

第5章 *ASP.NET*的常用对象

5.1 ASP.NET对象概述

5.2 Page对象

5.3 Response对象

5.4 Request对象

5.5 Server对象

5.6 Application对象

5.7 Session对象

5.8 Cookie对象

5.9 配置Global.asax文件

5.1 ASP.NET对象概述

ASP.NET中常用的内置对象及其说明如下表所示。

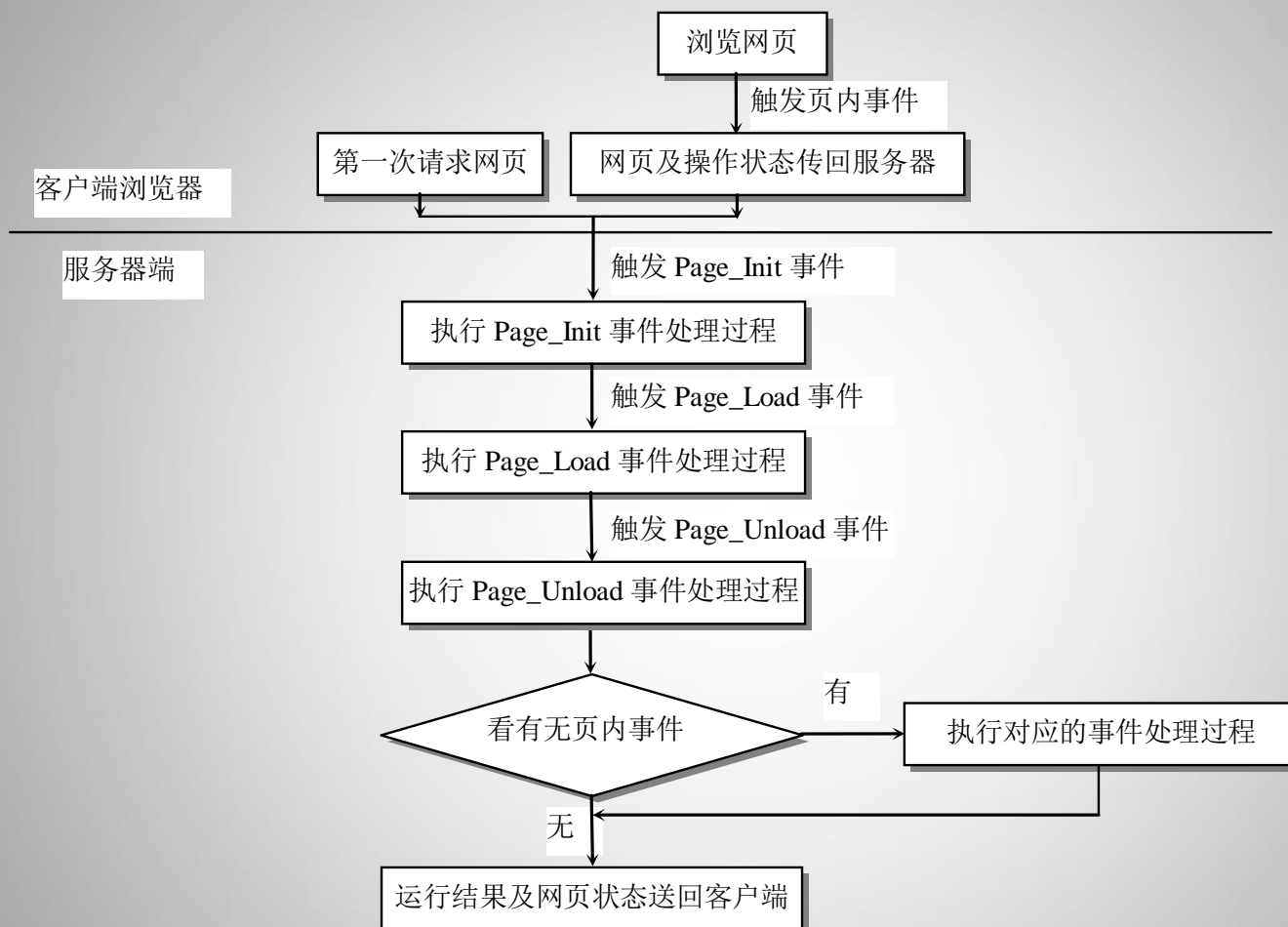
对象名	说明
Page	用于操作整个网页。
Response	用于向浏览器输出信息。
Request	提供对当前网页请求的访问。
Server	提供服务器端的一些属性和方法。
Application	提供对所有会话的应用程序范围的方法和事件的访问。还提供对可用于存储信息的应用程序范围的缓存的访问。
Session	用于存储特定用户的会话信息。
Cookie	用于设置或获取Cookie信息。

5.2 Page对象

Page对象其实就是Web应用程序的.aspx文件，又称为页面。

也就是说，每一个ASP.NET网页都是一个Page对象，Page对象是由System.Web.UI命名空间中的Page类来实现的，Page类与扩展名为.aspx的文件相关联，这些文件在运行时被编译为Page对象，并缓存在服务器内存中。

5.2.1 ASP.NET网页的处理过程



5.2.1 Page对象的属性

Page对象的常用属性及其说明如下表所示，除此之外，Page对象还包括Response、Request、Server、Session和Application对象属性。

属性	说明
QueryString	获取请求的URL的查询字符串部分。
ErrorPage	获取或设置错误页，在发生未处理的页异常的事件时请求浏览器将被重定向到该页。
Form	获取网页的HTML窗体。
IsPostBack	获取一个值，该值指示该页是否正为响应客户端回发而加载，或者它是否正被首次加载和访问。
IsValid	获取一个值，该值指示页验证是否成功。
Master	获取确定页的整体外观的母版页。
MasterPageFile	获取或设置母版页的文件名。

1. IsPostBack属性

获取一个布尔值，为True表示当前网页是为响应客户端回发（PostBack，指网页及操作状态传回服务器）而加载，为False时表示首次加载和访问网页。

在Page_Load事件处理过程中，通过该属性可以实现首次加载和回发时执行不同的程序代码，例如：

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    If Page.IsPostBack = True Then
        Label1.Text = TextBox1.Text & ":您好，已经提交了!"
    Else
        Label1.Text = "您还没有提交!" // //如果网页为首次加载    End
    If
End Sub
```

2. IsValid属性

获取一个布尔值，指示网页上的验证控件是否验证成功。若网页验证控件全部验证成功，该值为**True**，否则为**False**。

IsValid属性在网页验证中起着重要作用。例如，以下事件过程通过**mylabel**标签输出验证结果：

```
void Button1_Click(Object Sender, EventArgs E)
{
    if (Page.IsValid) //也可写成if (Page.IsValid = true)
        mylabel.Text="信息验证成功!";
    else
        mylabel.Text="信息验证失败";
}
```

5.2.2 Page对象的方法

Page对象的常用方法及其说明如下表所示。

方法	说 明
DataBind	将数据源绑定到被调用的服务器控件及其所有子控件
FindControl	在页面中搜索指定的服务器控件
RegisterClientScriptBlock	向页面发出客户端脚本块
MapPath	检索虚拟路径（绝对的或相对的）或应用程序相关的路径映射到的物理路径。
Validate	指示页面中所有验证控件进行验证

5.2.3 Page对象的事件

Page的常用事件及其说明如下表所示，

事件	说 明
Init	当服务器控件初始化时发生,只触发一次，可用于初始化
Load	当服务器控件加载到Page对象中时发生,可以触发多次
Unload	当服务器控件从内存中卸载时发生

1. Init事件

Init事件对应的事件处理过程为Page_Init。在初始化网页时触发该事件。Init事件只触发一次。Init事件通常用来完成系统所需的初始化，如设置网页、控件属性的初始值。

2. Load事件

Load事件对应的事件处理过程为Page_Load。当在内存中加载网页时触发该事件。Load事件可以触发多次。不管是首次加载，还是按用户要求回送信息再次调用网页的回发加载，Page_Load事件处理过程都会被执行。

3. Unload事件

Unload事件对应的事件处理过程为Page_Unload。当网页从内存中卸载并将输出结果发送给浏览器时触发该事件。

Unload事件主要用来执行最后的资源清理工作，如关闭文件、关闭数据库连接和释放对象等。由于这个事件是最后事件，网页的所有内容已经传到客户端浏览器，所以不能使用它来改变控件。这个事件并不是指用户在浏览器端关闭网页，而是从IIS角度讲，网页从内存中卸载时发生这个事件。

【例5.1】 设计一个使用Page对象的IsPostBack属性的网页WForm5-1。

其设计步骤如下：

(1) 在Myaspnet网站的ch5文件夹中添加一个名称为WForm5-1的空网页。

(2) 其设计界面中包含一个文本框TextBox1、一个按钮Button1和一个标签Label1。在该网页上设计如下事件过程：

