Contents

[Introduction 3](#_Toc511365710)

[History of Visual Basic 3](#_Toc511365711)

[Power of Visual Basic 6.0 3](#_Toc511365712)

[How to learn VB effectively 3](#_Toc511365713)

[The IDE 4](#_Toc511365714)

[The Integrated Development Environment 4](#_Toc511365715)

[View Code Window 5](#_Toc511365716)

[Save your project 5](#_Toc511365717)

[The Tool Bar 5](#_Toc511365718)

[The Tool Box 5](#_Toc511365719)

[Placing controls on the form 6](#_Toc511365720)

[Writing the Code for a Control 6](#_Toc511365721)

[The Properties Window 6](#_Toc511365722)

[The Project Explorer Window 7](#_Toc511365723)

[The Code Window 7](#_Toc511365724)

[The Form Layout Window 7](#_Toc511365725)

[The Immediate Window 7](#_Toc511365726)

[The Object Browser 7](#_Toc511365727)

[The Locals Window 7](#_Toc511365728)

[The Call Stack Window 7](#_Toc511365729)

[Controls Overview 8](#_Toc511365730)

[Label 8](#_Toc511365731)

[TextBox 8](#_Toc511365732)

[CommandButton 8](#_Toc511365733)

[Option Button 8](#_Toc511365734)

[CheckBox 8](#_Toc511365735)

[VscrollBar & HscrollBar 8](#_Toc511365736)

[Frame 8](#_Toc511365737)

[PictureBox & Image 9](#_Toc511365738)

[ListBox & ComboBox 9](#_Toc511365739)

[DriveListBox, DirListBox & FileListBox 9](#_Toc511365740)

[Timer 9](#_Toc511365741)

[Shape & Line 9](#_Toc511365742)

[The Data Control 9](#_Toc511365743)

[OLE(Object Linking & Embedding) 9](#_Toc511365744)

[Your First Program 10](#_Toc511365745)

[Explanation 10](#_Toc511365746)

[Writing comments 11](#_Toc511365747)

# Introduction

Visual Basic, derived from the Basic language, is an object based and even-driven programming language from Microsoft. This language is relatively easy to learn.

It enables you to create GUI (graphical user interface) applications easily using the Rapid Application Development (RAD) technique.The one most interesting feature of this language is , it comes with an Integrated Development Environment (IDE). The easy-to-use tools of the IDE enable you to create buttons, textbox, and other controls for your applications with minimum efforts.

## History of Visual Basic

We are not exploring the history of VB in depth, but a little knowledge is necessary. Visual Basic 1.0 was released in May 1991 while Visual Basic 6.0 came out in 1998. The following table represents the versions of Visual Basic according to their release year.

|  |  |
| --- | --- |
| Version | Year of Release |
| 1.0 | 1991 |
| 2.0 | 1992 |
| 3.0 | 1993 |
| 4.0 | 1995 |
| 5.0 | 1997 |
| 6.0 | 1998 |

## Power of Visual Basic 6.0

Visual Basic 6.0 is a very powerful programming language. It enables GUI application development, access to databases using Data Access Objects, Remote Data Objects or ActiveX Data Objects, and creation of ActiveX controls. Visual Basic supports API programming that lets you handle the windows.

You can develop simple GUI program, large and complex commercial applications, applications for enterprise solutions and internet based applications.

## How to learn VB effectively

The main purpose of learning VB is obviously to achieve the ability to develop software programs. So you need to learn almost everything of it. Though this is quite difficult to master the language in its entirety as because Visual Basic is too large to complete, you should keep on exploring different features of the language everyday.

One best way to learn VB is to learn with sample programs. Edit the sample programs how you want and see what happens after that, and you have to try to develop unique but small applications on your own after studying each lesson of this tutorial. You must first study the lessons to have a good concept of the topic before working with the sample programs. You will find many downloadable sample programs with source code in this tutorial.

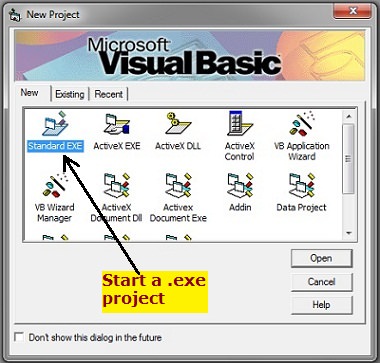
After reading some lessons, you have to revise the previous lessons on a regular basis so that you can easily remember the use of functions, properties, methods and other features.

