



# Fundamentals of Programming in Visual Basic (VB)

---

- Visual Basic Events
- Simple Statement



# Event

---

- An **event** is an action, such as the user clicking on a button
    - Usually, nothing happens in a Visual Basic program until the user does something and generates an event.
    - What happens is determined by statements.
- ## Visual Basic Events



# Assignment Statements

---

- Assign a value to a property.

General Form:

*source = value*

- A value on the right-hand side of = is assigned to the left-hand side of =.





# Sample Statements

- `textBox.ForeColor = Color.Red` —
- `textBox.Visible = True`
- `textBox.Text = "Hello World"`

**Color** is a structure allowing us to specify various colors

General Form:

***`controlName.property = setting`***

Value represented by **setting** is stored into **controlName.property**



# Lab

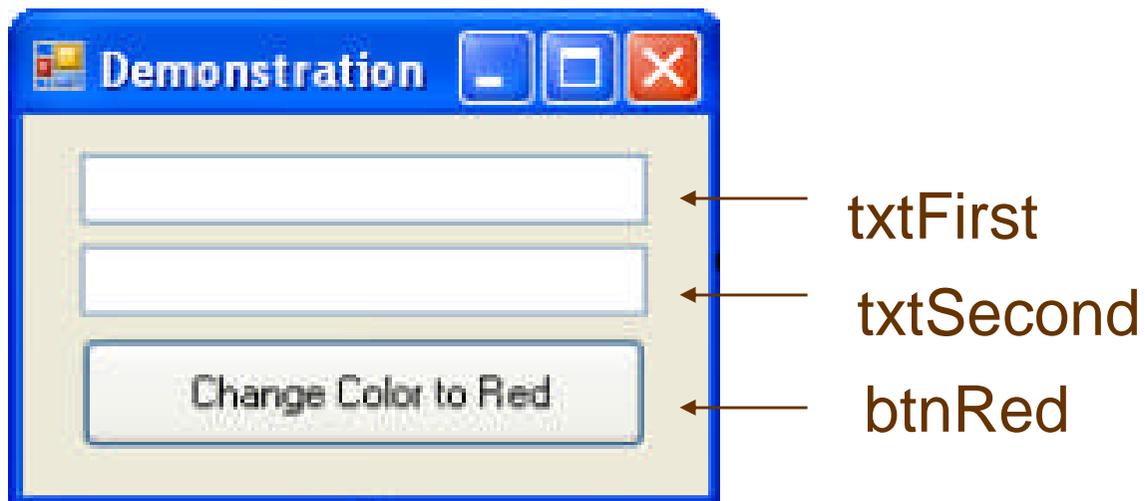
---

- Refers to the examples in the Lab



# Sample Form

---





# Focus

---

- When you click on a text box, a cursor appears in the text box, and you can type into the text box.
- Such a text box is said to have the **focus**.
- If you click on another text box, the first text box loses the focus and the second text box receives the focus.



# Examples of Events

- `btnShow.Click`
- `textBox.TextChanged`
- `textBox.Leave`

This event occurs when a mouse is clicked on the button `btnShow`

This event occurs when user changes the text of `textBox`

This event occurs when the input focus leaves **`textBox`**

General Form:

