



# ОСНОВЫ ASP.NET 3

# Элементы управления для проверки

RequiredFieldValidator

Запрещает пустые поля

CompareValidator

Сравнивает два поля

RangeValidator

Проверяет на заданный диапазон

RegularExpressionValidator

Сравнивает значение с регулярным выражением

CustomValidator

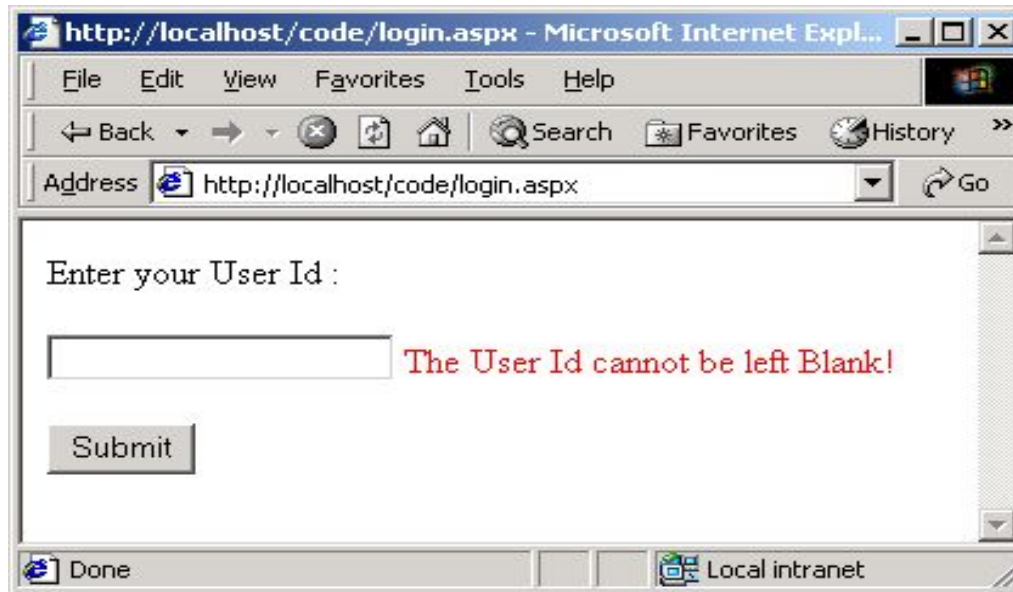
Использует для проверки функции клиентской или серверной стороны

ValidationSummary

Перечисляет ошибки, найденные при проверке всех элементов на странице

# RequiredFieldValidator

Не введено  
никакого  
значения



The screenshot shows a Microsoft Internet Explorer window with the address bar displaying 'http://localhost/code/login.aspx'. The page content includes the text 'Enter your User Id :', an empty text input field, and a 'Submit' button. A red error message, 'The User Id cannot be left Blank!', is displayed next to the input field. The browser's status bar at the bottom shows 'Done' and 'Local intranet'.

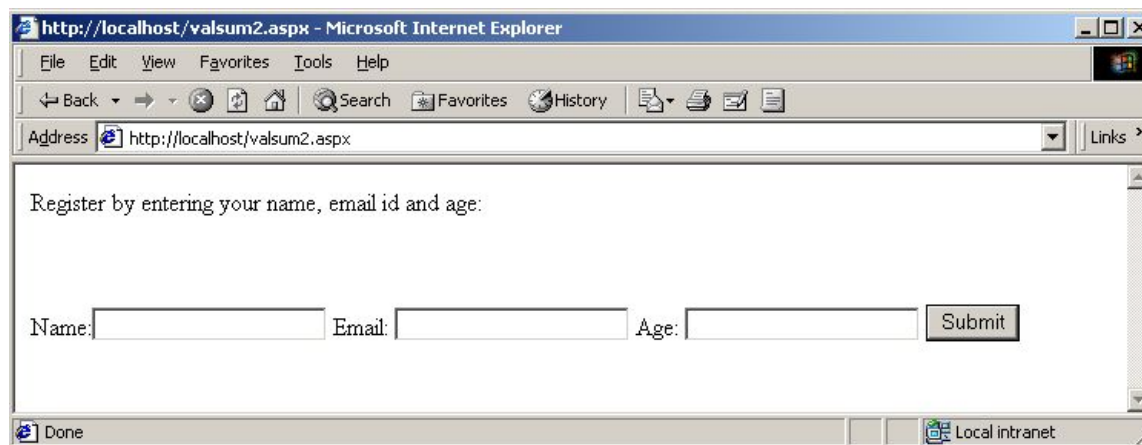
Сообщени  
е об  
ошибке

```
<asp:requiredfieldvalidator controltovalidate="userid"  
display="static" errormessage="You must enter your user  
id." runat=server>
```

