Developing A Name List

In this lesson we will develop a list of names in a list box and then display every 2nd name in another list box. We will use a **While Loop** to input the names and then a **For Loop** to display them.

RGB (red, green, blue)

Virtually any shade can be reproduced using a combination of the colours red, green and blue. Artist mix paint to produce varied shades.
Televisions (pre-digital) use red green and blue tubes in combination to create the various colours and pictures on screen.

In Visual Basic the 3 colours are given a brightness value between 0-255.
An rgb setting of (0,0,0,) displays an absence of colour which is black.

```
Private Sub btnStart_Click(ByVal sender As System.Object, ByVal
    Me.BackColor = System.Drawing.Color.FromArgb(0, 0, 0)
End Sub
```
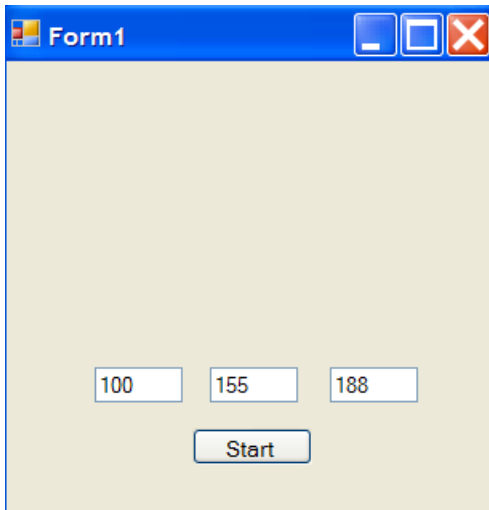
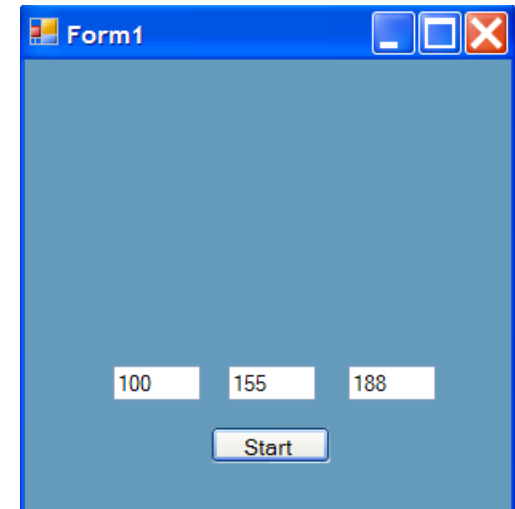red value

blue value

Me refers to the Form

green value

Form1
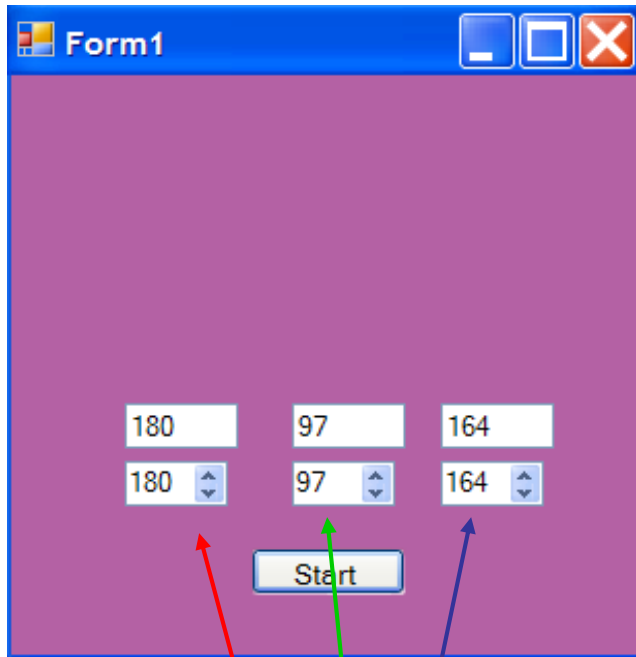
Start
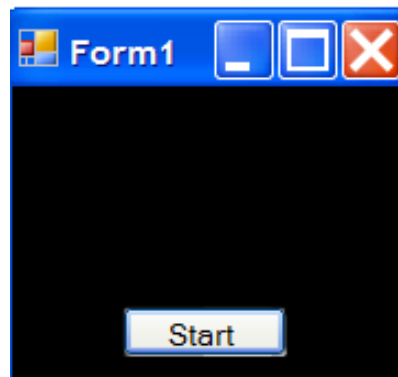
User Control Over Colours

In this modification we will allow the user to enter values between 0 and 255 for each rgb value.

```
Private Sub btnStart_Click(ByVal sender As System.Object,
    Dim r As Integer
    Dim g As Integer
    Dim b As Integer
    r = Val(txtRed.Text)
    g = Val(txtGreen.Text)
    b = Val(txtBlue.Text)
    Me.BackColor = System.Drawing.Color.FromArgb(r, g, b)
End Sub
```

Numeric Up Down Counter

97

Form1

180   97   164

180 ⇕   97 ⇕   164 ⇕

Start

Increment	1
InterceptArrowKeys	True
⊞ Location	**208, 187**
Locked	False
⊞ Margin	3, 3, 3, 3
Maximum	**255**
⊞ MaximumSize	0, 0
Minimum	0
⊞ MinimumSize	0, 0
Modifiers	**Friend**
ReadOnly	False
RightToLeft	No
⊞ Size	**50, 22**
TabIndex	**6**
TabStop	True
Tag	
TextAlign	Left

Set minimum and maximum values
of the counters in properties.

```
r = Val(numRed.Value)
g = Val(numGreen.Value)
b = Val(numBlue.Value)
```