***`controlName.event`***



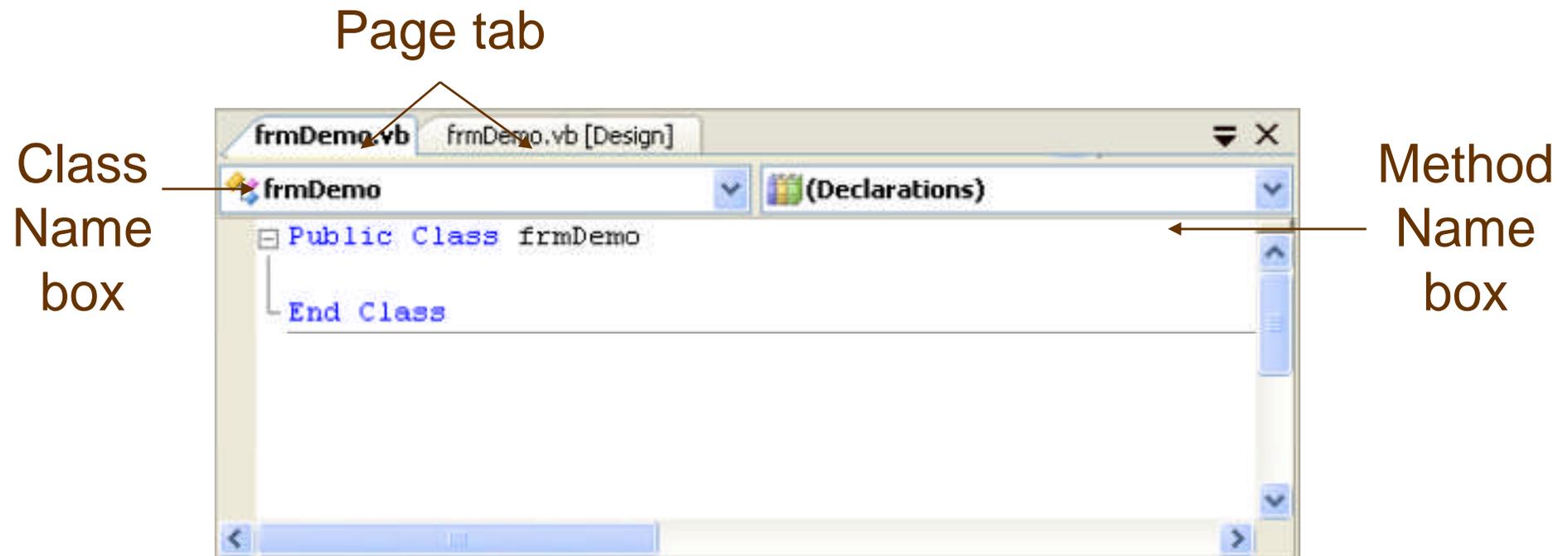
# The three steps in creating a Visual Basic program:

---

1. Create the interface; that is, generate, position, and size the objects.
2. Set properties; that is, configure the appearance of the objects.
3. Write the code that executes when events occur.



# Code Window

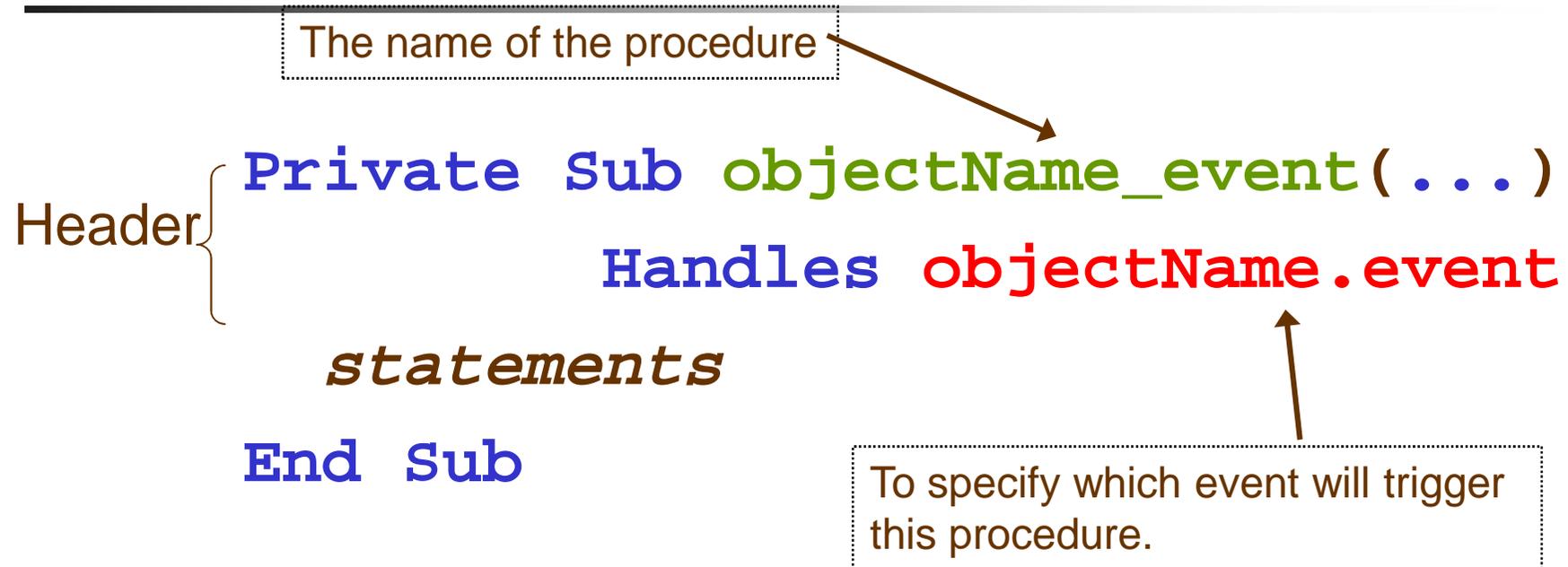


Page tab:

