**Variables**

* Syntax
Dim *varName* As *DataType*
	+ *varName* – You should give your variable a name that represents the value that it holds
	+ *DataType* – Your data type should represent the value
		- Integer (int) whole numbers
		- String (str) names/anything with 1+ letters
		- Float (flt) small number with decimals
		- Double (dbl) larger number with decimals
		- Decimal (dec) larger number with decimals - currency
		- Char (chr) single letter, number or symbol
		- Boolean (bln) true or false
	+ Examples
	 Dim intYear As Integer
	 Dim strLastName As String
	 Dim dblGpa as Double
	 Dim chrInitial As Character
* Assignment
	+ Assigning a value to a variable
	+ Uses =
	+ Value always moves left to right

	 strLastName = “Smith”
	+ Always put “” around strings and characters only
	+ You should assign a starting value when you declare your variables. You can two lines or one line.

	 Dim intYear As Integer
	 intYear = 2011
	 Dim intYear As Integer = 2011
* Concatenation
	+ Means merging
	+ Use &
	+ Example
	 strName = strFName & “ “ & strLName
	 lblInfo.Text = “My name is “ & strName
* Variable Scope
	+ Scope is where the variable is available to use
	+ Lifetime is the time the variable is available to use.
	+ Types
		- Global
			* Declaring a variable just below the public class Form1 statement at the top of your code – not in an event, like a button click
			* The variable can be used anywhere
		- Local
			* Declaring a variable in an event like a button click
			* The variable can only be used where it was declared
		- Procedural
			* Declaring a variable in a specific block of code where it will be available.

**Numeric, Integers**

The following types will hold a numeric value between two given numbers. They CANNOT hold decimal values (I.E. 3.14).

* Byte - Holds a number from 0 to 255
* Integer - Holds a number from -32768 to 32767
* Long - Holds a number from -2,147,483,648 to 2,147,483,647

**Numeric, Decimal**

The following types will hold a numeric value between two given numbers. They CAN hold decimal values, but only with a certain amount of precision.

* Single - Up to 7 digits, can hold exponents up to +-38
* Double - Up to 14 digits, can hold exponents up to +-300

**Other**

* Boolean - Is either True or False
* Currency - Holds a number between +-922,337,203,685,477.5807
* String - Holds text, such as a word or sentence
* Date - Holds a date from Jan 1, 100 to Dec 31, 9999

The following table shows the Visual Basic data types, their supporting common language runtime types, their nominal storage allocation, and their value ranges.

|  |  |  |  |
| --- | --- | --- | --- |
| **Visual Basic type** | **Common language runtime type structure** | **Nominal storage allocation** | **Value range** |
| [Boolean](http://msdn.microsoft.com/en-us/library/wts33hb3.aspx)  | [Boolean](http://msdn.microsoft.com/en-us/library/system.boolean.aspx)  | Depends on implementing platform | True or False |
| [Byte](http://msdn.microsoft.com/en-us/library/e2ayt412.aspx)  | [Byte](http://msdn.microsoft.com/en-us/library/system.byte.aspx)  | 1 byte | 0 through 255 (unsigned) |
| [Char](http://msdn.microsoft.com/en-us/library/7sx7t66b.aspx) (single character) | [Char](http://msdn.microsoft.com/en-us/library/system.char.aspx)  | 2 bytes | 0 through 65535 (unsigned) |
| [Date](http://msdn.microsoft.com/en-us/library/3eaydw6e.aspx)  | [DateTime](http://msdn.microsoft.com/en-us/library/system.datetime.aspx)  | 8 bytes | 0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999 |
| [Decimal](http://msdn.microsoft.com/en-us/library/xtba3z33.aspx)  | [Decimal](http://msdn.microsoft.com/en-us/library/system.decimal.aspx)  | 16 bytes | 0 through +/-79,228,162,514,264,337,593,543,950,335 (+/-7.9...E+28) † with no decimal point; 0 through +/-7.9228162514264337593543950335 with 28 places to the right of the decimal; smallest nonzero number is +/-0.0000000000000000000000000001 (+/-1E-28) † |
| [Double](http://msdn.microsoft.com/en-us/library/x99xtshc.aspx) (double-precision floating-point) | [Double](http://msdn.microsoft.com/en-us/library/system.double.aspx)  | 8 bytes | -1.79769313486231570E+308 through -4.94065645841246544E-324 † for negative values; 4.94065645841246544E-324 through 1.79769313486231570E+308 † for positive values |
| [Integer](http://msdn.microsoft.com/en-us/library/06bkb8w2.aspx)  | [Int32](http://msdn.microsoft.com/en-us/library/system.int32.aspx)  | 4 bytes | -2,147,483,648 through 2,147,483,647 (signed) |
| [Long](http://msdn.microsoft.com/en-us/library/y595sc15.aspx) (long integer) | [Int64](http://msdn.microsoft.com/en-us/library/system.int64.aspx)  | 8 bytes | -9,223,372,036,854,775,808 through 9,223,372,036,854,775,807 (9.2...E+18 †) (signed) |
| [Object](http://msdn.microsoft.com/en-us/library/twcxd6b8.aspx)  | [Object](http://msdn.microsoft.com/en-us/library/system.object.aspx) (class) | 4 bytes on 32-bit platform 8 bytes on 64-bit platform | Any type can be stored in a variable of type Object |
| [Short](http://msdn.microsoft.com/en-us/library/7tb7bdw6.aspx) (short integer) | [Int16](http://msdn.microsoft.com/en-us/library/system.int16.aspx)  | 2 bytes | -32,768 through 32,767 (signed) |
| [Single](http://msdn.microsoft.com/en-us/library/xay7978z.aspx) (single-precision floating-point) | [Single](http://msdn.microsoft.com/en-us/library/system.single.aspx)  | 4 bytes | -3.4028235E+38 through -1.401298E-45 † for negative values; 1.401298E-45 through 3.4028235E+38 † for positive values |
| [String](http://msdn.microsoft.com/en-us/library/thwcx436.aspx) (variable-length) | [String](http://msdn.microsoft.com/en-us/library/system.string.aspx) (class) | Depends on implementing platform | 0 to approximately 2 billion Unicode characters |

