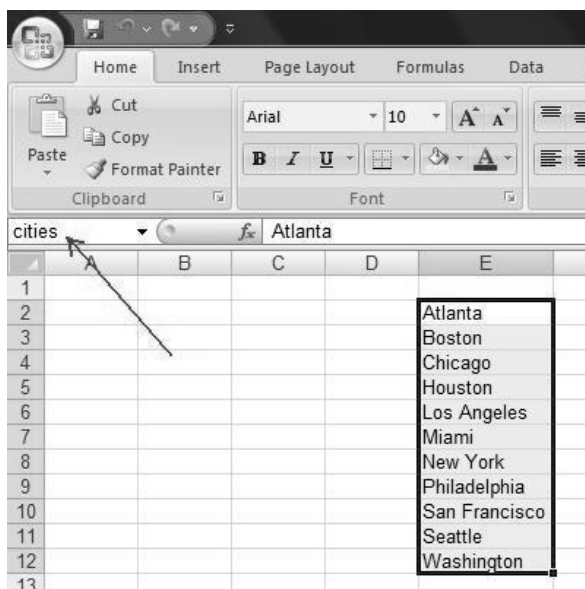


Creating a drop down list in Excel 2010:

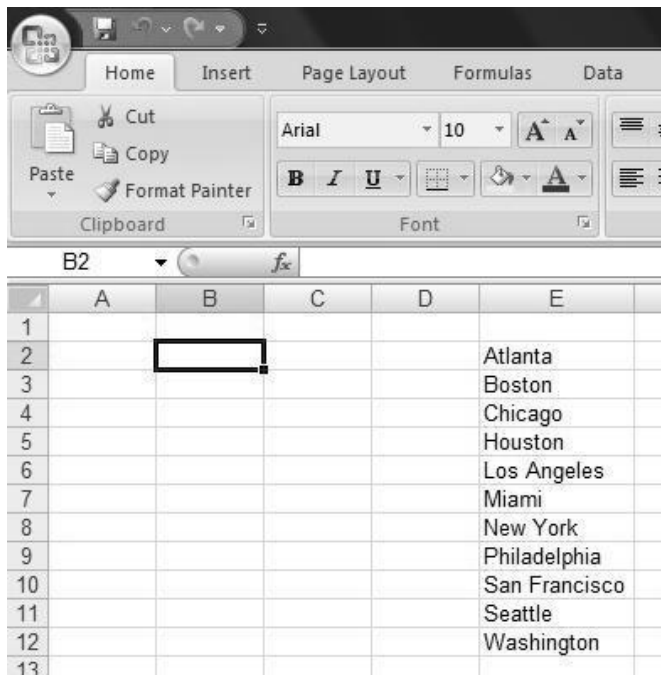
1. Write the list somewhere inside the worksheet. It is recommended to have it sorted alphabetically in ascending order, for this is the order in which they will appear in the drop down list.

	A	B	C	D	E
1					
2					Atlanta
3					Boston
4					Chicago
5					Houston
6					Los Angeles
7					Miami
8					New York
9					Philadelphia
10					San Francisco
11					Seattle
12					Washington
13					

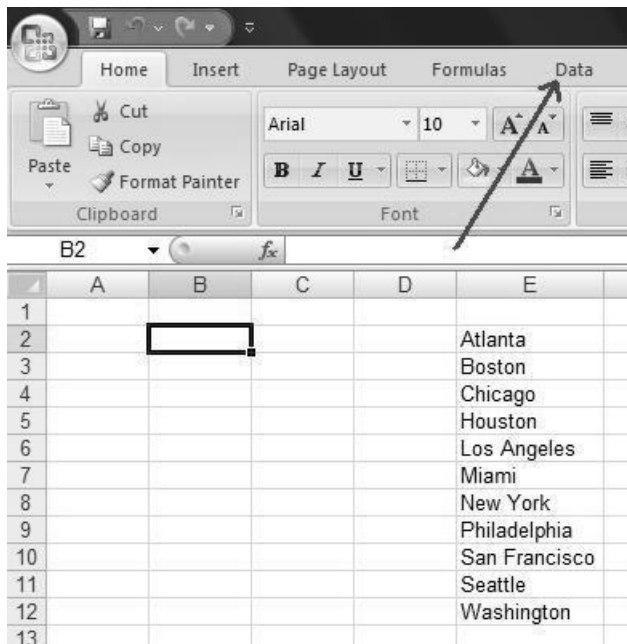
2. Name the list, by selecting its range and writing its name in the name box. In this example we named it "cities":



3. Select the cell in which you want the drop down list to appear. In this example we selected cell B2:

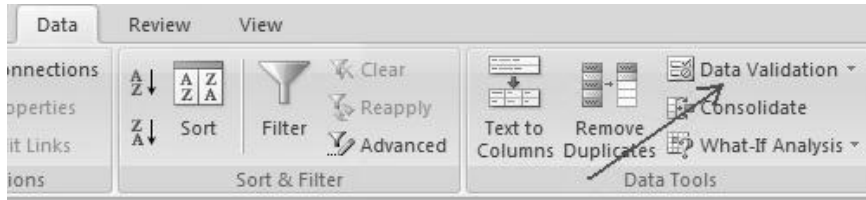


4. Choose the "Data" tab from the ribbon.

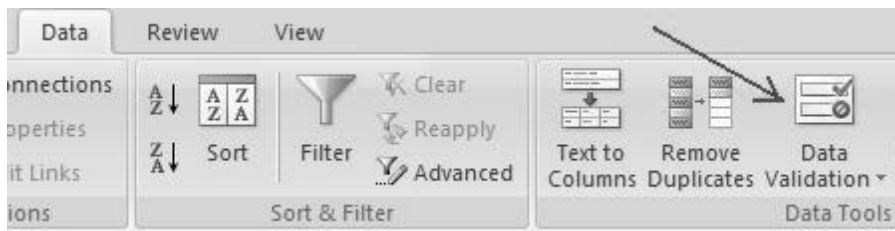


5. Click the "Data Validation" button.

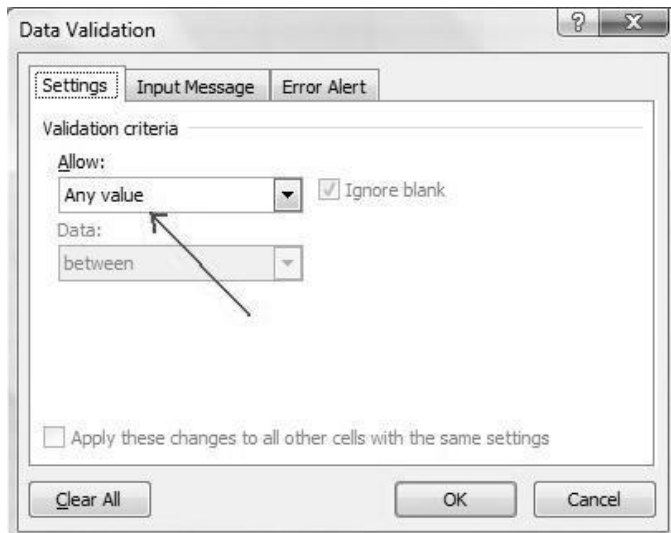
It will look like one of the two following examples, depends on your screen resolution:



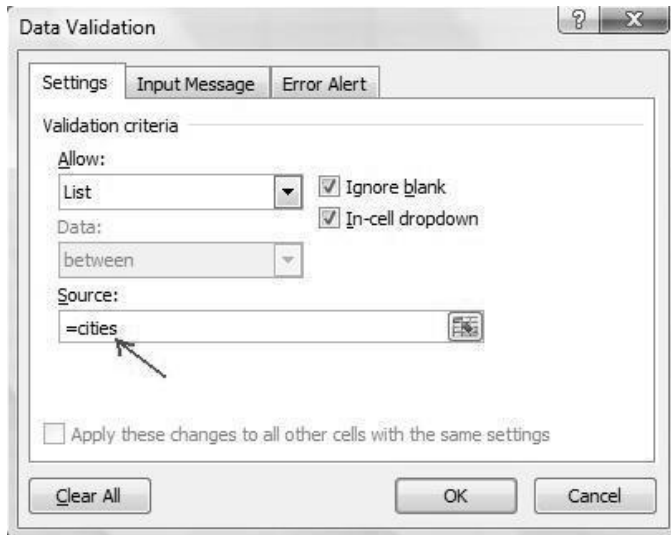
OR:



6. A dialog box will appear. Click the words "Any value" and choose "List".

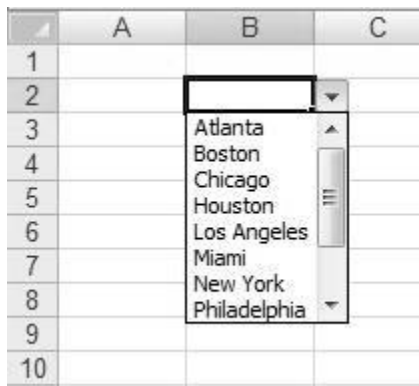


7. Click inside the "Source" text field, and write =[the name you gave to the list] (in our example: =cities).



8. Click the OK button.

Done! Now you have a drop down list inside your selected cell.



How to Import a Delimited File into Excel

Whether they are created manually or exported from another software application, delimited files are a commonly used to store important data. The trouble with data is that it is not very valuable unless you have some way to manipulate and study it. While delimited files are a nice way to store data, it's not very practical to try to study the data in the raw form of these files. Importing the delimited file into Microsoft [Excel](#) will allow you to use all the tools of Excel to analyze your data and to present the findings of your analysis in a clear manner.

Step 1



Choose “Open” from Microsoft Excel’s File menu, and navigate to the folder that contains your delimited file. Make sure that you have chosen the option to view all file types. If you don’t do this, you will not be able to see your file in the folder. Double-click on the file name to open it. Excel will automatically enter “Text Import Wizard” mode.

Step 2



Choose “Delimited” as the file type that best describes your data on the first screen of the Text Import Wizard. Then click “Next” to go to the next screen.

Step 3



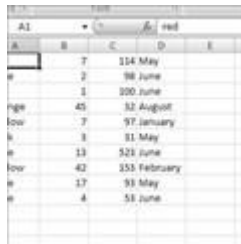
Select the delimiter that is used in your delimited file. If you didn’t know what the delimiter was before beginning the file import, you can usually scan the data and determine which character is being used. In our example, the file is comma-delimited. After selecting the delimiter, click “Next” to proceed.

Step 4



Assign the data format to each field that you are importing. The default data format is “General,” and this is the format you should keep unless any of your data fields contains dates or consists of data that you want to keep as text. Click “Next” to finish the import after you have finished assigning data formats to all fields.

Step 5



Save the resulting spreadsheet as an Excel file. Now you are ready to manipulate and study your data with the tools in Excel.

Excel Scenario Manager

Excel's **Scenario Manager** is a tool that can be used to determine different projected outcomes of data by changing different cells within a Worksheet model.

A **scenario** is a specific set of values that Excel can save for you and automatically substitute into your Worksheet. This means that you could have a spreadsheet displaying numerical data that is relevant to a certain date, month, topic or whatever and using the **Scenario Manager** you can enter different values into the worksheet to forecast the outcome of the data. These values (or Scenarios) can be retained for future use and are stored in a hidden part of the workbook which can be retrieved by asking the Scenario Manager to show the Scenario that uses those specific values.

Remember, all that Scenarios are are just a different version of the same worksheet with each different version using a different set of input values in the nominated cells.

For Scenarios to work correctly, you should first set up a base or **default Scenario**, on a worksheet in Excel. It is from this default Scenario that all other Scenarios are defined.

- Go to **Tools>Scenarios** to activate the Scenario Manager. You will see a message telling you "No Scenarios are defined".
- Choose **Add** to add your default Scenario.
- Give your Scenario a name, Base, Original or Default or a name you can easily identify will be fine.
- Click in the next box **Changing cells**: You will notice here that the cell that was your active cell in the Workbook will be referenced here.
- Click the collapse dialog button to collapse this dialog box up so that you can easily select the cells from the Worksheet that you wish to reference.
- Click on the collapse dialog button again to expand the full Scenario Manager box.
- If you wish to type a comment in, then click into the **Comment**: box and do so.

There are two options at the bottom of this dialog box. They are **Prevent changes** and **Hide**, with **Prevent changes** being the default. If you select **Prevent changes**, then all your Scenarios will be locked and will be unable to be edited. It is important to note here that you **MUST** also protect your Worksheet via the **Tools>Protection>Protect sheet** option for this option to take effect. If you then wish to edit your Scenarios, you must first unprotect your Worksheet, then de-select the **Prevent changes** option to proceed. The **Hide** option when selected will do exactly as said and Hide your Scenarios. This option also requires sheet protection for it to take effect.

- Click the **OK** button. Once you click on the **OK** button the **Scenario Values** dialog box will appear. This is where you must enter values into the scenario cells. As the first scenario is your default Scenario, the values in the cells that we specified in the **Changing Cells**: box have been picked up so we need to make no changes here, we need only click the OK button. This will now take us back to the Scenario Manager and you will see the name of your Scenario.

Adding

Scenarios

There is no limit to the number of Scenarios that you can apply to your worksheet model. Adding a scenario is done in basically the same way as creating a default scenario.

- Go to **Tools>Scenarios** to display the Scenario Manager dialog box and select **Add** to display the **Add Scenario** dialog box.
- Under **Scenario name**: give your new Scenario a name. You need make no changes to the **Changing cells**: box as the cells we nominated in your Default scenario show here.
- Click **OK**. This will show the Scenario values dialog box and this time you will need to change the values.
- Click **OK**.

You will now have two Scenarios available you.

Displaying Scenarios

Now you can display your Scenarios to show how they change the outcome of your data by asking the Scenario Manager to show a particular scenario.

- Select **Tools>Scenarios**
- Click on the Scenario name you want to see
- Click **Show**.
- Drag the dialog box out of the way and have a look at the values in the worksheet. Once you have done this, click on the next Scenario name and then **Show**, and again peruse the values in the worksheet.
- Click on **Close** when you no longer wish to view your Scenarios.

Excel Consolidation to Consolidate and Summarize Data from Different Worksheets

Consolidation is the process of combining values from several ranges of data either from within the same or different workbooks. It can be used to summarize data from different worksheets into master worksheet and create a report using a variety of calculations. Consolidate is found under the Data Menu; Data>Consolidate.

Consolidate by Position

Used when worksheet data is identical in order and location. This type of Consolidation is the easiest and works using a layering operation. This means that once you open the workbook(s) you wish to consolidate, you specify the ranges to consolidate, then the values in one worksheet are overlaid on those of another worksheet until all the worksheets are overlaid. It is then that you can select the calculation to be performed on your data. No formulas are used to perform this type of Consolidation, although you can set up the consolidation to update automatically when the source data changes, by checking the create links to source data box, rather than manually which is the default.

Consolidate by Category

Used when data is organized differently but has identical row and column labels. This type of Consolidation is a little trickier to perform, but works in nearly the same way. When you select your data to consolidate, you must this time include your row and/or column headings. Excel will then examine the row and/or column headings and will be able to plot the layout of your Worksheets and Consolidate your data for you by examining the contents of the ranges to be used.

Consolidate using 3-D Formulas

Used when worksheet data does not have a consistent layout or pattern. 3-D formulas are formulas that refer to multiple worksheets and can be used to combine this type of data. A formula such as **=SUM(Sheet1!A6,Sheet2!C12, Sheet3!H9)** can be used to add cells from different worksheets to consolidate into a Summary sheet. This type of Consolidation does not use the Consolidate dialog box found under **Data>Consolidate**, but is created entirely as a formula, so therefore will automatically update if the data it is dependent on changes

Formulas

Calculations using addition, subtraction, division and multiplication

Excel is an electronic spreadsheet that can be likened to a great sheet of paper, divided into 256 columns and 16,384 rows, in which we can store texts and numbers. But the great advantage of this program is how values and texts, stored within the sheet, can be manipulated in such a way that the user finds best, by means of a great number of a valuable formula that can be used at any given moment.

But first, what exactly are formulas?

Formulas are expressions that allow precise and speedy calculations and operations to be made very simply. With them, we can work with numbers, cell addresses, texts and conditions, just to mention a few. If you accurately define a formula, it will calculate the correct answer when introduced into a cell and will further provide constant updating, recalculating the result every time any value is modified. We can use mathematical functions to create formulas that make simple or complex calculations and those which may be used in a formula are as follows: It is a logical operators used in Comparison.

Operators	Used for	Example
H >	Greater than	F5>H7
<	Less than	D4<H3
=	Equal to	G5=H6
<>	Different from	G5oC10
>=	Greater or equal to	14>=D2
<=	Equal or less than	13<=G6
Tor +	Addition	B1+B7
-	Subtraction	X4-9
*	Multiplication	C9*D4
/	Division	D7/A2
A	Exponential	G5A9
%	Percentage	235*10%

Functions

Excel has pre-defined functions that facilitate a wide range of calculations and tasks. All you need to do to make a function work is to supply it with the appropriate values in order to make the calculation. The functions must start with the equals sign (=) ininterpret them as formulas and not as text. It is also advisable to type the name of the function and of the cells in capital letters.

SUM Formula

i.e., =SUM (A1:A7) The formula will add all the values that it finds in address A1 until address A7. The formula will always be the same; all that will change, are the addresses of the values that you wish to add up.

See the example:

In this example, all the values from A1 to D1 were added. The formula used was =SUM(A1:D1) and, on pressing Enter, the result appeared. In this case the result was **45**.

To reach this result, you need to follow these steps:

1. Place the cursor in the cell that will store the result of the calculation.
2. Type in the equals sign (=) to indicate that a formula is to be entered into the cell.
3. Insert the address of the cell that contains the first data value to be added.
4. Type the plus sign (+) to show that an addition is to be made.
5. Type the cell address that contains the next result to be added and so on and so forth.
6. Press the Enter key.

However, there is an easier way to add up by using the AutoSum button %.

To work with the AutoSum button, do the following:

1. Select the values that you wish to add up.
2. Then click the AutoSum button and it will show the answer.

Subtracting Cells

Imagine you have created a complex table with all your expenditure over the last six months and wish to know what balance will be left over, in order to make an investment. The procedure is quite simple:

1. Select the cell which will hold the result of the operation
2. Type in the equals sign (=) to indicate that a formula is to be entered into the cell.
3. Insert the address of the cell that contains the first data value from which it is to be subtracted.
4. Type the minus sign (-) to show that a subtraction is to be made.
5. Type the cell address that contains the value to be subtracted.
6. Press the Enter key to conclude the operation. The result should be as shown in the image.

Multiplication Formula

The four basic mathematical operations have similar formulas in Excel. In case of multiplication, you just need to use the same scheme you used for subtraction, changing just the subtraction symbol for that of multiplication (*).

See the example:

1. Select the cell which will hold the result of the operation.
2. Type in the equals sign (=) to indicate that a formula is to be entered into the cell.
3. Insert the address of the cell that contains the first data value to be multiplied
4. Type the sign (*) to show that a multiplication is to be made.
5. Type in the cell address that contains the value to be multiplied.
6. Press Enter to conclude the operation.

Division Formula

To make a division, you must use the formula in much the same manner as before. You just need to change the sign to division (/).

Procedure:

1. Select the cell which will hold the result of the operation.
 2. Type in the equals sign (=) to indicate that a formula is to be entered into the cell.
 3. Insert the address of the cell that contains the first data value to be divided.
 4. Type the division sign (/) to show that a division is to be made.
 5. Type in the cell address that contains the value to be divided.
 6. Press Enter to conclude the operation.
- Copying and Moving Formulae** If you make a lot of calculations in a spreadsheet, you don't need to type in the formula every time you use it. You can either copy or just move it.

Copying Formula

To copy a formula the procedure is as follows:

1. Click on the cell to be copied.
2. Access the Edit menu and choose Copy, or use the shortcut Ctrl + C.
3. Click on the cell or region where you wish the formulas to be copied.
4. Open the Edit menu and choose Paste, or use the shortcut Ctrl + V

Moving Formula

You can also move a formula, which will save having to retype it every time it is needed.

1. Select the cell to be moved.
2. Open the Edit menu and choose Cut, or use the shortcut Ctrl + X.
3. Click on the cell or region where you wish the formulas to be moved.
4. Open the Edit menu and choose Paste, or use the shortcut Ctrl + V.

Formulas**A simple step-by-step project that explains the basic formulas**

In the previous pages you have been introduced to some of the basic concepts about the application of formulas within Microsoft Excel. Now, we are going to learn how to

put into practice the four basic operations in mathematics: addition, subtraction, multiplication and division.

Addition

1. With Excel open, create a sheet with the items Products, Value and Quantity.
2. Now, fill in the items with the values you wish to add up. You can also use numbers with decimal points.
3. To apply the formula for addition, click on cell C4 and type the formula =C2+C3. It will give you the answer to the sum of the two figures.

Subtraction

1. To learn how to use the formula for subtraction, make a sheet with the items Products, Value, Quantity and Sold.
2. Fill in the Sold field with the number of products that are no longer in stock.
3. Click on cell E2 and type the following formula =C2-D2. In this case the answer will give the result of the subtraction

Multiplication

1. To multiply numbers, let us use the same sheet from the previous example.
2. Click on cell E2 and type the following formula; =B2*D2 and press Enter.
3. Repeat the same operation in order to carry out the calculation for black pen by changing just the cells (=B3*D3).

Division

1. Now we shall make a division with values that we have in our spreadsheet. (Products, Price of Box, Quantity in Box and Unit Value).
2. Fill in the corresponding numbers of the items (Products, Value, Quantity in Box) to discover the unit value.
3. Now, just click in cell D2 and type the formula =B2/C2. This will return the result of the division of the two numbers.

