Jessica Davies

# Visual Basic Tutorial

## Hello World Program



The simplest program that beginners make when they first learn a programming language. The words outputted when a beginner learns to code is usually:

**"Hello World"**

## Variables



A variable is usually defined as an "empty box" that stores any type of data, like strings and integers. When creating variables in VB.NET, you always need to underline declare them. To do so, type:

**Dim** [varName] **As** [DataType]

| Data Type (VB.net) | Meaning |
|---|---|
| Boolean | True or False |
| Date | DD/MM/YYYY (e.g. 23/07/2014) |
| Decimal | Non-whole number (e.g. 0.75) |
| Integer | Whole Number (e.g. 7) |
| Char | Charater (e.g. J, &) |
| String | A word (e.g. 'Pizza') |
| Double | Decimals (+ve and -ve) (e.g. -7.5) |

# If/Else Statements

```vbnet
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If TextBox1.Text = "1" Then
            Label1.Text = "Message 1"
        End If
        If TextBox1.Text = "2" Then
            Label1.Text = "Message 2"
        End If
        If TextBox1.Text = "3" Then
            Label1.Text = "Message 3"
        End If
    End Sub
End Class
```

If statements are used EVERYWHERE in coding. They are so important for making applications and games, and are used for running a different piece of code if a user clicks a button, selects an option from a menu or something else.
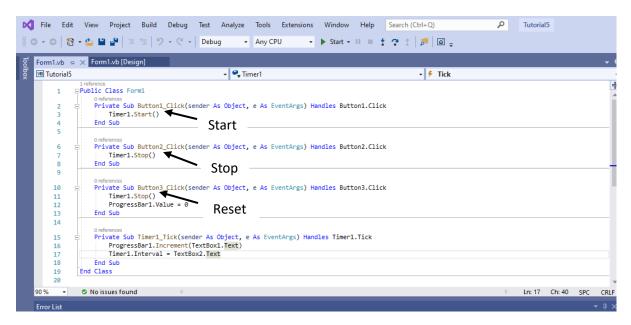
# Adding 2 Numbers

Add Button

```vbnet
Public Class Form1
    Dim num1 As Integer
    Dim num2 As Integer
    Dim result As Integer

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles btnAdd.Click
        num1 = TextBox1.Text
        num2 = TextBox2.Text
        result = num1 + num2
        MsgBox(result)
    End Sub
End Class
```
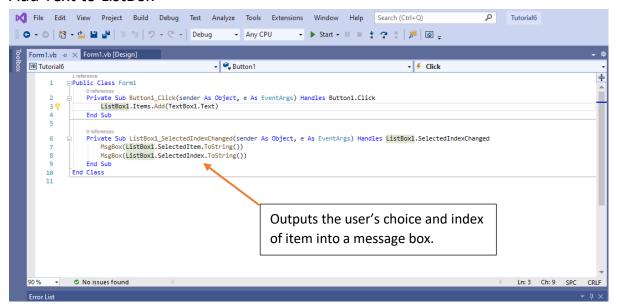
# Timer Control

```vb
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Timer1.Start()          ← Start
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Timer1.Stop()           ← Stop
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Timer1.Stop()
        ProgressBar1.Value = 0   ← Reset
    End Sub

    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        ProgressBar1.Increment(TextBox1.Text)
        Timer1.Interval = TextBox2.Text
    End Sub
End Class
```

# Add Text to ListBox

```vb
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        ListBox1.Items.Add(TextBox1.Text)
    End Sub

    Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ListBox1.SelectedIndexChanged
        MsgBox(ListBox1.SelectedItem.ToString())
        MsgBox(ListBox1.SelectedIndex.ToString())
    End Sub
End Class
```

Outputs the user's choice and index of item into a message box.

# Check Boxes, Radio Buttons & Group Boxes



Show Age in MsgBox

# Text to Speech



Button to speak text in TextBox

# Multiple Forms



Button to switch to second form

# Password Character



You can type a character here to create a password protection system.

To put a picture on a Visual Basic form application, you use a PictureBox (found in your toolbox)

# TabControl (+ PictureBox)
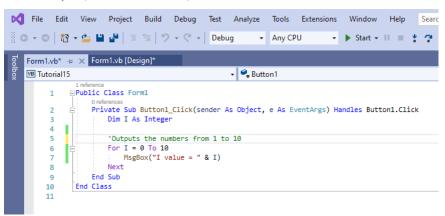
# Adding Nodes to a TreeView



Here's your image list. The image tool can be found in your toolbox. Just double click it to add it to your VB application.

