

## **What is a Variable?**

DocumentID : What\_is\_a\_Variable.PDF  
Author : Michele Harris  
Version : 1.0  
Date : 07-03-2009

## Contents

So What is a Variable? .....	2
The Syntax of a Variable .....	2
Fundamental Data Types in Visual Basic 2008.....	3
Smart Practices.....	5
Declaring and Using Variables Yourself .....	5
For More Practice .....	8

## So What is a Variable?

In Visual Studio, variables are used to temporarily store information.

Whether just allowing data to be stored in the computer's memory, or serving as a component of a function, variables let us keep track of our program's information at run time.

## The Syntax of a Variable

To use a variable, you must first declare it, which tells Visual Studio what sort of variable it will be and how/when you want to use it.

The basic syntax for a variable is as follows:

*Dim VariableName as Type*

The "Dim" statement, short for "dimension," informs Visual Basic that you're declaring a new variable. *VariableName* is exactly that—the name of your variable. You can name your variables whatever you'd like to. But keep in mind, however, that your variable names should be relevant to their operations. You don't want to create a program with a number of vague variables floating around and risk forgetting just which variable does what.

The "Type" designation lets Visual Basic know what sort of data to expect. It also informs Visual Studio how much memory will be allocated for this

---

particular variable. For instance, if we want our variable to handle text, we'd declare the type as a string. If we wanted our variable to handle reasonably small whole numbers, we might declare our variable type as an integer.

Below are many of the variable types in Visual Basic.

## Fundamental Data Types in Visual Basic 2008

Below are the basic types you'll use to declare your variables. Don't be alarmed by the size of some of the numbers—some variables types are just meant to handle very large numbers (like Long), and some, like Byte, are meant to handle fairly small numbers. Some are optimized for numbers with many decimals (such as Decimal), and some are meant to store something as straight forward as a date (such as Date.)

Data Type	Size	Range	Sample Usage
Short	16-bit	-32,768 through 32,767	Dim NumofSenators As Short NumofSenators= 100
UShort	16-bit	0 through 65, 535	Dim Salary As UShort Salary = 50,000
Integer	32-bit	-2,147,483,648 through 2,147,483,647	Dim Age as Integer Age = 22
UInteger	32-bit	0 through 4,294,967,295	Dim Lottery As UInteger Lottery = 400,000,000

Long	64-bit	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807	Dim NationalDebt As Long NationalDebt = -11,505,401,038,909
ULong	64-bit	0 through 18,446,744,073,709,551,615	Dim Stars As ULong
Single	32-bit floating point	-3.4028235E38 through 3.4028235E38	Dim CompPrice As Single CompPrice = 699.99
Double	64-bit floating point	-1.79769313486231E308 through 1.79769313486231E308	Dim Pi As Double Pi = 3.1415926535
Decimal	128-bit	0 through +/- 79,228,162,514,264,337,593,543,950,335 (+/-7.9...E+28) with no decimal point. 0 through +/- 7.922816251426433759354395	Dim HappyMeal as Decimal HappyMeal = 1.99
Byte	8-bit	0 through 255	Dim KiloRun As Byte KiloRun = 11
SByte	8-bit	-128 through 127	Dim ApplesOwed as SByte ApplesOwed = -5
Char	16-bit	Any Unicode symbol in the range of 0-65,535. Append "c" when initializing a Char.	Dim UnicodeChar As Char UnicodeChar = "c"
String	Usually 16-bit	0 to up to 2 billion 16-bit Unicode characters	Dim Greeting as String Greeting = "Hello!"
Boolean	16-bit	True or False	Dim ColorBlue as Boolean ColorBlue = True
Date	64-bit	January 1, 0001, through December 31, 9999	Dim IndependenceDay as Date IndependenceDay = #3/1/1963#
Object	32-bit	Any type can be stored as an Object	Dim MinutesPerDay as Object MinutesPerDay = 1440

As you can see, there exist several different variable types playing different roles—though it's important to note that the variable types *aren't* mutually exclusive. For instance, a variable with the value of 10 could technically fall under ALL the above types, excluding Date and Boolean. However, to streamline our programs, we want to maximize speed and minimize RAM allocation. Therefore, it doesn't make much sense to declare a variable that will never reach beyond a value of 10 as Long, which takes up 8 times the amount of space as does a Byte.

