

HTML Controls

Introduction

- The HTML server controls are defined in the `System.Web.UI.HtmlControls` namespace. This namespace includes more than 20 different HTML controls' classes.

The `HtmlControl` Class

- The `HTMLControl` class is the base class for all the HTML controls' classes. The `HtmlInputControl` and the `HtmlContainerControl` classes extend `HTMLControl` and they are the base classes for many other classes.
- The important properties defined in `HTMLControl` include the following: `Attributes`, `Disabled`, `Style` and `TagName`.

The `HtmlControl` Class

- We can use the `Attributes` property for accessing as well as for adding attributes to the control tag.
- The `Disabled` property sets whether the control should be rendered as a disabled control. Disabled controls are usually rendered in gray.
- The `Style` property holds a collection of CSS attributes that are applied to the control.
- The `TagName` property holds the control tag name (e.g. `table`).

The HtmlControl Class

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication16.WebForm1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<body>
  <form id="form1" runat="server">
    <input id="SampleText" type="text" value="bla bla bla" style="font: 24pt
      verdana;width:320; background:yellow" runat="server"/>
  </form>
  <asp:DataList id="OurList" runat="server">
    <ItemTemplate>
      <%# DataBinder.Eval(Container.DataItem, "CSSAttributeName") %>
      :
      <%# DataBinder.Eval(Container.DataItem, "CSSAttributeValue") %>
    </ItemTemplate>
  </asp:DataList>
</body>
</html>
```



The HtmlControl Class

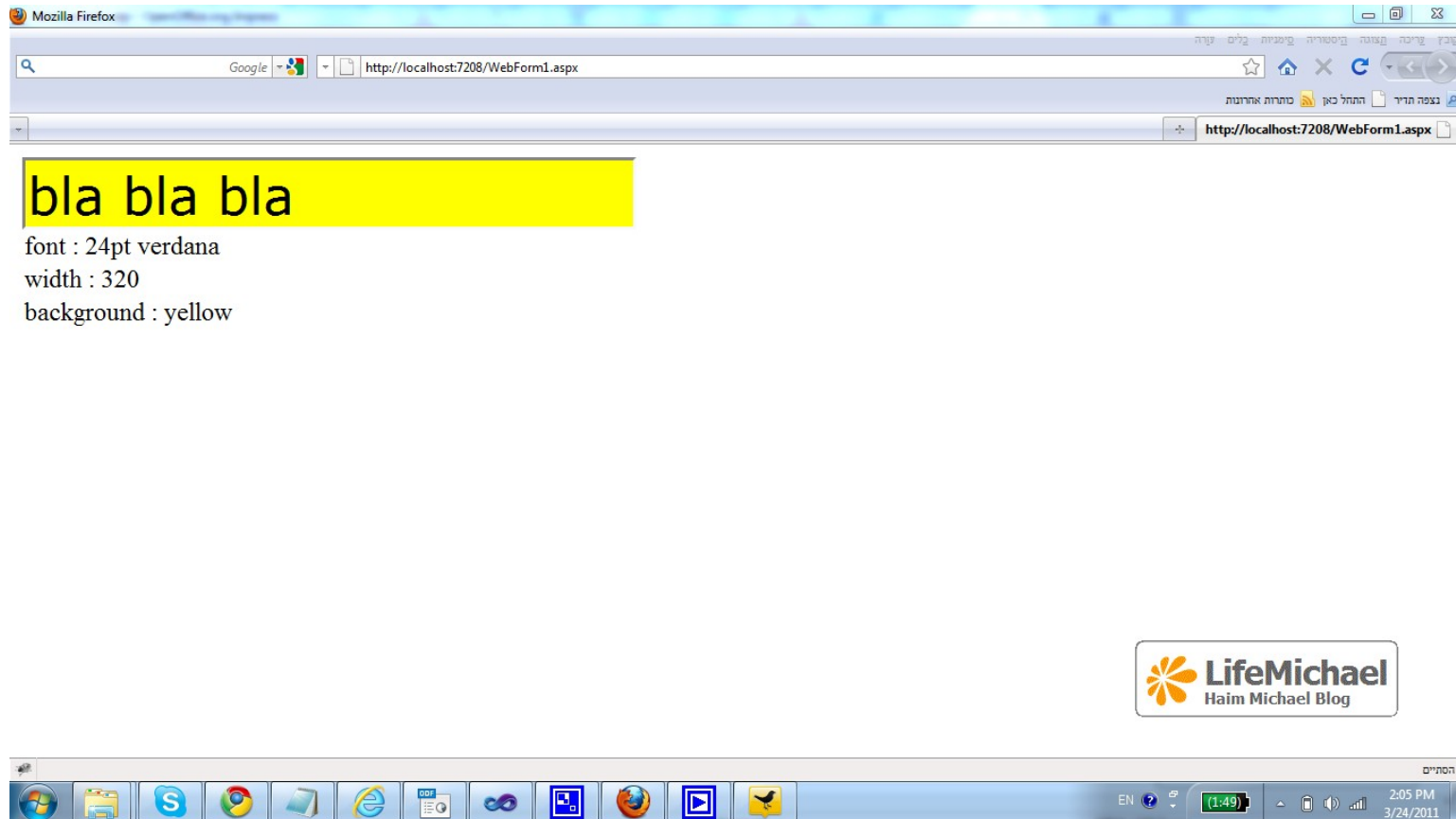
```
namespace WebApplication16
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            DataTable table = new DataTable();
            DataRow row;
            table.Columns.Add(
                new DataColumn("CSSAttributeName", typeof(String)));
            table.Columns.Add(
                new DataColumn("CSSAttributeValue", typeof(String)));