This is how to add images to your image list. When you add a node, your image will appear to the left of it.

# Adding a Splash Screen



To add a splash screen, right click the name of your file (in bold), then click Add, then New Item.
After that, scroll down until you see "Splash Screen". Click it, then press "Add"

# For Loops (Back 2 Basics)



```vb
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim I As Integer

        'Outputs the numbers from 1 to 10
        For I = 0 To 10
            MsgBox("I value = " & I)
        Next
    End Sub
End Class
```

# While Loops, Do While & Do Until

```vb
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim x As Integer = 1

        Do While x < 10
            MsgBox("X value = " & x)
            x += 1
        Loop
    End Sub


    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Dim y As Integer = 1
        Do While Not y = 10
            MsgBox("Y value = " & y)
            y += 1
        Loop
    End Sub


    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Dim z As Integer = 0
        Do Until z = 20
            MsgBox(z)
            z += 1
        Loop
    End Sub
End Class
```

Do the action **while** the condition is true

Do the action while the condition is **not** true

Do the action **until** the condition is true

# Subroutines & Functions

```vb
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        add(TextBox1.Text, TextBox2.Text)
    End Sub

    Private Function addFunc(ByVal a As Integer, ByVal b As Integer) As Integer 'we are expecting this function to return an integer, hence why we wrote 'as integer' at the end
        Dim sum As Integer
        sum = a + b
        Return sum
    End Function

    Private Sub add(ByVal a As Integer, ByVal b As Integer)
        Dim result As Integer
        result = a + b
        MsgBox("The sum of the two numbers is " & result)
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Dim ans As Integer
        ans = addFunc(TextBox1.Text, TextBox2.Text) 'to call a function, you need to create a variable as the data type of the value we want to return (integer), then set the...
        '...variable equal to the function
        MsgBox(ans)
    End Sub
    'if the sub is private, you cannot use it in other classes
End Class
```

This is a SUBROUTINE

This is a FUNCTION

If the keyword "return" is in the code, then it's a function. If not, it's a subroutine.

# Outputting the Date & Time (On a label)



```
1 reference
1  ☐Public Class Form1
       0 references
2  ☐    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3          Timer1.Enabled = True
4
5          End Sub
6
       0 references
7  ☐    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
8          Label1.Text = Date.Now.ToString("dd-MM-yyyy  hh:mm:ss")
9
10         End Sub
11    End Class
12
```

mm = Minutes
hh = Hours
ss = Seconds
dd = Day
MM = Month
yyyy = Year

# Open File Dialog Control



Button to open any EXE file

```
1 reference
1  ☐Public Class Form1
       0 references
2  ☐    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3          'this will open any exe file
4          OpenFileDialog1.Filter = "ALL Files |*.*"
5  ☐        If OpenFileDialog1.ShowDialog = DialogResult.OK Then
6              System.Diagnostics.Process.Start(OpenFileDialog1.FileName)
7          End If
8          End Sub
9
       0 references
10 ☐    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
11         OpenFileDialog1.Filter = "TEXT FILE | *.txt |PDF FILE |.pdf |ALL FILES |*.*"
12 ☐        If OpenFileDialog1.ShowDialog = DialogResult.OK Then
13 ☐            For x = 0 To OpenFileDialog1.FileNames.Count - 1
14                 MsgBox(OpenFileDialog1.FileNames(x))
15             Next
16         End If
17         End Sub
18    End Class
19
```

Button to open the file path

After choosing a file, the file's path is outputted onto a Message Box



When you run the application, you'll first see the form at the top right corner of this picture. When you click Open File Path, your files are shown. Choose your file, then the file path is outputted in a Message Box (code in picture above)
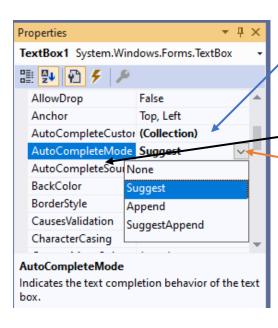
# Opening PDF Files in a VB Form Application



```
1    Public Class Form1
2        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3            OpenFileDialog1.Filter = "PDF Files |*.pdf"
4            If OpenFileDialog1.ShowDialog = DialogResult.OK Then
5                AxAcroPDF1.src = OpenFileDialog1.FileName
6            End If
7        End Sub
8    End Class
9
```

"AxAcroPDF1" is the name of the PDF reader added to the form to the right. Check out next picture to find out how to add it to your toolbox.