```
Protected Sub Page_Load(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Me.Load  
    If Page.IsPostBack = True Then  
        Label1.Text = TextBox1.Text & ":您好，已经提交了!"  
    Else  
        Label1.Text = "您还没有提交!"  
    End If  
End Sub  
Protected Sub Button1_Click(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Button1.Click  
    '不含任何代码  
End Sub
```



设计界面



运行界面

5.3 Response对象

Response对象用于控制服务器发送给浏览器的信息，包括直接发送信息给浏览器、重定向浏览器到另一个URL或设置cookie的值。

5.3.1 Response对象的属性

属性	说明
Buffer	获取或设置一个值，该值指示是否缓冲输出，并在完成处理整个响应之后将其发送
BufferOutput	获取或设置一个值，该值指示是否缓冲输出，并在完成处理整个页之后将其发送
Cache	获取网页的缓存策略（过期时间、保密性、变化子句）
Cookies	获取响应Cookie集合
Expires	获取或设置在浏览器上缓存的页过期之前的分钟数。如果用户在页面过期之前返回该页，则显示缓存版本。提供Expires 是为了与以前版本的 ASP 兼容
IsClientConnected	获取一个值，通过该值指示客户端是否仍连接在服务器上

5.3.2 Response对象的方法

方法	说明
Output	启用到输出HTTP响应流的文本输出
OutputStream	启用到输出HTTP内容主体的二进制输出
RedirectLocation	获取或设置HTTP“位置”标头的值
Status	设置返回到客户端的Status栏
AppendCookie	将一个HTTP Cookie添加到内部Cookie集合
AppendToLog	将自定义日志信息添加到Internet信息服务(IIS)日志文件
BinaryWrite	将一个二进制字符串写入HTTP输出流
Clear	清除缓冲区流中的所有内容输出
ClearContent	清除缓冲区流中的所有内容输出
ClearHeaders	清除缓冲区流中的所有头
Close	关闭到客户端的套接字连接
End	将当前所有缓冲的输出发送到客户端，停止该页的执行，并引发EndRequest 事件
Redirect	将客户端重定向到新的URL
Write	将信息写入HTTP响应输出流
WriteFile	将指定的文件直接写入HTTP响应输出流

1. Write方法

Write方法可以将一个字符串写入HTTP响应输出流。例如：

```
Response.Write("现在时间为: " + DateTime.Now.ToString());
```

用于输出当前的时间。

实际上Write方法将指定的字符串输出到客户端，由客户端浏览器解释后输出，所以这个输出字符串中可以包含一些HTML格外输出标记。

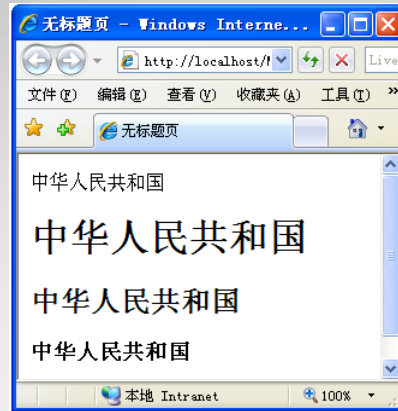
【例5.2】 设计一个使用Response对象的Write方法输出若干文字的网页窗体WForm5-2。

其设计步骤如下：

(1) 在Myaspnet网站的ch5文件夹中添加一个名称为WForm5-2的空网页。图5.5 WForm5-2网页运行界面

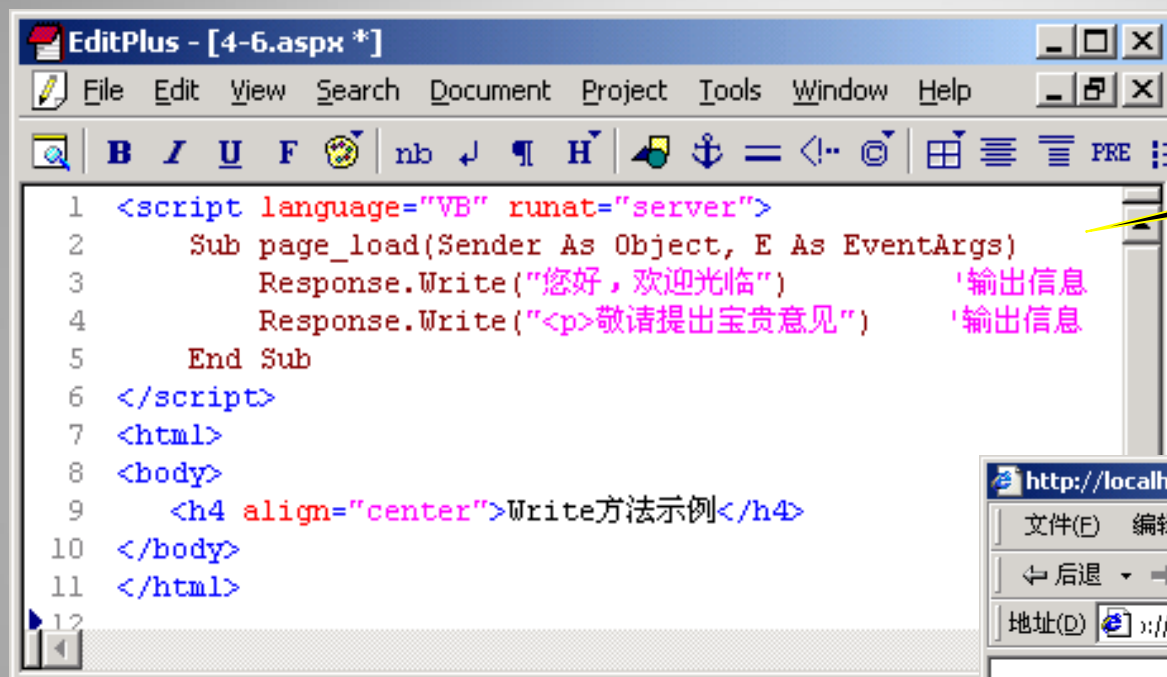
(2) 其设计界面中不包含任何内容。在该网页上设计如下事件过程：

```
Protected Sub Page_Load(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Me.Load  
    Response.Write("中华人民共和国<br>")  
    Response.Write("<h1>中华人民共和国</h1>")  
    Response.Write("<h2>中华人民共和国</h2>")  
    Response.Write("<h3>中华人民共和国</h3>")  
End Sub
```

运行界面

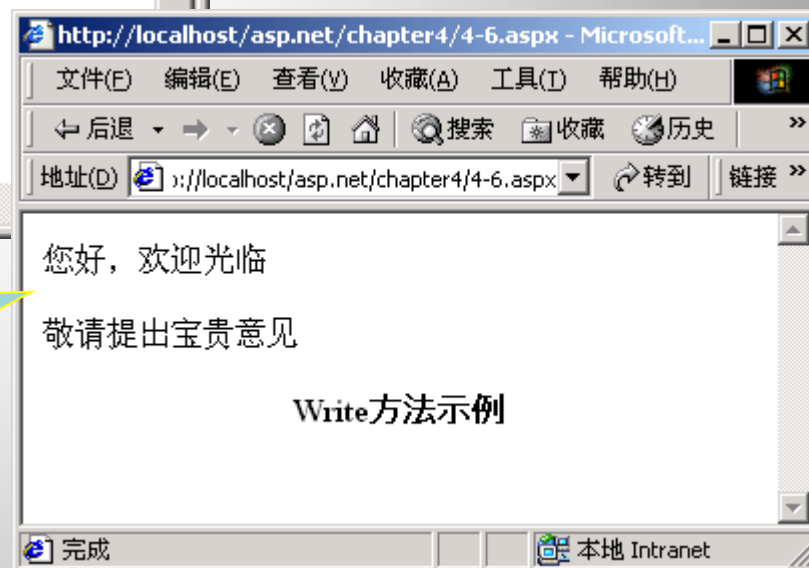
例:使用write方法输出信息



```
1 <script language="VB" runat="server">
2     Sub page_load(Sender As Object, E As EventArgs)
3         Response.Write("您好，欢迎光临") '输出信息
4         Response.Write("<p>敬请提出宝贵意见") '输出信息
5     End Sub
6 </script>
7 <html>
8 <body>
9     <h4 align="center">Write方法示例</h4>
10 </body>
11 </html>
```

4-6.aspx

运行结果：注意是先执行Page_load事件后，然后才执行Body标记中的内容。



Write方法的其它方式

- 利用该方法还可以直接在HTML代码中动态输出信息，此时需要将语句用`<%和%>`括起来，例如：

```
<h4 align="center">Write方法示例</h4>
```

```
<%
```

```
Response.Write("您好，欢迎光临")
```

```
Response.Write("<p>敬请提出宝贵意见")
```

```
%>
```

- 此时，它还有一种省略方式，在输出变量或字符串的两端加`<%和%>`，例如：

```
<h4 align="center">Write方法示例</h4>
```

```
<%= "您好，欢迎光临"%>
```

```
<%= "<p>敬请提出宝贵意见"%>
```

2. Redirect方法

使用**Redirect**方法可以实现在不同页面之间进行**跳转**的功能，也就是可以从一个网页地址转到另一个网页地址，可以是本机的网页，也可以是远程的网页地址。例如，输入以下代码：

Response.Redirect(网址变量或字符串)

例子

```
Response.Redirect("http://www.edu.cn")
```

```
Response.Redirect("other.aspx")
```

```
theURL="http://www.pku.edu.cn"
```

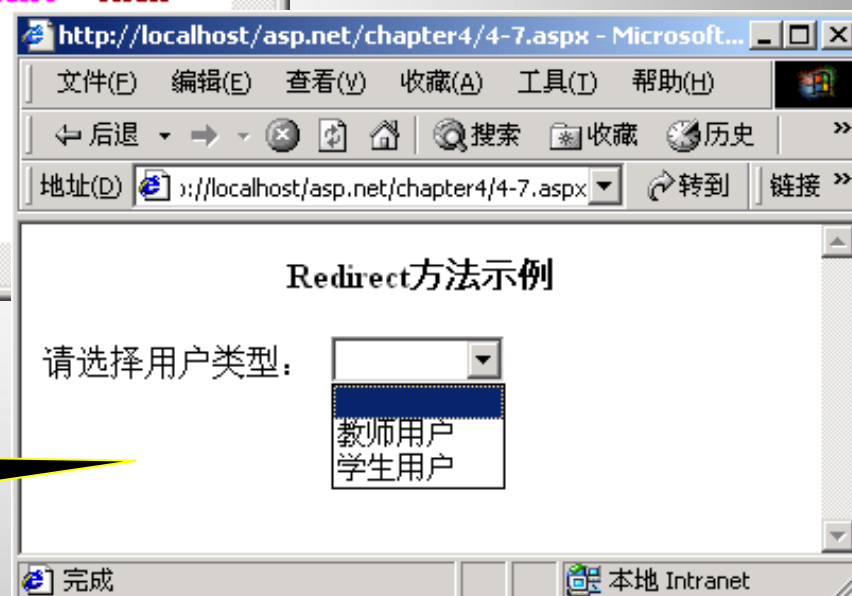
```
Response.Redirect(theURL)
```

例:重定向至另一个页面