The User Id cannot be left Blank!

```
</asp:requiredfieldvalidator>
```

# Сообщения об ошибках при проверке



http://localhost/valsum2.aspx - Microsoft Internet Explorer

File Edit View Favorites Tools Help

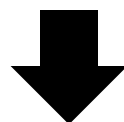
Back Forward Stop Home Search Favorites History Print Mail

Address http://localhost/valsum2.aspx Links >>

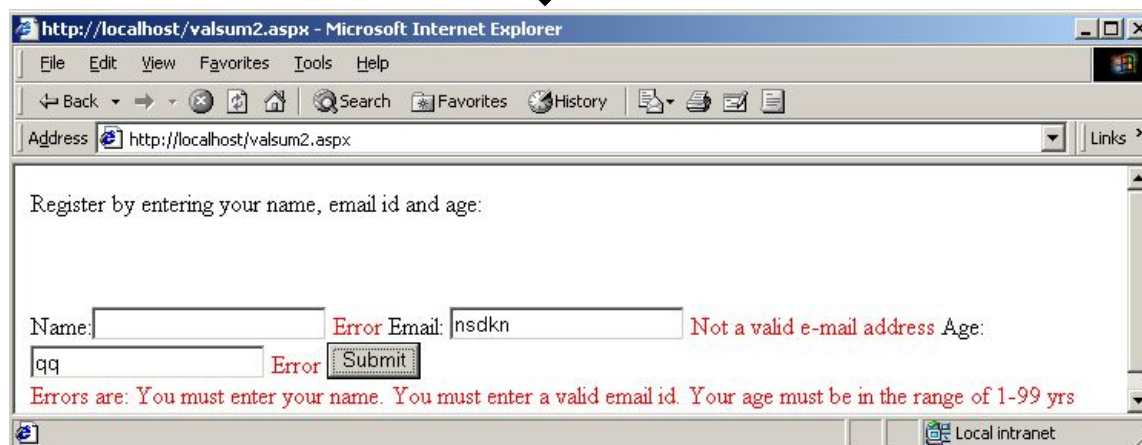
Register by entering your name, email id and age:

Name:  Email:  Age:  Submit

Done Local intranet



Динамическое отображение



http://localhost/valsum2.aspx - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Mail

Address http://localhost/valsum2.aspx Links >>

Register by entering your name, email id and age:

Name:  Error Email: nsdkn Not a valid e-mail address Age:

qq Error Submit

Errors are: You must enter your name. You must enter a valid email id. Your age must be in the range of 1-99 yrs

Local intranet

# CompareValidator

```
<asp:comparevalidator controltovalidate="pwd_con" display="static"
errormessage="the confirmation password does not match."
controltocompare="pwd" type="String" operator="Equal"
runat=server> *
</asp:comparevalidator>
```

= , < , > , <= ,  
>= , Not Equal

String, Integer, DateTime,  
Currency, Double

```
<asp:comparevalidator controltovalidate="bid" display="static"
errormessage="You cannot enter a bid for lesser than $100."
valuetocompare=100 type="Integer" operator="GreaterThanOrEqual"
runat="server">*
</asp:comparevalidator>
```

Сравнение со статическим  
значением, 100

# RangeValidator

```
<asp:rangevalidator controltovalidate="r3" type="Integer"  
minimumvalue="1" maximumvalue="99"  
errormessage="Your age must be in the range of 1-99 yrs"  
display="static" runat="server" > *  
</asp:rangevalidator>
```

Задание границ  
диапазона

Задание имени  
элемента управления

```
<asp:rangevalidator controltovalidate="r4" type="Integer"  
minimumcontrol="r1" maximumcontrol="r2"  
errormessage="Your age must be in the range of 1-99 yrs"  
display="static" runat="server" > *  
</asp:rangevalidator>
```

# RegularExpressionValidator - 1

Значёк	Смысл значка
<b>^</b>	Значёк ^ определяет, что проверка начинается с этого места
<b>\$</b>	Значёк "\$" определяет, что проверка заканчивается в этом месте
<b>[]</b>	Квадратные скобки "[]" служат для проверки, совпадает ли введенное значение с любым из символов в квадратных скобках.
<b>\w</b>	"\w" позволяет ввести любое значение
<b>\d{}</b>	"\d" определяет, что вводимое значение должно быть цифрой, а {} задают, сколько раз встречается заданный тип выражения
<b>+</b>	Значёк + означает добавление одного или более элементов в выражение для проверки

