

# Visual Web Development

Terry Marris October 2007

## 5 The Primitive Data Types

We look at some of the fundamental data types provided by Visual Basic.

### 5.1 The Number Types

We have already met the types Integer (whole numbers) and Double (numbers with a decimal point) in Chapter 4 - Number Input Output. VB provides a range of number types suitable for different applications.

<b>Signed Integer Type</b>	<b>Range</b>	<b>Application</b>
Short	from -32,768 up to 32,767	counting, small numbers, loop control
Integer (Int32)	from $-2.1 \times 10^9$ up to $+2.1 \times 10^9$	most commonly used, general purpose
Long (Int64)	from $-9.2 \times 10^{18}$ up to $+9.2 \times 10^{18}$	largest integers that can be stored in a computer

<b>Unsigned Integer Type</b>	<b>Range</b>	<b>Application</b>
Byte	from 0 up to 255	small numbers and PC character set
UShort (UInt16)	from 0 up to 65,535	larger numbers and loop control
UInteger (UInt32)	from 0 up to $4.29 \times 10^9$	large numbers, populations
ULong (UInt64)	from 0 up to $1.8 \times 10^{19}$	astronomical distances

At one time, when lack of memory was an issue, us programmers took care to minimise our data storage requirements by using Byte, UShort and Short integer types wherever possible. Nowadays, memory is plentiful and cheap and preserving memory is not so much of a problem. For most purposes, Integer is the type of choice.

<b>Real Type</b>	<b>Range</b>	<b>Application</b>
Single	from $-3.4 \times 10^{38}$ to $3.4 \times 10^{38}$	Scientific - 7 digit precision
Double	from $-1.7 \times 10^{308}$ to $1.7 \times 10^{308}$ , with 17 places after the decimal point	most commonly used general purpose
Decimal	from $-7.9 \times 10^{28}$ to $7.9 \times 10^{28}$	financial transactions

For most purposes involving floating point numbers, Double is the type of choice.

## 5.2 The Non-Number Types

The non-number primitive types include Char and Boolean.

<b>Sundry Types</b>	<b>Range</b>	<b>Application</b>
Char	from 0 up to 65535	full PC character set
Boolean	True, False	boolean expressions

Non-primitive types provided by VB include String and Date. We shall cover these later.

## 5.3 Choosing a Type

In general, we use the unsigned number types when we can guarantee that values will never become negative, and the signed number types otherwise. We use smaller types when we can guarantee that values will not fall outside the type's range. For example, we would not use a variable of type Byte (range 0..255) to hold the population size of the UK (some 60 million).

A character is a letter of the alphabet, a punctuation mark, a digit, a space or special characters such as a tab and carriage return/line feed. We use variables of type Char when we want to store a single character. If we choose to represent a persons gender with 'f' for female, 'm' for male and 'x' for unknown, we might write:

```
Dim chrGender As Char = "x"
```

Notice the double quotation marks around the value x.

We use Boolean when a value is either true or false. For example, you are either married, or you are not. So we might write:

```
Dim boolsMarried As Boolean = False
```

Note that there are no quotation marks around *False*. It is customary to include the word *Is* in a boolean variable name.

## 5.4 Conversion Between Types

In general, we input values as Text and then convert the text input to the required type - if we can.

The Convert class provides a rich range of conversion methods.

```
Dim intAge As Integer = Convert.ToInt32(txtAge.Text)
Dim dblHeight As Double = Convert.ToDouble(txtHeight.Text)
```

Some conversions result in a loss of precision (e.g. number of digits after a decimal point) without throwing an error or exception. Some conversions result in an overflow exception being thrown if the result is too large.

For example, when a Double is converted to a Single, a loss of precision may occur and an exception may not be thrown. But if the size of the Double value is too large to be stored in a Single variable then an overflow exception is thrown.