```
EditPlus - [4-7.aspx *]
File Edit View Search Document Project Tools Window Help
B I U F nb ↵ ¶ H
1 <script language="VB" runat="server">
2 Sub usertype_change(Sender As Object, E As EventArgs)
3     If usertype.SelectedItem.value="teacher" Then
4         Response.Redirect("teacher.aspx")
5         '如是教师用户，重定向到教师页面
6     ElseIf usertype.SelectedItem.value="student" Then
7         Response.Redirect("student.aspx")
8         '如是学生用户，重定向到学生页面
9     End If
10 End Sub
11 </script>
12 <html>
```

4-7.aspx

运行结果



3. End方法

End方法用来输出当前缓冲区的内容，并中止当前页面的处理。例如：

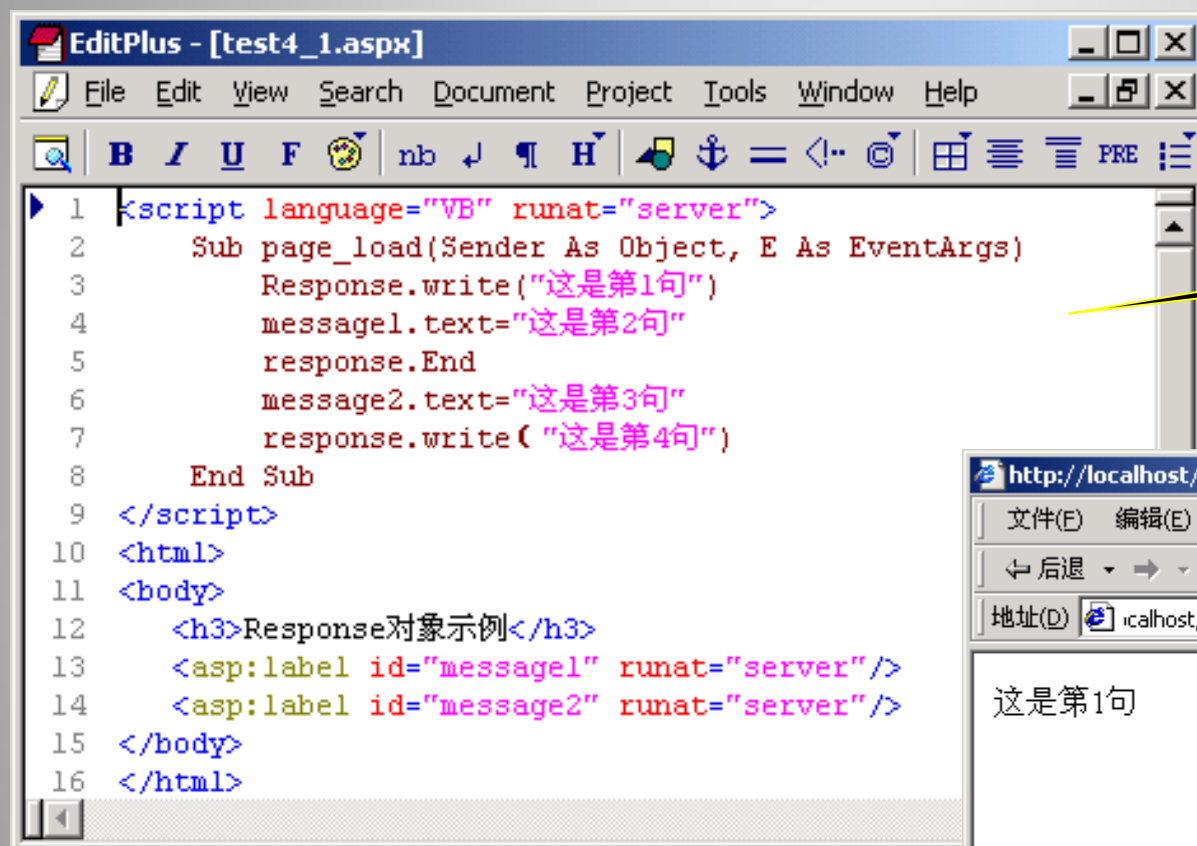
```
Response.Write("欢迎光临");
```

```
Response.End();
```

```
Response.Write("我的网站！");
```

只输出“欢迎光临”，而不会输出“我的网站！”。End方法常常用来帮助调试程序。

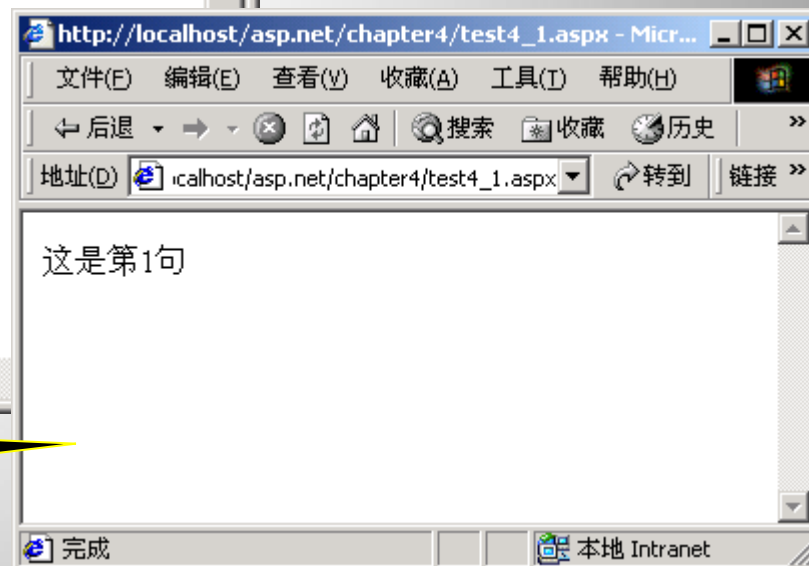
使用End方法终止程序，用于调试



```
1 <script language="VB" runat="server">
2     Sub page_load(Sender As Object, E As EventArgs)
3         Response.write("这是第1句")
4         message1.text="这是第2句"
5         response.End
6         message2.text="这是第3句"
7         response.write("这是第4句")
8     End Sub
9 </script>
10 <html>
11 <body>
12     <h3>Response对象示例</h3>
13     <asp:label id="message1" runat="server" />
14     <asp:label id="message2" runat="server" />
15 </body>
16 </html>
```

Test4_1.aspx

运行结果



5.4 Request对象

Request对象的主要功能是从客户端获取数据。

使用该对象可以访问任何HTTP请求传递的信息，包括使用POST方法或者GET方法传递的参数、cookie和用户验证。

5.4.1 Request对象的属性

属性	说明
ApplicationPath	获取ASP.NET应用的虚拟目录（URL）
PhysicalPath	获得ASP.NET应用的物理目录
Browser	获取有关正在请求客户的客户端的浏览器功能的信息
Cookies	获取在请求中发送的Cookies集。
FilePath	获取当前请求的虚拟路径
Form	获取回传到网页的窗体变量集。
Headers	获取HTTP头部
ServerVariables	获取服务器变量的名字/值集
QueryString	获取HTTP查询字符串变量集合
Url	获取有关当前请求的URL的信息
UserHostAddress	获取客户端主机的地址

5.4.2 Request对象的方法

1. MapPath方法

其使用语法格式如下：

MapPath(VirtualPath)

该方法将当前请求的URL中的虚拟路径VirtualPath映射到服务器上的物理路径。参数VirtualPath用于指定当前请求的虚拟路径（可以是绝对路径，也可以是相对路径）。返回值为与VirtualPath对应的服务器端物理路径。

例如，语句：

Response.Write(Request.MapPath("aa"));

在浏览器中输出aa所在的物理路径。

2. SaveAs方法

其使用语法格式如下：

SaveAs(filename, includeHeaders)

该方法将客户端的HTTP请求保存到磁盘。参数filename用于指定文件在服务器上保存的位置；布尔型参数includeHeaders用于指示是否同时保存HTTP头。

例如：

Request.SaveAs("H:\aaa", True);

则执行后在H盘根目录产生aaa文件。

5.4.3 Request对象的应用

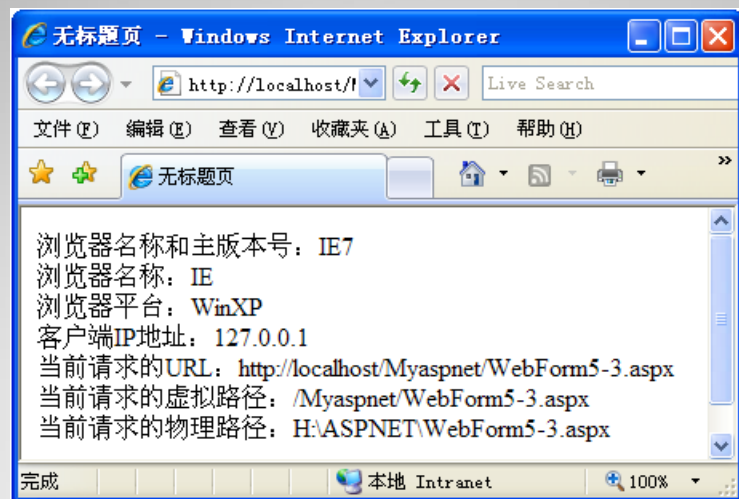
1. 获取客户端机器和浏览器的相关信息

【例5.3】 设计一个获取客户端机器和浏览器的信息的网页WForm5-3。

(1) 在Myaspnet网站的ch5文件夹中添加一个名称为WForm5-3的空网页。

(2) 其设计界面中不包含任何内容。在该网页上设计如下事件过程：

