Learning VB.Net

Tutorial 08 – For statement

Hello everyone… welcome to vb.net tutorials. These are going to be very basic tutorials about using the language to create simple applications, hope you enjoy it. If you have any notes about it, please send them to **notes@mka-soft.com** I will be happy to answer them. Finally if you find these tutorials are useful, it would be nice from you to send a small donation via PayPal to **donation@mka-soft.com**.

Tutorial posted on 2009-December-05.

For Loop

Almost every language has some kind of looping statement (in case you don't know what that does, it allows the execution of a number of statements several times). In VB.NET there are a number of looping statements, these are REPEAT, DO and FOR. We will talk about the easiest of them all which is the FOR loop. The FOR loop is written like this:

```
For variable = Min To Max Step JumpStep
    Statement
    Statement
    …
Next
```

The code will execute the statements between the For and Next parts by setting the variable to Min, increasing it by one every time until it reaches Max. To make things clear consider this example

```
For A = 1 To 10
    MsgBox("The value of A is:" & A)
Next
```

The result of executing the code above is ten message boxes telling you the value of A every time.

Now let us consider another example. Here you have a form with a textbox and a ComboBox. You select font size from the combo box and the text size changes accordingly.

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
                       System.EventArgs) Handles MyBase.Load
    ' this part fills the combobox with the sizes of font that we
    ' can pick from
    Dim I As Double
    For I = 12 To 70
        ComboBox1.Items.Add(I)
    Next
End Sub

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As
            System.Object, ByVal e As System.EventArgs) Handles
            ComboBox1.SelectedIndexChanged
    ' this part changes font size
    Dim F As Font
    F = New Font("COURIER NEW", ComboBox1.Text)
    TextBox1.Font = F
End Sub
```

You notice two things in the example, first the loop does not start from 1, it starts from 12, you can start from any value you like, for example start from 283732, -12, 0, 88888, etc. Second the data type of the variable `I` is double. You can use Single, Double, Integers, Long… You are not restricted here.

If we want to display the numbers between 5 and 50 by adding 5 to the previous in each step then:

```
Dim Counter As Integer
For Counter = 5 To 50 Step 5
    MsgBox(Counter)
Next
```

Assume we need the values 0, 0.1, 0.2, 0.3, 0.4, 0.5… 1.0. This can be done in two ways:

```
Dim Counter As Integer
Dim V As Double
For Counter = 0 To 10
    V = Counter / 10.0
    MsgBox(V)
Next
```

This method requires extra variable, and does not take advantage of the for loop. A better way is to use the STEP keyword with double or single data type to make it easy for us:

```
Dim Counter As Double
For Counter = 0 To 1 Step 0.1
    MsgBox(Counter)
Next
```

One last important thing to notice is that the initial value of the variable should always be smaller than or equal to the value after the `To` keyword, otherwise the for loop does not get executed and it is skipped. For example:

```
For Counter = 10 To 1
    MsgBox(Counter)
Next
```

Will never give you message box at all. To fix this and make the count down work, just put a negative step value:

```
For Counter = 10 To 1 Step -1
    MsgBox(Counter)
Next
```

These are most of the details needed to work with the For loop. The next example is a simple one showing how to use the FOR loop to identify Prime number.

Prime numbers are numbers that can only be divided by themselves and 1 with remainder=0. So this means if we have number 9212, we should check the remainder of dividing this number over all the values from 9212 to 2 and it should never give a zero if

it is a prime. Without for loop this is very hard to compute. The code to calculate the prime number is:

```
Dim MyNumber As Integer
Dim RemainderIsZeroFlag As Boolean
Dim I As Integer

' read a number from the screen
MyNumber = InputBox("Enter a number")

' this is a flag to tell us when the condition
' of prime number is not satisfied
RemainderIsZeroFlag = False

' start checking all the numbers
For I = 2 To MyNumber - 1
    ' if the condition is not satisfied
    If MyNumber Mod I = 0 Then
        ' mark that the remainder is not zero
        RemainderIsZeroFlag = True
    End If
Next

' if there was any remainder then tell the user
' that the number is not prime, else it is.
If RemainderIsZeroFlag Then
    MsgBox("The number is not prime")
Else
    MsgBox("The number is prime")
End If
```

Next tutorial we will start working with arrays and collections, and things will get more exciting. If you have any notes or questions send them to notes@mka-soft.com.

Thank you.

mkaatr.