            // The Style property of the MyText control returns
            // a CssStyleCollection object.
            System.Collections.IEnumerator keys =
                SampleText.Style.Keys.GetEnumerator();
        }
    }
}
```

The HtmlControl Class

```
while (keys.MoveNext())
{
    String cssStyleKey = (String)keys.Current;
    row = table.NewRow();
    row[0] = cssStyleKey;
    row[1] = SampleText.Style[cssStyleKey];
    table.Rows.Add(row);
}
DataView dataView = new DataView(table);
OurList.DataSource = dataView;
OurList.DataBind();
}
}
```

The HtmlControl Class



The `HtmlInputControl` Class

- The `HtmlInputControl` is the base class HTML controls' classes that are based on the HTML well known Input element.
- The `HtmlInputControl` has two important properties. The `Type` property that gets the type of the control (e.g. 'text' when the control is a simple text field) and the `Value` property that allows us setting the value.

The `HtmlContainerControl` Class

- The `HtmlContainerControl` is the base class HTML controls' classes that can have child controls.
- Many of the HTML tags that function as containers are represented (on the server side) using objects instantiated from the `HtmlGenericControl` class, that extends `HtmlContainerControl`. When a tag isn't recognized as a tag for which there is a control class then the server side uses the `HtmlContainerControl` class.

```
<div runat="server"></div>
```

Programmatically Controls Creating

- We can programmatically create controls during the execution of the Page_Load function.

Programmatically Controls Creating

```
namespace WebApplication17
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            HtmlTable table = new HtmlTable();
            table.Border = 2;
            table.BorderColor = "blue";
            HtmlTableRow row = new HtmlTableRow();
            HtmlTableCell cell11 = new HtmlTableCell();
            cell11.InnerHtml = "blabla11";
            HtmlTableCell cell12 = new HtmlTableCell();
            cell12.InnerHtml = "blabla12";
            row.Cells.Add(cell11);
            row.Cells.Add(cell12);
            table.Rows.Add(row);
            this.Controls.Add(table);
        }
    }
}
```



Programmatically Controls Creating