```
Protected Sub Page_Load(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Me.Load  
    Response.Write("浏览器名称和主版本号: " & Request.Browser.Type &  
"<br>")  
    Response.Write("浏览器名称: " & Request.Browser.Browser & "<br>")  
    Response.Write("浏览器平台: " & Request.Browser.Platform & "<br>")  
    Response.Write("客户端IP地址: " & Request.UserHostAddress &  
"<br>")  
    Response.Write("当前请求的URL: " & Request.Url.ToString() & "<br>")  
    Response.Write("当前请求的虚拟路径: " & Request.Path & "<br>")  
    Response.Write("当前请求的物理路径: " & Request.PhysicalPath &  
"<br>")  
End Sub
```



运行界面

2. 使用QueryString属性在网页之间传递数据

在上网的过程中，经常发现网址后面跟一串字符，这就是通过URL后面的字符串在两个网页之间传递参数，QueryString属性保存这些参数和值，因此可以通过Request的QueryString在两个网页之间传递信息。

其设计步骤如下:

(2) 在其中添加一个文字串和一个超链接。



(3) 在Myaspnet网站中添加一个名称为WForm5-4-1的空网页。其中包含一个文字串和一个标签Label1，在该网页上设计如下事件过程：

```
Protected Sub Page_Load(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Me.Load  
    Dim uname As String, uage As String  
    uname = Request.QueryString("uname")  
    uage = Request.QueryString("uage")  
    Label1.Text = uname & ":您好! 您的年龄为" & uage & "岁"  
End Sub
```

IE: <http://localhost/Myaspnet/ch5/WForm5-4-1.aspx?uname=王华&uage=20>

提示：后面可有多个变量参数，参数之间用&连接。



3. 使用Form属性在网页之间传递数据

使用Request的Form属性可以获取客户端通过**POST**方式传递的表单数据，从而实现网页之间的数据传递。

【例5.5】 设计两个网页WForm5-5和WForm5-5-1，说明Form属性的使用方法。

其设计步骤如下：

（1）在Myaspnet网站的ch5文件夹中添加一个名称为WForm5-5的空网页（可以是静态网页）。

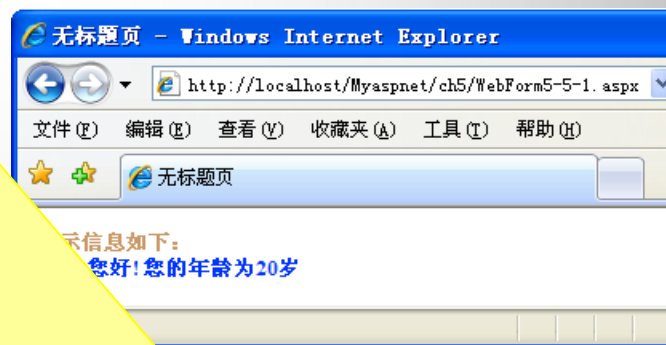
(2) 其中包含一个文字串、一个提交按钮和两个隐蔽文本框，对应的视图代码如下：

```
<body>
  <form id="form1" method="post" action="WForm5-5-1.aspx">
    <div>
      <strong><span style="font-size: 10pt; color: #cc33ff">
        请单击提交按钮<br />
      <br /></span></strong>
      <input id="Submit1" type="submit" value="提交" /><br />
      <input id="Hidden1" name="uname" value="王华"
        style="width: 40px" type="hidden" />
      <input id="Hidden2" name="uage" value="20" style="width: 36px;
        height: 17px" type="hidden" />
    </div>
  </form>
</body>
```

通过Submit按钮提交表单，采用POST方法，使用action属性直接转向网页
method = "post" action="WForm5-5-1.aspx"

(3) 在Myaspnet网站的ch5文件夹中添加一个名称为WForm5-5-1的空网页。其中包含一个文字串和一个标签Label1，在该网页上设计如下事件过程：

```
Protected Sub Page_Load(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Me.Load  
    Dim uname As String, uage As String  
    uname = Request.Form("uname")  
    uage = Request.Form("uage")  
    Label1.Text = uname & ":您好! 您的年龄为" & uage & "岁"  
End Sub
```



使用FORM属性获取客户端通过POST方式传递的表单数据，从而实现静态网页和动态网页之间传递数据

5.5 *Server*对象

Server对象提供了对服务器的方法和属性的访问，可以获取服务器的信息，对HTML文本进行编码和解码等，如文件的物理路径等。

5.5.1 Server对象的属性

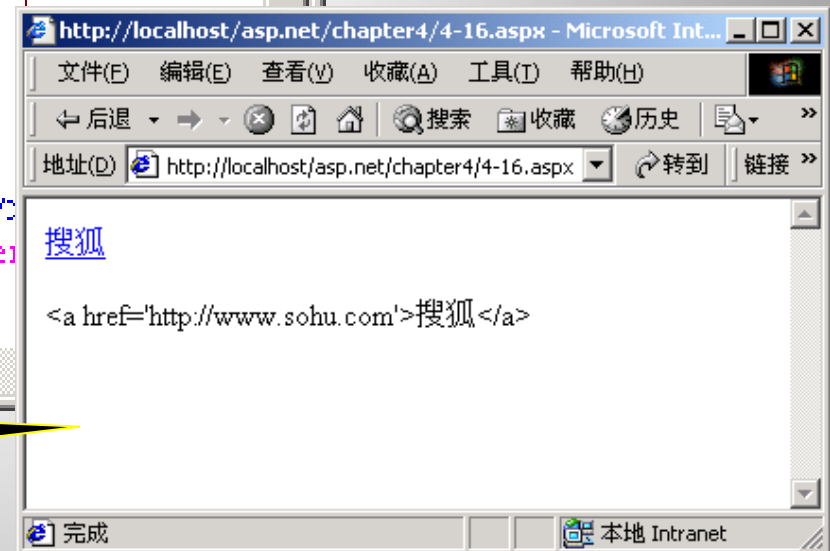
属性	说明
MachineName	作用是获取服务器的名称
ScriptTimeOut	<p>获取和设置请求超时值（以秒计）。该属性用来规定脚本文件执行的最长时间，默认为90秒。主要是用来防止某些可能进入死循环的错误导致服务器过载问题。</p> <p>语法： Server.ScriptTimeout=300</p>

例:显示HTML语句

```
1 <script language="VB" runat="server">
2     Sub Page_Load(Sender As Object, E As EventArgs)
3         message1.text="<a
4             href='http://www.sohu.com'>搜狐</a>"
5         message2.text=Server.HtmlEncode ("<a
6             href='http://www.sohu.com'>搜狐</a>")
7     End Sub
8 </script>
9 <html>
10 <body>
11     <asp:label id="message1" runat="server"/>
12     <p><asp:label id="message2" runat="server"/>
13 </body>
14 </html>
```

4-16.aspx

运行结果

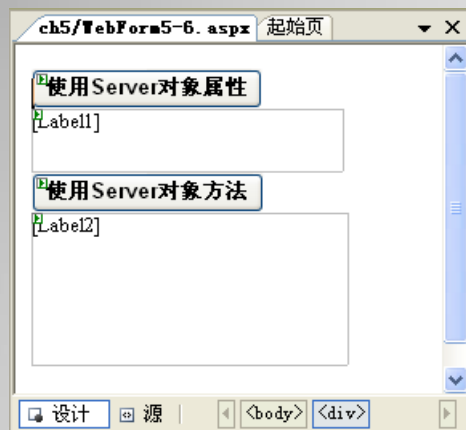


5.5.2 Server对象的方法

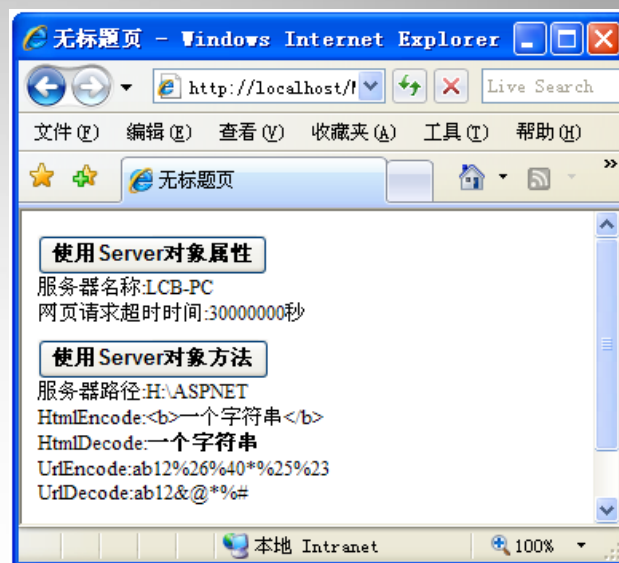
方法	说明
CreateObject	创建COM对象的一个服务器实例。
Execute 方法	运行另一个网页以执行当前请求
HtmlEncode 方法	对要在浏览器中显示的字符串进行编码，转换成字符实体
HtmlDecode 方法	对已被编码以消除无效HTML字符的字符串进行解码
UrlEncode 方法	对指定字符串以URL格式进行编码
UrlPathEncode	对URL字符串的路径部分进行URL编码，并返回已编码的字符串。
MapPath	返回与Web服务器上的指定虚拟路径相对应的物理文件路径。
Transfer	终止当前网页的执行，并开始执行新的请求网页

【例5.6】 设计一个获取服务器端相关信息的网页WForm5-6。