1. frmDemo.vb is a code window
2. frmDemo.vb [design] is a design window (to manipulate the window appearance)



# Structure of an Event Procedure



(...) is filled automatically with (ByVal sender As System.Object, ByVal e As System.EventArgs)



# Code Window

Page tab

Class Name box

Method Name box

```
frmDemo.vb frmDemo.vb [Design]
frmDemo
Public Class frmDemo
End Class
(Declarations)
```



## Create an Outline for an Event Procedure; i.e. header and End Sub

---

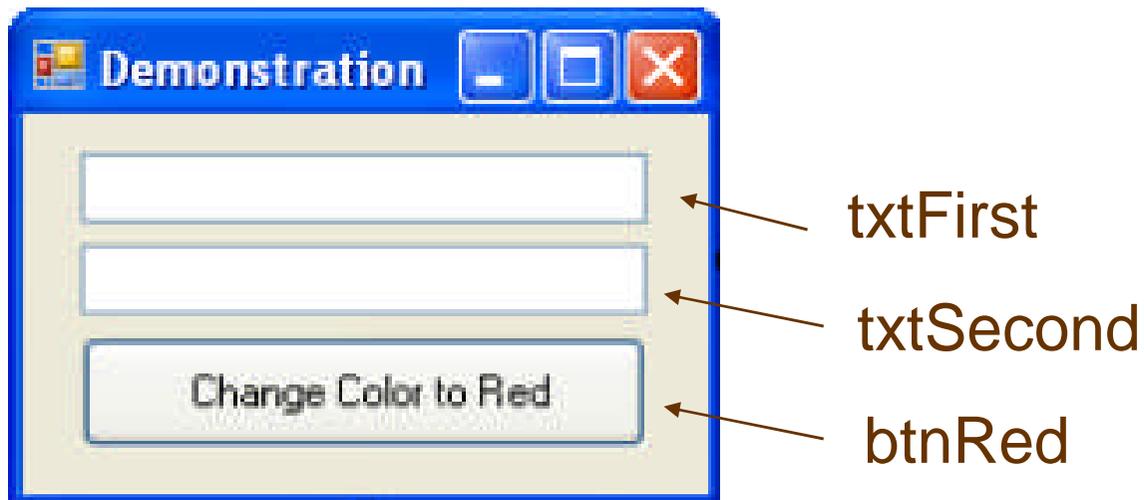
- 1. Double-click on a control  
or**
- 2. Use the Class Name and Method  
Name boxes.**

**(We primarily use the first  
method.)**



# Sample Form

---



**Double Click on txtFirst**



# Code for Walkthrough

---

```
Public Class frmDemo
```

```
    Private Sub txtFirst_TextChanged(...)
```

```
        Handles txtFirst.TextChanged
```

```
    End Sub
```

```
End Class
```

TextChanged event occurs when the user changes the text of a [TextBox](#)



# Code for Walkthrough

---

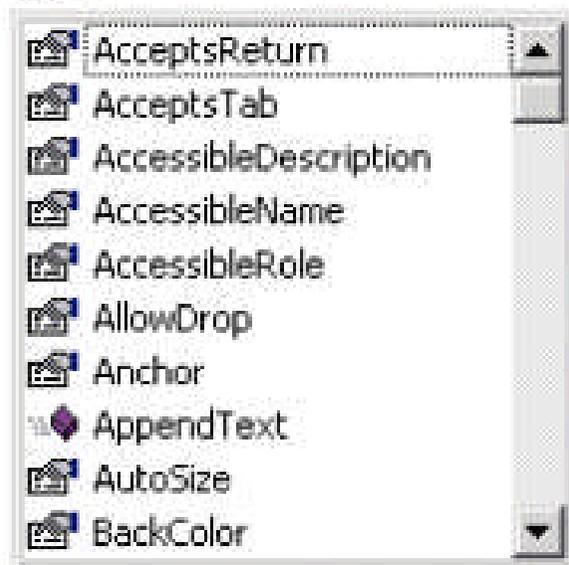
```
Public Class frmDemo
    Private Sub txtFirst_TextChanged(...)
        Handles txtFirst.TextChanged
            txtFirst.ForeColor = Color.Blue
        End Sub
    End Class
```