## **Smart Practices**

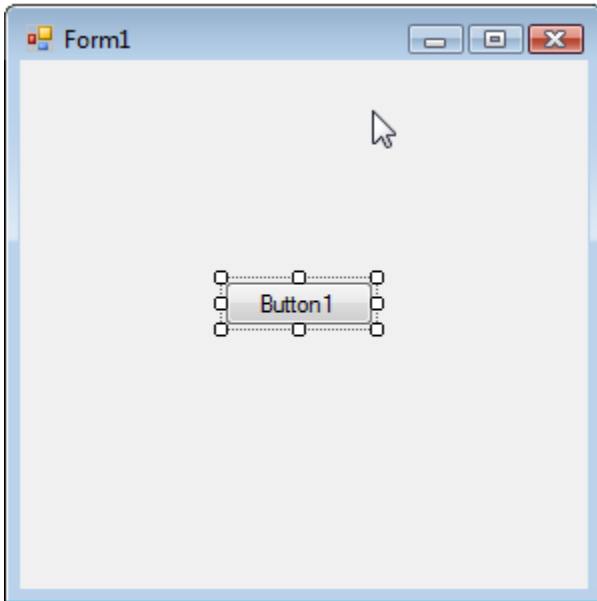
You should always make a practice of declaring your variables at the beginning of a procedure or class. Such a step isn't logistically necessary, but believe me, you'll be glad you declared all your variables in one place (rather than random, separate points in your code) when you debug your program or go back and make changes. And doesn't it make sense to declare your variable before actually using it?

## **Declaring and Using Variables Yourself**

Now that you have some grasp of just what a variable is, let's get some practice! Open Visual Studio 2008, and in Visual Basic, create a new Windows Form called BooleanTest. After you've created the form, go to the

---

Toolbox and add a button, to be placed in the center of your form. You should be seeing something much like the following:



Now to arrive at the Code Editor—specifically at the Button1\_Click procedure—double click the new button on your form. You should see the following code already on the screen:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    End Sub
End Class
```

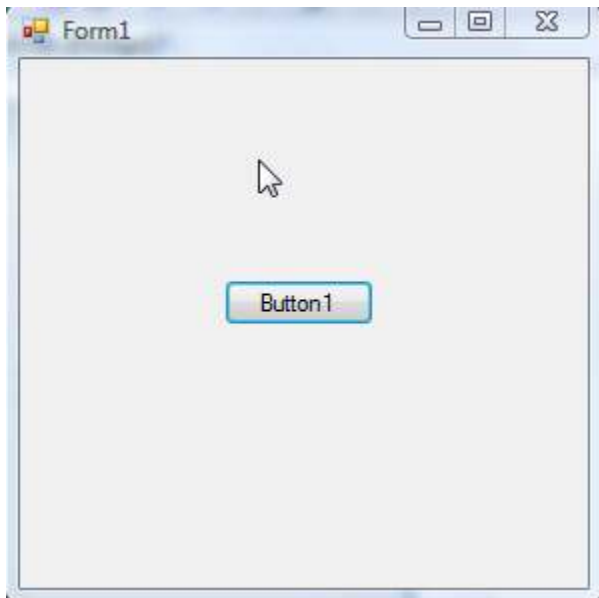
Place your cursor after “Handles Button1.Click” and hit return. On the new line, enter the following code:

---

```
Dim ClickMe As Boolean
ClickMe = True
Button1.Text = ClickMe
```

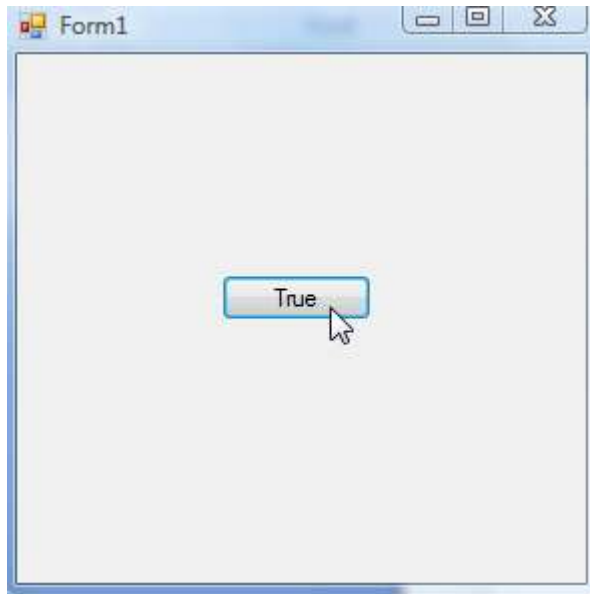
That's it! Our first line declares a variable named *ClickMe* as a Boolean Type—which means it can either be *True* or *False*. On the next line, we state that our variable's value will be *True*. The third line simply makes Button1's text property change to the value of *ClickMe* (which is, in this case, *True*).

Press F5 to debug and try it out! First, the following screen should appear:



Now click your newly created and programmed button. Its text should now read "True." Congratulations! You just declared and used your first Boolean type variable.

---



### **For More Practice**

For more practice declaring and utilizing different types of variables—as well as subroutines—see exercises in the document, “Using Sub Procedures and Variables to Create Your Own Calculator.”

---