# The IDE

This lesson will give a quick tour around the Integrated Development Environment, or IDE.

Start a New Project

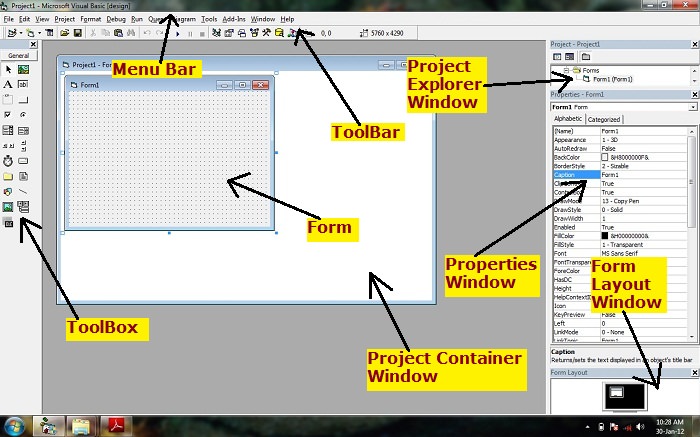
Run the Visual Basic software. A window (see picture below) will appear.



Click "Standard EXE". You can start a standard .exe type of project. In the beginner level, you will only learn about this type of project.

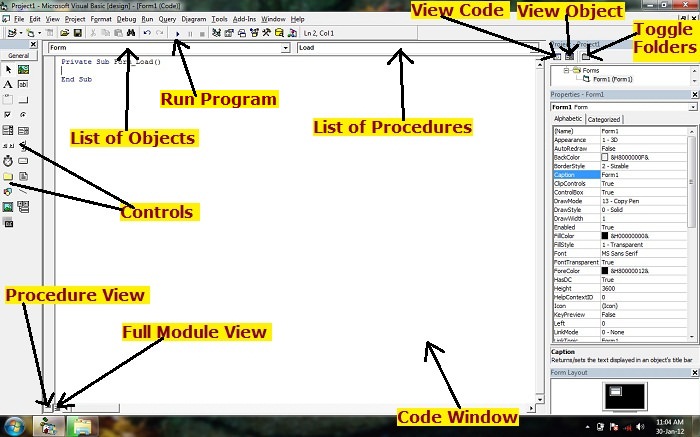
Other project types on this window are for the advanced learners.

## The Integrated Development Environment

After this, the main workspace appears where you will develop your application with the tools in IDE (Integrated Development Environment).

## View Code Window

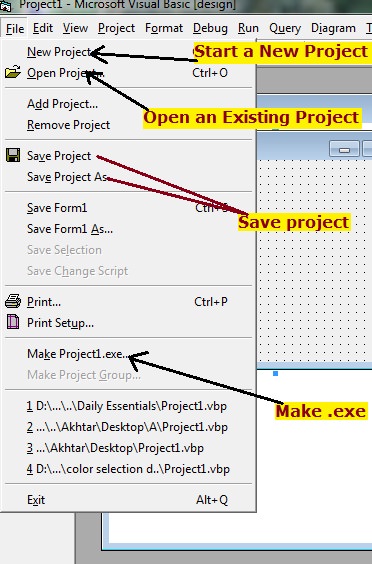
Double-click on form or any control on the form or click "view code" icon in explorer window to view the code window.



## Save your project

After developing your application, save your project in order to modify or improve it later.

The project, form and module are saved in .vbp, .frm and .bas extensions respectively.

If you click on "Make Project1.exe", it will make an executable file that will run like any other software.

## ToolboxThe Tool Bar

The Tool Bar contains all the tools such as Open, Save, Copy, Cut, Start( to run a program) and so on.

## The Tool Box

Tool Box that contains the visual basic controls. You can also add other ActiveX controls to enhance your program.

## Placing controls on the form

Select a control from Toolbox, click on form and drag until you have got the shape of the control you want.

Alternatively, u may double-click any control to add to form.