To add the PDF reader to your toolbox, open your toolbox. Then, right-click anywhere in your toolbox and click "Choose Items". You will see the interface on the right of *this* picture. Click 'COM Components', tick the box next to the PDF reader (highlighted), then press Ok.

Jessica Davies

# AutoComplete ComboBox & TextBox



This is your collection of strings which are either suggested and/or appended to text in a text box. This can be changed to either search history, websites, file system and many other sources by looking here (auto complete source)

When you click here, it will open a drop-down menu with 3 modes: Suggest, Append or both.
Suggest is basically showing you a suggestion for what you are typing into the text box, whereas Append is when the suggestion gets added to the rest of your text.

Here's your choice of items. If you want to search/select from your own list of strings, select 'Custom Source' from the drop-down menu.
You can also select none if you're just using a textbox as input/output.
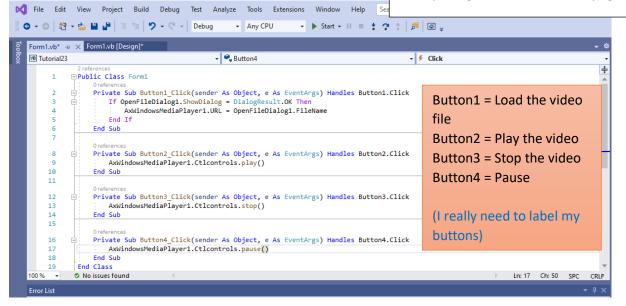
Important Info:

This searching system with a textbox has the same properties as a combo box, so if you click the combo box while you're editing your form, it will also show the auto complete options seen on the picture on the left.

## Playing an MP4 file in your Form Application
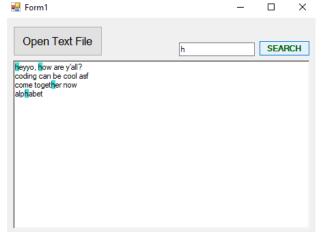
To see how to add Windows Media Player to your form application, follow the steps on the Opening a PDF File section. (page 10)



Button1 = Load the video file
Button2 = Play the video
Button3 = Stop the video
Button4 = Pause

(I really need to label my buttons)

## Open a text file and search for text in Visual Basic
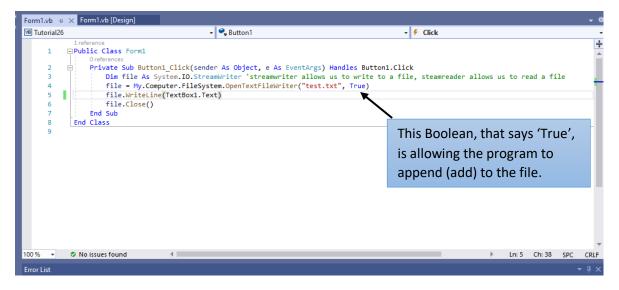
Button to open the text file



Button2 = Search for text
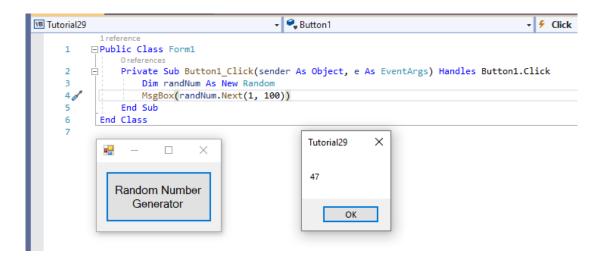


When you click the Open Text File button, it will run the open file dialog tool and open your file path. You proceed to choose the text file, and then it appears in a rich text box. Type some text in the search bar then hit search. All results will be highlighted in a specific colour.

# Write to a Text File

```vb
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim file As System.IO.StreamWriter 'streamwriter allows us to write to a file, steamreader allows us to read a file
        file = My.Computer.FileSystem.OpenTextFileWriter("test.txt", True)
        file.WriteLine(TextBox1.Text)
        file.Close()
    End Sub
End Class
```

This Boolean, that says 'True', is allowing the program to append (add) to the file.