Beyond the E=MC² Basics

Learn to use the most important functions of Excel

If you want to know how many students didn't reach the final average from a table consisting of several names, you can use the formula _COUNTBLANK. It counts the cells that have no value. Follow these steps:

1. Create a sheet with the columns 'Student' and 'Average'. Fill the cells with the names and the values of their scores.
2. To know how many students failed to reach the required average in the sheet, type the following formula: =COUNTBLANK(B2:B7)

3. To understand the function: =COUNTBLANK is the name of the formula; (B2:B7) refers to the addresses of the cells.

Other formulas

COUNT: Calculates how many cells contain numbers and the numbers contained in the lists of arguments.

Example: If all the cells in A1:A10 contain numbers except the cell A5. Which is empty and A7, which contains a word, then: COUNT(A1:A10) will give the answer 8.

DCOUNTA: Calculates the number of cells that are not empty, and the existing values in the argument list.

Example: If all the cells in A1 :A10 contain numbers, except the cell A5, which is empty and the cell A7, which contains a word, then: DCOUNTA(A1 :A10) will return the answer 9.

STDEV: This calculates the standard deviation from a mean within a sample. The standard deviation is a measure of the degree of dispersion of values in relation to an average value.

Example: Let us suppose that 10 tools made in the same machine during production are collected as a random selection and are then measured to check their resistance to breakage. The values of the sample (1345, 1301, 1368, 1322, 1310, 1370, 1318, 1350, 1303, 1299) are stored in cells A2:E3, respectively. STDEV estimates the standard deviation of resistance to breakage for the entire sample. STDEV(A2:E3) equals 27.46.

ABS: Provides the absolute value of a number. The absolute value is the same figure without a plus or minus sign.

Examples: =ABS(-4) will give us 4
=ABS(4) will also give us 4

ROUND: Rounds up the number to the nearest determined number of decimal places.

Examples: =ABS(-4) will give us 4

Examples: =ROUND(2.149;1) will give us 2.1;
=ROUND (1.475;2) will give us 1.48; =ROUND (2,5;-1) will give us 20.

INT: Rounds a figure down to the nearest whole number.

Example: =INT(8.9) will give us 8.

TRUNCATE: Truncates the figure in accordance with the number of decimal places required.

Examples: =TRUNC (8.97;0) will give us 8;
=TRUNC (8.97; 1) will give us 8.9.

Conditional Functions

Learn how to transform your Excel spreadsheets into efficient tools for analyzing data

On opening Excel for the first time, none of us ever imagined the power and capacity of all those rows and columns. Behind this great table, exist hundred of formulas, functions, and filters amongst other recourses that help build Excel's reputation as the best spreadsheet editor. Nevertheless, to get the most out of this tool it is necessary to know what these recourses are and how to use them: otherwise Excel becomes nothing more than an enormous database, built up of hundreds of rows and columns.

In this step-by-step tutorial, we shall deal with conditional functions - formulae that allow us to create sheets that give us results based on certain given conditions. Check out the examples of the If, And & Or functions that we introduce here, in this step-by-step guide, and adapt their usage in accordance to your needs.

Function: AND & OR

In the following example, we shall define the status of each student in a college according to their final mark and their number of absence. To pass, the student must attain an average higher than 7 and the number of absence must be less than 10. Only when meeting both conditions will their status be TRUE, which means , the student's performance will have met both arguments conditions. Should the student not meet both requirements, the message returned will be FALSE. Let us also give an example in which it is not necessary to meet all the conditions. In this case, if just one of them is true, then the message returned will now be TRUE.

First Step: To create the status of a student, as stated, you have to use the function AND. In cell A1 , type a value for the final average mark of the student, and the number of absence in B1. In cell C1, type =AND(A1>7,B1<10).

Second Step: You can add more arguments to the function using, Continue typing the function, putting in the first argument: A1>7. Type ; and continue with the second

argument: B1>10 and close the brackets. The final formula should look like this:
=AND(A1>7,B1<10).

Third Step: To finish, press Enter. Now, using the function AND, the result will only return TRUE if all the arguments comply.

IF

Test: In this attribute, an expression will be considered to see if it has either a TRUE or FALSE value

Value if true: The returned value, if the test argument proves to be TRUE. For example, if we have an argument as a sequence of letters such as “Approved” and this argument for the test was put as TRUE, the function IF will show “Approved” in the corresponding cell of the formula. If the test is TRUE and Value if true is blank, the argument will return the value zero.

Value if false: The returned value if the test is FALSE. For example, if this argument is a sequence of characters “Failed” and the test argument is FALSE, the function IF will show “Failed”. If the test is FALSE and value if false is blank, the value zero will be returned

AND(argument1; argument2..)

In this function, all the arguments must be considered for the function to return the value TRUE. The function AND can also be used in conjunction with the function IF, transforming it into a twin function.

Example

=AND(5<8;13>2)~> Function will return the value True, as all arguments are true.

=AND(4>5;7>6)) —> Function will return the value False, as only one of the arguments proves to be true.

OR

Unlike the function AND, the function OR requires at least one of the tested arguments to be true for it to return the value TRUE. This function will only return the value FALSE, should all of the tested arguments prove to be false.

Examples

=OR(1 <2;8>4) —> Function returns the value True, because all the arguments are true.

=OR(1>2;8>4)) -> Function returns the value True, because two of the arguments are true.

=OR(1>2;8<4) —> Function returns the value False, because all the arguments are false.

Fourth Step: For the formula to return TRUE. even when only one of the two arguments is TRUE, use the function OR. Change just the formula to OR instead of AND.

Fifth Step: There is a further logical function used to invert the logical value of an argument, which means, if the value of a cell is true, the return will be FALSE and if the value is false the return will be TRUE. This function is called NO.

Sixth Step: Type, for example, in cell D1: =NO C1). Apart from this, it is also possible to mix all these functions to a more specific result. i.e., =ANDORAVE D8:D16 >7,E8=10).

Functions: IF & AND

Calculate the retention of the income tax in accordance with the salaries of each employee.

First Step: Taking advantage of the same table of data for the positions and salaries we saw before, we shall now calculate the retention of income tax in accordance with the different salary bands.

Second Step: For this task, add a new column which should be placed next to the Salary column. Give it the tide Tax, as shown in the illustration.

Third Step: On the “Tax” column, we calculate the tax retention for each employee. In this case, we have to use the combination of functions IF and AND so that the condition deals with these three conditions.

Fourth Step: Click on the cell D4 and type in part of the formula “=IF(C4<=500;Exempt;” (Without the inverted commas). If the condition holds true for the tested cell, there will be no tax retained for this position.

Fifth Step: To test the second salary band between INR 500 and INR 1,000, add the function AND, so that the formula becomes

=IF(C4<=500;”Exempt”;IF(AND(C4>500:C4<= 1 000 ;50;”.

Sixth Step: To finish the formula, add the argument 100 to the formula so that there is tax retention for the upper salary band over INR 1,000. The complete formula is “=IF(C4<=500:”Exempt”;IF(AND(C4>500;C4<= 1000 :50: 100))”

The Functions IF & OR

Employee benefits in accordance with the salary band and department in which each employee works.

First Step: Now, to illustrate the combined use of the functions IF and OR, we shall insert two columns in the position and salary table - Health Plan and Department.

Second Step: In this case, the company offers the benefit of a health plan only for those workers who receive up until INR 1,000 or who work in the Sales Department.

Third Step: To check which employees have the right to this benefit, select the first cell in the column Health Plan, which in our case here is the cell E4 to insert the function. `IF(B4="Manager";2500))` which corresponds to the condition test for the three positions and their respective salaries.

Fourth Step: With the numbers now different, try changing the second sign = in the formula in C1 for <>. The result changes to TRUE.

Fifth Step: To copy the function to all the rows of the Salary field, just position the pointer of the mouse in the lower right hand corner of the cell and drag until you reach the end of the table-

;Logical Operators

In order to make conditional calculations, you need to pay particular attention to the logical operators. Open a blank spreadsheet and type in the formulas for the test. In this example, you will be able to check how two of the operators word = equal) and <> (different).

First Step: Select the cell A1 from your table and type the value 10. Then, select cell B and type the value 10 as well.

Second Step: In cell C1, type the following formula `=A1=B1`. It will test the values, returning TRUE.

Third Step. If you alter the value in one of the cells, the message in C1 automatically changes to FALSE.

Operators

In the table opposite (below/above), you will find all the logical operators used in conditional functions, complete with a description, examples and the results that are generally obtained. ft*

Operator	Description		
=	Equals returns TRUE when the two values are equal	=A2=A5	TRUE
>	Greater than returns TRUE when the first value is greater than the second	=A2>A5	TRUE
<	Less than returns TRUE when the first value is less than the second	=A2<A5	TRUE
>=	Greater than or equal to returns TRUE when the first value is greater than or equal to the second	=A2>=A5	TRUE
<=	Less than or equal to returns TRUE when the first value is less than or equal to the second	=A2<=A5	TRUE
<>	Different returns FALSE when the two compared values are different	=A2oA5	FALSE

Percentage Formulas

Discover how to use this simple but useful function, step by step >>

Percentages are frequently used in several Activities in our day to day routines. They very often arise in expressions which reflect _ increases or reductions in prices, figures or quantities. We frequently hear phrases such as: petrol has gone up by 10% (which means that for every INR 100 worth of petrol, there has been an increase of INR 10 ; the client received a 20% discount on all the goods (which means that for each INR 100, a discount of INR 20 dollars was given on the price); of all the players that play for Brazil. 90° o are crack players (meaning in 100 players that play for the squad, 90 are exceptional goal scorers).

Two simple examples:

1. A shop launches a promotion of “10% OFF” on all its products. If one item of merchandise originally costs INR 120.00. how much will the discount price be? The discount will be 10% of the value of INR 120.00. Therefore, $120 \times 10/100 = 1200/100 = 12$ Therefore, we deduct INR 12.00 from INR 120.00: $120 - 12 = 108$ With the sale promotion price, we pay just INR 108.00.

2. A class has 100 students, 40% of which are girls. What is the number of girls and boys? The number of girls is: $100 \times 40/100 = 40$ And the number of boys can be determined as:
 $100 - 40 = 60$.

Complicated? Although the examples are easy, Excel allows even such obvious calculations to be made quickly and easily. The calculation is the same as if you were

using a calculator but the difference is that you add addresses into the formula. Follow the step-by-step tutorial.

1. Imagine that a client makes a purchase of INR 1,500 and that you wish to give a discount of 5% on the value of the purchase. Make a table with the fields: Clients, Value, Discount and Price to Pay.
2. After having formatted the spreadsheet, fill it in with the name of the client and the value of the product. In the column marked Discount, insert the formula for determining percentage.
3. In the cell below Discount, enter the formula $=B2*5/100$ or even $=B2*5\%$. Press Enter to check the result.
4. To discover the real value of the product to be purchased by the client, enter the subtraction formula $= B2-C2$.
5. So that you may understand the formula: B2 refers to the address of the purchase value, * is the multiplication sign, whilst 5 /100 is the value of the discount, divided by 100.
6. In this formula you will be multiplying the address of the purchase value by 5 and dividing it by 100, thus generating the value of the discount.

INSERTING DATES

Automate the insertion of days, months and years in your spreadsheets. If you work with Excel spreadsheets and frequently need to insert dates, then follow this tutorial. The Date formula allows you to put a date into a cell, without the need of typing it in. Furthermore, you can also discover any day of the week that corresponds to a particular date.

Formulas for Dates and Days of the Week.

1. To insert today's date in a spreadsheet, all we need to do is position the cursor in the cell in which we wish the date to appear and type $=\text{TODAY}()$.
2. You can also insert the day of the week corresponding to a date. The day is given by a whole integer which varies between 1 (Sunday) and 7 (Saturday), by default.
3. Select a cell and click on it with the right hand button of the mouse. In the tab Number, choose the item Date and, in Type, select the format with the day of the week and the date.
4. In the cell, type the date to discover the day of week it corresponds to. Press Enter.
5. Note that Excel will supply the day of the week of any date you require.
6. You can further discover the day of the week of several dates all at the same time. Select all the cells with the dates and change their format into day of the week.

To calculate the number of days between two dates, simply subtract them.

For example: B2=08/08/2000 and B3=27/08/2000. The formula B2-B3 will provide us with the resulting number of days. In this example, the result will be 19. Note that the date in the cell needs to be formatted as a number; otherwise Excel will provide a date as an answer instead.

Maximum, Minimum, And, Average

Learn how to discover the extreme values and averages in spreadsheets

Excel offers a range of tools to facilitate the daily routine of the user. One of these is the option of obtaining the average from a list of ranging values. Apart from this, you can also discover Which figures in the lists are the maximum and minimum values? Follow the next step-by-step tutorial and see how easy it is to use this resource

The Formula for Maximum

Imagine you have a sheet with names and ages of various students in a school and need to know what the maximum age of any student is. To discover this, all you need to use is the maximum formula.

1. Create a table with the names and ages of students from a school. In column A, enter the names and in column B. the ages.
2. Now let us type Maximum Age in cell A15. Cell B15 will be where we want to show the result of the maximum figure.
3. In cell B15, type the formula =MAX (B2:B14). Note that B2 corresponds to the first cell of the figures and B14 is the last.

Whenever subtracting dates and hours, check that you have created the formula correctly. The dates and hours must be positive values. If a date or hour formula produces a negative result, Excel will display in the length of the cell.

Formula for Determining the Minimum

This formula shows the minimum value of a range of cells. To discover the minimum weight of any of the children in our list, do as follows.

1. Let us use the same spreadsheet that we saw for the maximum formula but we shall add another column to include the relative weights of the students.
2. Now, we shall enter the data concerning each student's weight. In cell C15, write: Minimum Weight.
3. In cell D15, you must enter the formula =MIN(C2:C 14). Should you use the same values as shown in the image, then the result will be 25.

Formula for Determining the Average

The formula for determining the average is very useful one to know in order to discover the average value (age, for example) in a long list or table.

1. Let us now discover the average age of students in the school. To do this, we shall use the same sheet we used in the previous examples.
2. Now we shall enter the data concerning each student's weight. In cell C15, write:

Minimum Weight. In cell D15 you must enter the formula =ML\ (C2:C 14 .Should you use the same values as shown in the image, then the result will be 25.

1. Insert a row in the sheet and write Average in cell A15. In cells B15 and C15, you will obtain the average age and weight of the students.
2. In cell B15, type =AVERAGE(B2:B14) to discover the average age, whilst in cell C15, type =AVERAGE(C2:C14).

Interaction Between Tables

Create a formula to make calculations using data from separate spreadsheets Imagine you have a plan for household expenditure and that in each sheet of the file there are the sums of incomes and expenditure of each member of the family. You can link the result of two or more sheets to facilitate things, for example, to see the running total of the balance after every payment by each person.

In this step-by-step tutorial, we shall use a plan with the expenses of a couple and we are going to include, in our sheet, the balance of funds after the payment of **each** of the bills. Follow the steps given below:

Office Assistant and Functions

If you wish to include functions into your spreadsheet and can't easily remember them, you can take advantage of Excel's Functions Assistant. You can access it by means of the Insert menu, Function.

There, you will find a whole range of formulas and explanations about what each of them does. To access any of them, click on its name and then click on OK. A screen will appear in which you can determine which cells will make up the calculation.

Excel also shows the formulas most recently used by the user.

1. In the spreadsheet that is to contain the formula, select a cell in which you want to insert the result of another sheet.
2. If you are creating a new formula, enter = (equals sign).
3. To create a link with another sheet in the open workbook, click on the one which contains the cells you wish to link.
4. After opening the sheet that will receive the result, insert the formula as in the example: =Cristina!C 10+Marcelo!C 14.
5. So as to understand this formula: =Cristina! is the name of the table in which the figure to be added is: C10 is the cell that contains this value in the sheet;

Marcelo ! is the name of the second sheet and C14 is the cell with the second value to be added.

6. To conclude the formula, press Enter. Note that: if you created a different table from the example, the cells inserted in the formula

ASSIGNMENTS

Assignment:1

NAME OF THE STUDENT	HINDI	ENGLISH	HISTORY	MARATHI	GEOGRAPHY
RAHUL	58	48	74	85	71
RAJ	74	85	48	74	65
ROCKY	65	58	54	82	25
RAHIM	86	69	71	65	74
WAHAB	57	68	91	58	47
ANIL	56	67	81	45	55
SACHIN	75	74	58	75	65

STEPS TO BE DONE: Take a new sheet and create the data as shown. And alot do the formula like (Total, Avg., Min, Max, Grade, Result)

Option to be Used: Home panel is to be used

Assignment 2:

STEPS TO BE DONE: Open a new file and do the editing for the data. Cut the marks of the Sachin and paste it into Wahab data. As shown above.

Option to be used: Home panel clipboard option is to be used

Assignment: 3

STEPS TO BE DONE:

1. Insert a column.
2. Delete cell.
3. Delete row.
4. Change the size of the rows, columns to 25" and 23"

Option to be used: Home panel

Assignment: 4**STEPS TO BE DONE:**

1. Open a file and enter the data as given below.
2. Use different Text Alignment for first row.
3. Change the Font and Font Size of the second row.
4. Use Border for outline the text.

Option to be used: Home panel font, paragraph

Assignment: 5**STEPS TO BE DONE:**

Open the file and enter the data as given below. Sum the products of all four Products and prepare a chart showing details of sales of all three products for all four products.

Option to be used: Insert panel Charts option. And sum formula

Assignment: 6**STEPS TO BE DONE:**

Open a new file and enter the data as given below.
Use the pivot table and pivot chart and analyze the data

Option to be used:

Insert panel > Tables > Pivot Table

Assignment: 7**STEPS TO BE DONE:**

Create data sheet as shown below and do the following

1. Using page break option makes the columns B-3 to be printed on page 1 and F5 on page 2
2. Set margins of the page 0.75" left & right and 0.75" top & bottom
3. Print the grid lines
4. Display IMAC SOLUTIONS PVT. LTD. Corporation Ltd in the header and page numbers in the footer
5. Preview the document in print preview

OPTION TO BE USED:

1. Page Break
2. Page Setup
3. Print Preview

Assignment: 8**STEPS TO BE DONE:**

1. Select one column and sort by ascending order.
2. Again select other column sort by descending order.
3. Select three columns and sort the first column by ascending order, other two by descending order

OPTION TO BE USED:

Data panel > Sort and Filter

Assignment: 9**STEPS TO BE DONE:**

Create a data sheet as shown below and validate Rate field to accept minimum 1 to 2 month dates.

Option to be used:

Data panel > Data tools

Solve the question

1. Which of the following is the latest version of Excel ?
Excel 2000
Excel 2002
Excel ME
Excel XP
2. Excel files have a default extension of ?
XLS
XLW
WK1
123
3. A typical worksheet has ____ number of columns ?
128
256
512
1024

4. All formula in Excel start with ?
%
+
=
-
5. The divide symbol is ?
. /
D
D
)
6. The multiplication arithmetic operator is represented by which of the following symbols?
^
*
/
X
7. On an excel sheet the active cell is indicated by ...?
A dark wide border
A dotted border
A blinking border
None of above
8. Using the F11 shortcut key to create a chart on chart sheet creates?
A default chart
A 2-dimensional column chart
A 2-dimensional bar chart
A 3-dimensional line chart
9. You can print ?
A range of cells by range name
An entire worksheet
A single worksheet
All of the above
10. You can create only a horizontal page break by first selecting?
A row below the row where you want the page break to occurs
A cell in row 1
A cell in column A
a and c

11. You can create hyperlinks from the Excel workbook to?
 - A webpage on company internet
 - A web page on the internet
 - Other Office 97 application documents
 - All
12. The advantage of using a spreadsheet is?
 - Calculations can be done automatically.
 - changing data automatically updates calculations
 - more flexibility
 - all of the above
13. The intersection of a row and column is called?
 - Data
 - a field.
 - a cell
 - an equation
14. There are three types of data found in a spreadsheet.?
 - data, words, numbers
 - equations, data, numbers
 - words, numbers, labels
 - numbers formulas, labels
15. If you press, the cell accepts your typing as its contents.?
 - Enter
 - Ctrl+Enter
 - Tab
 - Insert
16. Which If you press, the cell accepts your typing as its contents.?
 - Esc
 - Shift
 - Return
 - Tab
17. Which is not an advantage of using computerized spreadsheets?
 - Flexibility of moving entries
 - Speed of calculation
 - Ability of generate tables
 - cost of initial setup

18. To select several cells or ranges that are not touching each other, you would ... while selecting?
- Hold down the Ctrl key
 - Hold down the Shift key
 - Hold down the Alt key
 - Hold down Ctrl + Shift key
19. A constant is another name for this type of data:?
- Number
 - Equation
 - Formula
 - Description

MODULE II

Building macros in Excel

Building macros in Excel

Visual Basic Editor (VBE) in Excel

The Visual Basic Editor (VBE) is the friendly programming environment within Excel. You can program VBA in every version of Microsoft Office, including MS Office 97, MS Office2000, MS Office2002, MS Office2003 and MS Office XP. The reason VBA is needed is due to the limitations in using the built-in functions of VB and macro recording. By using VBA, you can build some very powerful tools in MS Excel, including financial and scientific applications such as getting financial data from the Internet as well as linear programming.

The VBE is integrated into Excel and you can open it from the Excel menu bar "Tools/Macro/Visual Basic Editor" .