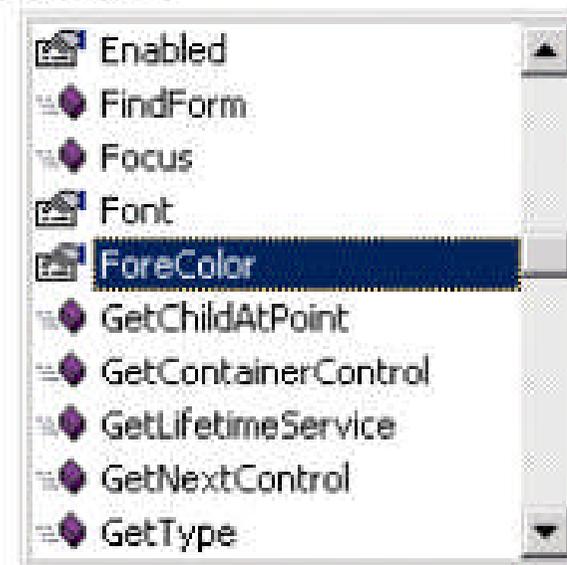
# IntelliSense

Automatically pops up to give the programmer help.

`txtFirst.`



`txtFirst.For`

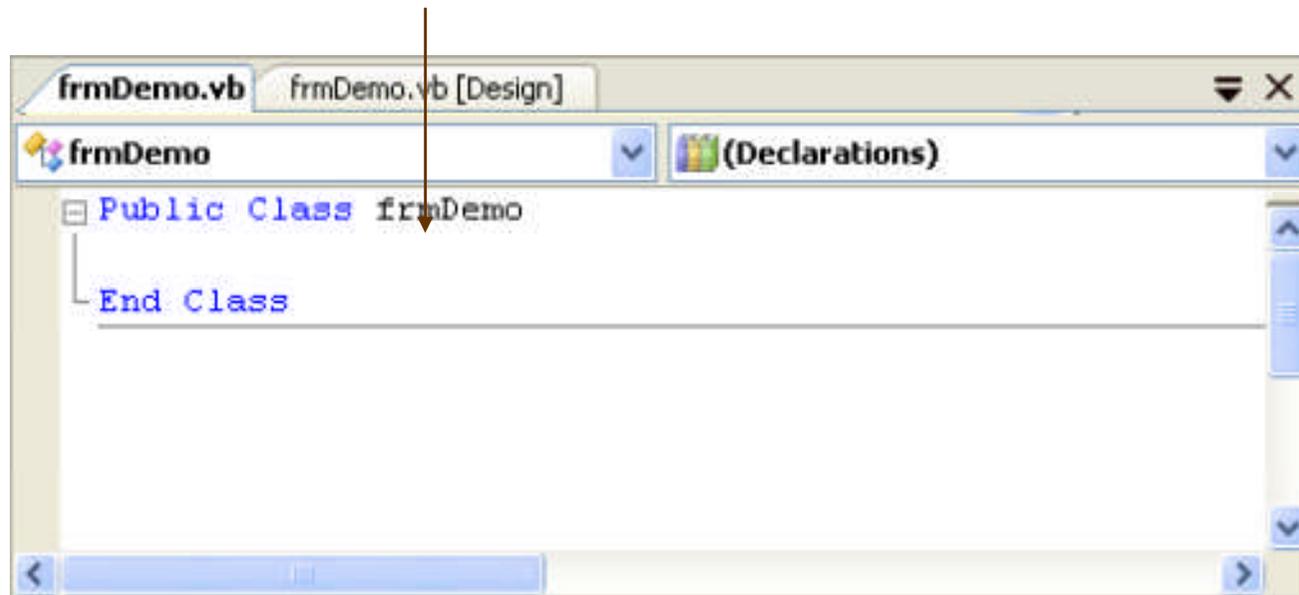




# Code Window

---

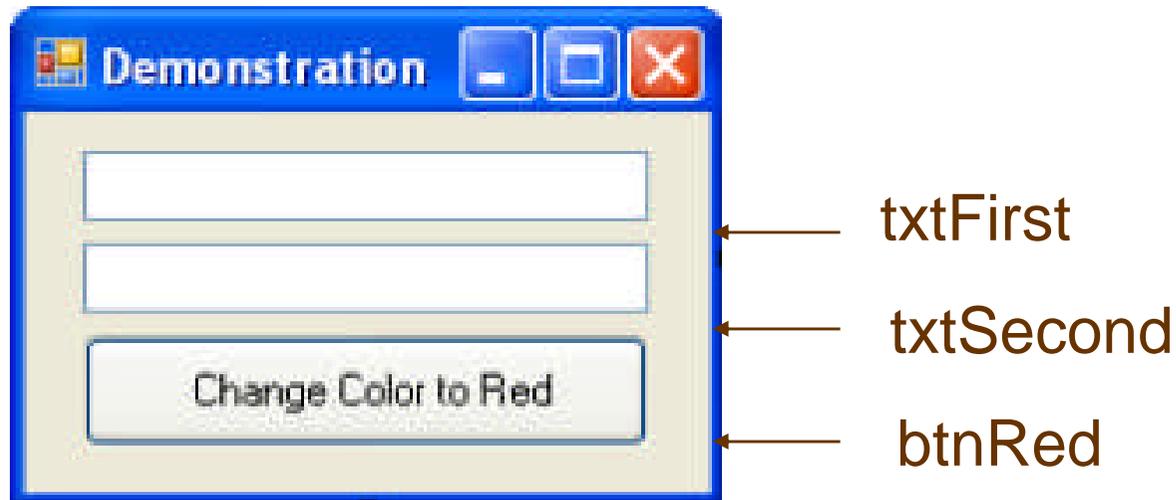
Click tab to return to Form Designer





# Sample Form

---



txtFirst

txtSecond

btnRed

**Double-click on btnRed**



# Code for Walkthrough

---

```
Public Class frmDemo
    Private Sub txtFirst_TextChanged(...)
        Handles txtFirst.TextChanged
        txtFirst.ForeColor = Color.Blue
    End Sub

    Private Sub btnRed_Click(...)
        Handles btnRed.Click

    End Sub
End Class
```



# Code for Walkthrough

---

```
Public Class frmDemo
    Private Sub txtFirst_TextChanged(...)
        Handles txtFirst.TextChanged
        txtFirst.ForeColor = Color.Blue
    End Sub

    Private Sub btnRed_Click(...)
        Handles btnRed.Click
        txtFirst.ForeColor = Color.Red
    End Sub
End Class
```



## Event Procedure txtFirst.Leave

---

- Select txtFirst from Class Name box drop-down list.
- Select Leave from Method Name box drop-down list.



# Code for Walkthrough

---

```
Private Sub txtFirst_Leave(...) Handles txtFirst.Leave
```

```
End Sub
```

```
Private Sub txtFirst_TextChanged(...)