After adding a control to the form, u need to set its property and then write code for the control to work how you want.

|  |
| --- |
| **QuickHint** |
| There are two ways you can set the value of a property:   * You can set property in Design Time from the Properties Window. * Or, you may wish to set property at run time by writing code. |

## Writing the Code for a Control

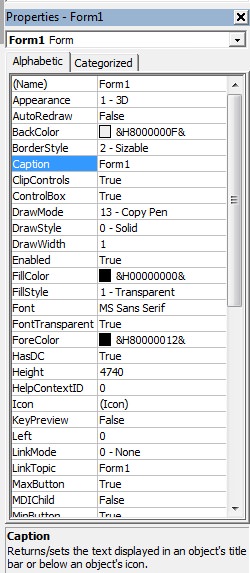
Simply double-click the control (which is on the form) to view the code window and write code to specify how this control will work.

|  |
| --- |
| **QuickHint** |
| * Pointer is not a control. Click this icon to select controls already on the form. * All controls are objects * Form is an object, but it is not a control. |

'Caption' is the property of the Form object. 'Form1' is the value of the property.

In the same way, 'Appearance' is the property. '1-3D' is the value.

## The Properties Window

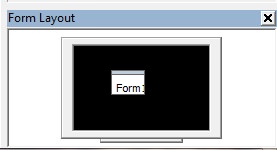
In the Properties Window, notice the help information about the object. This helps in learning new properties. (See Picture) Help information for the 'Caption' property is shown.

## Project Explorer WindowThe Project Explorer Window

Press Ctrl+R if this window is not showing.

The Project Explorer Window gives you a view of the modules or forms which are contained in your VB application. You can switch from one form to another or from one module to another from the Project Explorer Window. You can view the code window of a particular form or module as well.

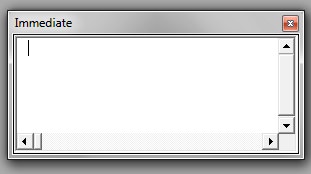
## The Code Window

We need the Code Window to write code that will specify the behavior of the forms and the objects. Remember that the Form is an object.

## The Form Layout Window

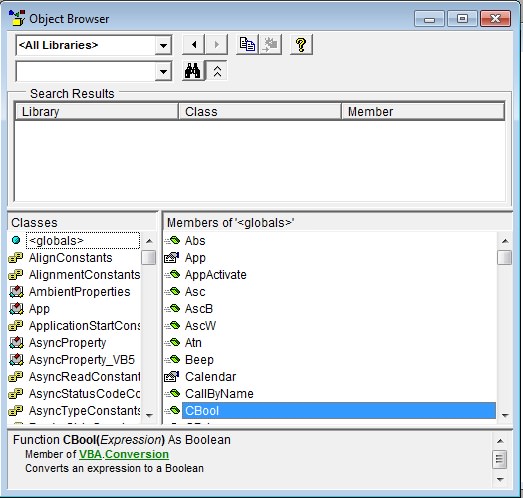
The Form Layout Window shows where on the screen the form will be displayed when the program will be executed. Simply drag on it so that the form appears on the position where you want.

## The Immediate Window

Press Ctrl+G to show the Immediate Window.

The Immediate Window allows to enter a command. When you have stopped a program, you can use these commands to display the current value of variables or expressions.

## The Object Browser

Press F2 to show the Object Browser Window. Object Browser is very useful because you can learn about all the methods, functions, properties and events of the objects. If you find any property, method, event or function unknown, simply search it in the object browser.

## The Locals Window

The Locals Window becomes active when a program is running. It shows the values of variables that are local to a module or procedure.

## The Call Stack Window

It is a useful debugging tool. It comes when you break the program execution and press Ctrl+L.

# controls toolboxControls Overview

Before getting into the actual programming work, you need to have a knowledge of VB controls. In this lesson, a very short description has been given about the controls.

In later chapters, you will learn about the VB controls in some detail.

This is a picture of the VB6 Controls Toolbox. Following that, description of each of the controls is given.

## Label

The label control is used to display text. It is also used to label other controls. The end user cannot edit the label text.

## TextBox

The TextBox control contains characters. End-users can edit the characters contained in the TextBox.

## CommandButton

The CommandButton control is simply a button that we see in our daily-use software. When the end-user clicks the CommandButton, the program behaves according to the code written for the CommonButton.

## Option Button

Option Button is also known as radio button. This control enables the end-user to select one among several options. Only one option button among others in a group can be on at the same time. You can name an option using the Caption property.

## CheckBox

The CheckBox control is used to make a yes/no or true-false selection. You can check more than one CheckBox at the same time that lets you make multiple choices. You can give label to this control using the Caption property.