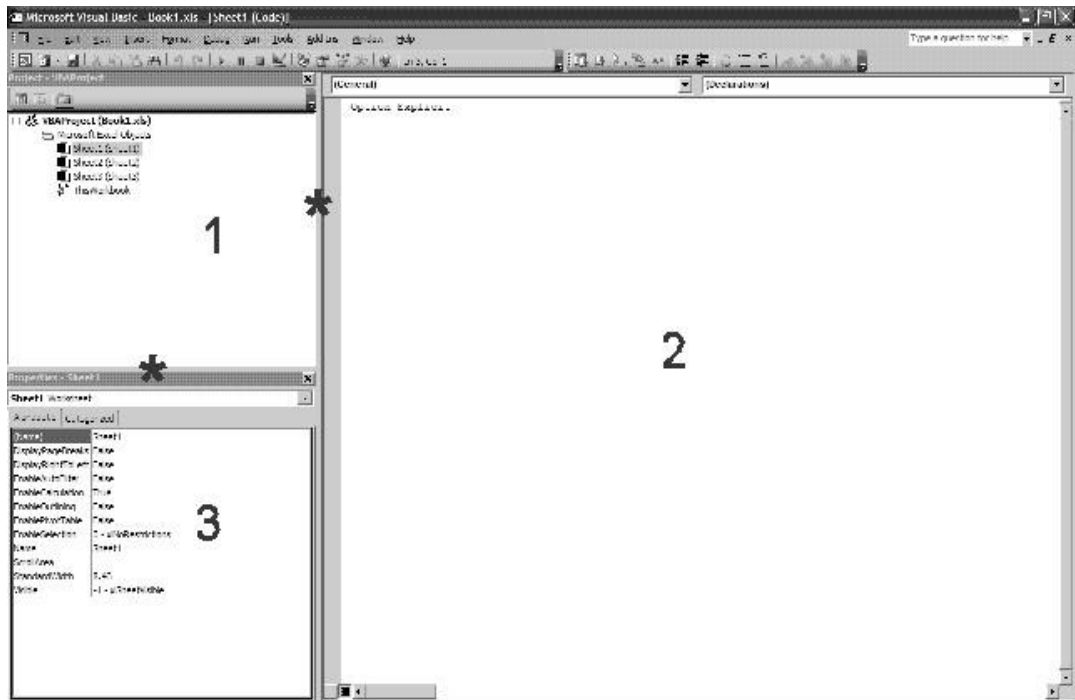
Open Excel and on your keyboard click on the "ALT" key (left of the space bar), hold and click on the "F11" key. Here is the Visual Basic Editor. Click again and you are back to Excel.

The Three Windows in the Visual Basic Editor

When you want somebody to do some work for you open your Email program and you send him a message in a language that he understands (English, Spanish, French...). When you want Excel to do some work for you open the Visual Basic Editor and you write the instructions in a language that Excel understands VBA (Visual Basic for Application).

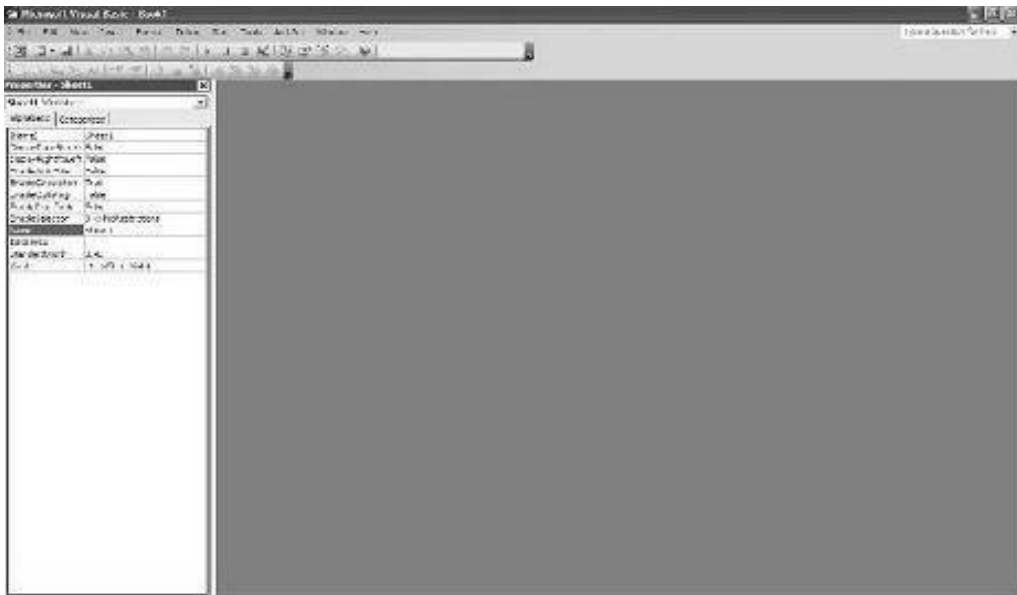
You will develop, test and modify VBA procedures (macros) in the Excel Visual Basic Editor (VBE). The VBA procedures are developed in the Excel Visual Basic Editor become part of the workbook in which they are developed and when the workbook is saved the VBA components (macros, modules, user forms. etc.) are saved at the same time. When you send the workbook to the "Recycling bin" the VBA procedures are gone.

When you work with the VBE there always should be 3 windows that are showing. The **Project window** (1), the **Code window** (2) and the **Properties window** (3) like below.

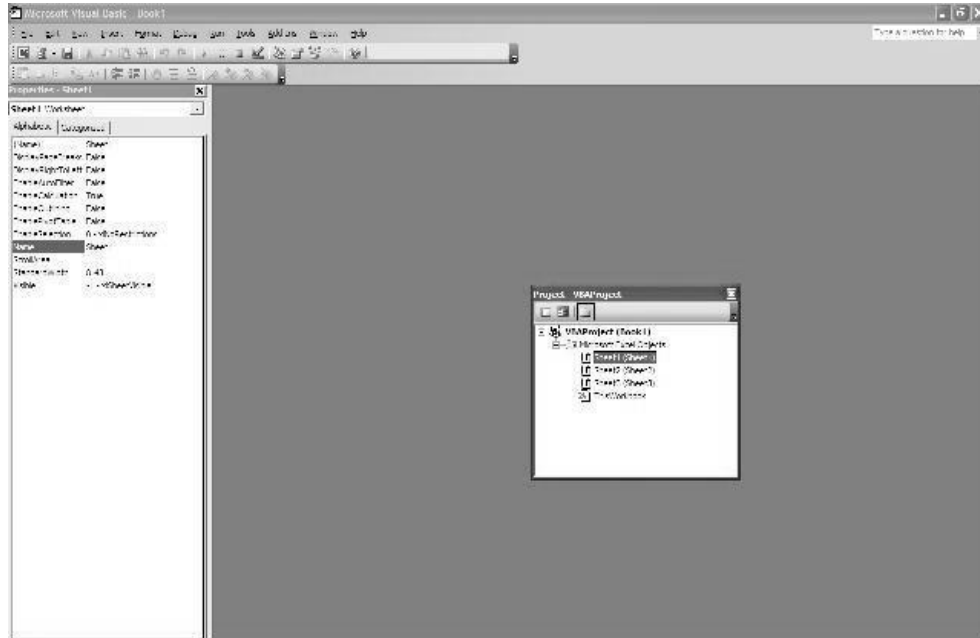


You might not see the 3 windows on your screen. We will fix that right now and it can be a little tricky. But once you have set them they will stay in their position always.

If there are any window open under the tool bars close them. On the menu bar of the VBE choose "View" and select "Properties Window". The window can appear minimized, it can be full width at the top of the screen or many other ways. You want it from top to bottom on the right of the screen. To do so click on its blue line at the top, hold and drag it to the bottom right corner. It is now minimized. Click again on its blue header, hold and drag it full left (it will even disappear) toward the vertical middle of the screen. It is now in the following position:



Then go back to the menu bar and select "Project Explorer". From wherever it is click on its blue header, hold and drag it all the way down in the bottom right corner. It is now free.



Reduce the size of the "Project Explorer" to about the width of the "Properties Window" and one third the height by using the double arrows that appear when you move the

mouse over the border of the "Project Explorer" . Click again on its blue header, hold and bring the top left corner of the "Project Explorer" over the top left corner of the "Properties Window". Then let go.

You now have the "Project Explorer" over the "Properties Windows" in the left part of your screen. The "Code Window" appears when you double click on a sheet name in the "Project Explorer" . You will see later that the "Code Window" also appears when you double click on a user form name or a module name.

You can resize the 3 windows by placing the cursor over the borders (*) and dragging them right, left, up or down.

You can use "ALT/F11" to navigate from Excel to the VBE and back.

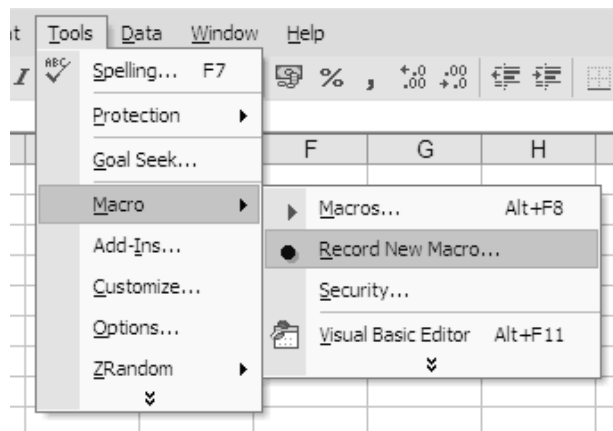
Creating or Developing Your First Macro

Recording Your First Macro

Microsoft Excel has a build-in macro recorder that translates your actions into VBA macro commands. After you recorded the macro, you will be able to see the layout and syntax. Before you record or write a macro, plan the steps and commands you want the macro to perform. Every action that you take during the recording of the macro will be recorded - including the correction that you made.

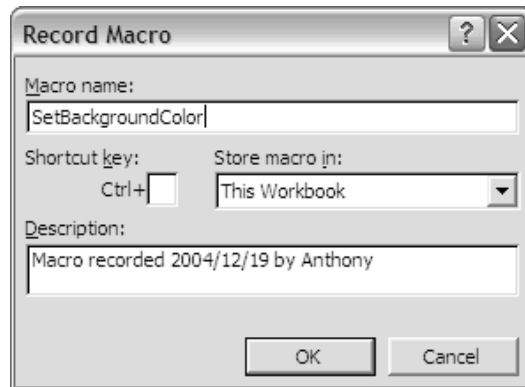
In this example, we will record a macro that sets the cell background color to light yellow. To record the macro, follow the steps below:

1. Select Record New Macro... under Tools...Macro

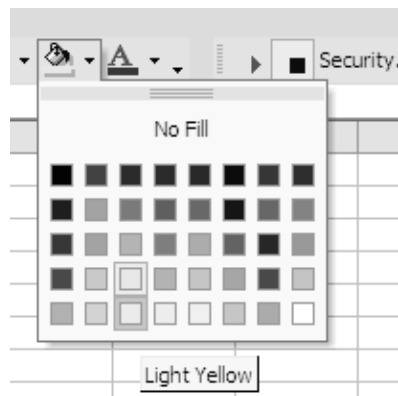


2. In the Record Macro dialog box, type "SetBackgroundColor" in the Macro Name textbox to set the macro name. Leave all other option by default then click the Ok

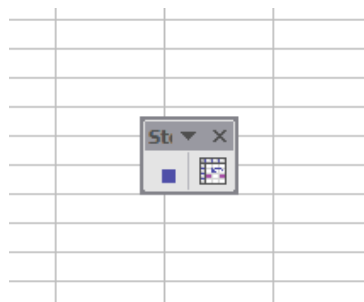
button. This will start the macro recording.



3. In the Background Color Panel, select the Light Yellow color box. This action will set the background of the current cell (A1) in light yellow color.



4. To stop the macro recording, click the Stop button (the navy blue rectangle) on the Macro Recorder toolbar.

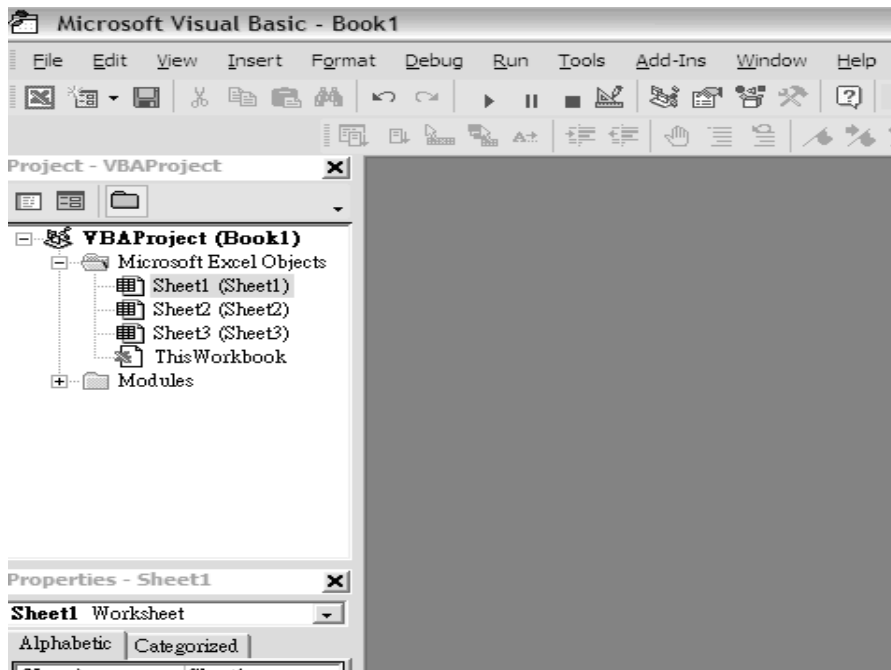


Now you have recorded a macro that set cell background to light yellow.

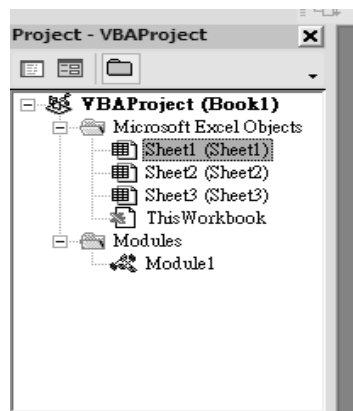
See the Recorded Syntax

The recorded macro is ready for use. Before we run the macro, let's look into the syntax.

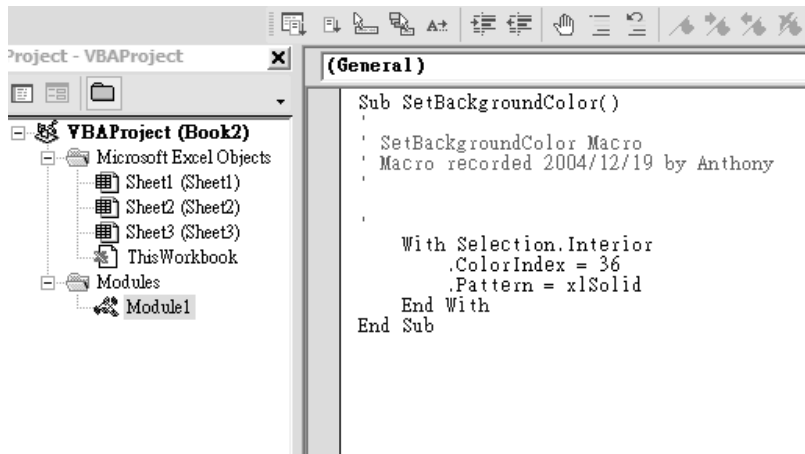
1. To load the Visual Basic Editor, press [Alt] and [F11] at the same time. (Remember from our prior lesson?) The Visual Basic Editor comes up.



2. Expand the Modules folder in the Project Explorer by clicking on the plus (+) sign.



3. Double click the **Module1** folder to see the sub routine (marco).

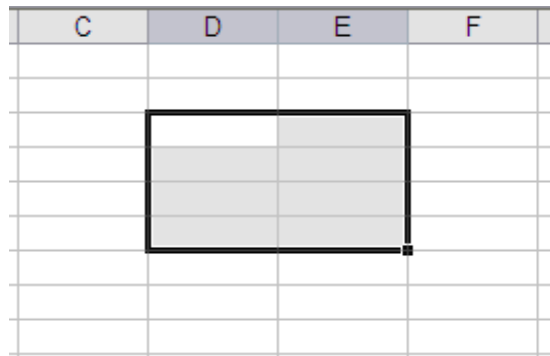


As the figure shows, the name of the sub routine is "SetBackgroundColor". The color index for the light yellow is 36. The background pattern is solid.

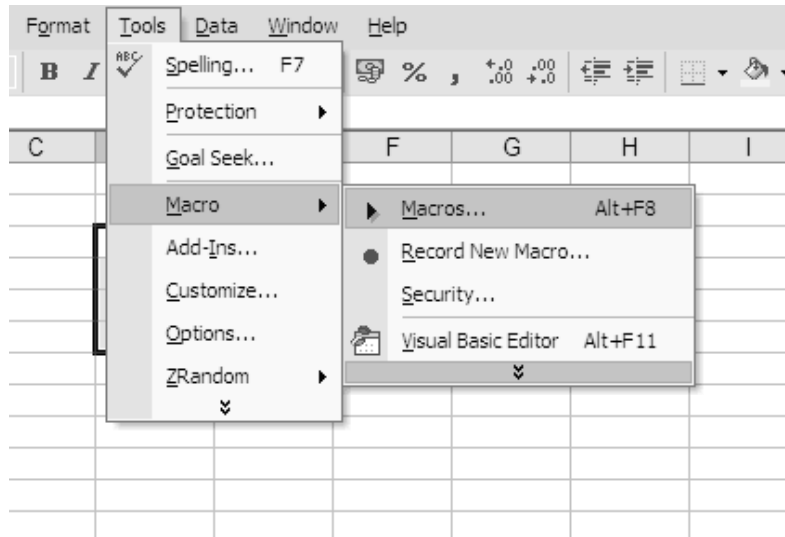
Run the Recorded Macro

In our prior example, we created the "Hello World!" macro. We ran the macro within the Visual Basic Editor. This time we will run the recorded macro in the worksheet.

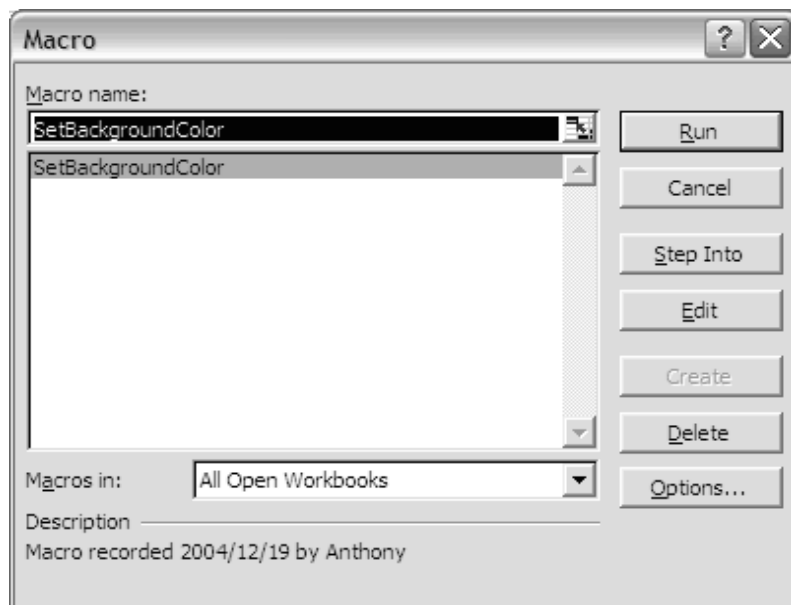
1. On any worksheet, select from D3 to E6.



2. Run the recorded macro by select Tools...Macro...Macros... or press [Alt] and [F8] at the same time.



3. The Macro dialog box displayed. Since there is only one macro in the module, by default the only macro, SetBackgroundColor is selected. Click the Run button to run the macro.



4. Cells D3 to E6 now have light yellow background color.

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						
8						

Let 's Lea rn some basi cs befor e wr it ing

VBA Code:

Range Object and Cells Property

Range represents a cell, a row, a column, a selection of cells containing one or more contiguous blocks of cells, or a 3-D range. We will show you some examples on how Range object can be used.

The following example places text "AB" in range A1:B5, on Sheet2.

Worksheets ("Sheet2").Range("A1:B5") = "AB"

	A	B
1	AB	AB
2	AB	AB
3	AB	AB
4	AB	AB
5	AB	AB
6		

Note that, Worksheets.Range("A1", "B5") = "AB" will yield the same result as the above example.

The following place "AAA" on cell A1, A3, and A5 on

Sheet2.Worksheets("Sheet2").Range("A1, A3, A5") = "AAA"

	A
1	AAA
2	
3	AAA
4	
5	AAA

Range object has a Cells property. This property is used in every VBA projects (very important). The Cells property takes one or two indexes as its parameters.

For example,

Cells(index) or Cells(row, column)

where row is the row index and column is the column index.

The following three statements are interchangeable:

ActiveSheet.Range.Cells(1,1)

Range.Cells(1,1)

Cells(1,1)

The following returns the same outcome:

Range("A1") = 123 and Cells(1,1) = 123

The following puts "XYZ" on Cells(1,12) or Range("L1") assume cell A1 is the current cell:

Cells(12) = "XYZ"

The following puts "XYZ" on cell C3:

Range("B1:F5").cells(12) = "XYZ"

	A	B	C	D	E	F	G
1		1	2	3	4	5	
2		6	7	8	9	10	
3		11	XYZ 12	13	14	15	
4		16	17	18	19	20	
5		21	22	23	24	25	
6							

* The small gray number on each of the cells is just for reference purpose only. They are used to show how the cells are indexed within the range.