# RegularExpressionValidator - 2

Проверка адреса электронной почты email id

```
<asp:regexexpressionvalidator controltovalidate="emailid"  
display="static"  
validationexpression="^[\\w-]+@[\\w-]+\\. (com|net|org|edu|mil)$"  
runat=server>
```

Not a valid e-mail address

```
</asp:RegularExpressionValidator>
```



# CustomValidator

```
<asp:customvalidator runat="server"  
controltovalidate="grade"  
clientvalidationfunction="clval"  
onservervalidate="serval" display="static">
```

Wrong value

```
</asp:customvalidator>
```

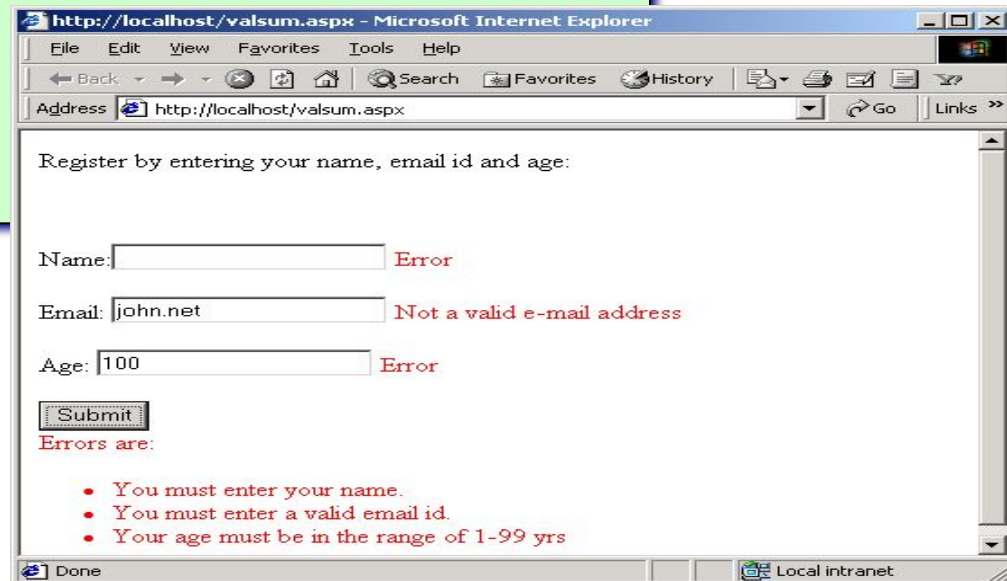
Функция  
клиентской  
стороны

# ValidationSummary

```
<asp:validationsummary id="vs1" headertext="The  
errors found are: " displaymode="singleparagraph"  
runat="server"/>  
</asp:customvalidator>
```

# СВОЙСТВО Page.IsValid

```
<script language="C#" runat="server" >
void validate_page(Object Src, EventArgs E){
    if (Page.IsValid == true) {
        lbl.Text = "Page is Valid!";}
    else {
        lbl.Text = "Page is not Valid!";    }
    }
</script>
```



The screenshot shows a Microsoft Internet Explorer window with the address bar displaying `http://localhost/valsum.aspx`. The page content includes a registration form with the following fields and errors:

- Name:  Error
- Email:  Not a valid e-mail address
- Age:  Error

Below the form is a  button. Underneath the button, the text "Errors are:" is followed by a bulleted list of error messages:

- You must enter your name.
- You must enter a valid email id.
- Your age must be in the range of 1-99 yrs

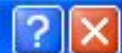
The status bar at the bottom of the browser window shows "Done" and "Local intranet".



# Global.asax

- Хранится в корневом каталоге приложения
- Определяет границы приложения
- Инициализирует переменные уровня приложения  
или сессии
- Содержит полезные события

Add New Item - C:\Documents and Settings\Александр\Мои документы\Visual Studio 2005\WebSi...