# Random Number Generator

```vb
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim randNum As New Random
        MsgBox(randNum.Next(1, 100))
    End Sub
End Class
```

Random Number Generator

Tutorial29

47

OK

# Opening a Webpage using the Web Browser tool

```vb
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        WebBrowser1.Navigate(TextBox1.Text)
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        WebBrowser1.GoBack()
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        WebBrowser1.GoForward()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        WebBrowser1.Refresh()
    End Sub
End Class
```

Jessica Davies

**Toolbox**

Search Toolbox

SplitContainer
Splitter
StatusStrip
TabControl
TableLayoutPanel
TextBox
ToolStrip
ToolStripContainer
ToolTip
Timer
TrackBar
TreeView
VScrollBar
WebBrowser
Adobe PDF Reader
Windows Media Player
▷ Common Controls
▷ Containers
▷ Menus & Toolbars
▷ Data
▷ Components
▷ Printing
▷ Dialogs
▷ WPF Interoperabi
◢ General

resh    Forwards    Back

This is the web browser tool. It's mainly used for opening webpages, and usually appears in your toolbox when you open Visual Studio. If it's not there, follow the steps on page 10.

rnings    ⓘ 0 Messages    Build + IntelliSense
Project ▲

When running this HTML editor, the user can enter HTML into the rich text box (black arrow), and when they press Run Code, it will output the website on the web browser (orange arrow).

Form1.vb*    For

**Form1**

**Run Code**

```vb
Public Class Form1
    0 references
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        WebBrowser1.DocumentText = RichTextBox1.Text
    End Sub
End Class
```

# Arrays (Back 2 Basics)

```vb
1 reference
Public Class Form1
    0 references
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim names(3) As String

        names(0) = "Jessica"
        names(1) = "Evan"
        names(2) = "Leo"
        names(3) = "Lewis"

        'this will show all names in the array one by one
        For i = 0 To names.Length - 1
            MsgBox(names(i))
        Next
    End Sub
End Class
```
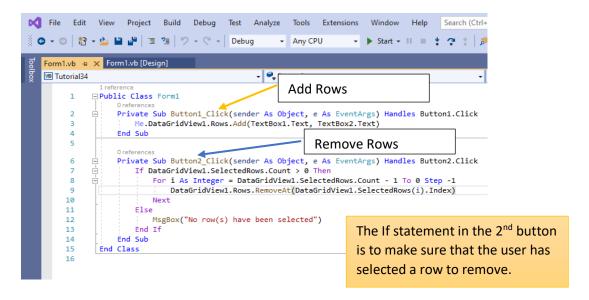
# Adding Items to a Combo Box

```vb
Form1.vb -⊐ ×  Form1.vb [Design]
VB Tutorial33                              ⌄ ⌀ ComboBox1                                dexChanged
                1 reference
1           Public Class Form1
                    0 references
2           ⊟        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
3                        ComboBox1.Items.Add(TextBox1.Text)
4                    End Sub
5
                    0 references
6           ⊟        Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ComboBox1.SelectedIndexChanged
7 💡              MsgBox(ComboBox1.Text)
8                    End Sub
9           End Class
10
```

Button to add items to Combo Box

The item selected then appears in a message box

# Data Grid View Control
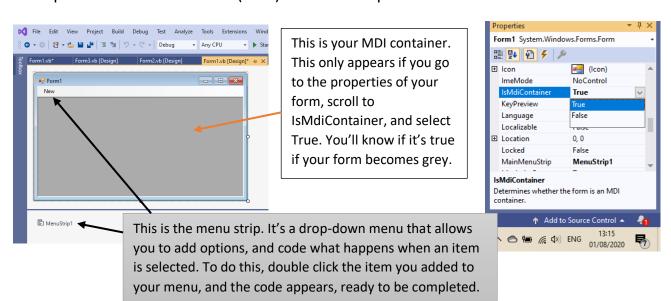
This is the data grid view tool. It's similar to Excel, but obviously you need to add code to it. To add a row, you need to fill both text boxes with strings, then click Add Rows. To remove a row, there should be a little arrow to the left of the row that you've added. Click the arrow, then press Remove Rows.

```
1 reference
Public Class Form1
    0 references
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.DataGridView1.Rows.Add(TextBox1.Text, TextBox2.Text)
    End Sub

    0 references
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        If DataGridView1.SelectedRows.Count > 0 Then
            For i As Integer = DataGridView1.SelectedRows.Count - 1 To 0 Step -1
                DataGridView1.Rows.RemoveAt(DataGridView1.SelectedRows(i).Index)
            Next
        Else
            MsgBox("No row(s) have been selected")
        End If
    End Sub
End Class
```

**Add Rows**

**Remove Rows**

The If statement in the 2nd button is to make sure that the user has selected a row to remove.