```
Protected Sub Button1_Click(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Button1.Click  
    Label1.Text = "服务器名称:" & Server.MachineName & "<br>" & _  
        "网页请求超时时间:" & Server.ScriptTimeout.ToString() & "秒"  
End Sub  
  
Protected Sub Button2_Click(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Button2.Click  
    Dim mystr1 As String = "<b>一个字符串</b>"  
    Dim mystr2 As String = "ab12&@*%#"  
    Label2.Text = "服务器路径:" & Server.MapPath(".") & "<br>" & _  
        "HtmlEncode:" & Server.HtmlEncode(mystr1) & "<br>" & _  
        "HtmlDecode:" & Server.HtmlDecode(mystr1) & "<br>" & _  
        "UrlEncode:" & Server.UrlEncode(mystr2) & "<br>" & _  
        "UrlDecode:" & Server.UrlDecode(mystr2)  
End Sub
```

设计界面



运行界面

1. MapPath方法

使用MapPath方法可以获得服务器文件的物理路径。其使用语法格式如下：

```
Server.MapPath(虚拟路径字符串);
```

2. Transfer方法

用户可能希望将用户从一个ASP.NET网页重定向到另一个网页。重定向页的方法很多，使用Server.Transfer方法就是其中的一种方法，其语法格式如下：

```
Server.Transfer(URL);
```

Transfer方法执行完新的网页后，不再返回原网页执行。

3. Execute方法

有时用户希望在网页运行时执行其他网页的内容后继续执行当前网页的内容，可以使用Server.Execute方法。其语法格式如下：

Server.Execute(URL);

Execute方法执行完新的网页后再返回原网页执行。

例:重定向到新的文件

The image illustrates a file redirection example in ASP.NET. It consists of three main components:

- 4-18.aspx (Left):** The source file in EditPlus. It contains the following code:

```
1 <html>
2 <body>
3     欢迎光临我的主页
4     <%
5     Server.Execute("4-19.aspx")
6     %>
7     <p>谢谢，再见
8 </body>
9 </html>
```
- 4-19.aspx (Bottom):** The target file in EditPlus. It contains the following code:

```
1 <html>
2 <body>
3     <p>敬请提出宝贵意见
4 </body>
5 </html>
6
```
- 运行结果 (Right):** A screenshot of a web browser displaying the output of 4-18.aspx. The browser address bar shows `http://localhost/asp.net/chapter4/4-18.aspx`. The page content is:

欢迎光临我的主页
敬请提出宝贵意见
谢谢，再见

Yellow callout boxes with red text identify the files: **4-18.aspx** points to the first code editor, **4-19.aspx** points to the second code editor, and **运行结果** (Execution Result) points to the browser window.

5.6 *Application*对象

Application对象是运行在Web应用服务器上的虚拟目录及其子目录下所有文件、页面、模块和可执行代码的总和。

一旦网站服务器被打开，就创建了**Application**对象；所有的用户共用一个**Application**对象并可以对其进行修改；**Application**对象的这一特性使得网站设计者可以方便地创建诸如聊天室和网站计数器等常用Web应用程序。

5.6.1 Application对象的属性

属性	说明
Count	返回Application集合中的对象个数
Contents	表示Application对象中对象集合，主要是为了与以前版本的ASP兼容

5.6.2 Application对象的方法

方法	说明
Add	向Application集合中添加新对象
Clear	从Application集合中移除所有对象
Remove	从Application集合中移除指定名称的对象
RemoveAt	从Application集合中移除指定索引的对象
RemoveAll	从Application集合中移除所有对象
Lock	禁止其他用户修改Application集合中的对象
Unlock	允许其他用户修改Application集合中的对象

1. Add方法

用于将新对象添加到Application集合中。其语法格式如下：

Application.Add(对象值,字符串)

Application(“Application名字”)=表达式

其中，“字符串”指定对象名。例如：

```
string str1="mystr"
```

```
int int1=34
```

```
Application.Lock
```

```
Application.Add("var1",str1)
```

```
Application.Add("var2",int1)
```

```
Application.Unlock
```

这样Application集合中新增了var1和var2两个对象，它们的值分别是“mystr”和34。

2. Remove和RemoveAt方法

它们都用删除Application集合中的指定对象，其使用语法格式如下：

```
Application.Remove(对象名);
```

```
Application.RemoveAt(对象索引);
```

例如：

```
Application.Remove("var1")
```

```
//删除var1对象
```

```
Application.RemoveAt(1);
```

```
//删除var2对象
```

5.6.3 Application对象的事件

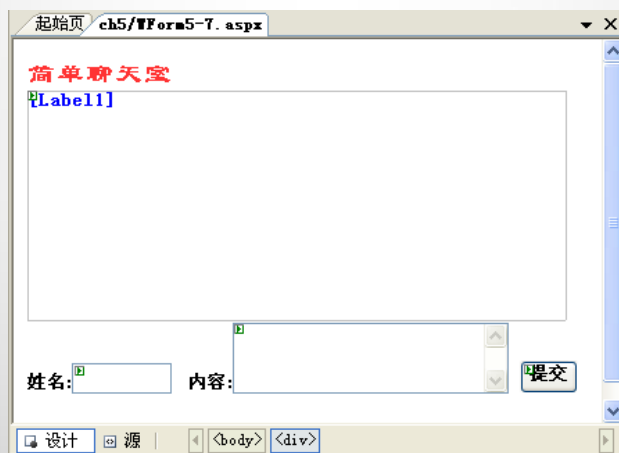
事件	说明
Start	在整个ASP.NET应用程序第一次执行时引发
End	在整个ASP.NET应用程序结束时引发

【例5.7】 设计一个实现简单聊天功能的网页WForm5-7。

其设计步骤如下：

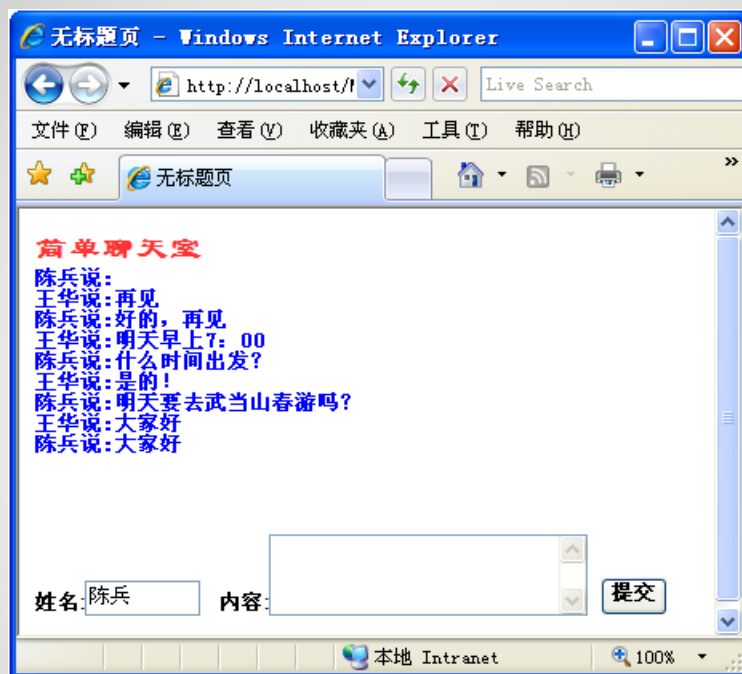
（1）在Myaspnet网站的ch5文件夹中添加一个名称为WForm5-7的空网页。

（2）其设计界面如下图所示，其中包含一个标签（Label1，用于显示聊天内容）、两个文本框（TextBox1和TextBox2，分别用于输入姓名和聊天记录，TextBox2的TextMode属性设为MultiLine）和一个命令按钮（Button1，用于提交聊天记录）。在该网页上设计如下事件过程：



设计界面

单击工具栏中的▶按钮运行本网页，输入姓名开始聊天。再次启动IE浏览器，输入地址http://localhost/Myaspnet/WForm5-7.aspx启动本网页，这样两个人就可以相互聊天了，如下图所示。



运行界面

例2:利用框架制作的聊天室

共分3个文件：4-13.aspx(框架页文件)、4-14.aspx(保存发言信息)、4-15.aspx (读取发言信息)。

—<frameset> 元素可定义一个框架集。它被用来组织多个窗口（框架）。

—<frame> 标签定义 frameset 中的一个特定的窗口（框架）。