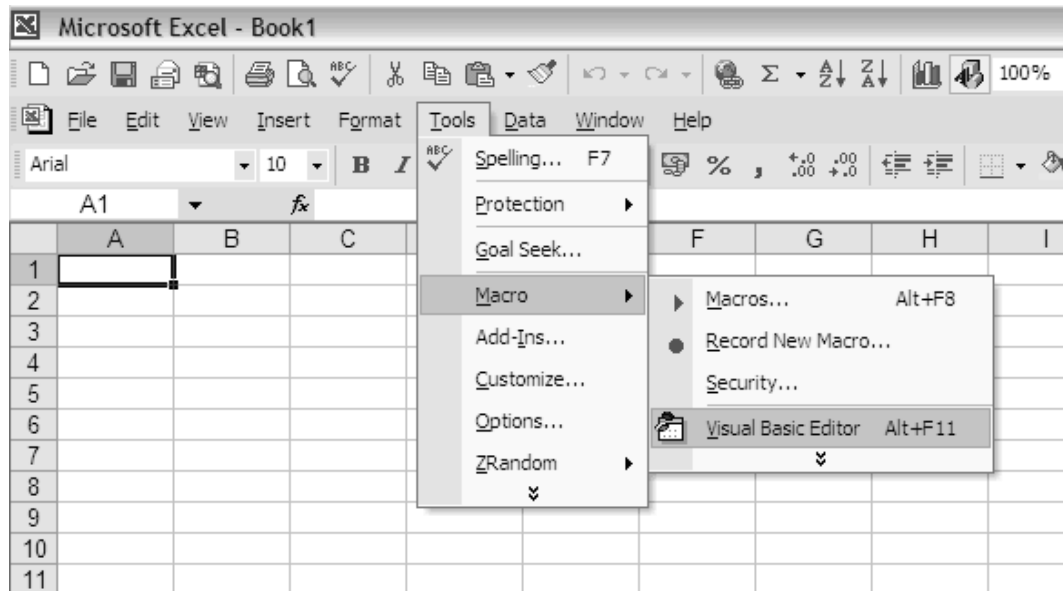
Practice a simple VBA program

There are two ways which you could program a VBA

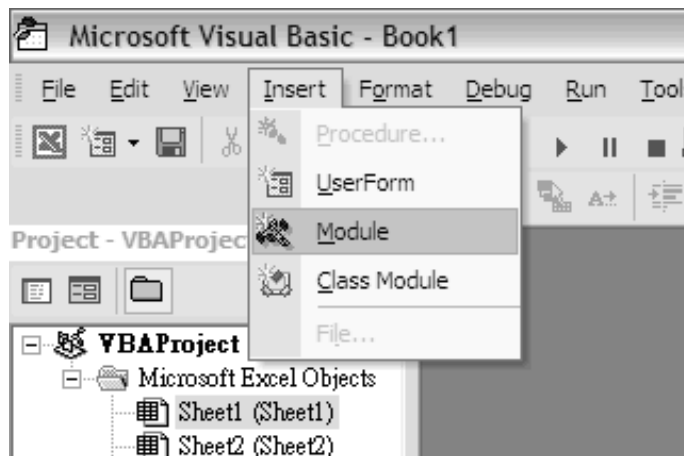
1. To write Visual Basic functions inside the VB Editor.

In this sub section, we will show you how to create your first macro (VBA program). We will use the world classic "Hello World!" example. To create the example, please follow the following steps:

1. Open Visual Basic Editor by go to Tools...Macro...Visual Basic Editor or just simply press the [Alt] and [F11] keys at the same time.

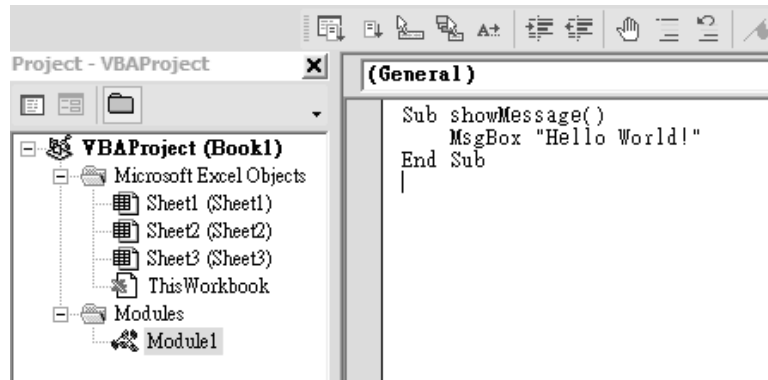



2. In the Insert menu on top of the Visual Basic Editor, select Module to open the Module window (code window).

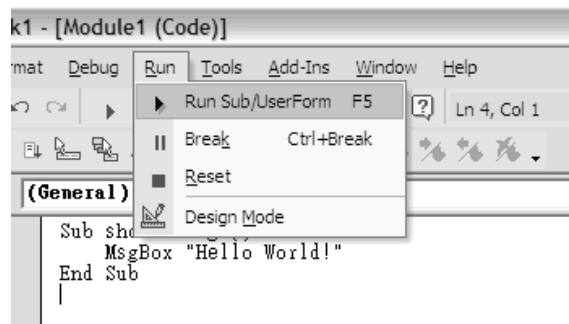


3. In the Module window, type the following:

```
Sub showMessage()  
    MsgBox "Hello World!"  
End Sub
```

4. Click the Run button, , press [F5], or go to Run..Run Sub/UserForm to run the program.

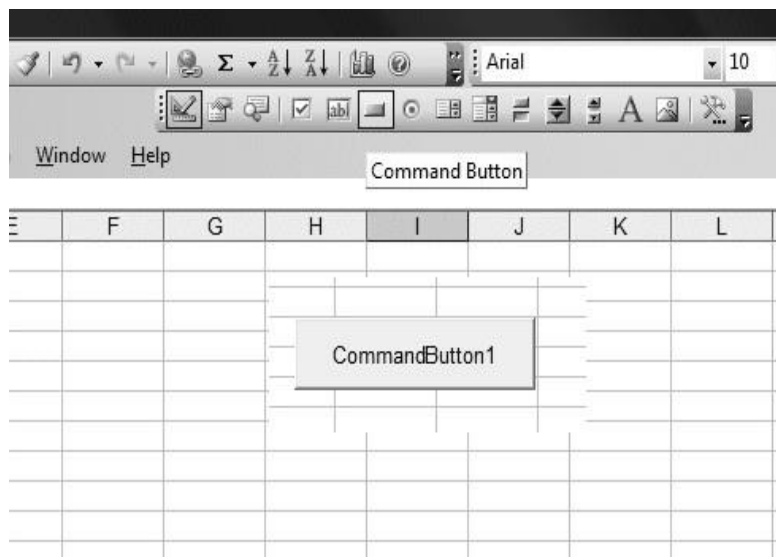
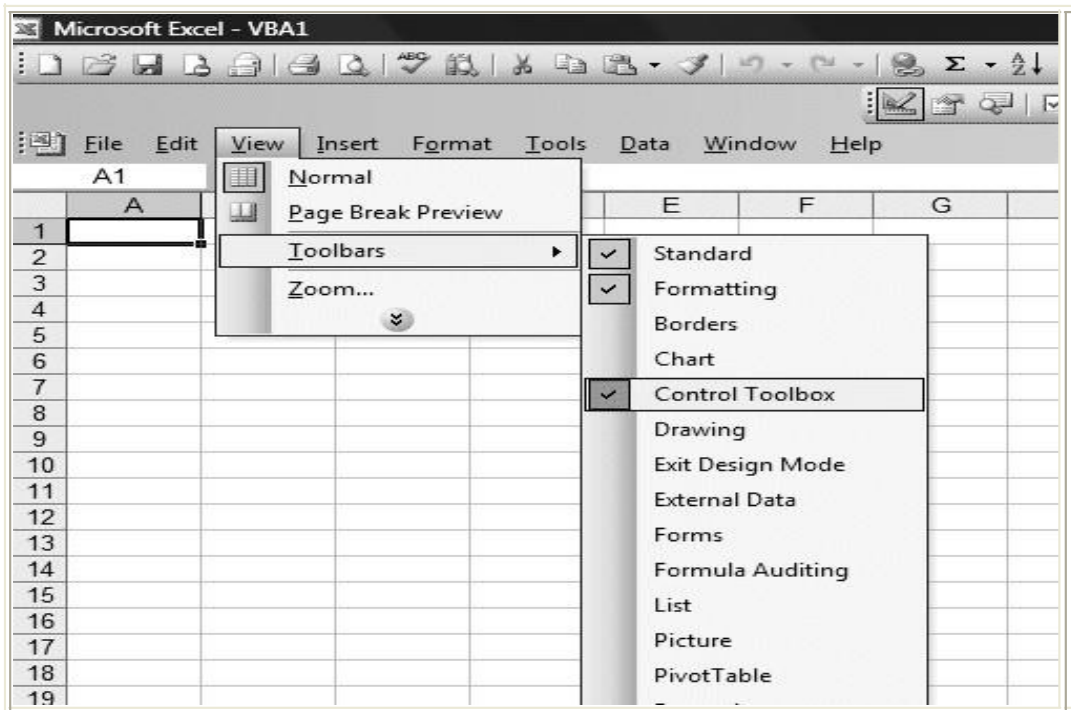


5. The message box pops up with the "Hello World!" greeting.

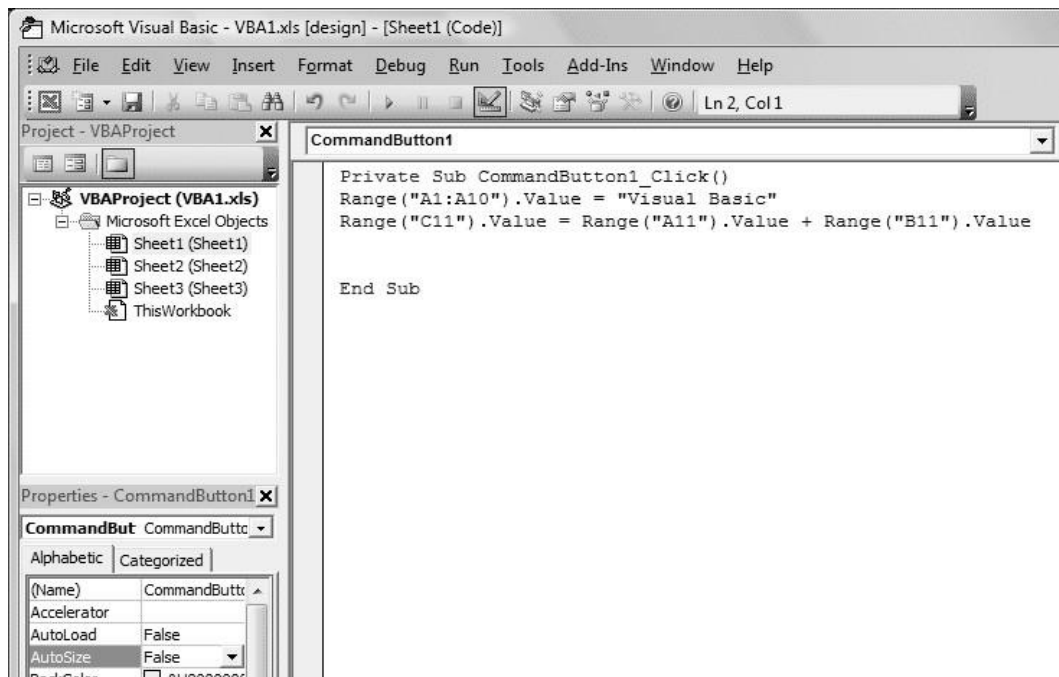


2. To place a command button on the spreadsheet and start programming by clicking the command button.

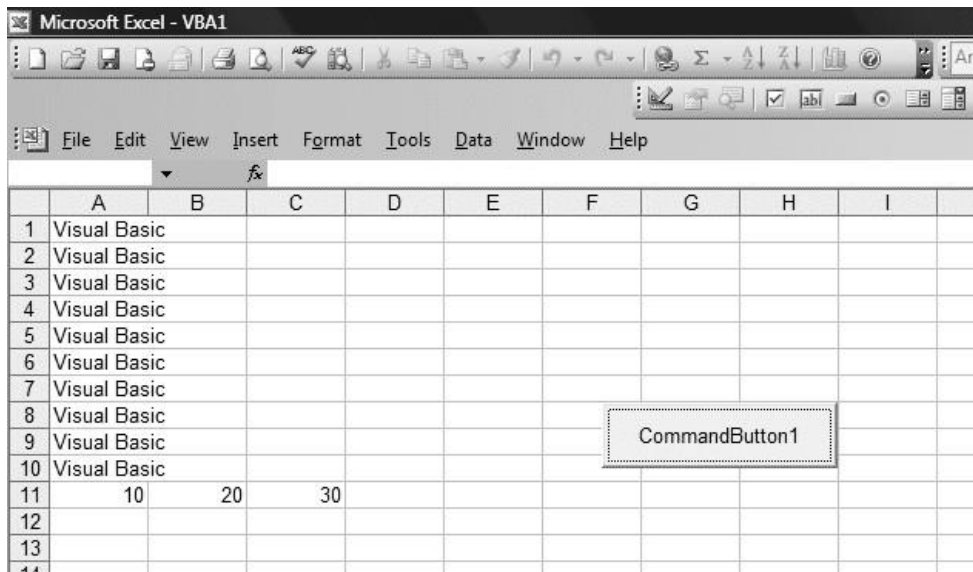
In order to place a command button on the spreadsheet, you need to click View on the MS Excel menu bar and then click on toolbar and finally select the Control Toolbox after which the control toolbox bar will appear. Then you click on the command button and draw it on the spreadsheet.



Next, you click on the command button and the Visual Basic Editor will appear. Then you enter the statement as shown in the figure. The first statement will fill up cell A1 to cell A10 with the phrase "Visual Basic" while the second statement add the value in cell A11 and cell B11 and then show the sum in cell C11. It is that simple.



The output:



Programming Structure in Excel VBA

Working with Variables in Excel VBA

The Concept of Variables

Variables are like mail boxes in the post office. The contents of the variables changes every now and then, just like the mail boxes. In VBA, variables are areas allocated by the computer memory to hold data. Like the mail boxes, each variable must be given a name. To name a variable in VBA, you have to follow a set of rules, as follows:

a) Variable Names

The following are the rules when naming the variables in VBA

- It must be less than 255 characters
- No spacing is allowed
- It must not begin with a number
- Period is not permitted

Examples of valid and invalid variable names are displayed in Table.

Valid Name	Invalid Name
My_Car	My.Car
ThisYear	1NewBoy
Long_Name_Can_beUSE	He&HisFather * & is not acceptable
Group88	Student ID * Spacing not allowed

b) Declaring Variables

In VBA, one needs to declare the variables before using them by assigning names and data types. There are many VBA data types, which can be grossly divided into two types, namely the numeric data types and non-numeric data types

i) Numeric Data Types

Numeric data types are types of data that consist of numbers, which can be computed mathematically with various standard operators such as add, minus, multiply, divide and so on. In VBA, the numeric data are divided into 7 types, which are summarized in Table.

Type	Storage	Range of Values
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,648

Single	4 bytes	-3.402823E+38 to -1.401298E-45 for negative values 1.401298E-45 to 3.402823E+38 for positive values.
Double	8 bytes	-1.79769313486232e+308 to -4.94065645841247E-324 for negative values 4.94065645841247E-324 to 1.79769313486232e+308 for positive values.
Currency	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Decimal	12 bytes	+/- 79,228,162,514,264,337,593,543,950,335 if no decimal is use +/- 7.9228162514264337593543950335 (28 decimal places).

Numeric Data Types

ii) Non-numeric Data Types

The nonnumeric data types are summarized in Table.

Data Type	Storage	Range
String(fixed length)	Length of string	1 to 65,400 characters
String(variable length)	Length + 10 bytes	0 to 2 billion characters
Date	8 bytes	January 1, 100 to December 31, 9999
Boolean	2 bytes	True or False
Object	4 bytes	Any embedded object
Variant(numeric)	16 bytes	Any value as large as Double
Variant(text)	Length+22 bytes	Same as variable-length string

Table: Nonnumeric Data Types

You can declare the variables implicitly or explicitly. For example, `sum=text1.text` means that the variable `sum` is declared implicitly and ready to receive the input in `Text1` textbox. Other examples of implicit declaration are `volume=8` and `label=|±Welcome|±`. On the other hand, for explicit declaration, variables are normally declared in the general section of the codes' windows using the `Dim` statement. The format is as follows:

`Dim variableName as DataType`

Example 2.1

`Dim password As String`

`Dim yourName As String`

`Dim firstnum As Integer`

`Dim secondnum As Integer`

`Dim total As Integer`

`Dim BirthDay As Date`

You may also combine them in one line, separating each variable with a comma, as follows:

Dim password As String, your Name As String, Fortnum As Integer.

If the data type is not specified, VB will automatically declare the variable as a Variant. For string declaration, there are two possible formats, one for the variable-length string and another for the fixed-length string. For the variable-length string, just use the same format as Example 2.1 above. However, for the fixed-length string, you have to use the format as shown below:

Dim VariableName as String * n

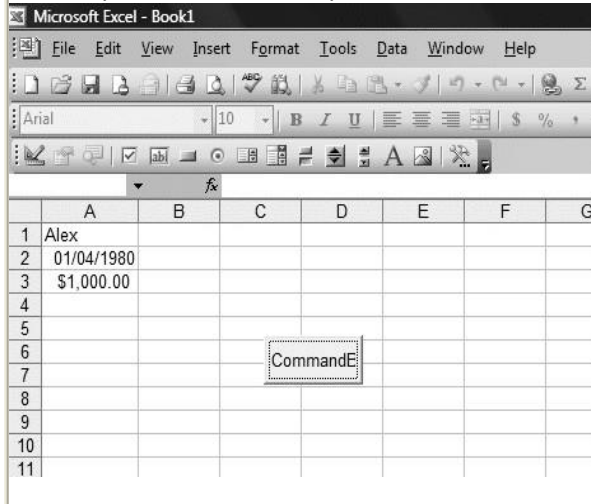
where n defines the number of characters the string can hold. For example, Dim yourName as String * 10 mean yourName can hold no more than 10 Characters.

Example

In this example, we declared three types of variables, namely the string, date and currency.

```
Private Sub CommandButton1_Click()
    Dim YourName As String
    Dim BirthDay As Date
    Dim Income As Currency
    YourName = "Alex"
    BirthDay = "1 April 1980"
    Income = 1000
    Range("A1") = YourName
    Range("A2") = BirthDay
    Range("A3") = Income
End Sub
```

The output screen of Example



Message Boxes (MsgBox) in VBA for Excel

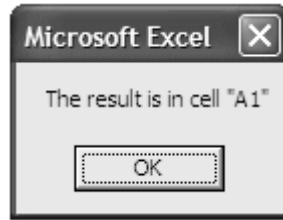
In VBA for Excel the message box (MsgBox) is the primary tool to interact with the user. You can use it to inform, alert or ask the user (with a Yes/No message box) if he is sure that he wants a certain critical procedure to run (deleting things).

The code in VBA for Excel to generate the following basic message box is:

MsgBox "Thank you"



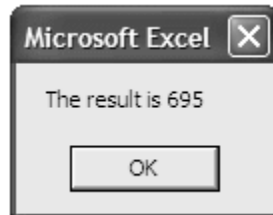
If you want you want part of your message to be between quotes you need to double the quotes as in:



The line of code would be:

```
MsgBox "The result is in cell ""A1"""
```

If you don't want to show the cell's address but its value as in:



The line of code would be:

```
MsgBox "The result is " & Range("A1").Value
```

Don't forget the space after "is" .

Input Boxes

You need to declare (create) a variable to receive the answer to a VbYesNo message box or an input box.

For the input box you will write:

```
varAnswer = InputBox("For how many years?", "Duration")
```



and for the message box you will write

```
varAnswer = MsgBox("Do you want to continue?", vbYesNo, "Alert")
```

Practice: Message Box

Yesterday I have shown that how we can display phrases in a range of cells and also perform arithmetic operations in MS Excel. Today, I shall demonstrate how we can display message boxes in a MS Excel worksheet. A message box normally act as a dialog box where users can interact with the computer, it is able to perform certain actions in response to what the user clicks or selects. The format for a message box is as follows:
 message=MsgBox(Prompt, Style Value,Title)

The first argument, Prompt, will display the message in the message box.

The Style Value determines what type of command button will appear in the message box. .

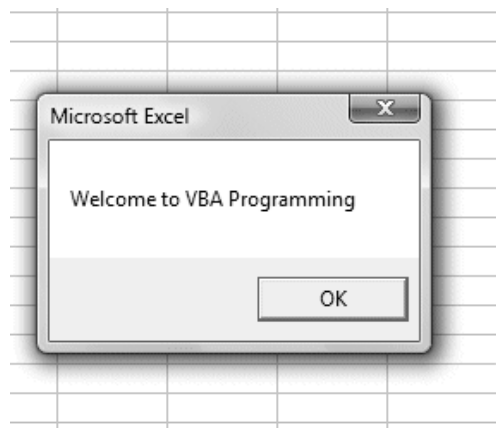
The Title argument will display the title of the message board. message is a variable that holds values that are returned by the MsgBox () function.

The values are determined by the type of buttons being clicked by the users. It has to be declared as Integer data type in the procedure or in the general declaration section.