## VscrollBar & HscrollBar

VscrollBar and HscrollBar controls let you create Vertical scroll bar and Horizontal scroll bar respectively.

## Frame

The Frame control is used as a container of other controls. This is also used to group different controls especially in Option Button controls when you wish to select more than one option. The Caption property associated with it is useful to name the frame.

## PictureBox & Image

These controls are used to display images (e.g company logo). The supported picture formats are BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF (metafile), EMF (enhanced metafile), GIF, and JPEG. But PNG format is not supported.

## ListBox & ComboBox

The ListBox control contains a number of items. The user can select one or more items from the list.

The comboBox control has the feature of ListBox and TextBox. This control does not support multiple selections.

|  |
| --- |
| **QuickHint** |
| ComboBoxes and ListBoxes can have their data loaded from any source, such as a text file, a database, or even on-the-fly. |

## DriveListBox, DirListBox & FileListBox

These controls are often used together to perform file related tasks like opening or selecting files that are stored in the secondary memory.

## Timer

The Timer control is not visible on the form when you run the program. It is used to execute lines of code repeatedly at specific intervals.

## Shape & Line

These controls do not raise events. Shape and Line are used to draw line, rectangle, circle etc on the form.

## The Data Control

The Data control is used for database programming.

## OLE(Object Linking & Embedding)

You can connect other programs to your application provided with a window.

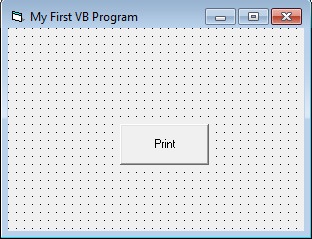
There are also many other controls other than these. The other controls are discussed in the appropriate chapters. You can also add external ActiveX controls that will enhance the interface and functionality of your program.

# Your First Program

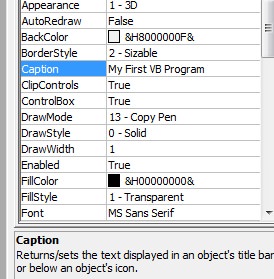
This lesson will walk you through building your first Visual Basic 6 (VB6) program.

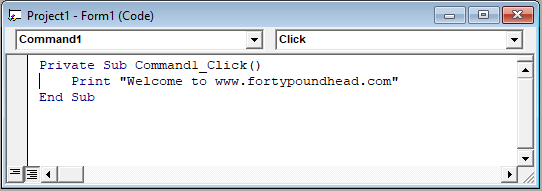
Start a .exe project.

Place a CommandButton on the form from ToolBox.



Click on Form and edit the caption property of the form. Write "My First VB Program".

Edit the caption property of the CommandButton from the Properties Window. Write "Print".

Double-Click the CommandButton to open the Code Window. Write the following code:

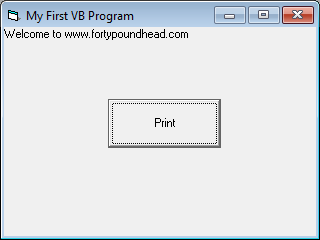
|  |
| --- |
| **QuickHint** |
| Write code between the two lines 'Private Sub Command1\_Click()' and 'End Sub'. These two lines will be created automatically after double-clicking the CommandButton. |

## Explanation

'Sub' means Sub-Routine or Sub-Procedure or Function. 'Command1\_Click()' is the sub-procedure name where 'Command1' is a control and 'Click' is an event. '( )' this sign indicates that 'Command1\_Click()' is a function. 'Private' is the Scope. You will learn about scope in the next chapters. 'Print' is a Method.

You could also write Form1.print, where Form1 is an object and print is its method. 'End Sub' indicates the end of the sub-procedure. You will later learn about sub-procedures.

Now you are done!

Press F5 to run the program or click the start icon button from the toolbar. The code will be executed and the string "Welcome to www.fortypoundhead.com" will be displayed on the form as shown in the shown.

## Writing comments

Comments are the lines of text that are not executed but used for the advantage of the programmers. Comments are written so that the other programmers can easily understand your program and you can better understand the code in case that it becomes complex or large. When you write comments, it becomes easy to maintain the code of your application. Commenting is a part of Documentation and it is a good practice to write comments.

### How to write comments?

Comments are written using the apostrophe ( ' ). That means if you write anything after apostrophe that becomes your comment and text color of the comments becomes green. See the picture below.