    Handles txtFirst.TextChanged
```

```
    txtFirst.ForeColor = Color.Blue
```

```
End Sub
```

```
Private Sub btnRed_Click(...) Handles btnRed.Click
```

```
    txtFirst.ForeColor = Color.Red
```

```
End Sub
```



# Code for Walkthrough

---

```
Private Sub txtFirst_Leave(...) Handles txtFirst.Leave
    txtFirst.ForeColor = Color.Black
End Sub
```

```
Private Sub txtFirst_TextChanged(...)
    Handles txtFirst.TextChanged
    txtFirst.ForeColor = Color.Blue
End Sub
```

```
Private Sub btnRed_Click(...) Handles btnRed.Click
    txtFirst.ForeColor = Color.Red
End Sub
```



# Header of Event Procedure

---

```
Private Sub btnRed_Click(...) Handles btnRed.Click
```

↑  
Name, can  
be changed.

↑  
Identifies the  
event that  
triggers the  
procedure

```
Private Sub Button_Press(···) Handles btnRed.Click
```



# Handling Multiple Events

---

Event procedure can be invoked by two events.

```
Private Sub Button_Click(...)  
    Handles btnRed.Click, txtSecond.Leave  
    txtFirst.ForeColor = Color.Red  
End Sub
```



# Altering Properties of the Form

---

- The following won't work:  
**frmDemo.Text = "Demonstration"**
- The form is referred to by the keyword *Me*.  
**Me.Text = "Demonstration"**