Some examples of different types of message box

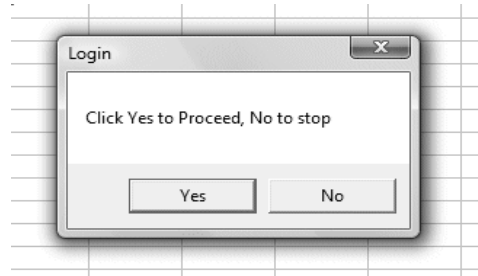
The codes are as follows:

```
Private Sub CommandButton1_Click()
MsgBox ("Welcome to VBA Programming")
End Sub
```



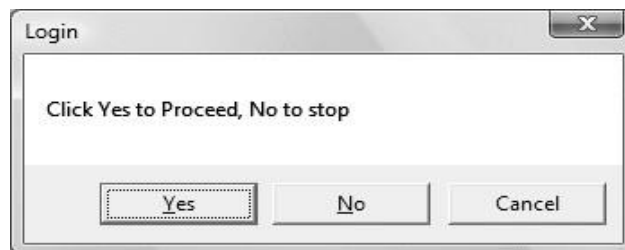
The code for message box with Yes No button:

```
Private Sub CommandButton3_Click()
Dim message As Integer
message = MsgBox("Click Yes to Proceed, No to stop", vbYesNo, "Login")
If message = 6 Then
Range("A1").Value = "You may proceed"
ActiveWorkbook.Activate
ElseIf message = 7 Then
ActiveWorkbook.Close
End If
End Sub
```

The code for message box with Yes No Cancel button:

```
Private Sub CommandButton1_Click()
Dim message As Integer
message = MsgBox("Click Yes to Proceed, No to stop", vbYesNoCancel, "Login")
If message = 6 Then
Range("A1").Value = "You may proceed"
ActiveWorkbook.Activate
ElseIf message = 7 Then
ActiveWorkbook.Close
End If
End Sub
```



Creating and Managing Array

Declaring an Array with Dim Statement

An array is a set of sequentially indexed elements having the same intrinsic data type. Each element of an array has a unique identifying index number. Changes made to one element of an array don't affect the other elements.

Before signing values to an array, the array needs to be created. You can declare the array by using the Dim statement.

For example, to declare a one-dimensional array with 5 elements, type the following:

Dim Arr(4)

The element's index of the array starts from 0 unless Option Base 1 is specified in the public area (area outside of the sub procedure). If Option Base 1 is specified, the index will start from 1.

The following example assigns values to the array and displays all values in a message box :

Option Base 1

```
Sub assignArray( )  
    Dim Arr(5)  
    Arr(1) = "Jan"  
    Arr(2) = "Feb"  
    Arr(3) = "Mar"  
    Arr(4) = "Apr"  
    Arr(5) = "May"  
    MsgBox Arr(1) & "-" & Arr(2) & "-" & Arr(3) & "-" & Arr(4) & "-" & Arr(5)  
End Sub
```



* The number inside the array, i.e. Arr(1), is the index. One (1) is the index of the first element in the array.

Resize an Array with Redim Statement

The ReDim statement is used to size or resize a dynamic array that has already been formally declared.

For example, if you have already declared an array with an element value of 5 and decided to change the number of the element to 6, you can do the following to resize the array:

```
Redim Arr(6)
```

We incorporate it into our last example:

```
Option Base 1  
Sub assignArray( )  
    'Dim Arr(5)  
    Redim Arr(6)
```

```

Arr(1) = "Jan"
Arr(2) = "Feb"
Arr(3) = "Mar"
Arr(4) = "Apr"
Arr(5) = "May"
Arr(6) = "Jun"
Msgbox Arr(1) & "-" & Arr(2) & "-" & Arr(3) & "-" & Arr(4) & "-" & Arr(5)
End Sub

```

Note that the Dim Arr(5) statement is commoned out, because leaving this original statement in the sub will causing a compile error.

Manage Dynamic Array

A word of caution in using the Redim Statement to resize an array - resize the array can erase the elements in it. In the following example, all the values assigned prior to resize the array are erased. Only the value assigned to the array after resize remains.

```

Option Base 1
Sub assignArray( )
    Redim Arr(5)
    Arr(1) = "Jan"
    Arr(2) = "Feb"
    Arr(3) = "Mar"
    Arr(4) = "Apr"
    Arr(5) = "May"
    Redim Arr(6)
    Arr(6) = "Jun"
    MsgBox Arr(1) & "-" & Arr(2) & "-" & Arr(3) & "-" & Arr(4) & "-" & Arr(5) & "-" &
Arr(6)
End Sub

```



By replace the Redim Arr(6) with Redim Preserve Arr(6), all values will remain. For example:

```

Option Base 1
Sub assignArray( )

```

```
Redim Arr(5)
```

```
Arr(1) = "Jan"
```

```
Arr(2) = "Feb"
```

```
Arr(3) = "Mar"
```

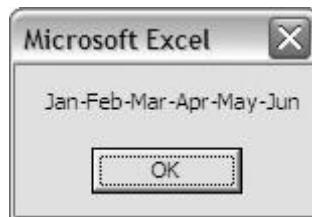
```
Arr(4) = "Apr"
```

```
Arr(5) = "May"
```

```
Redim Preserve Arr(6)
```

```
Arr(6) = "Jun"
```

```
Msgbox Arr(1) & "-" & Arr(2) & "-" & Arr(3) & "-" & Arr(4) & "-" & Arr(5) & "-"
& Arr(6)
End Sub
```



Create Multi-Dimensional Array

An array can also store multiple dimensional data. To simplify our tutorial, example on a two-dimensional array is used. Assume you have data of a local store's yearly sale in the following table and you want to store the data in a two-dimensional array:

	Year 2003	Year 2004
CD Sale	1,000	1,500
DVD Sale	1,200	2,000

First we create the array as follow:

```
Dim Arr(2,2)
```

Then we assign the values into the array. We treat the first dimension as the year and the second dimension as the product sale:

```
Arr(1,1) = 1000
```

```
Arr(1,2) = 1200
```

```
Arr(2,1) = 1500
```

```
Arr(2,2) = 2000
```

We now display the values of the array with a message box:

```
Msgbox "Sale of CD in 2003 is " & Arr(1,1) & vbCrLf & "Sale of CD in 2004 is " _ &  
Arr(2,1) & vbCrLf & "Sale of DVD in 2003 is " & Arr(1,2) & vbCrLf _& "Sale of DVD in  
2004 is " & Arr(2,2)
```

The complete procedure is as followed:

Option Base 1

```
Sub multDimArray( )
```

```
    Dim Arr(2,2)
```

```
    Arr(1,1) = 1000
```

```
    Arr(1,2) = 1200
```

```
    Arr(2,1) = 1500
```

```
    Arr(2,2) = 2000
```

```
Msgbox "Sale of CD in 2003 is " & Arr(1,1) & vbCrLf & "Sale of CD in 2004 is " _  
& Arr(2,1) & vbCrLf & "Sale of DVD in 2003 is " & Arr(1,2) & vbCrLf _& "Sale of DVD in  
2004 is " & Arr(2,2)
```

```
End Sub
```



* vbCrLf stands for VB Carriage Return Line Feed. It puts a return and a new line as shown in the

message box above. The underscore "_" on the back of the first line of the message box means

"continue to the next line"

Decision Structures

Using If.....Then....Else

Visual Basic Editor in MS Excel is just as powerful as the stand alone Visual Basic compiler in the sense that you can use the same commands in programming. For example, you can use If..Then...Else to control program flow and display certain output based on certain conditions in MS Excel. Here, I am going to demonstrate the concept using one example.

IF ... Then Statement

The IF ... Then is a single condition and run a single statement or a block of statement.

Example, the following statement set variable Status to "Adult" if the statement is true:

```
If Age >= 18 Then Status = "Adult"
```

You can also use multiple-line block in the If statement as followed:

```
If Age >= 18 Then  
    Status = "Adult"  
    Vote = "Yes"  
End If
```

Note that in the multiple-line block case, End If statement is needed, where the single-line case does not.

IF ... Then ... Else

The If ... Then ... Else statement is used to define two blocks of conditions - true and false.

Example:

```
If Age >= 22 Then  
    Drink = "Yes"  
Else  
    Drink = "No"  
End If
```

Again, note that End If statement is needed in this case as well since there is more than one block of statements.

Nested IF

It is possible to nest multiple IF functions within one Excel formula. You can nest up to 7 IF functions to create a complex IF THEN ELSE statement.

The IF ... Then ... Elseif is used to test additional conditions without using new If ... Then statements.

For Example:

```

If Age >= 18 and Age < 22 Then
    MsgBox "You can vote"
Elseif Age >=22 and Age < 62 Then
    MsgBox "You can drink and vote"
Elseif Age >=62 Then
    MsgBox "You are eligible to apply for Social Security Benefit"
Else
    MsgBox "You cannot drink or vote"
End If

```

Note that the last condition under Else is, implicitly, Age < 18.

Let's take one more example :

The Code

```

Private Sub CommandButton1_Click()
    Dim mark As Integer
    Dim grade As String
    Randomize Timer
    mark = Int(Rnd * 100)
    Cells(1, 1).Value = mark
    If mark < 20 And mark >= 0 Then
        grade = "F"
    Cells(2, 1).Value = grade
    Elseif mark < 30 And mark >= 20 Then
        grade = "E"
    Cells(2, 1).Value = grade
    Elseif mark < 40 And mark >= 30 Then
        grade = "D"
    Cells(2, 1).Value = grade
    Elseif mark < 50 And mark >= 40 Then
        grade = "C-"
    Cells(2, 1).Value = grade
    Elseif mark < 60 And mark >= 50 Then
        grade = "C"
    Cells(2, 1).Value = grade
    Elseif mark < 70 And mark >= 60 Then
        grade = "C+"
    Cells(2, 1).Value = grade
    Elseif mark < 80 And mark >= 70 Then

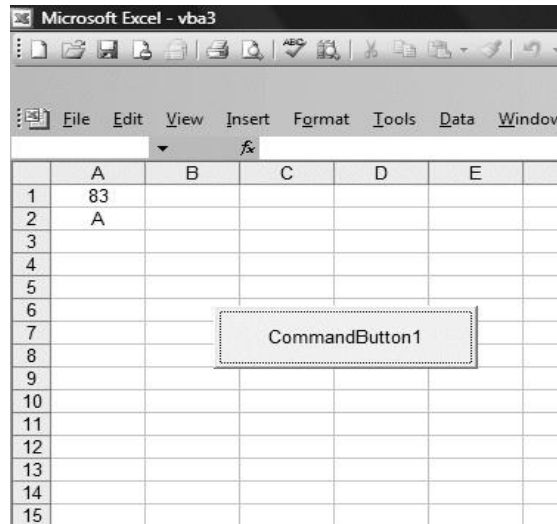
```

```

grade = "B"
Cells(2, 1).Value = grade
ElseIf mark <= 100 And mark > -80 Then
grade = "A"
Cells(2, 1).Value = grade
End If

```

Output :



Select Case.....End Select

Normally it is sufficient to use the conditional statement If....Then....Else for multiple options or selections programs. However, if there are too many different cases, the If...Then...Else structure could become too bulky and difficult to debug if problems arise. Fortunately, Visual Basic provides another way to handle complex multiple choice cases, that is, the Select Case.....End

In short Select Case statement is an alternative to the Elself statement. This method is more efficient and readable in coding the If ... Then ... Elself statment.

Select Case...End Select structure is as follow:

Select Case variable

Case value 1

Statement

Case value 2

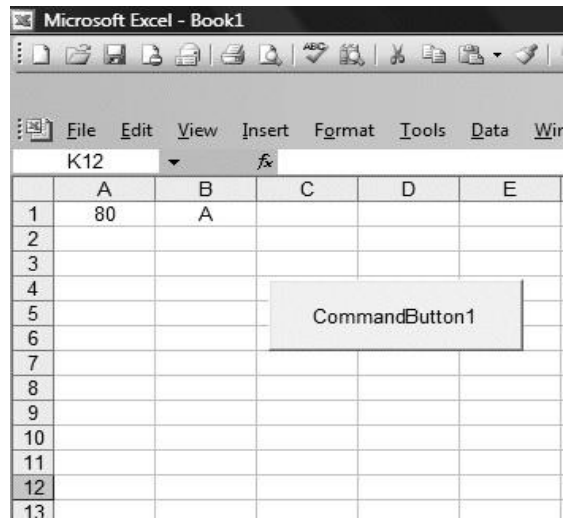
Statement

Case value 3

Statement


```
Private Sub CommandButton1_Click()  
Dim mark As Single  
Dim grade As String  
mark = Cells(1, 1).Value  
'To set the alignment to center  
Range("A1:B1").Select  
With Selection  
.HorizontalAlignment = xlCenter  
End With  
  
Select Case mark  
Case 0 To 20  
grade = "F"  
Cells(1, 2) = grade  
Case 20 To 29  
grade = "E"  
Cells(1, 2) = grade  
Case 30 To 39  
grade = "D"  
Cells(1, 2) = grade  
Case 40 To 59  
grade = "C"  
Cells(1, 2) = grade  
Case 60 To 79  
grade = "B"  
Cells(1, 2) = grade  
Case 80 To 100  
grade = "A"  
Cells(1, 2) = grade  
Case Else  
grade = "Error!"  
Cells(1, 2) = grade  
End Select  
End Sub
```

The diagram on the lower left illustrates the output of this example.



Explanation:

To set the cell align alignment to center, we use the following procedure:

Range("A1:B1").Select

With Selection

.HorizontalAlignment = xlCenter

End With

We can use the statement case value1 to value 2 to specify the range of values that fulfill the particular case.

You should also include the error case where the values entered are out of the range or invalid. For example, if the examination mark is from 0 to 100, then any value out of this range is invalid. In this program, I use case else to handle the error entries.

One More Example:

Select Case Grade

Case Is >= 90

LetterGrade = "A"

Case Is >= 80

LetterGrade = "B"

Case Is >= 70

LetterGrade = "C"

Case Is >= 60

LetterGrade = "D"

Case Else

LetterGrade = "Sorry"

End Select

Loop Structures

This feature makes repetitive works easier.

For ... Next

Use For ... Next loop if the number of loops is already defined and known. A For ... Next loop uses a counter variable that increases or decreases in value during each iteration of the loop. This loop structure is being used the most for our examples on this site.

Here is an example of the For ... Next loop:

```
For i = 1 to 10  
    Cells(i, 1) = i  
Next i
```

	A
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	
12	

In this VBA program, you place the command button 1 on the spreadsheet then click on it to go into the Visual Basic editor. When you click on the button, the VBA program will fill cells(1,1) with the value of 1, cells(2,1) with the value of 2, cells(3,1) with the value of 3.....until cells (10,1) with the value of 10. The position of each cell in the Excel spreadsheet is referenced with cells(i,j), where i represents row and j represent column.

Note that the counter variable i, by default, increase by an increment of 1.

For ... Next Loop With Step

You can use the Step Keyword to specify a different increment for the counter variable.

For example:

```
For i = 1 to 10 Step 2  
    Cells(i,1) = i  
Next i
```

This looping process will print values with an increment of 2 on row 1, 3, 5, 7 and 9 on column one.

	A
1	1
2	
3	3
4	
5	5
6	
7	7
8	
9	9
10	

You can also have decrement in the loop by assign a negative value after the Step keyword.

For example:

For i = 10 to 1 Step -2

Cells(i,1) = i

Next i

This looping process will print values with an increment of -2 starts from 10 on row 10, 8, 6, 4 and 2 on column one.

	A
1	
2	2
3	
4	4
5	
6	6
7	
8	8
9	
10	10

DO.....LOOP

In the previous chapter, you have learned to use the For.....Next loop to execute a repetitive process. In this chapter, you will learn about another looping method known as the Do Loop. There are four ways you can use the Do Loop as shown below.

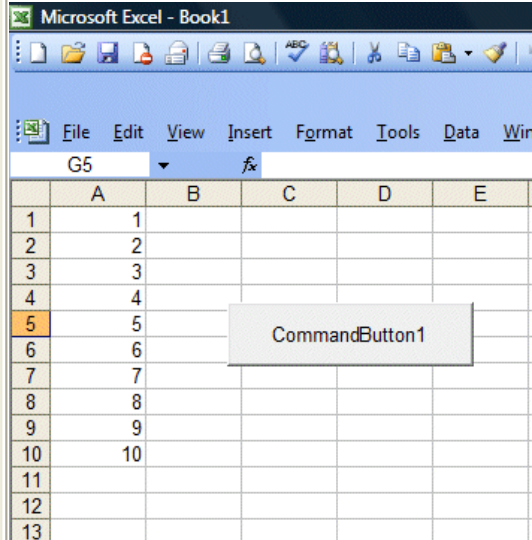
ANIMATIX COMPUTER EDUCATION

- i) Do.....Loop While
- (ii) Do until.....Loop
- (iii) Do while.....Loop
- (iv) Do.....Loop until

<p>Example 1</p> <pre>Private Sub CommandButton1_Click() Dim counter As Integer Do counter = counter + 1 Cells(counter, 1) = counter Loop While counter < 10 End Sub</pre> <p>In this example, the program will keep on adding 1 to the preceding counter value as long as the counter value is less than 10. It displays 1 in cells(1,1), 2 in cells(2,1) i.e. until 10 in cells (10,1).</p>	<p>Example 2</p> <pre>Private Sub CommandButton1_Click() Dim counter As Integer Do Until counter = 10 counter = counter + 1 Cells(counter, 1) = 11 - counter Loop End Sub</pre> <p>ii In this example, the program will keep on adding 1 to the preceding counter value until the counter value reaches 10. It displays 10 in cells(1,1), 9 in cells(2,1) i.e. until 1 in cells (10,1).</p>	<p>Example 3</p> <pre>Private Sub CommandButton1_Click() Dim counter, sum As Integer 'To set the alignment to center Range("A1:C11").Select With Selection .HorizontalAlignment = xlCenter End With Cells(1, 1) = "X" Cells(1, 2) = "Y" Cells(1, 3) = "X+Y" Do While counter < 10 counter = counter + 1 Cells(counter + 1, 1) = counter Cells(counter + 1, 2) = counter * 2 sum = Cells(counter + 1, 1) + Cells(counter + 1, 2) Cells(counter + 1, 3) = sum Loop End Sub</pre> <p>In this example, the program will display the values of X in cells(1,1) to cells(11,1). The value of Y is X2 and the values are display in column 2, i.e. from cells(2,1) to cells(2,11). Finally, it shows the values of X+Y in column 3, i.e. from cells(3,1) to cells(3,11) ii</p>
---	--	---

ANIMATIX COMPUTER EDUCATION

Example 1



Microsoft Excel - Book1

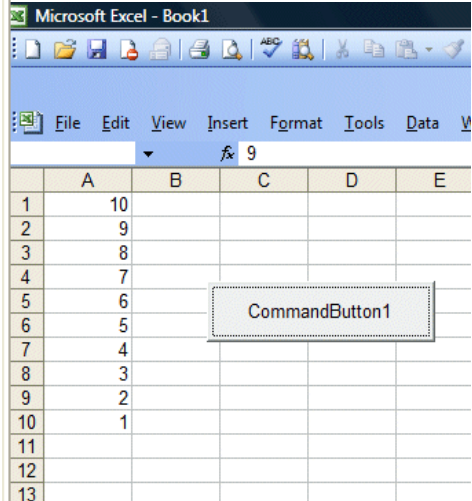
File Edit View Insert Format Tools Data Win

G5 fx

	A	B	C	D	E
1	1				
2	2				
3	3				
4	4				
5	5				
6	6				
7	7				
8	8				
9	9				
10	10				
11					
12					
13					

CommandButton1

Example 2



Microsoft Excel - Book1

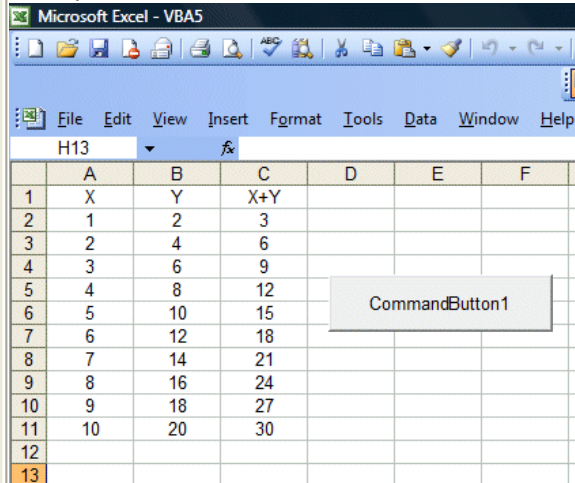
File Edit View Insert Format Tools Data V

fx 9

	A	B	C	D	E
1	10				
2	9				
3	8				
4	7				
5	6				
6	5				
7	4				
8	3				
9	2				
10	1				
11					
12					
13					

CommandButton1

Example 3



Microsoft Excel - VBA5

File Edit View Insert Format Tools Data Window Help

H13 fx

	A	B	C	D	E	F
1	X	Y	X+Y			
2	1	2	3			
3	2	4	6			
4	3	6	9			
5	4	8	12			
6	5	10	15			
7	6	12	18			
8	7	14	21			
9	8	16	24			
10	9	18	27			
11	10	20	30			
12						
13						

CommandButton1

One thing to be caution is that sometimes the loop might be a infinite loop. And it happens when the condition never becomes false. In such case, you can stop the loop by press [ESC] or [CTRL] + [BREAK].

Font and Background Color

Now we will explore how to create VBA that can format the color of a MS Excel spreadsheet. Using Visual Basic codes, we can actually change the font color as well as the the background color of each cell effortlessly.

Alright, I am going to create a program that can create random font and background colors using a randomize process. Colors can be assigned using a number of methods in VBA, but it is easier to use the RGB function. The RGB function has three numbers corresponding to the red, green and blue components. The range of values of the three numbers is from 0 to 255. A mixture of the three primary colors will produce different colors.

The format to set the font color is

`cells(i,j).Font.Color=RGB(x,y,x)`, where x , y , z can be any number between 1 and 255

For example

`cells(1,1).Font.Color=RGB(255,255,0)` will change the font color to yellow

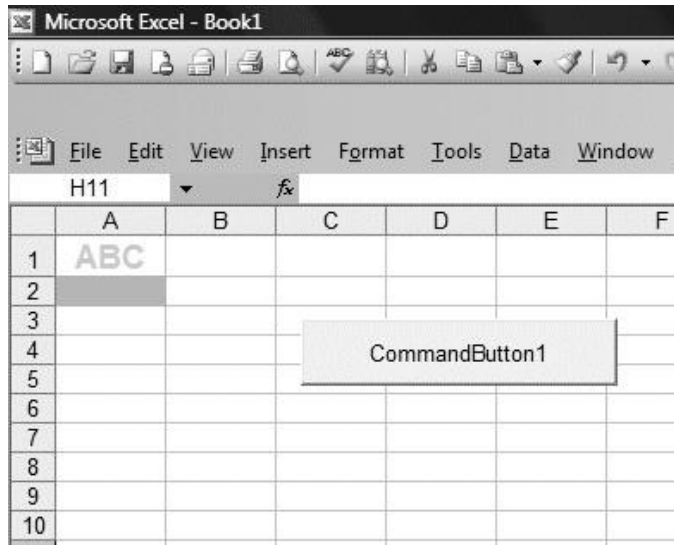
The format to set the cell's background color is

`cells(i,j).Interior.Color=RGB(x,y,x)`, where x , y , z can be any number between 1 and 255

In the following example, the font color in cells(1,1) and background color in cells(2,1) are changing for every click of the command button due to the randomized process.