**Formatting Output**

* ToString Method
	+ Allows you to format a numeric variable for display
	+ Does not actually change the variable’s value
	+ Use when displaying the value in a label.
	+ Syntax
	 lblYourLabel.Text = varNum.ToString(**“**format**”**)
	+ Example
	 lblTotal.Text = decTotal.ToString("$###.##")
* Format Function
	+ Allows you to format a numeric variable for display
	+ Does not actually change the variable’s value
	+ Use when displaying the value in a label.
	+ Syntax
	 label.Text = Format(number, “format type”)
	+ Examples
	 Me.lblAnswer.Text = Format(numVar, "General Number") '8789
	 Me.lblAnswer.Text = Format(8789, "Currency") '$8,789.00
	 Me.lblAnswer.Text = Format(8789, "Fixed") '8789.00
	 Me.lblAnswer.Text = Format(8789, "Standard") '8,789.00
	 Me.lblAnswer.Text = Format(89, "Percent") '8900.00%
	 Me.lblAnswer.Text = Format(8789, "Scientific") '8.79E+3
	 Me.lblAnswer.Text = Format(8, "Yes/No") 'Yes
	 Me.lblAnswer.Text = Format(0, "True/False") 'False
	 Me.lblAnswer.Text = Format(1, "On/Off") 'On

**Textboxes**

* Allow you to get input from the user when your programming in running (runtime)
* Properties
 (Name) – start with txt
 Text – what is displayed inside the text box
 Alignment – aligns the text relative to the text box.
 PasswordChar – Sets a character to be displayed in the textbox as the user types.
* Prompt
	+ The label next to the textbox to tell the user what to enter.
* Syntax
 varName = txtTextBox.Text
* TextChanged Event
	+ Executes when the user types in the textbox
	+ Use to clear answers or messages from the label
	+ Create the TextChanged event
		- Design Window
			* Double click the textbox
		- Code Window
			* Select the TextBox from the Class List
			* Select TextChanged from the Method List

**MessageBox**

* Pre-set form with an OK button
* Syntax
 MessageBox.Show (“string here”)

**Advanced Variables**

* Static Variables
	+ Hold its value between runs of the program.
	+ Use Static instead of Dim
	+ Static variables cannot be defined globally.
	+ Example
	 Static intNumber As Integer
* Constant Variables
	+ A variable that once declared cannot be changed by the program.
	+ Constant variables should be declared using all uppercase letters.
	+ Use Const instead of Dim
	+ Use all capitals for the variable name (except for the prefix)
	+ Example
	 Const decTAXRATE As Decimal = 0.775
* Counter
	+ A variable that is used to determine how many times something happens
	+ Declare as Static
	+ Example – Can set up either way
	 Counter = Counter + 1
	 Counter +=1

**Name Objects**

* Use Hungarian Notation prefixes when naming control objects
	+ Form- frm (frmMain)
	+ Button- btn (btnSubmit)
	+ Label- lbl (lblTotal)
	+ Text Box- txt (txtAge)
	+ Radio Button- rdo (rdoAdd)
	+ Check box- chk (chkDivide)
	+ Image- img (imgMegaMan)
	+ Combo Box- cbo (cboState)
	+ Picture Box- pic (picFlower)
	+ List box- lst (lstState)
	+ Menu- mnu (mnuFile)