Templates:

**Visual Studio installed templates**

- |                         |                               |                        |
|-------------------------|-------------------------------|------------------------|
| Web Form                | Master Page                   | Web User Control       |
| HTML Page               | Web Service                   | Class                  |
| Style Sheet             | Global Application Class      | Web Configuration File |
| XML File                | XML Schema                    | Text File              |
| Resource File           | SQL Database                  | DataSet                |
| Generic Handler         | Site Map                      | Mobile Web Form        |
| VBScript File           | Report                        | JScript File           |
| Mobile Web User Control | Mobile Web Configuration File | XSLT File              |
| Skin File               | Browser File                  | Class Diagram          |

**My Templates**

Search Online Templates...

A class for handling Web Application events

Name:

Global.aspx

Language:

Visual C#

- ☐ Place code in separate file
- ☐ Select master page

Add

Cancel

## События в Global.asax

```
void Application_Start(object sender, EventArgs e)
{
    // Code that runs on application startup
}

void Application_End(object sender, EventArgs e)
{
    // Code that runs on application shutdown
}

void Application_Error(object sender, EventArgs e)
{
    // Code that runs when an unhandled error occurs
}

void Session_Start(object sender, EventArgs e)
{
    // Code that runs when a new session is started
}

void Session_End(object sender, EventArgs e)
{
    // Code that runs when a session ends.
    // Note: The Session_End event is raised only when the sessionstate mode
    // is set to InProc in the Web.config file. If session mode is set to StateServer
    // or SQLServer, the event is not raised.
}
```

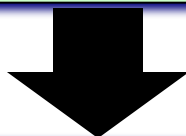
# Объект Application - 1

Представляет экземпляр приложения  
ASP.NET



Синтаксис

Object[varName]



```
Application ["greeting"] = "Welcome to my World";
```

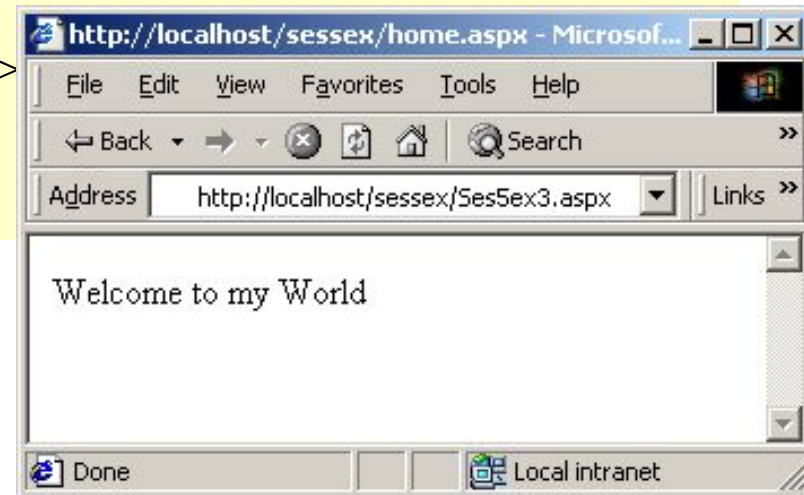
# Объект Application - 2

Global.asax

```
<script language="C#" runat="server">  
void Application_OnStart(Object sender, EventArgs E) {  
    Application ["greeting"] = "Welcome to my World";  
}</script>
```

page.aspx

```
<HTML>  
<script Language="C#" runat="server" Debug="true">  
void Page_Load(Object Src, EventArgs E) {  
    Response.Write(Application ["greeting"]);  
}</script>  
<form runat="server" >  
</form>  
</HTML>
```