Increment RGB Value with Button Click

In this example, each time that the button is clicked the red value is incremented by five.

```
Private Sub btnStart_Click(ByVal sender As System.Object, ByVal
    txtRed.Text = r
    r = r + 5
    Me.BackColor = System.Drawing.Color.FromArgb(r, g, b)
End Sub
```
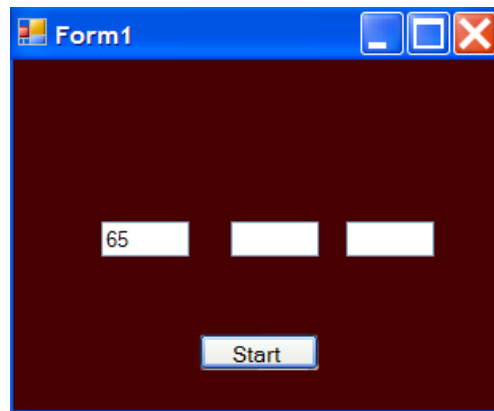
Don't Go Over 255!

```
Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Ha
    txtRed.Text = r
    r = r + 5
    Me.BackColor = System.Drawing.Color.FromArgb(r, g, b)
End Sub
Class
```

⚠ **ArgumentException was unhandled**   ✕

Value of '260' is not valid for 'red'. 'red' should be greater than or
equal to 0 and less than or equal to 255.

Troubleshooting tips:

Visual Basic will complain if you try and use a number greater than a 255 rgb value.
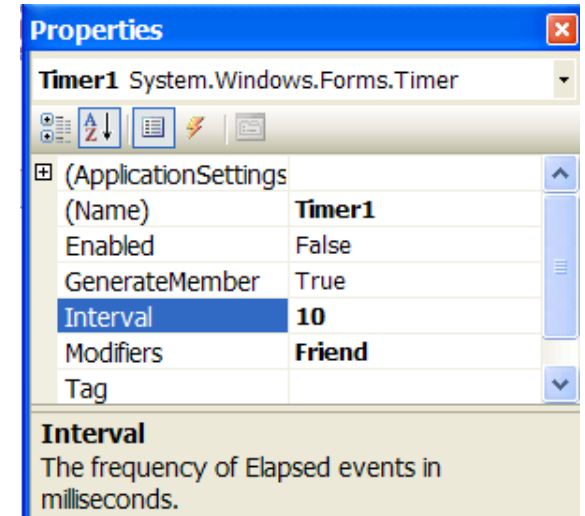
Add Timer to Fire A Sub()

Button click causes timer to start.

```
Private Sub btnStart_Click(ByVal sender As System.Object, By'
    Timer1.Start()
End Sub
Sub ColourChange()
    txtRed.Text = r
    txtGreen.Text = g
    txtBlue.Text = b
    r = r + 1
    g = g + 1
    b = b + 1
    Me.BackColor = System.Drawing.Color.FromArgb(r, g, b)
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal
    ColourChange()
End Sub
```
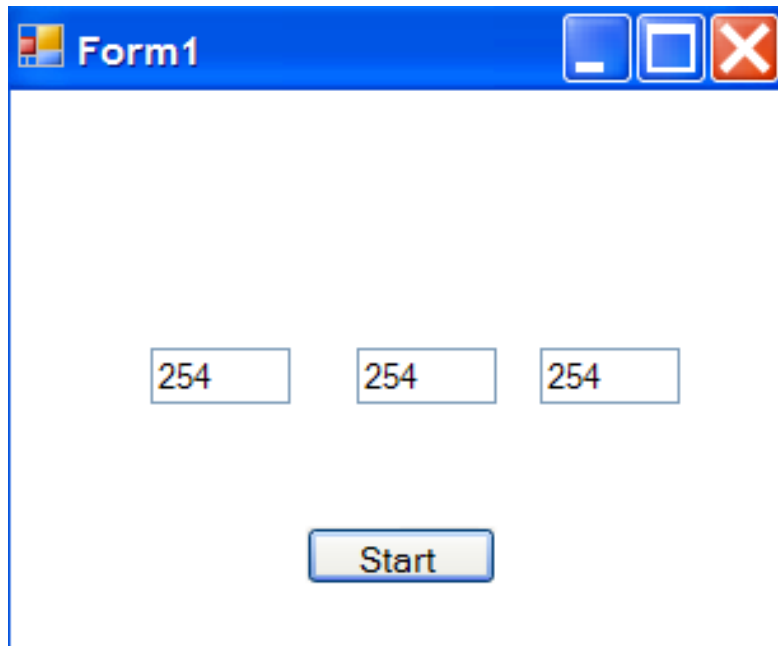
ColourChange() gradually changes rgb values.

Timer fires every 10 milliseconds calling ColourChange().

Program will run until rgb values reach 255 at which point it crashes. Use while statements to constrain values.

Stop Timer

```
Private Sub Timer1_Tick(ByVal sender
    If (r < 255) Then
        ColourChange()
    Else
        Timer1.Stop()
    End If

End Sub
```

Add A 2ⁿᵈ Timer and Reverse Values

```vb
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal
    If (r < 255) Then
        ColourChange()
    Else
        Timer1.Stop()
        Timer2.Start()
    End If
End Sub
Sub ColourChangeBack()
    txtRed.Text = r
    txtGreen.Text = g
    txtBlue.Text = b
    r = r - 1
    g = g - 1
    b = b - 1
    Me.BackColor = System.Drawing.Color.FromArgb(r, g, b)
End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal
    If (r > 0) Then
        ColourChangeBack()
    Else
        Timer2.Stop()
        Timer1.Start()
    End If
End Sub
```

Psychedelic Programming