Private Sub CommandButton1_Click() Randomize Timer Dim i, j, k As Integer i = Int(255 * Rnd) + 1 j = Int(255 * Rnd) + 1 k = Int(255 * Rnd) + 1 Cells(1, 1).Font.Color = RGB(i, j, k) Cells(2, 1).Interior.Color = RGB(j, k, i) End Sub	Explanation: Rnd is a random number between 0 and 1 255* Rnd will produce a number between 0 and 255 Int(255*Rnd) will produce integers that take the values from 0 to 254 So we need to add 1 to get random integers from 0 to 255. example; Rnd=0.229 255*Rnd=58.395 Int(58.395)=58
---	---

ANIMATIX COMPUTER EDUCATION



Modules and Procedures

A procedure is defined as a named group of statements that are run as a unit. A statement is simply 1 complete line of code. VBA procedures are used to perform tasks such as controlling Excel's environment, communicating with databases, calculating equations, analyzing data...etc .

VBA has two types of procedures:

Sub procedures and
Function procedures.

Sub Procedures

Sub procedures are written when you want to command Excel like creating a chart, analyzing data, coloring cells, copying and pasting data...etc.

Function Procedures

Function procedures are created when you want to make your own custom worksheet functions or perform a calculation that will be used over and over again. Note that Sub procedures can also do calculations.

Modules and Procedures and Their Scope

A module is a container for procedures as shown in our prior examples. A procedure is a unit of code enclosed either between the Sub and End Sub statement or between the Function and End Function statements.

ANIMATIX COMPUTER EDUCATION

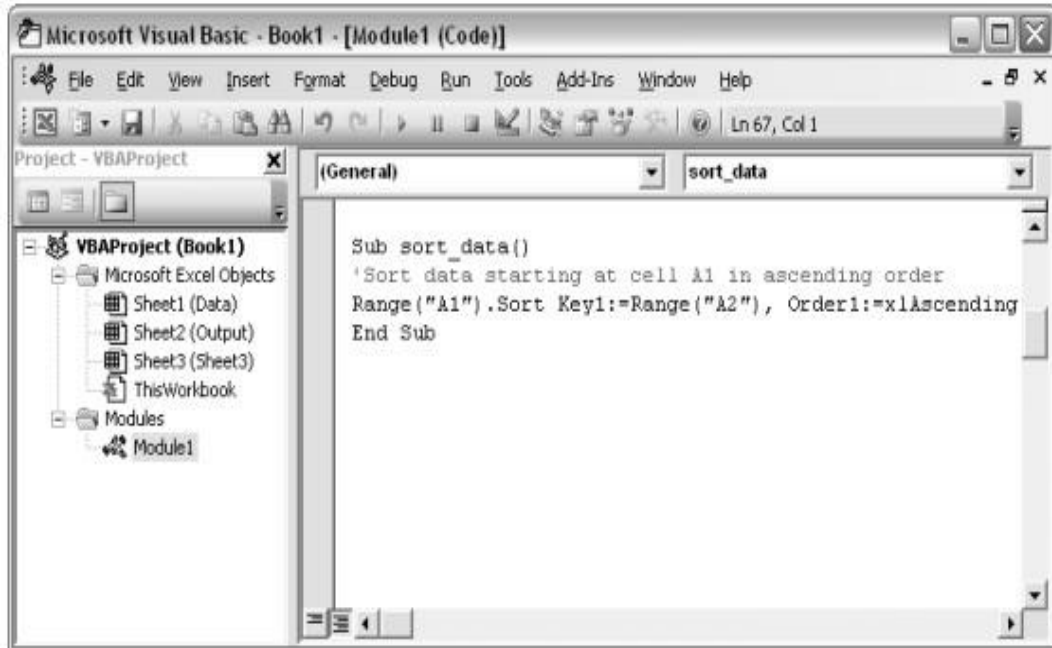
The following sub procedure (or sub routine) print the current date and time on cell C1:

```
Sub ShowTime()  
    Range("C1") = Now()  
End Sub
```

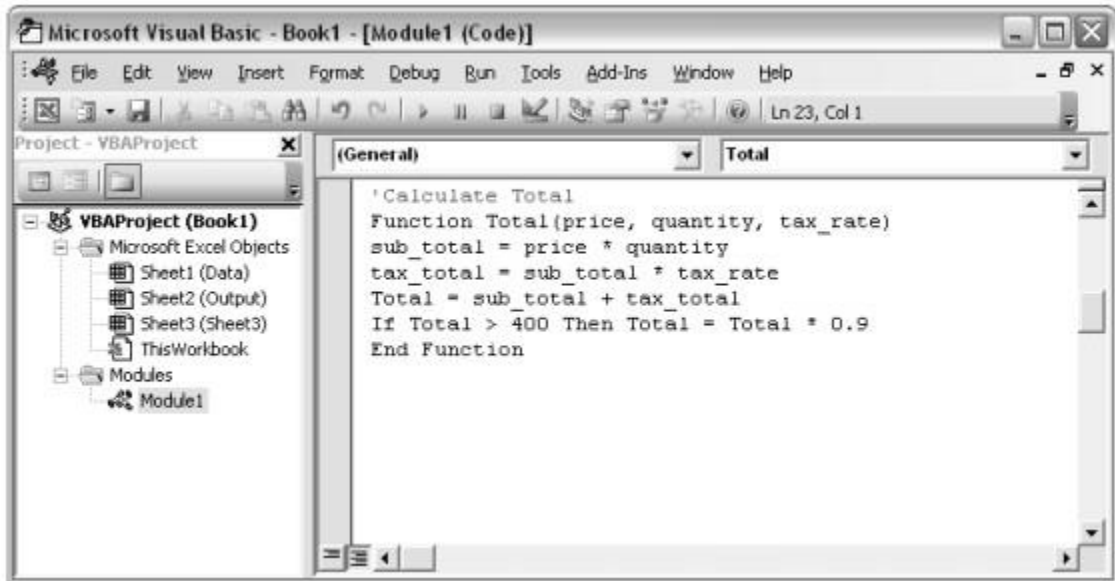
The following function sum up two numbers:

```
Function sumNo(x, y)  
    sumNo = x + y  
End Function
```

A Procedure:



A Function:



Excel VBA Procedures-Functions

The Concept of Functions

A VBA procedure is a block of code that performs certain tasks. We have actually learned about VBA procedures in our previous chapters, but all of them are event procedures. Event procedures are VBA programs that are associated with VBA objects such as command buttons, checkboxes, and radio buttons. However, we can also create procedures that are independent from the event procedures. They are normally called into the event procedures to perform certain tasks. There are two types of the aforementioned procedures, namely Functions and Sub Procedures. In this chapter, we will discuss functions. We will deal with Sub Procedures in the next chapter.

Types of Functions

There are two types of Excel VBA functions; one is the built-in functions while the other one is the user-defined functions. We can use built-in functions in Excel for automatic calculations. Some of the Excel VBA built-in functions are Sum, Average, Min (to find the minimum value in a range), Max (To find the maximum value in a range), Mode, Median and more. However, built-in functions can only perform some basic calculations, for more complex calculations, user-defined functions are often required. User-defined functions are procedures created independently from the event procedures. A Function can receive arguments passed to it from the event procedure and then return a value in the function name. It is normally used to perform certain calculations.

Writing Function Code

ANIMATIX COMPUTER EDUCATION

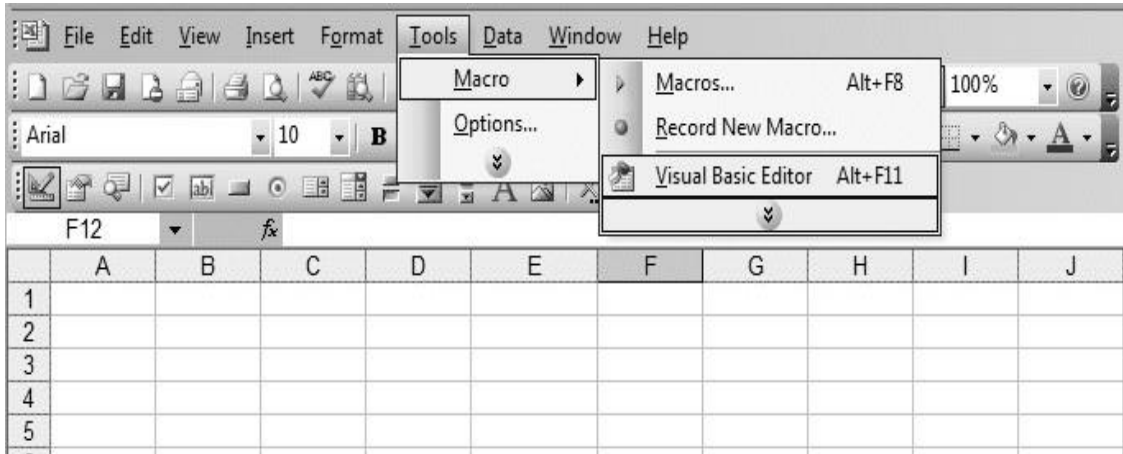
VBA Function begins with a Function statement and ends with an End Function statement. The program structure of a Function is as follows:

Function FunctionName (arguments) As DataType

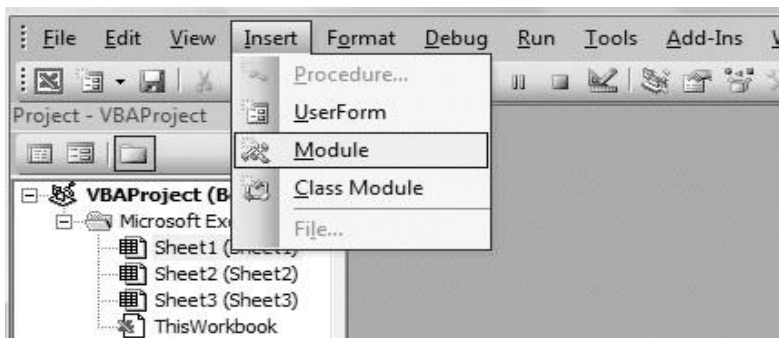
Statements

End Function

In Excel VBA, when you type the Function statement, the End Function statement will automatically appear.

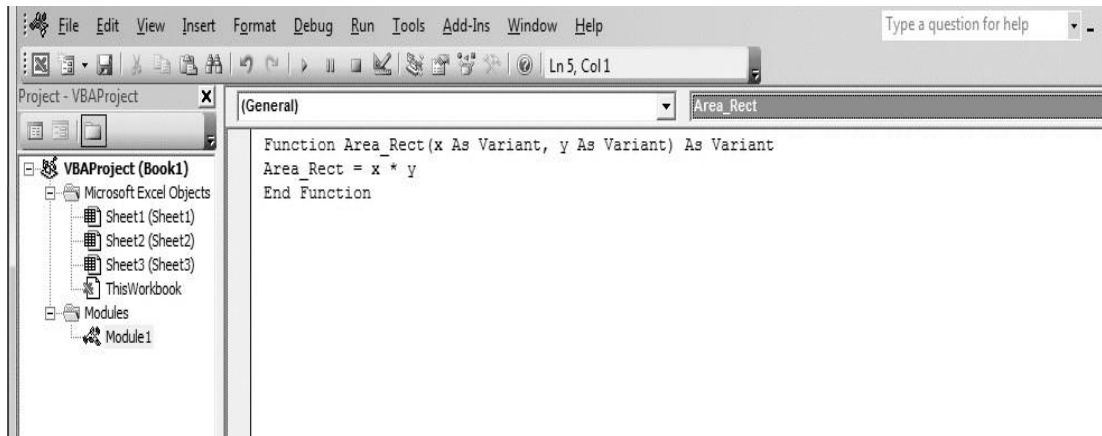


In the Visual Basic Editor window, insert a module by clicking Insert on the menu bar, and then click on Module, as shown in Figure.

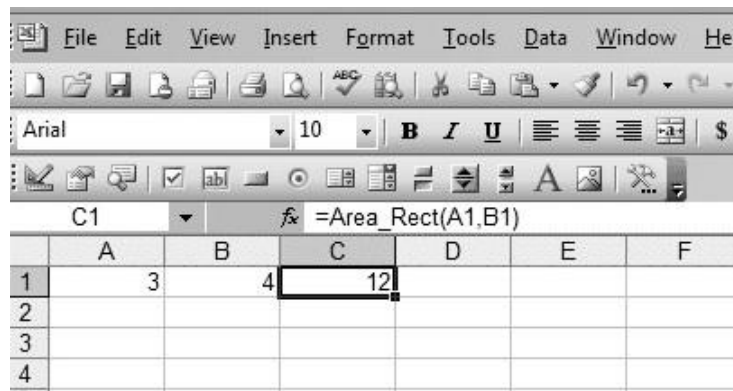


In the module environment, key in the function code for the function Area_Rect , as shown in the diagram below.

ANIMATIX COMPUTER EDUCATION

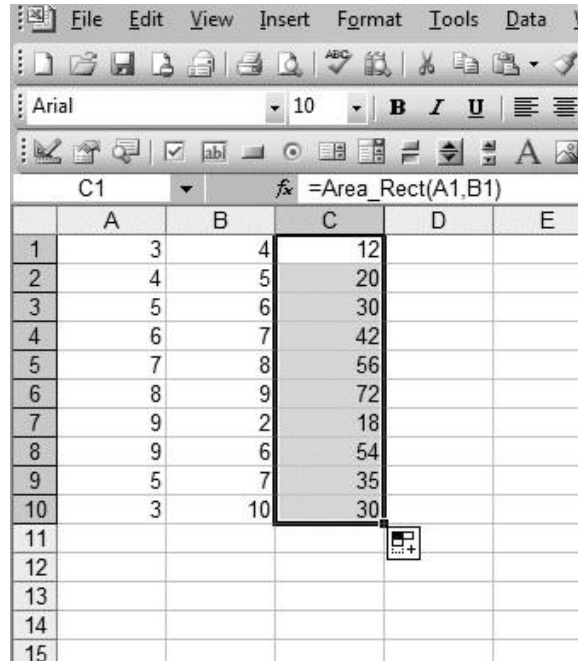


Now, you can return to the Excel spreadsheet and enter the function in any cell. In this Example, the function is entered in cell C1 and the values of width and height are entered in cell A1 and cell B1 respectively. Notice that the value of area is automatically calculated and displayed in cell C1.



The formula can be copied and updated to other cells by using the Autofill method, i.e. by dragging the place holder on the bottom right corner of the cell, as shown in Figure below.

ANIMATIX COMPUTER EDUCATION



	A	B	C	D	E
1	3	4	12		
2	4	5	20		
3	5	6	30		
4	6	7	42		
5	7	8	56		
6	8	9	72		
7	9	2	18		
8	9	6	54		
9	5	7	35		
10	3	10	30		
11					
12					
13					
14					
15					

Few Examples:

Methods normally do something or perform certain operations. For example, ClearContents is a method of the range object. It clears the contents of a cell or a range of cells. You can write the following code to clear the contents:

```
Private Sub CommandButton1_Click()  
Range(A1:A6).ClearContents  
End Sub
```

You can also let the user select his own range of cells and clear the contents by using the InputBox function, as shown in Example

```
Private Sub CommandButton1_Click()  
Dim, selectedRng As String  
selectedRng = InputBox("Enter your range")  
Range(selectedRng).ClearContents  
End Sub
```

In order to clear the contents of the entire worksheet, you can use the following code:
Sheet1.Cells.ClearContents

But if you only want to clear the formats of an entire worksheet, you can use the following syntax:

```
Sheet1.Cells.ClearFormats
```

ANIMATIX COMPUTER EDUCATION

To select a range of cells, you can use the Select method. This method selects a range of cells specified by the Range object. The syntax is

```
Private Sub CommandButton1_Click()
```

```
Range("A1:A5").Select
```

```
End Sub
```

This example allows the user to specifies the range of cells to be selected.

```
Private Sub CommandButton1_Click()
```

```
Dim selectedRng As String
```

```
selectedRng = InputBox("Enter your range")
```

```
Range(selectedRng).Select
```

```
End Sub
```

To deselect the selected range, we can use the Clear method.

```
Range(CiRj:CmRn).Clear
```

In this example, we insert two command buttons, the first one is to select the range and the second one is to deselect the selected range.

```
Private Sub CommandButton1_Click()
```

```
Range("A1:A5").Select
```

```
End Sub
```

```
Private Sub CommandButton2_Click()
```

```
Range("A1:A5").Clear
```

```
End Sub
```

Instead of using the Clear method, you can also use the ClearContents method.

Another very useful method is the Autofill method. This method performs an Autofill on the cells in the specified range with a series of items including numbers, days of week, months of year and more. The format is

```
Expression.AutoFill(Destination, Type)
```

Where Expression can be an object or a variable that returns an object. Destination means the required Range object of the cells to be filled. The destination must include the source range. Type means type of series, such as days of week, month of year and more. The AutoFill type constant is something like XlFillWeekdays, XlFillDays, XlFillMonths and more.

Example

```
Private Sub CommandButton1_Click()
```

```
Range(A1)=1
```

```
Range(A2)=2
```

```
Range("A1:A2").AutoFill Destination:=Range("A1:A10")
```

```
End Sub
```

ANIMATIX COMPUTER EDUCATION

In this example, the source range is A1 to A2. When the user clicks on the command button, the program will first fill cell A1 with 1 and cell A2 with 2, and then automatically fills the Range A1 to A10 with a series of numbers from 1 to 10.

Example

```
Private Sub CommandButton1_Click()  
Cells(1, 1).Value = "monday"  
Cells(2, 1).Value = "Tuesday"  
Range("A1:A2").AutoFill Destination:=Range("A1:A10"), Type:=xlFillDays  
End Sub
```

This example allows the user to select the range of cells to be automatically filled using the Autofill method. This can be achieved with the use of the InputBox. Since each time we want to autofill a new range, we need to clear the contents of the entire worksheet using the Sheet1.Cells.ClearContents statement.

```
Private Sub CommandButton1_Click()  
Dim selectedRng As String  
Sheet1.Cells.ClearContents  
selectedRng = InputBox("Enter your range")  
Range("A1") = 1  
Range("A2") = 2  
Range("A1:A2").AutoFill Destination:=Range(selectedRng)
```

Calling Sub Procedures and Function Procedures

There are two ways to call a sub procedure. The following example shows how a sub procedure can be called by other sub procedures.

```
Sub z(a)  
    MsgBox a  
End Sub
```

```
Sub x()  
    Call z("ABC")  
End Sub
```

```
Sub y()  
    z "ABC"  
End Sub
```

Sub z procedure takes an argument (a) and display the argument value ("ABC") in a message box. Running either Sub x or Sub y will yield the same result.

ANIMATIX COMPUTER EDUCATION

The following example calls a function procedure from a sub procedure.
The ShowSum sub procedure calls the sumNo function and returns an "8" in a message box.

```
Sub ShowSum()  
    msgbox sumNo(3,5)  
End Sub
```

```
Function sumNo(x, y)  
    sumNo = x + y  
End Function
```

If there are procedures with duplicate names in different modules, you must need to include a module qualifier before the procedure name when calling the procedure.

Objects and Collections

Workbook and Worksheet Object

Workbooks

Each object contains its own methods and properties.

A Property represents a built-in or user-defined characteristic of the object. A method is an action that you perform with an object. Below are examples of a method and a property for the Workbook Object:

Workbooks. Close
Close method close the active workbook

Workbooks. Count
Count Property returns the number of workbooks that are currently opened

A lot of VBA beginners start their career using Cells. For example:

Cells(1,1).Select is the same thing as Range("A1").Select and
Cells(11,31).Select is the same as Range("AE11").Select.

I strongly recommend that you use Range instead of Cells to work with cells and groups of cells. It makes your sentences much clearer and you are not forced to remember that column AE is column 31.

ANIMATIX COMPUTER EDUCATION

The only time that you will use Cells is when you want to select all the cells of a worksheet like in:

`Cells.Select`

To select all cells and then to empty all cells of values or formulas you will use:

`Cells.ClearContents`

worksheets

You access a worksheet named "Balance" with:

`Sheets("Balance").Select`

Note that the word " Sheets" is plural and never forget the quotes within the parenthesis

You cannot select a sheet that is hidden so you will need to write:

`Sheets("Balance").Visible= True`

`Sheets("Balance").Select`

and then if you want to hide the sheet again:

`Sheets("Balance").Visible= False`

The name of a sheet must not have more than 31 characters and cannot include certain special characters like ? : \ / [] . If you don't respect these rules your procedure will crash. The following lines of code will generate an error message:

`Sheets("Sheet1").Name= "Balance and Introduction to Numbers"` because there are more than 31 characters including the spaces

`Sheets("Sheet1").Name= "Balance: Introduction"` because of the special character :

`Sheets("Sheet1").Name= ""` because the name cannot be blank

You cannot go directly from a sheet to a cell on another sheet. For example if the active sheet is "Balance" and you want to go to cell A1 of a sheet named "Results" you cannot write:

`Sheets("Results").Range("A1").Select`

You must take two steps:

`Sheets("Results").Select`

`Range("A1").Select`

Range Object and Cells Property

We have already seen basics for the range object. Let's learn few more properties.