```
1 <html>
2 <head>
3     <title> Application对象示例 </title>
4 </head>
5 <frameset rows="*, 60">
6     <frame name="message" src="4-15.aspx">
7     <frame name="say" src="4-14.aspx">
8 </frameset>
9 </html>
```

框架

- 框架是把浏览器的窗口分割成一个个小的子窗口，这些子窗口称为框架，每一个框架都相当于一个浏览器窗口，这样就使一个浏览器窗口可以显示多个网页。在VS中，**frame**框架只能在视图中用直接输入代码的方式创建。
- **frameset**是确定网页分框的定义
 - **name**属性设置**frame**的名字
 - **Border=0**则不显示边界
 - **scrolling**属性决定**frame**是否可以使用滚动条
 - **noresize**属性决定**frame**是否可以改变大小，默认为true

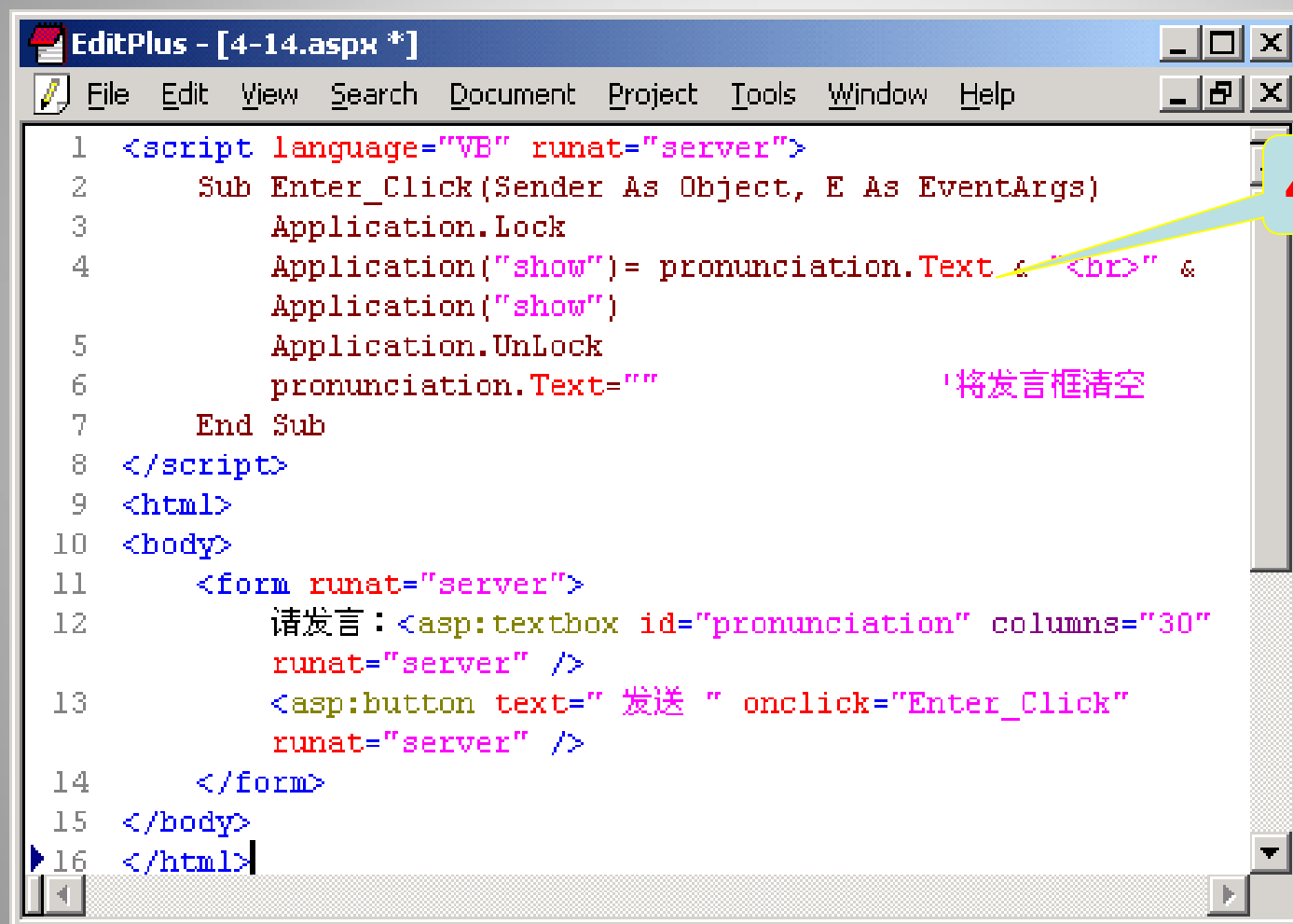
```
<frameset rowss="60%," border=0>
```

```
    <frame name="message" src="4-15.aspx" scrolling=yes >
```

```
    <frame name="say" src="4-14.aspx" noresize=true>
```

```
</frameset>
```

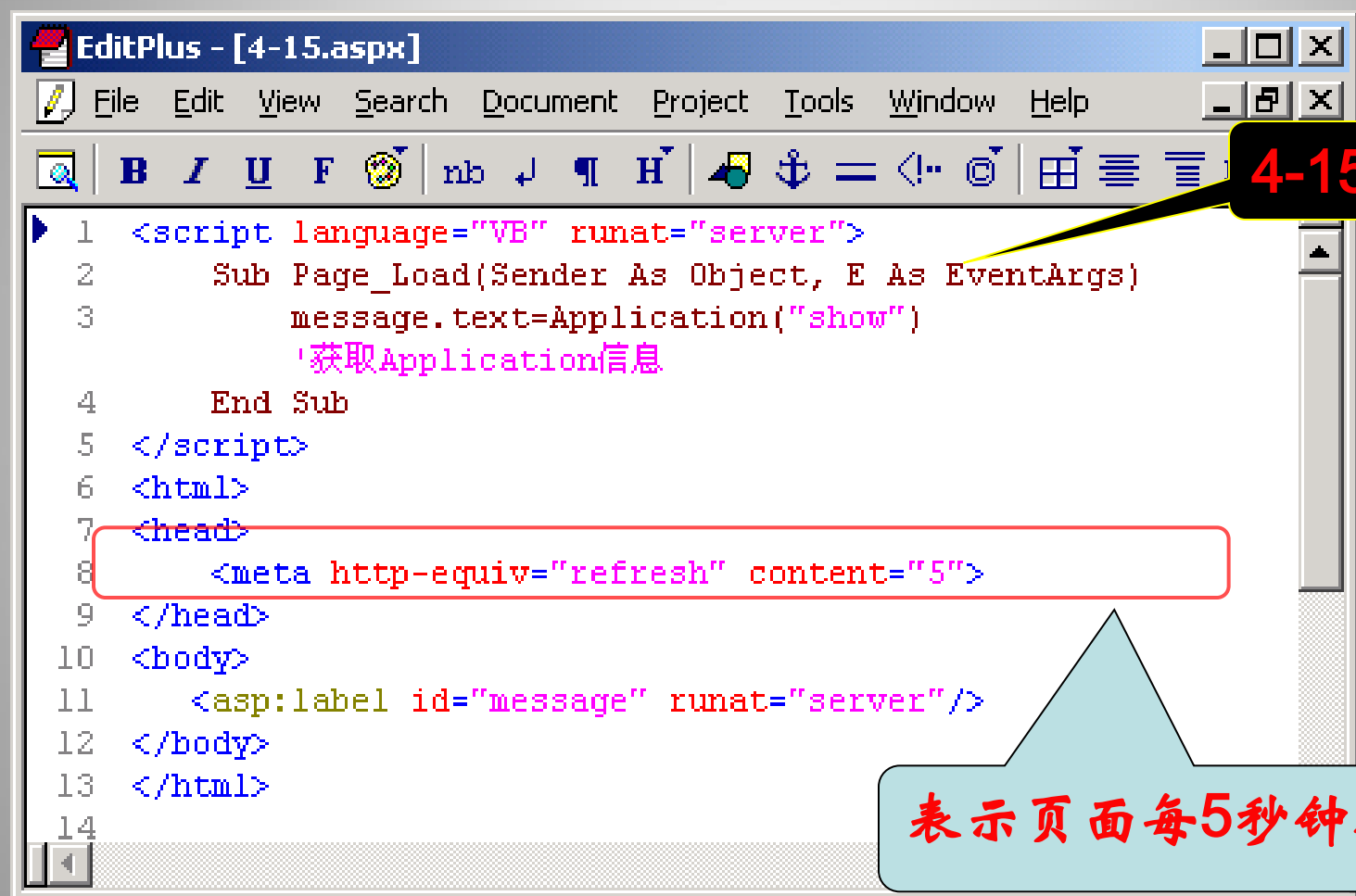
4-14.aspx:保存发言信息



```
1 <script language="VB" runat="server">
2     Sub Enter_Click(Sender As Object, E As EventArgs)
3         Application.Lock
4         Application("show")= pronunciation.Text & "<br>" &
5         Application("show")
6         Application.Unlock
7         pronunciation.Text="" '将发言框清空
8     End Sub
9 </script>
10 <html>
11 <body>
12     <form runat="server">
13         请发言:<asp:textbox id="pronunciation" columns="30"
14         runat="server" />
15         <asp:button text=" 发送 " onclick="Enter_Click"
16         runat="server" />
17     </form>
18 </body>
19 </html>
```

4-14.aspx

4-15.aspx:读取发言信息



The screenshot shows the EditPlus editor window titled "EditPlus - [4-15.aspx]". The menu bar includes File, Edit, View, Search, Document, Project, Tools, Window, and Help. The toolbar contains various icons for editing and formatting. The code is as follows:

```
1 <script language="VB" runat="server">
2     Sub Page_Load(Sender As Object, E As EventArgs)
3         message.text=Application("show")
4         '获取Application信息
5     End Sub
6 </script>
7 <html>
8 <head>
9     <meta http-equiv="refresh" content="5">
10 </head>
11 <body>
12     <asp:label id="message" runat="server"/>
13 </body>
14 </html>
```

A yellow callout bubble points to the `runat="server"` attribute in the `Page_Load` event handler, with the text "4-15.aspx".

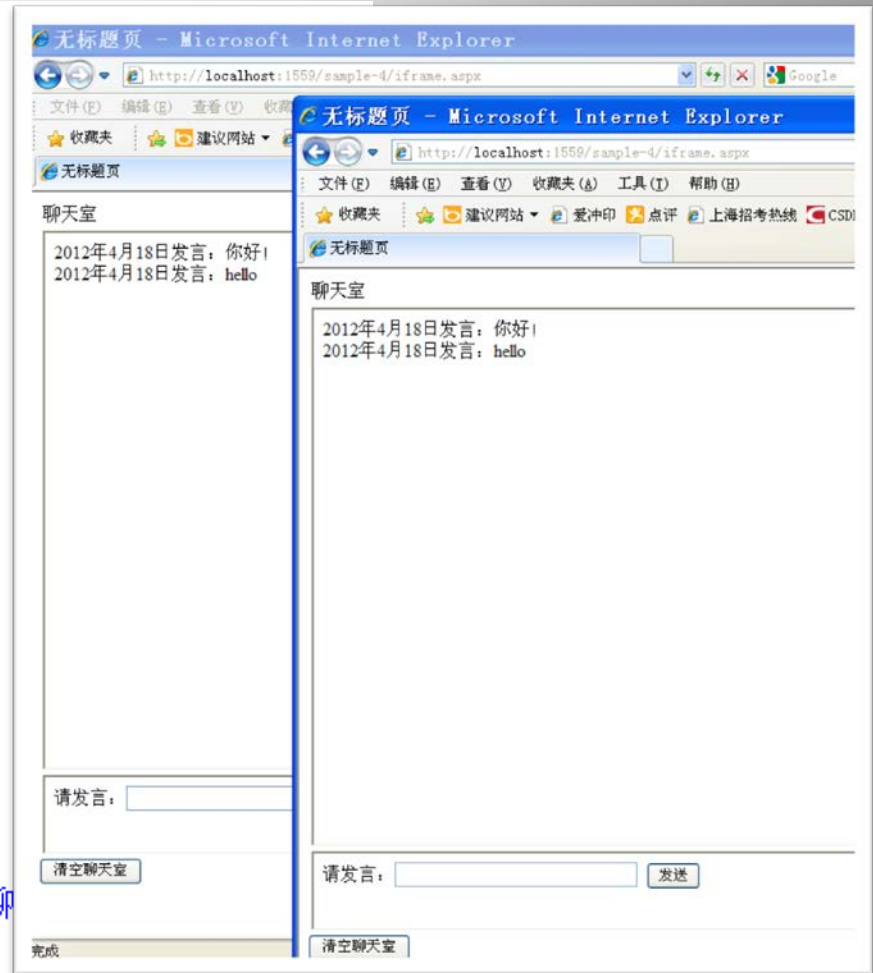
A red rectangle highlights the `<meta http-equiv="refresh" content="5">` tag in the `<head>` section.

A light blue callout bubble points to the `<meta http-equiv="refresh" content="5">` tag, with the text "表示页面每5秒钟刷新一次".

iframe框架 (iframe.aspx) ?

iframe.html

```
<body>
  <form id="form1" runat="server">
    <div>
      <table style="width: 90%; height: 90%;">
        <tr>
          <td>
            聊天室</td>
          </tr>
          <tr>
            <td style="height: 148px" >
              <iframe id="Iframe1" src="test4_15.aspx"
                style="width: 700px; height: 478px">
              </iframe>
            </td>
          </tr>
          <tr>
            <td style="height: 44px" >
              <iframe id="Iframe2" src="test4_14.aspx"
                style="width: 710px; height: 71px">
              </iframe>
            </td>
          </tr>
        </table>
      </div>
      <asp:Button ID="Button1" runat="server" Text="清空聊
    </form>
  </body>
```



利用Application存储数组信息

- 存储

```
Dim usern(2) As String  
usern(0) = "白芸"  
usern(1) = "海霞"  
Application.Lock  
Application("arry_user") = usern  
Application.Unlock
```

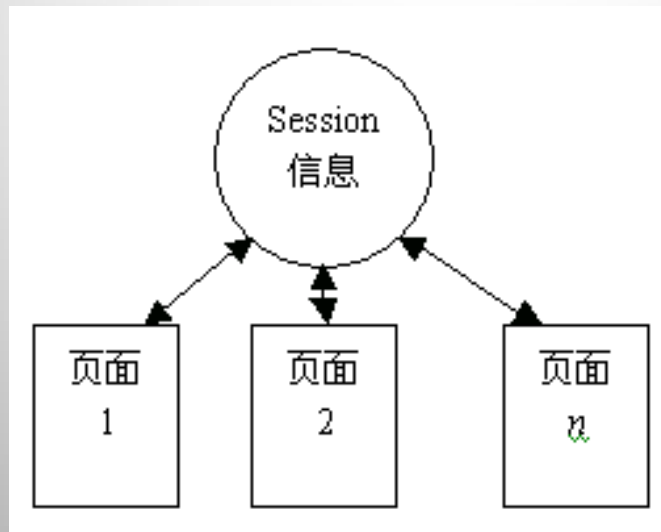
- 读取

```
Dim usern () As String  
usern = Application("arry_user")  
message.text = usern(0) & "您好<p>" & usern(1) & "您好"
```

5.7 Session对象

当用户请求一个ASP.NET页面时，系统将自动创建一个Session（会话），退出应用程序或关闭服务器时该会话撤销。默认生命周期为**20分钟**。

系统在创建会话时将其分配一个长长的字符串（SessionID）标识，以实现对话进行管理跟踪。该字符串中只包含URL中所允许的ASCII字符。SessionID具有的随机性和唯一性保证了会话不会冲突，也不会被怀有恶意的人利用新SessionID推算出现有会话的SessionID。



5.7.1 Session对象的属性

属性	说明
SessionID	用来标识一个Session对象
TimeOut	获取并设置会话状态提供程序终止会话之前各请求之间所允许的超时期限（以分钟为单位）

5.7.2 Session对象的方法

方法	说明
Add	将新的项添加到Session集合中
Clear	从Session集合中清除所有对象，但不结束会话
Abandon	强行结束用户会话，并清除会话中所有信息
CopyTo	将Session集合复制到一维数组中

1. Add方法

用于将新对象添加到Session集合中。其语法格式如下：

- **Session.Add(对象值,字符串)**
- **Session(“Session名字”)=变量/常量/字符串/表达式**

其中，“字符串”指定对象名。例如：

```
string str1="mystr"
```

```
int int1=34
```

```
Session.Add(“var1”,str1) 或Session(“var1”) = str1
```

```
Session.Add("var2",int1) 或Session(“var2”) = int1
```

2. Clear方法

用于清除Session集合中所有对象，其使用语法格式如下：

```
Session.Clear();
```

5.7.3 Session对象的事件

事件	说明
Start	建立Session对象时发生
End	结束Session对象时发生

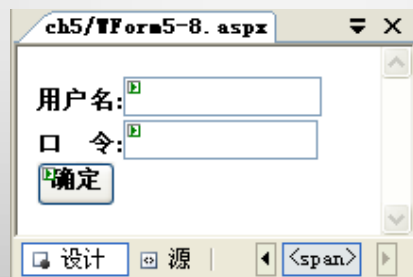
【例5.8】 设计一个采用Session对象在网页之间传递数据的网页WForm5-8和WForm5-8-1.aspx。

其设计步骤如下：

(1) 在Myaspnet网站的ch5文件夹中添加一个名称为WForm5-8的空网页。

(2) 其设计界面如下图所示，其中包含两个文本框（TextBox1和TextBox2，分别用于输入用户名和口令，TextBox2的TextMode属性设置为Password）和一个命令按钮（Button1）。在该网页上设计如下事件过程：

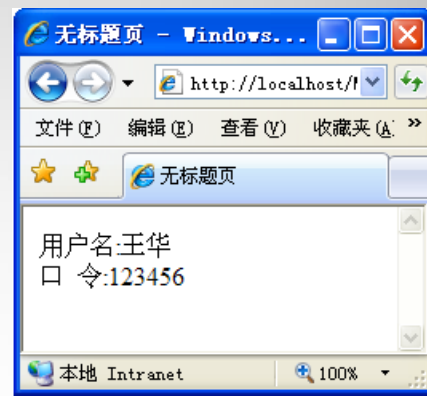
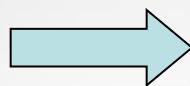
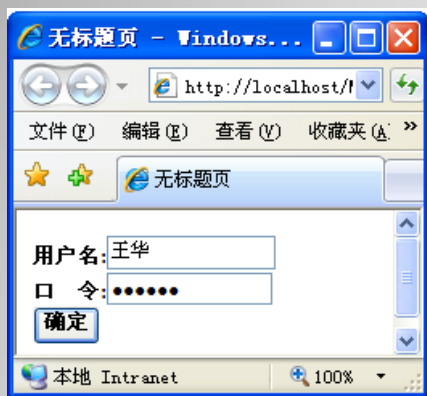
```
Protected Sub Button1_Click(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Button1.Click  
    Session("uname") = TextBox1.Text  
    Session("upass") = TextBox2.Text  
    Server.Transfer("WForm5-8-1.aspx")  
End Sub
```



设计界面

(3) 再添加一个名称为WForm5-8-1.aspx的空网页，不放置任何控件，在该网页上设计如下事件过程：