## 5.5 The Assignment Operator

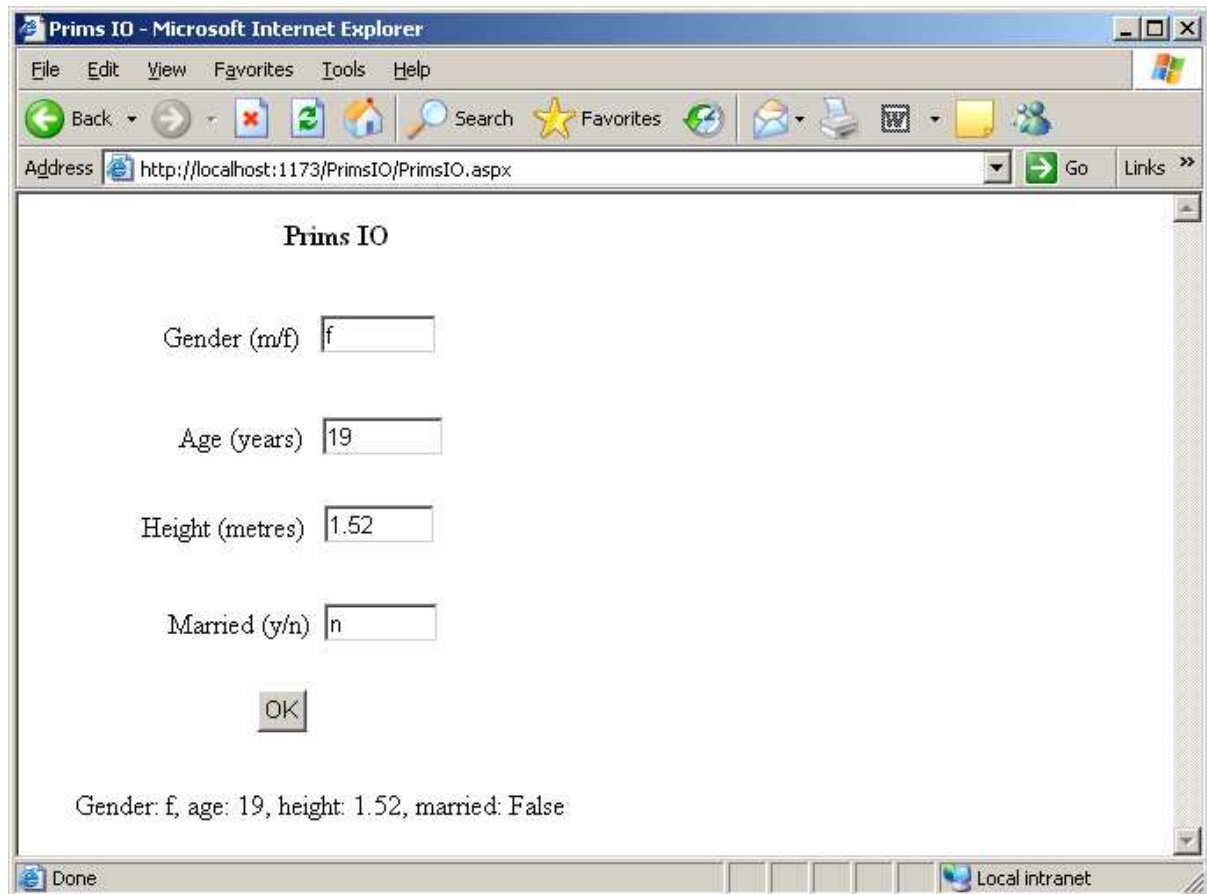
In

```
Dim intNum As Integer = 50
```

the = symbol is known as the assignment operator. The assignment operator copies the value on its right into the variable on its left. So, in our example, it copies the value 50 into the variable intNum.

## 5.6 Primitives Input Output

An example of input-output using some primitive data types is shown below.



The user enters f, 19, 1.52 and n, and clicks OK. The computer displays Gender: g, age: 19, height: 1.52, married: false.

The code behind the OK button is shown on the next page.

```

Protected Sub btnOK_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnOK.Click
    Dim chrGender As Char = "x"
    Dim intAge As Integer = 0
    Dim dblHeight As Double = 0.0
    Dim boolIsMarried As Boolean = False

    Try
        chrGender = Convert.ToChar(txtGender.Text)
        chrGender = Char.ToLower(chrGender)
        If chrGender <> "m" And chrGender <> "f" Then
            chrGender = "x"
        End If

        intAge = Convert.ToInt32(txtAge.Text)
        dblHeight = Convert.ToDouble(txtHeight.Text)

        txtIsMarried.Text = txtIsMarried.Text.ToLower()
        If txtIsMarried.Text.Equals("y") Then
            boolIsMarried = True
        Else
            boolIsMarried = False
        End If

        lblSummary.Text = "Gender: " + chrGender.ToString() + _
            ", age: " + intAge.ToString() + _
            ", height: " + dblHeight.ToString() + _
            ", married: " + boolIsMarried.ToString()

    Catch ex As Exception
        lblSummary.Text = "Errors in input"
    End Try
End Sub

```

Declare and initialise variables

Convert Text to lower case Char. Ensure chrGender is either m, f or x

Convert Text to Integer and Double respectively

Convert Text to lower case and assign True to boolIsMarried if input is 'y'

Convert each item to a String

Catch some input errors

Notice the use of the continuation operator, `_`, (underscore). This allows program code to be continued over several lines. There must be a space immediately before the continuation operator.

## 5.7 Exercise

1. Try out the program PrimsIO specified on pages four and five above.
2. Identify appropriate data types for each of the following:
  - a. bags of beans, kilograms per bag of beans and the total number of kilograms of beans that might be carried by a container ship.
  - b. number of heartbeats in a lifetime. Assume a lifetime is 100 years and a heart rate is 72 beats per minute.
  - c. temperature of a live human in degrees Celsius
  - d. your bank balance
  - e. The accounts of a large company such as Microsoft
3. Create a program to display fltValue shown in the following code fragment:

```
Dim shrtNum As Short = 3  
Dim shrtDenom As Short = 4  
Dim fltValue As Single = shrtNum / shrtDenom
```

Explain each line of the program.