To select a single cell you will write:

`Range("A1").Select`

ANIMATIX COMPUTER EDUCATION

To select a set of contiguous cells you will use the colon and write:

```
Range("A1:G5").Select
```

To select a set of non contiguous cells you will use the comma and write:

```
Range("A1,A5,B4").Select
```

To select a set of non contiguous cells and a range you will use both the colon and the comma:

```
Range("A1,A5,B4:B8").Select
```

Some objects have default properties. For example, Range's default property is Value.

The following yields the same outcome.

```
Range("A1") = 1 and Range("A1").Value = 1
```

Here are examples on how to set and to get a Range property value:

The following sets the value of range A1 or Cells(1,1) as "2005". It actually prints "2005" on A1.

```
Range("A1").Value = 2005
```

The following gets the value from range A1 or Cells(1,1).

```
X = Range("A1").Value
```

Method can be used with or without argument(s).

The following two examples demonstrate this behavior.

Methods That Take No Arguments:

```
Worksheets("Sheet").Column("A:B").AutoFit
```

Methods That Take Arguments:

```
Worksheets("Sheet1").Range("A1:A10").Sort _
```

```
Worksheets("Sheet1").Range("A1")
```

Worksheets("Sheet1").Range("A1") is the Key (or column) to sort by.

Sometime a method takes more than one argument. For example, the Open method for the Workbook object, takes 12 arguments. To open a workbook with password protection, you would need to write the following code:

```
Workbooks.Open "Book1.xls", , , , "pswd"
```

Since this method takes so many arguments, it is easy to misplace the password argument. To

overcome this potential problem, one can use named arguments like the following example:

```
Workbook.Open fileName:="Book1.xls", password:="pswd"
```

ANIMATIX COMPUTER EDUCATION

You can also assign an object to an object variable using the Set Statement.

For example:

```
Dim myRange as Range  
Set myRange = Range("A1:A10")
```

Project Assignments

Creating a Financial Calculator

Excel VBA can be programmed to perform complex financial calculations. Here we have created a financial calculator to calculate the periodic payment for a loan taken from the bank. Below is the Excel VBA code for the financial calculator and its output interface.

```
Private Sub CommandButton1_Click()
```

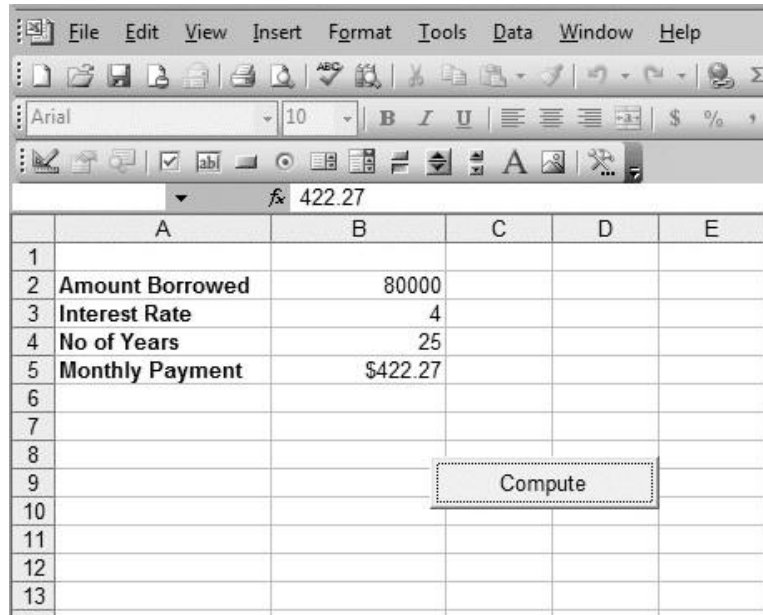
```
Dim N As Integer  
Dim p, pmt, rate, I, PVIFA As Double  
p = Cells(2, 2)  
rate = Cells(3, 2)  
N = Cells(4, 2) * 12  
I = (rate / 100) / 12  
PVIFA = 1 / I - 1 / (I * (1 + I) ^ N)  
pmt = p / PVIFA  
Cells(5, 2) = Format(pmt, "$#,##0.00")
```

```
End Sub
```

The formula to calculate periodic payment is $\text{payment} = \text{Initial Principal} / \text{PVIFA}$, where PVIFA is known as present value interest factor for an annuity. The formula to compute PVIFA is $1/i - 1/i(1+i)^n$ where n is the number of payments. Normally you can check up a financial table for the value of PVIFA and then calculate the payments manually.

The function Format is to determine the number of decimal places and the use of the \$ sign.

ANIMATIX COMPUTER EDUCATION



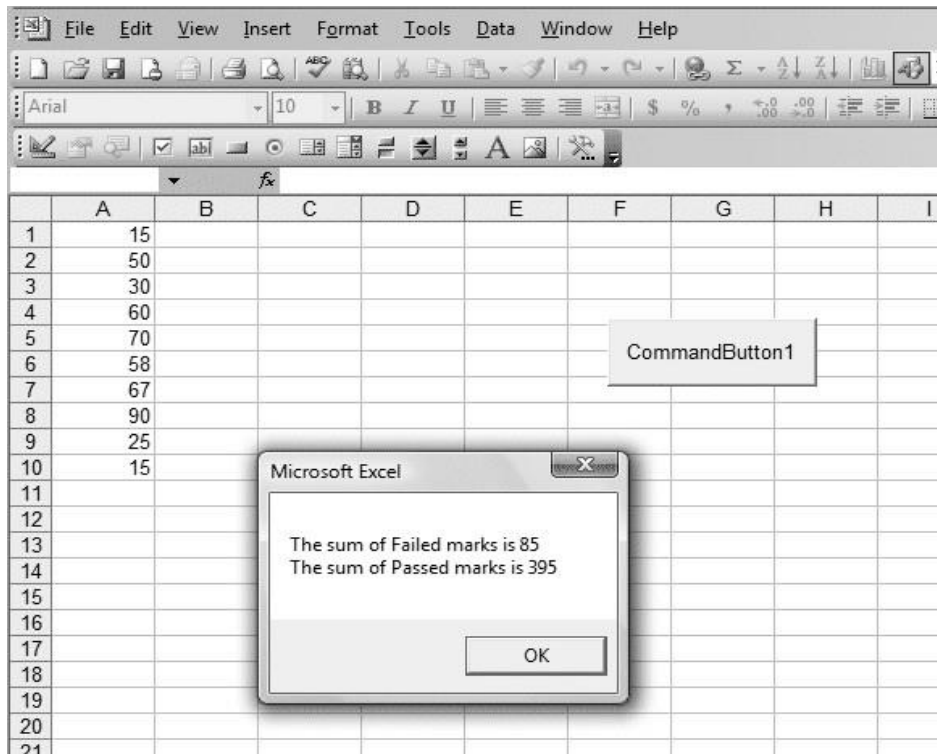
Selective Summation using VBA

In the assignment we have created a VBA that can perform selective summation according to a set of conditions. For example, you might just want to sum up those figures that have achieved sales target and vice versa. The VBA program I am showing you can sum up marks that are below 50 (which considered as failed) as well as those marks which are above 50 (which considered as passed). Here is the program

```
Private Sub CommandButton1_Click()  
Dim rng As Range, i As Integer  
Dim mark, sumFail, sumPass As Single  
sumFail = 0  
sumPass = 0  
Set rng = Range("A1:A10")  
For i = 1 To 10  
mark = rng.Cells(i).Value  
Select Case mark  
Case Is < 50  
sumFail = sumFail + mark  
Case Is >= 50  
sumPass = sumPass + mark  
End Select  
Next i  
MsgBox "The sum of Failed marks is" & Str(sumFail) & vbCrLf & "The sum of Passed
```

ANIMATIX COMPUTER EDUCATION

```
marks is" & Str(sumPass)  
End Sub
```



Explanation:

rng is declared as range and we can set it to include certain range of cells, here the range is from A1 to A10.

Then I used the ForNext loop to scan through the selected range

rng.Cells(i).Value read the value in cells(i) and then passed it to the variable mark.

To do selective addition, I used the statement Select Case....End Select

Finally, the results are shown in a message box

Excel VBA Controls' Properties and VBA Code

Creating and Using form

When the message box or the input box are not sufficient anymore to communicate with the user you need to start developing userforms.

The form or userForm is also known as a GUI (Graphical User Interface). The form is used to require values, parameters and information from the user to feed the VBA procedure. Different basic controls can be added to the userform they are called: Label, TextBox, ComboBox, ListBox, CheckBox, OptionButton, Frame, CommandButton, SpinButton and Image . You can find all kinds of other controls on the Internet but the problem is that if you send your workbook to other people and the new control is not installed on their computer it won't work.

Tab Order

Once you have added the controls here is an important feature. The user can use the mouse to move from one control to the other but he should also be able to move from one control to the other by entering a value in one and clicking "Enter" or "Tab" and the focus will be set on the next control where he is supposed to enter a value not on a label. More importantly when the user enters a value in the final control you want the focus to be on the right command button (Submit) and not on another one like " Close Form" .

To make sure that the user moves from one control to the next one in a set order you need to set the tab order. To do so, right click on the form itself and select the "Tab Order" item. Follow the instructions. The first control in the list will be the one that is active (flashing cursor within) when the form is activated. Bring the controls that are not to be used by the user (labels) at the end of the list. For the controls that are not to be used you can also set the "TabStop" property of the individual control to "False" in the properties window.

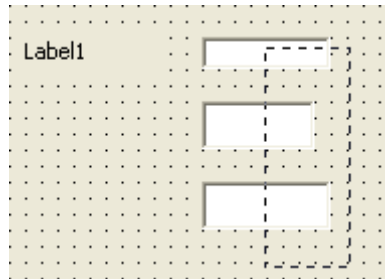
Managing Controls

You can move the controls by clicking on them holding and moving them around. You can resize them by selecting them and using the different handles around them. You can copy or cut them by right clicking on them and choosing the right menu item.

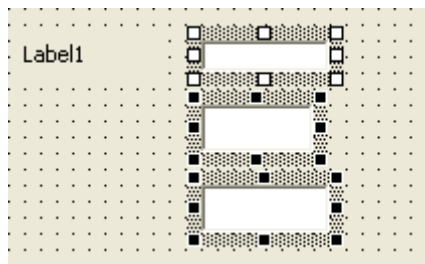
Once you have added your controls you might want to align a few of them or resize a few so that they are all the same size. To do so you first need to select many controls at

ANIMATIX COMPUTER EDUCATION

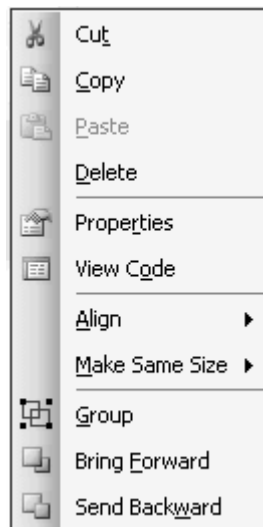
the same time. To do so left click on the form near one of the controls that you want to select. Hold and drag drawing a frame that includes many controls.



When you let go of the button all the controls that are touched by the frame are selected.



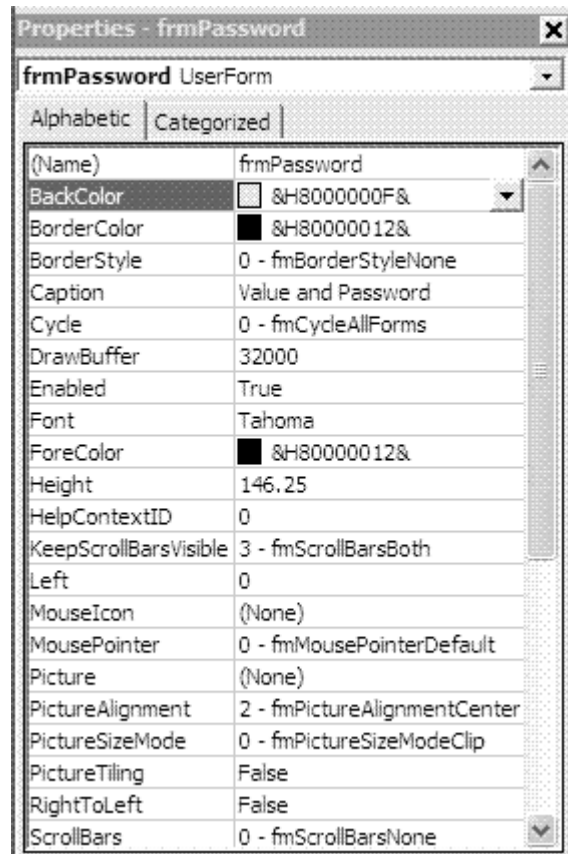
Right click on any of the selected controls and this contextual menu appears:



You can then align the controls (7th menu item) or make them the same size (8th menu item).

Userforms Properties and VBA Code

As you have seen in lesson 1 on the Visual Basic Editor you double click on the userform's name in the Project window and its properties appear in the Properties window:



In the Properties window of the VBE you MUST change the name "(Name)" of the form, its caption (the name in the blue band at the top of the UserForm) and you can also modify the default setting of any of the 32 other properties of the form.

When you name a form always use the prefix "frm" like in "frmDatabase" and be as descriptive as you can be so that your code will be easy to read. Always use one or more upper case letters in the name. When you write "frmdatabase.show" in lower case letters Excel will capitalize some letters "frmDatabase.Show" letting you know that the name is spelled correctly.

The caption is what your users will see at the top of the userform. Be as informative as possible and complete the information with a label if necessary.



Else than the Name and Caption there are just a few properties that you might want to modify. You can select a different color for the background with the property "BackColor".

By default the userform appears in the center of the screen. If you want it to show somewhere else set the "Start" property to "0-Manual" and use the "Top" and "Left" properties to set a new position.

The Code

Opening and Closing the Userform

The first thing to do is to create code to call your userform. Here is a basic line of code doing so:


```
frmCity.Show
```

The line of code to close the userform is:

```
frmCity.Hide or Me.Hide "Me" being the general name of the active form
```

The "Hide" sentence is usually part of the code of a command button on the form. A user clicks on a "GO", "SEND", "SUBMIT" or "CANCEL" button and part of what must happen is that the userform disappears from the screen.

Labels

In the toolbox the label has this icon . The label is a passive control meaning that the user never really acts on it. It is there to inform the user and to label other controls like text boxes, combo boxes or list boxes.

Properties

The other interesting properties of the label are:

- TabStop: To make the control invisible for the "Tab" and "Enter" keys (see Tab Order) set this property to "False".
- WordWrap: If you want to write more than one line of text in a label set this property to "True".

Code


There is not much coding developed for the labels although there are 8 events related to the label. For example there is an event named "MouseMove" . If you develop code within this event it is executed when the mouse moves over the label. If the code is the following:

```
MsgBox "Don't forget to..."
```

a message box will appear when the user moves the mouse over the label.

You can stack many labels one over the other and make their "Visible" property to "False" . You can then make any of the labels visible from an event related to another control. For example if a user chooses a certain value in a combo box a certain label appears.

Text Boxes

In the toolbox the text box has this icon .

The text box is the simplest control to require an entry from the user. The user types something in it and this value can then be used in your VBA procedure. You will usually add a label to accompany the text box.

For most controls including the VBA for Excel text box there are general properties that allow you to set the font, the color of the font, the color of the background, the type of background, the type of border and other design features. Using the 3 windows in the Visual Basic Editor you will see the following properties in the "Property" window when the text box is selected.

Properties

The other interesting properties of the text boxes are:

- WordWrap to be able to write more than one line on a button,
- ControlTipText which generates a small comment box when the user moves the mouse over the control. You can use this property to give explanations and instructions about the command button,
- Enabled and Visible are properties that you can change programmatically to disable or render invisible a command button following a previous selection in another control of the userform,
- TabIndex is a property that you change through the " Tab Order" functionality as shown in the UserForms section.
- MaxLength to limit the number of characters entered by the user,
- Value or Text which is the text shown in the text box when the userform is activated ("Enter your Name" for example)

To ask users to submit a password to run certain macros develop a userform with a text box and a command button. In the text box you can modify the "PasswordChar" property so that when the user enters the password nobody around can read it. Use an asterisk, an ampersand or any other character in it.


ANIMATIX COMPUTER EDUCATION

Code

The most important thing to remember is that a text box is what its name says it carries text. So if you want to send a numerical value from a text box to a cell you must use the "Value" thing:

```
Range(" A1" ).Value=tbxInput.Value
```

Command Buttons

In the toolbox the command button has this icon . The command button is a very active control and there is always VBA code behind it.

The command buttons are usually placed at the bottom of the form and serve to complete the transaction for which the form has been created. The caption of these buttons are usually "Go" , "Run" , "Submit" , "Cancel" , etc.

Properties

The other interesting properties of the command button are:

- WordWrap to be able to write more than one line on a button,
- ControlTipText which generates a small comment box when the user moves the mouse over the control. You can use this property to give explanations and instructions about the command button,

For advanced users there are the:

- Enabled and Visible properties that you can change programmatically to disable or render invisible a command button following a previous selection in another control of the userform.

Code

Name your command button before developing your code. VBA uses the name of the command button when it creates lines of code related to events. So if you don't name your command button VBA will create the private sub::

```
Private Sub CommandButton1_Click()
```

as if you name the command Button " cmbSubmit" for example the private sub will start with:

```
Private Sub cmbSubmit_Click()
```

If you name your command buttons after private subs have been created they won't work anymore.

A very simple VBA procedure for the command button would look like this:

```
Private Sub cmbSubmit_Click()
```

```
    Sheets("Code").Range("F1").Value = cbxInput.Value  
    frmPassword.Hide
```

```
End Sub
```

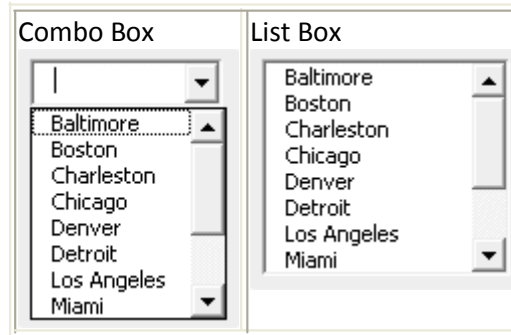
The content of the combo box "cbxInput" is entered in cell "F1" of the sheet "Code" and the form (frmPassport) is closed.


ANIMATIX COMPUTER EDUCATION

List Boxes

Before we begin on the List Box

The difference between a combo box and a list box is that the combo box is a drop-down list and the user can submit a single value from the drop-down list. The list box shows a certain number of values with or without a scroll bar and the user can select one or more values.



In the toolbox the list box has this icon .

No programming is needed to submit the list of values that will be offered to the user within the combo box. Look for the RowSource property.

The RowSource Property:

The values that should appear in the drop-down list of the combo box are submitted in the RowSource property. For example if the value of the RowSource property is Balance!A1:A12 The values residing in cell A1 to A12 of the sheet named Balance will be offered as choices to the user who clicks on the small arrow of the combo box.

The rules to submit the RowSource property is the name of the sheet where the list resides followed by an exclamation point (!), the address of the first cell, a colon and the address of the last cell.

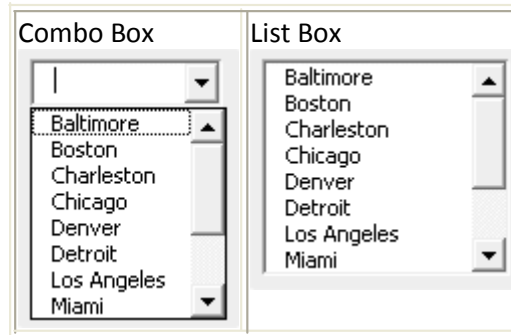
IMPORTANT NOTE: if there is a space or a special character within the name of the sheet where the list resides you must surround it with simple quotes like in 'New Balance'!A1:A12.

Combo Boxes

Before we begin on the Combo Box

The difference between a combo box and a list box is that the combo box is a drop-down list and the user can submit a single value from the drop-down list. The list box shows a certain number of values with or without a scroll bar and the user can select one or more values.

ANIMATIX COMPUTER EDUCATION



If you are looking for a drop-down list (also called pull-down lists) to use on a regular worksheet see the much easier and user friendly Excel drop-down lists in the website on Excel.

When you double click on the combo box in the Visual Basic Editor you will see all its properties in the Properties window .

No programming is needed to submit the list of values that will be offered to the user within the combo box. Look for the RowSource property.




The RowSource Property:

The values that should appear in the drop-down list of the combo box are submitted in the RowSource property. For example if the value of the RowSource property is Balance!A1:A12 The values residing in cell A1 to A12 of the sheet named Balance will be offered as choices to the user who clicks on the small arrow of the combo box.

The rules to submit the RowSource property is the name of the sheet where the list resides followed by an exclamation point (!), the address of the first cell, a colon and the address of the last cell.

IMPORTANT NOTE: if there is a space or a special character within the name of the sheet where the list resides you must surround it with simple quotes like in 'New Balance'!A1:A12.

Option Buttons, Check Boxes and Frames

In the toolbox the option button has this icon , the check box has this one  and, the frame this one .

You don't need to add a label to accompany the check box or the option button because they come with their own.

The check boxes and the option buttons are both used to offer the user a choice. The main difference between check boxes and option buttons is that if you have 5 of each

ANIMATIX COMPUTER EDUCATION

on a form a user can check all 5 check boxes but can only select one of the option buttons.

If you want to create two sets of option buttons read below on frames and option buttons. If you don't want to use frames to create groups of option buttons you will need to use the "GroupName" property of the option buttons. All option buttons with the same GroupName work together.

Properties

- WordWrap to be able to write more than one line in the caption,
- ControlTipText which generates a small comment box when the user moves the mouse over the control. You can use this property to give explanations and instructions about the option button or the check box.
- Enabled and visible are properties that you can change programmatically to disable or render invisible an option button or a check box following a previous selection in another control of the **userform**.