```
Protected Sub Page_Load(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Me.Load  
    Dim mystr As String  
    mystr = "用户名:" & Session("uname").ToString() & _  
        "<br>口 令:" & Session("upass").ToString()  
    Response.Write(mystr)  
End Sub
```



运行界面

5.8 Cookie对象

Response和Request对象都有一个Cookies属性，它是存放Cookie对象的集合。保存在客户的硬盘上

- Win XP中， c:\Documents and Settings\Administrator\Cookies）。
- Windows 7中， c:\Users\<用户名
>\AppData\Roaming\Microsoft\Windows\Cookies

Cookie是文本文件，能随着用户请求和页面在Web服务器和浏览器之间传递。用户每次访问站点时，Web应用程序都可以读取Cookie包含的信息，从而知道用户上次登陆的时间等具体信息。

5.8.1 Cookie对象的属性

属性	说明
Name	获取或设置Cookie的名称
Expires	获取或设置Cookie的过期日期和时间
Domain	获取或设置Cookie关联的域
Path	获取或设置要与Cookie一起传输的虚拟路径
Secure	获取或设置一个值，通过该值指示是否安全传输Cookie
Value	获取或设置单个Cookie值
Values	获取在单个Cookie对象中包含的键值对的集合

5.8.2 Cookie对象的方法

方法	说明
Equals	判断指定的Cookie对象是否等于当前的Cookie对象
ToString	返回此Cookie对象的一个字符串表示形式

5.8.3 Cookie对象的应用

1. 创建Cookie对象

Cookie对象是由HttpCookie类来实现的，创建一个Cookie对象就是建立HttpCookie类的一个实例。HttpCookie类具有以下构造函数：

```
public HttpCookie (string name)
```

```
public HttpCookie(string name,string value)
```

其中，name表示Cookie对象的名称（对应Name属性），value表示Cookie对象的值（对应Value属性）。例如：

```
HttpCookie cookie1 = new HttpCookie("mycookie1");
```

```
    //新建名称为mycookie1的Cookie对象
```

```
cookie1.Value="mystring";           //其值为设为"mystring"
```

```
Response.Cookies.Add(cookie1);      //添加cookie1对象
```

```
HttpCookie cookie2 = new HttpCookie("mycookie2","good");
```

```
    //新建名称为mycookie2的Cookie对象,其值为"good"
```

```
Response.Cookies.Add(cookie2); //添加cookie2对象
```

2. 设置多值Cookie

一个Cookie对象可以有多个值，通过子键区分。

例如，当一个名称为mycookie的Cookie对象已添加到Response对象中后，可以通过以下语句设置两个子键的值：

```
Response.Cookies("mycookie") ("uname")="Smith"
```

```
Response.Cookies("mycookie") ("uage")=23.ToString()
```

或者在创建Cookie对象同时设置多个值：

```
HttpCookie cookie=new HttpCookie("mycookie")
```

```
cookie.Values("uname")="Smith"
```

```
cookie.Values( "uage")=23.ToString()
```

```
Response.Cookies.Add(cookie);
```


3. 读取Cookie对象

对于单值Cookie对象，直接用Request.Cookies(Cookie的Name属性值)来读取其Cookie值。

对于多值Cookie对象，还需加上子键名称，例如，以下语句将Name为mycookie1的Cookie对象的两个子键值分别在两个文本框中输出：

```
TextBox1.Text = Request.Cookies("mycookie1")("uname")
```

```
TextBox2.Text = Request.Cookies("mycookie1")("uage")
```

4. Cookie的有效期

Cookie的**Expires**属性为DateTime类型的，用来指定Cookie的过期日期和时间即Cookie的有效期。浏览器在适当的时候删除已经过期的Cookie。如果不给Cookie指定过期日期和时间，则为会话Cookie，不会存入用户的硬盘，在浏览器关闭后就被删除。

应根据应用程序的需要来设置Cookie的有效期，如果用来保存用户的首选项，则可以把其设置为永远有效（例如100年），如果用来统计用户访问次数，则可以把有效期设置为半年。即使设置长期有效，用户也可以自行决定将其全部删除。

```
Response.Cookies("user_name").Expires=#1/1/2012#
```

```
Response.Cookies(" user_name").Expires=DateTime.Now.AddYears(1)
```

'设置有效期为一年后

5. 修改和删除Cookie

修改某个Cookie实际上是指用新的值创建新的Cookie，并把该Cookie发送到浏览器，覆盖客户机上旧的Cookie。

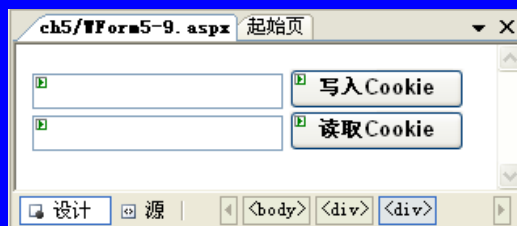
删除Cookie是修改Cookie的一种形式。由于Cookie位于用户的计算机中，所以无法直接将其删除。但是，可以修改Cookie将其有效期设置为过去的某个日期，从而让浏览器删除这个已过期的Cookie。

【例5.9】 设计一个说明Cookie对象使用方法的网页WForm5-9。

其设计步骤如下：

（1）在Myaspnet网站的ch5文件夹中添加一个名称为WForm5-9的空网页。

（2）其设计界面如下图所示，其中包含两个文本框（TextBox1和TextBox2）和两个命令按钮（Button1和Button2）。在该网页上设计如下事件过程：



设计界面



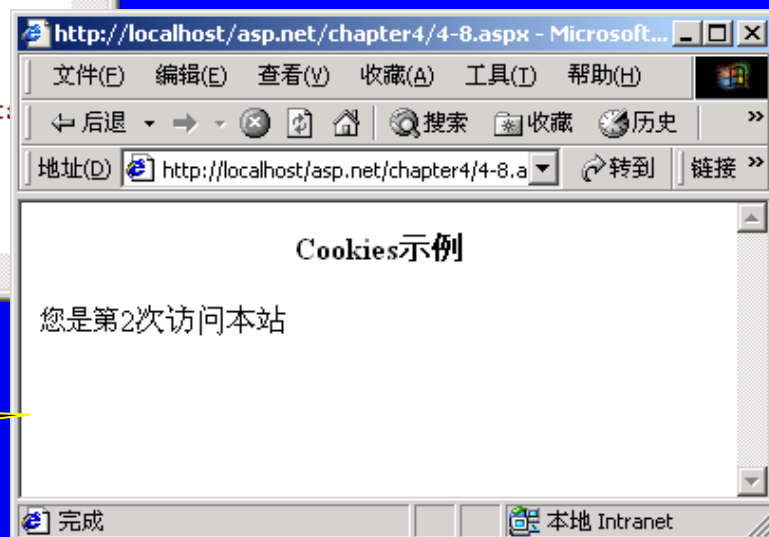
运行界面

Cookie的综合示例—显示第几次光临网站

```
EditPlus - [4-8.aspx *]
File Edit View Search Document Project Tools Window Help
B I U F nb ↓ H
2 Sub Page_Load(Sender As Object, E As EventArgs)
3   Dim vNumber As Integer '定义一个访问次数变量
4   If Request.Cookies("vNumber") Is Nothing Then
5     '如果是第1次访问，就令访问次数为1
6     vNumber=1
7   Else
8     '如果不是第1次访问，就令访问次数在原有次数上加1
9     vNumber=Request.Cookies("vNumber").value + 1
10  End If
11  '将新的访问次数保存到Cookies中
12  Response.Cookies("vNumber").value=vNumber
13  '设置有效期为1年后
14  Response.Cookies("vNumber").Expires=DateTime.Now.AddYear(1)
15  message.text="您是第" & vNumber & "次访问本站"
16 End Sub
17 </script>
18 <html>
```

4-8.aspx

运行结果



5.9 配置 *Global.asax* 文件

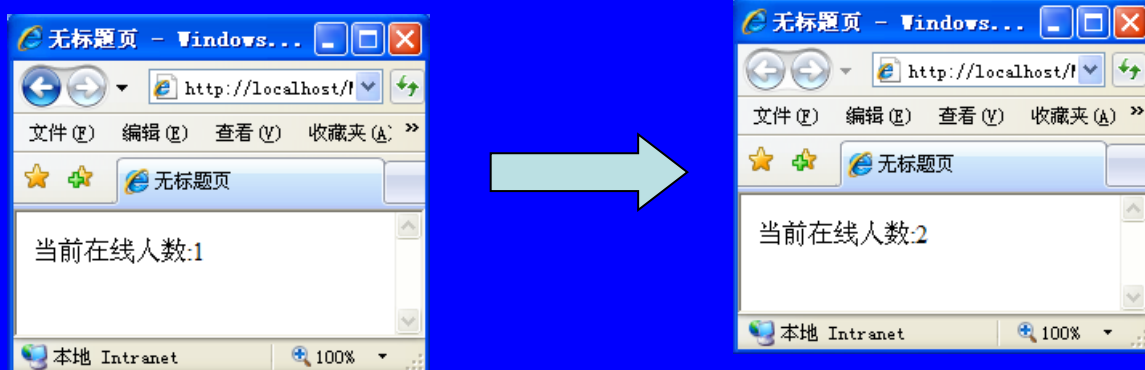
每个ASP.NET网页面中都会存在许多的事件，如Page_Load等，可以在网页中进行编程来处理这些事件。作为一个ASP.NET应用程序也存在这样的事件，如应用程序开始时要执行什么操作，一个新Session被创建的时候要进行什么操作等等。那么对这些事件的处理要写在什么地方呢？通常情况下这些事件处理过程应放在Global.asax和Web.config这两个文件中。有关Web.config文件的内容在以后介绍，这里只讨论Global.asax文件。

在ASP.NET中都不会自动创建Global.asax文件，如果要创建该文件，选择“网站”|“添加新项”菜单命令，在打开的“添加新项”对话框中选择“全局应用程序类”选项，单击“添加”按钮即可创建一个Global.asax文件。

该文件位于ASP.NET应用程序的根目录下，其作用就是用来处理与应用程序相关的一些事件。

(3) 在ch5文件夹添加一个名称为WForm5-10的空网页。在该网页上设计如下事件过程:

```
Protected Sub Page_Load(ByVal sender As Object,  
    ByVal e As System.EventArgs) Handles Me.Load  
    Response.Write("当前在线人数:" + Application("counter").ToString())  
End Sub
```



运行界面