

Life Cycle

Introduction

- Each and every ASP.NET web page has a life cycle. When the ASP.NET server side receives a request for a specific *.aspx web page the `Page` derived class is instantiated using the default constructor.
- Once the `Page` derived class is instantiated a series of events are triggered.

The PreInit Event

- This event takes place when the framework allocates the web controls, apply the themes, set the master page and set the user profiles.

The Init Event

- This event takes place when the framework sets the properties of the web controls with their previous values.

The Load Event

- The load event is the first event that takes place once the Page derived class is instantiated.
- We can override `Page_Load` in order to hook into this event.

```
protected void Page_Load(object sender, EventArgs e)
{
    ...
}
```

The PreRender Event

- This event takes place when the framework completes with binding and configuring the controls and they are finally ready to render their data back to the client.

The UnLoad Event

- This event takes place when the page and its controls finish with rendering their data back to the client. This event takes place just before the page object is about to be destroyed.
- We can use this event for cleaning resources, such as closing files.

```
protected override void Page_Unload(object sender, EventArgs e)
{
    ...
}
```

Sample

- The following code sample shows when the `Page_UnLoad` and the `Page_Load` two methods are invoked.

Sample

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication9.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">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>

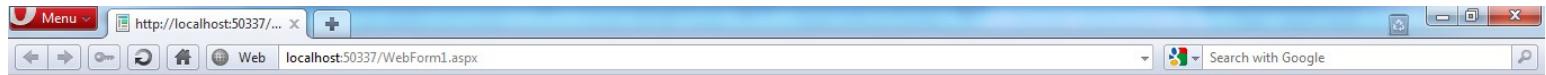
    </div>
  </form>
</body>
</html>
```



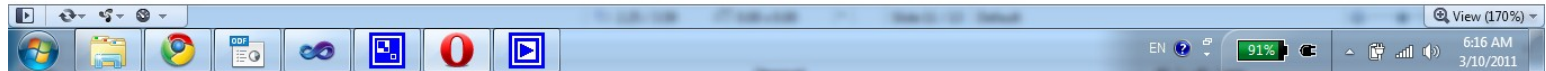
Sample

```
namespace WebApplication9
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        static int PageUnloadCounter = 0;
        static int PageLoadCounter = 0;
        protected void Page_Load(object sender, EventArgs e)
        {
            Response.Write("<br>hash code = " + this.GetHashCode());
            Response.Write("<br>page unload counter = " + PageUnloadCounter);
            Response.Write("<br>page load counter = " + PageLoadCounter);
            PageLoadCounter++;
        }
        protected void Page_UnLoad(object sender, EventArgs e)
        {
            PageUnloadCounter++;
        }
    }
}
```

Sample



hash code = 41102531
page unload counter = 10
page load counter = 10



The AutoEventWireup Attribute

- This attribute states whether we can hook up the life cycle events or not. Its default value is true.

The Error Event

- This event takes place when an error occurs in one of the methods that were defined in Page. If the error was hand-held through the exceptions handling mechanism then this event doesn't take place.

```
void Page_Error(object sender, EventArgs e)
{
    ...
    Exception e = Server.GetLastError();
    ...
    Server.ClearError();
    ...
}
```

The Error Event

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication10.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">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
      <asp:Label ID="Label1" runat="server" Text=" / "></asp:Label>
      <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
    </div>
  </form>
</body>
</html>
```



The Error Event

```
<asp:Label ID="Label2" runat="server" Text=""></asp:Label>  
<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>  
<asp:Button ID="Button1" runat="server" onclick="Button1_Click"  
    Text="calculate" />  
</div>  
</form>  
</body>  
</html>
```

The Error Event

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication10
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void Page_Error(object sender, EventArgs e)
        {
            Exception exception = Server.GetLastError();
            Response.Write("error happened, message is " + exception.Message);
            Server.ClearError();
        }
    }
}
```


The Error Event

```
protected void Button1_Click(object sender, EventArgs e)
{
    double numA = Double.Parse(this.TextBox1.Text);
    double numB = Double.Parse(this.TextBox2.Text);
    double result = numA / numB;
    this.TextBox3.Text = result.ToString();
}
}
```

The Error Event