Frames

Frames are also a passive control. Frames are used to improve the layout of the userform. You can use them around a group of controls that have something in common.

Frames become more important to manage option buttons. If you have two sets of option buttons on a userform and you don't place them within a frame they all work together and you can choose only one. If you put each set within a frame you can choose one in each set.

When you move a frame its entire controls move with it.
Advance form controls

Spin Button

In the toolbox the spin button has this icon .

You can ask a user to enter a value directly in a text box but you can make things a little fancier by using a text box and a spin button.

The spin button is not really used by itself. Because the spin button doesn't show its value it is usually used with a text box. The text box shows a number and by clicking on the arrows of the spin button the value in the text box is increased (or decreased) by 1, or 5 or 10...by whatever value that is set within the properties of the spin button.

Properties

The other interesting properties of the spin buttons are:

ANIMATIX COMPUTER EDUCATION

- Min is the minimum value of the spin button. It can be negative
- Max is the maximum value of the spin button. It can be negative
- Small is the value of the change when the user clicks on the arrows
- Large is the value of the change when the user clicks on the scroll bar of the spin button

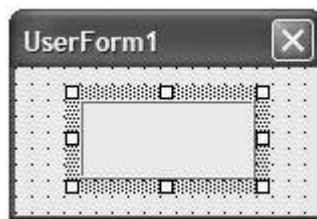
Image Control

There is a control in the toolbox called "Image" . Within this control you can show all kinds of pictures. You set an image control on a userform and you submit a picture in the property "Picture" . The picture becomes part of the control and userform.

Fitting the Picture

The first thing that you want to do is to fit the picture in the image control to make the size of the control adapt to the size of the picture.

When you are in the Visual Basic Editor and you single click on an image control a frame appears around it with 8 stretchers (picture below). If you double click on the middle stretcher (when a two tips arrow shows) of the right side or on the middle one at the bottom or on the bottom right corner stretcher the image control will adapt to the size of the image. Double clicking anywhere else will take you to the VBA code and will not adapt the control size to the picture size.



PictureSizeMode Property

Another interesting property of the image control is the PictureSizeMode.

If the property is set to the default value 0-fmPictureSizeModeClip the control size can be changed without the picture size being modified. So you can see only part of the picture or there can be a background behind it which color you can change at will.

If the property is set to the 1-fmPictureSizeModeStretch the picture is resized as the control is. The image fills the control.

If the property is set to the 3-fmPictureSizeModeZoom the picture is resized as the control is but the picture and background are present.

ANIMATIX COMPUTER EDUCATION

Event handling

When does the VBA procedure (macro) start? When an EVENT happens. The event is what triggers the VBA Excel procedure.

Clicking on a text box on the worksheet


95% of the VBA procedures that you develop are triggered by a click on a button located on a worksheet.

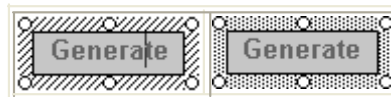
I prefer using text boxes rather than VBA command buttons because they are much easier to maintain and allow much more creativity in the design. You can use the font that you like and the background color that fits your needs. If you are a little creative you can add 3D effects, special borders and the likes.

A few note on Excel **text boxes**:

I always keep the Excel drawing toolbar visible at the bottom of my screen



You create text boxes by a left click on the icon  , let the button go, then go to the worksheet left click, hold and stretch the text box. When the border of the active text box is made of diagonal lines you can work the text inside the text box. If you click again on the border it becomes a set of dots and you then can work the text box itself. Right click on the border in any of the two states and you will see that the menus are different.



First you develop a macro in a module in the VBE. Then you click on the text box and when the border becomes a set of dots right click on it and select "Assign a macro" . Select a macro from the list that is offered to you. A simple macro to call a userform would look like this:

```
Sub proUserFormWeighing()  
    frmWeighing.Show  
End Sub
```

You can assign a VBA macro to a text box and also to a WordArt, a picture or any other shape from the " Drawing" toolbar.

Once a button (image, word art or text box) has been assigned a macro or an hyperlink you need to select it with a right click to modify it.

Download one of these buttons (right click on it in your browser and choose "Save image as"). Save it on your desktop:

ANIMATIX COMPUTER EDUCATION



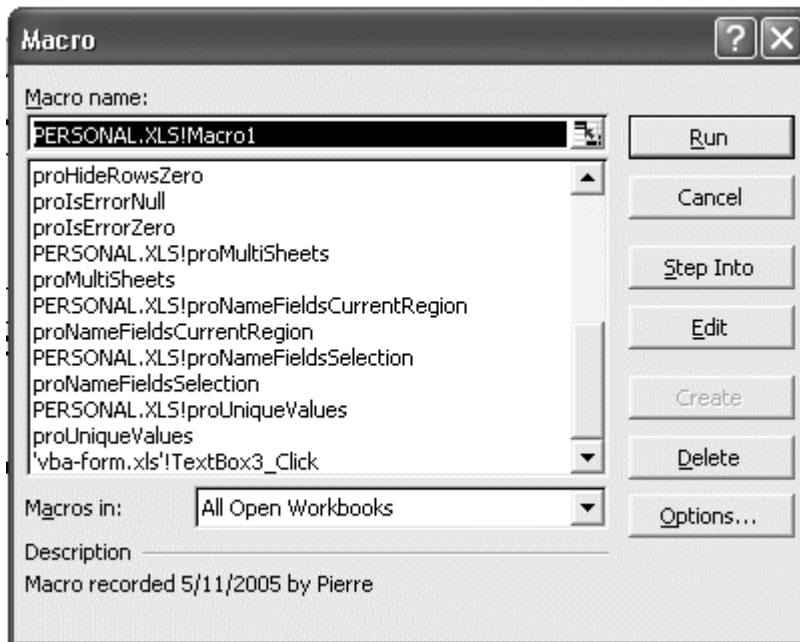
Insert the image that you have imported on the first sheet " Insert/Picture/From File/Desktop/.....gif" . Once the image has been added to the sheet, right click on the image, select "Assign Macro" and select a macro from the list. Click "OK" .

Now click on the image.

You can "borrow" all kinds of buttons from the Internet or create your own from the "Design" toolbar and use them as triggers for your VBA procedures.

From the Excel Menu

In Excel you can run an Excel VBA macro by going to the menu "Tool/Macro/Macros.." then select the macro from the list and click "Run" .



Open many workbooks with macros in them. When you go to the menu "Tool/Macro/Macros.." you will notice that you have access to all the macros from all the open workbooks. This means that you can store ALL your useful Excel macros in a single workbook (call it myMacros.xls) and have access to them while the workbook is opened. Let's say for example that you have designed a macro that multiplies the content of a cell by 2. If "myMacros.xls" is open you can call this Excel macro from any

ANIMATIX COMPUTER EDUCATION

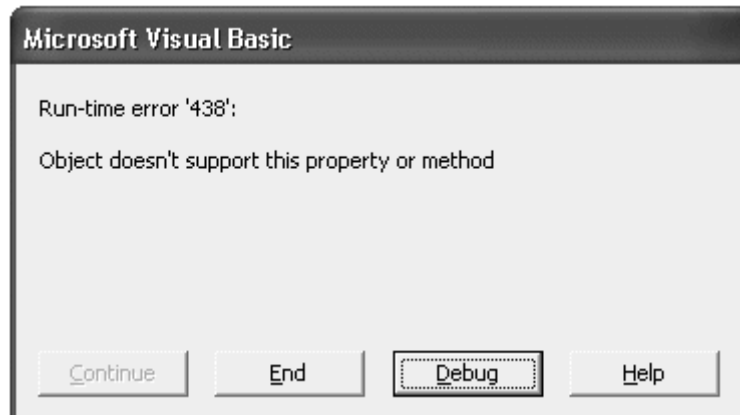
cell in any other workbook that is open. No need to copy your essential macros in all your workbooks just open myMacros.xls and put them to work.

Clicking on a Key of your Keyboard

First you need to program a key. To do so go to "Tool/Macro/Macros.." then select a macro from the list. Click on "Options" and follow the instructions. A suggestion, assign your macros to upper case keys ("Shift/A" instead of "Shift/a" for example) to make sure that you don't use one of the many lower case keys that are already used by Excel. You can also assign a macro to a key when you use the Excel Macro recorder.

Debugging and Dealing with Errors

After you have written a macro and VBE has helped you correct most the errors...after you have tested you macro there might still be error messages generated and you or the user will end up seeing this window:



This window is no problem if you or close colleagues are the only users of the workbook. But in other rare cases where you distribute your workbooks to many users you will probably prefer to have a more useful error message.

Here is the basic way to handle errors.

A VBA procedure usually runs from Sub to End Sub. In a VBA procedure that includes an error handler it will run from Sub to Exit Sub but if there is an error happening during the execution it will jump to a special address where the error handler resides.

So to add error handling capabilities to any of your VBA procedures you follow these steps:

- 1- Add an Exit Sub right before End Sub
- 2- Between these two lines add an address ending with a colon (addJump: for example)
- 3- Make room between the address line and End Sub to enter your code handling the error.

ANIMATIX COMPUTER EDUCATION

4- At the beginning of your code enter On Error GoTo addJump

5- As in the example below the error handler can be a message box giving your coordinates to the user: MsgBox "An error has occurred, call Peter at 1 613-749-4695"

Example 1:

```
Sub proTestErrorHandler()
```

```
    On Error GoTo addJump
```

```
    Workbooks.Open "xxxxxx"
```

```
    Exit Sub
```

```
addJump:
```

```
    MsgBox "An error has occurred, call Peter at 1 613-749-4695"
```

```
End Sub
```

Copy/Paste the examples in a module of your own and run them. As the workbook xxxxxx can't be found you will see a message box saying An error has occurred, call Peter at 1 613-749-4695 .

VBA Macros Security and Protection in Excel

Special note for users of Excel 2010: See how to install the Visual Basic Editor from your Office CD and set the security level of your Excel.

Sometimes you send a workbook with macros to a colleague. If he can't get them to work it is probably because his security setting is at "High" . Just tell him how to change his level by going to the menu bar "Tools/Macros/Security" and follow the instruction. Each time you open a workbook that contains macros the following dialog window will appear.



Adopt the same attitude as you have with documents attached to Emails. If you know the origin of the file you may enable the macros if not click on "Disable Macros" and you are fully protected. You can look at the workbook but the VBA procedures (macros) are not operational. You can go to the Visual Basic Editor to take a look at the macros. If nothing looks suspicious close the workbook and re-open it enabling the macros.

Password Protecting the code

As an Excel-VBA Developer you might want to protect your code so that nobody else may modify it. In the VBE editor go to "Tools/VBAProject Properties/Protection". Check the box and submit a password. Make sure that you save the password somewhere that you will remember because cracking Excel VBA passwords is expensive.

Forms controls and **ActiveX Controls**. The Forms controls aren't very complicated, and they have been part of Excel for longer. Being newer, the ActiveX controls have richer formatting possibilities. Both can link to cells and ranges in the worksheet. Some advanced users have attributed buggy behavior to the ActiveX controls from the Controls Toolbox.

There are outward similarities, mostly in appearance, between Excel's Forms controls and ActiveX controls. The ActiveX controls are actually more similar to the ActiveX controls available within VBA for use on UserForms. In Classic Excel and VBA, Forms and ActiveX controls are available on toolbars, while in Excel 2010, they've been moved to the Developer tab of the Ribbon.

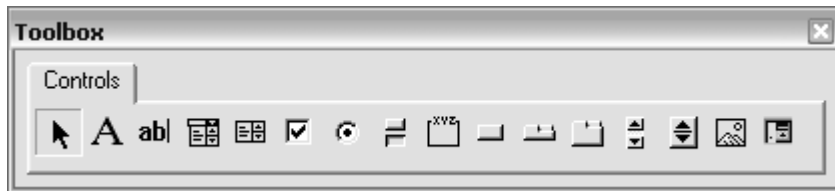
Forms Controls



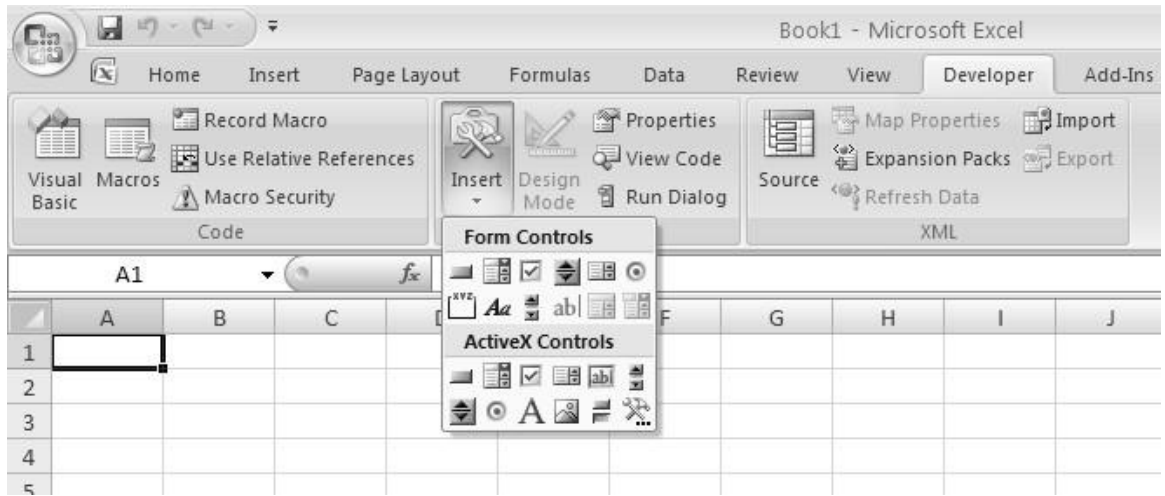
ActiveX Controls



VBA Controls



Forms and ActiveX Controls in Excel 2003



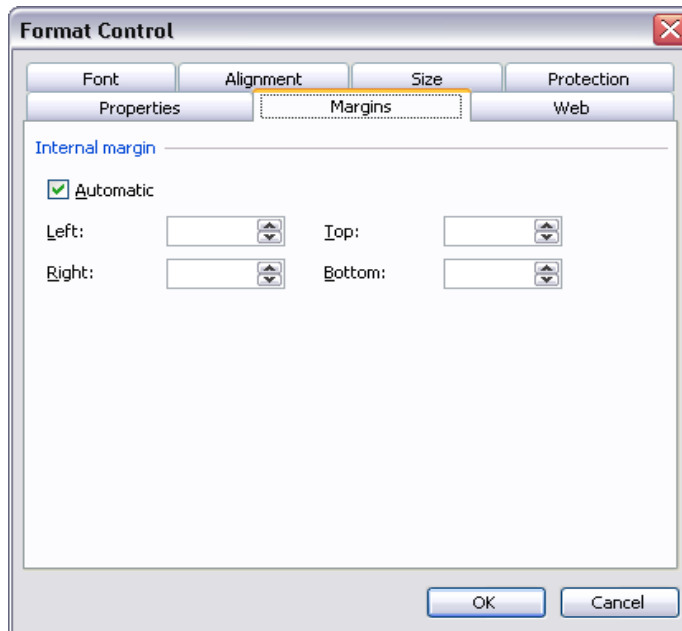
Forms controls can be used on worksheets and chart sheets. Forms controls can also be placed within embedded charts in Classic Excel (though not in Excel 2010). ActiveX controls can only be used on worksheets. Excel responds to a Forms control after the user finishes interacting with it (i.e., unclicks). Excel responds continuously to an ActiveX control. If a Forms slider control is linked to a cell's value, the user has to slide the bar on the control, then release it before the cell updates. If an ActiveX ScrollBar is linked to the cell's value, the cell updates continuously as the user slides the bar along the control.

In general I use the Forms controls, unless I need the continuous response of an ActiveX scrollbar.

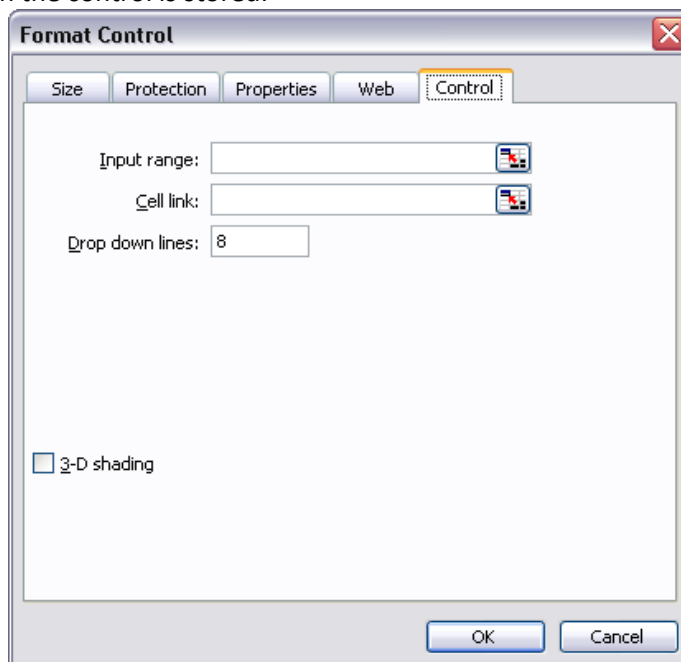
Formatting Controls

Forms controls can be formatted as if they were regular shapes (Excel 95 to Excel 2003 shapes; the new shapes in Excel 2010 are different animals altogether), although the options are not as extensive as for shapes, and vary for different control types. The button dialog (below) is much like that for shapes with various Font, Size, Alignment, and similar tabs, although the button is stuck with its gray color and rectangular shape. A regular shape can be assigned a macro in much the same way, with all the shape's formatting options.

ANIMATIX COMPUTER EDUCATION

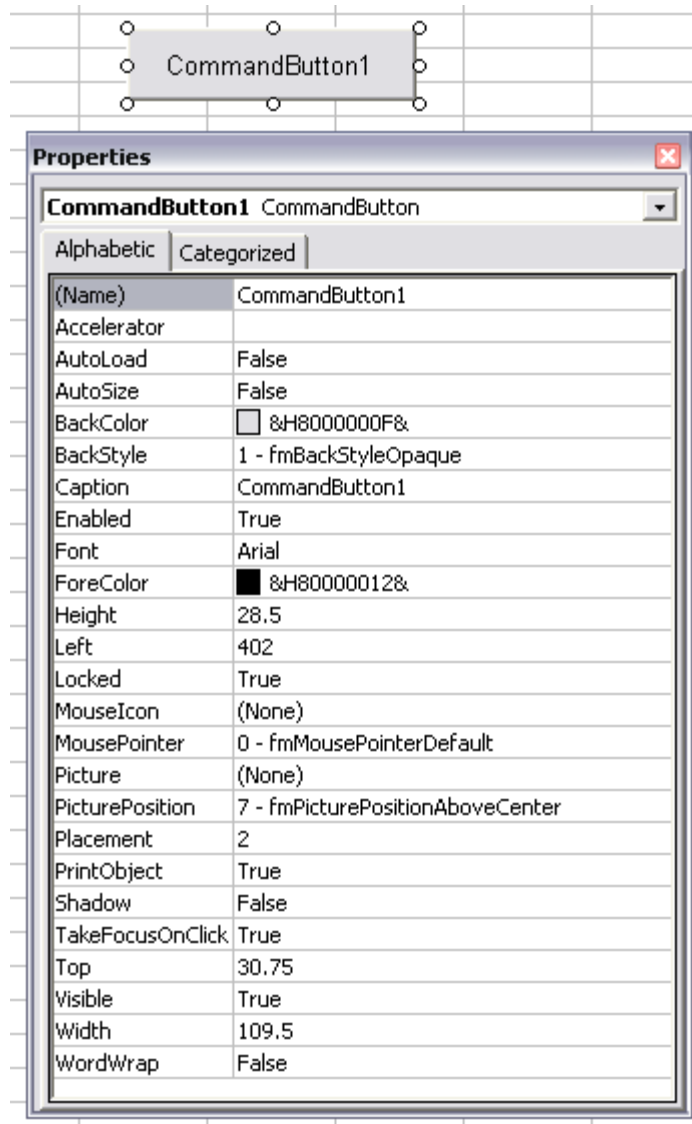


Also depending on the control, there is a tab for properties specific to the control. For a Drop Down or List Box control, there is a means to select an input range, containing the list of items appearing in the control, and another for a cell link, which is where the selected item in the control is stored.



ANIMATIX COMPUTER EDUCATION

ActiveX controls have a scaled back Format Control dialog, but a rich formatting environment offered by the Properties window. In addition, some controls have properties which indicate the addresses for List Fill Range and Linked Cell.



Project Assignment:

Make a project for a Shopping mall as per the following requirements:

Category	Items
Electronic Goods	Television Refrigerator Microwave
Clothing	Sari Shirts Panjabi Suits

UserForm1

WELLSHOPPIE Pvt.

Select Type

Electronic Goods

Clothings

Select Item

Television

Refrigerator

Microwave

Saree

Shirts

Panjabi Suits

Qunatity

Other Items -Fixed Rate Rs.50

☐ Soft Toys

☐ Accessories

☐ Keychains

Discount ?

☐ Yes ☐ No

Amount

Final Amount To be paid

Submit Your Order

Perform the following action:

Above data should be in excel sheet which can be modified.

ANIMATIX COMPUTER EDUCATION

Project form should display the items from the sheet .If user selects Electronics from List Box the next combo box should show the related list of electronics items like television, mixer etc. Same for clothing.

When user enters Quantity e.g. for Television id quantity is 4,Amount text Box should display the amount for 4 televisions. User may select Discount, Other items then final amount will be $\text{Amount} - \text{Discount} + \text{cost of other items}$.This should get displayed in Final amount to be paid.