## Opening a Website



```
1 reference
Public Class Form1
    0 references
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Process.Start(TextBox1.Text)
    End Sub
End Class
```

## Multiple Document Interface (MDI) & Menu Strip Control



This is your MDI container. This only appears if you go to the properties of your form, scroll to IsMdiContainer, and select True. You'll know if it's true if your form becomes grey.

This is the menu strip. It's a drop-down menu that allows you to add options, and code what happens when an item is selected. To do this, double click the item you added to your menu, and the code appears, ready to be completed.

```
Form1.vb*  ⊡ ×   Form3.vb [Design]      Form2.vb [Design]      Form1.vb [Design]*
VB Tutorial37                                             ▾  ⚙ Form3ToolStripMenuItem              ▾  ⚡ Click
                    1 reference
       1     ⊟Public Class Form1
                    0 references
       2     ⊟    Private Sub Form2ToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles Form2ToolStripMenuItem.Click
       3              Dim f2 As New Form2
       4              f2.MdiParent = Me
       5              f2.Show()
       6          End Sub
       7
                    0 references
       8     ⊟    Private Sub Form3ToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles Form3ToolStripMenuItem.Click
       9              Dim f3 As New Form3
      10              f3.MdiParent = Me
      11              f3.Show()
      12          End Sub
      13     End Class
      14
```

**Important Info:**

"Me" is the keyword given to the current form you are working on.

## Get Source Code from a Website

Button to get source code

**Important Info:**

The "Try…Catch" Statement is used to provide a way to handle some or all possible errors that may occur in a block of code, while still running the code.

```
Form1.vb  ⊡ ×   Form1.vb [Design]
VB Tutorial38                                             ▾  ⚙ Button1                          ▾
                    1 reference
       1     ⊟Public Class Form1
                    0 references
       2     ⊟    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
       3     ⊟        Try
       4                  Dim source As String
       5                  source = New System.Net.WebClient().DownloadString(TextBox1.Text)
       6                  RichTextBox1.Text = source
       7              Catch ex As Exception
       8                  MsgBox("ERROR: Enter Valid URL")
       9              End Try
      10          End Sub
      11     End Class
      12
```
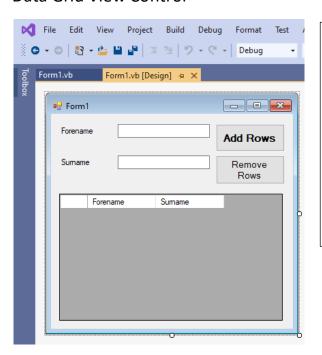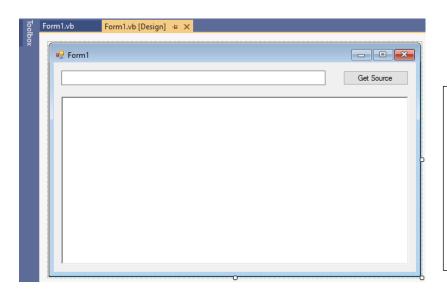
```
Form1.vb      Form1.vb [Design]  ⊡ ×
┌─────────────────────────────────────────────────────┐
│ 🖳 Form1                              [ − ] [ ▢ ] [ ✕ ]│
├─────────────────────────────────────────────────────┤
│  ┌──────────────────────────────────┐   ┌─────────┐  │
│  │                                  │   │Get Source│ │
│  └──────────────────────────────────┘   └─────────┘  │
│  ┌────────────────────────────────────────────────┐  │
│  │                                                │  │
│  │                                                │  │
│  │                                                │  │
│  │                                                │  │
│  │                                                │  │
│  │                                                │  │
│  │                                                │  │
│  │                                                │  │
│  │                                                │  │
│  └────────────────────────────────────────────────┘  │
└─────────────────────────────────────────────────────┘
```

The website is first entered into the text box, then the user presses the Get Source button, and the source code of the provided website is shown in the rich text box.