# Объект Application - 3

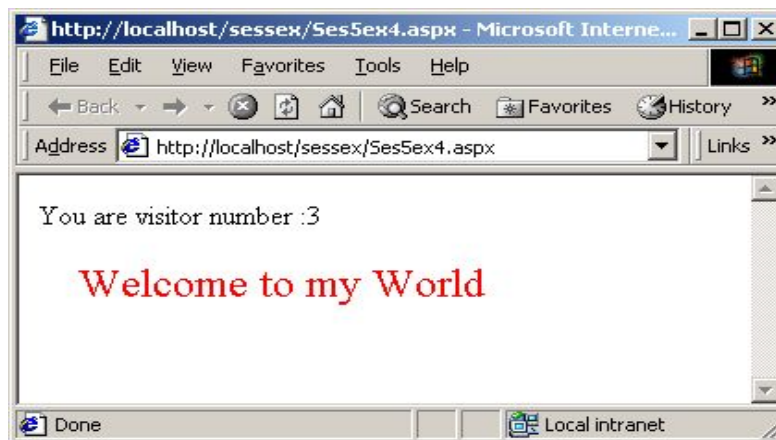
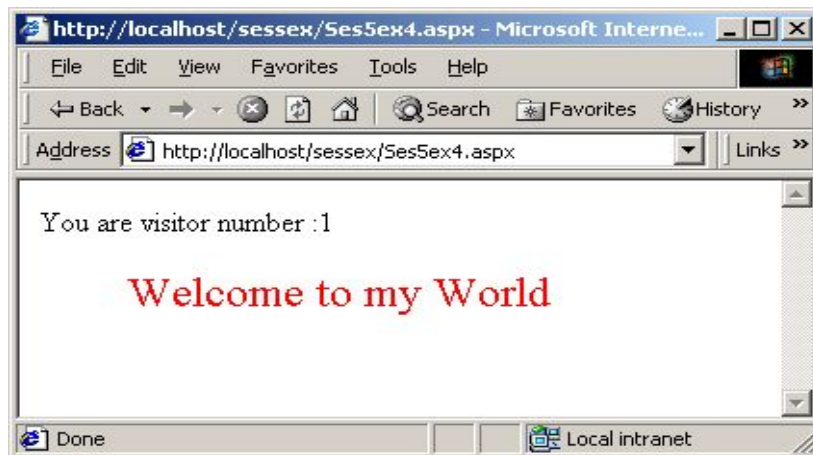
```
<script language="C#" runat="server">  
    void Application_OnStart(Object sender, EventArgs E) {  
        Application ["Counter"] = 0;  
    }</script>
```

Global.asax

```
<html>  
<script Language ="C#" runat ="server" Debug = "true">  
void Page_Load(Object Src, EventArgs E){  
    Application["Counter"] = (Int32) Application ["Counter"] + 1;  
    Response.Write("You are visitor number :" + Application ["Counter"]);  
}</script>  
<form runat= "server" >  
    <MARQUEE BEHAVIOR="scroll" Scrolldelay = 25><FONT SIZE = 5  
    COLOR  
    = RED>Welcome to my World</FONT></MARQUEE>  
</form></html>
```

page.aspx

# Объект Application - 4



# Управление доступом

```
<HTML>
<script Language ="C#" runat ="server" Debug = "true">
void Page_Load(Object Src, EventArgs E)
{
Application.Lock();
Application["Counter"] = (Int32) Application ["Counter"] + 1;
Application.Unlock();
}
</script>
<BODY>
<br>This page has been visited
<%Response.Write(Application ["Counter"]);%>
times!!
</BODY>
</HTML>
```

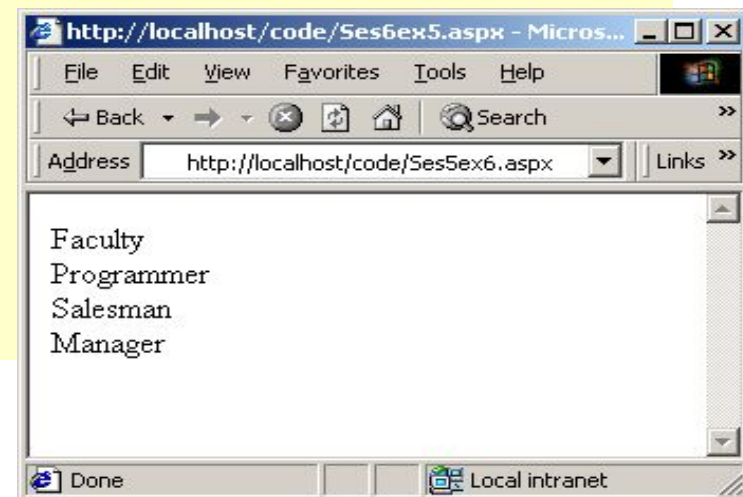
# Массивы

```
<script language="C#" runat="server">
    void Application_Start(Object sender, EventArgs E)
    { String [] job = new String [4];
      job[0]= "Faculty";  job[1]= "Programmer";
      job[2]= "Salesman"; job[3]= "Manager";
      Application ["j"] = job; } </script>
```

Global.asax

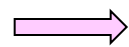
```
<<HTML> <script Language ="C#" runat ="server" >
void Page_Load(Object Src, EventArgs E) {
int i = 0; String[] k;
k = (String[])Application["j"];
for (i = 0; i<k.Length;i++)
{ Response.Write(k[i] + "<br>");}
}</script></HTML>
```

sess5ex6.aspx



# Объект Server

Синтаксис



`Server.property | method`

Свойства

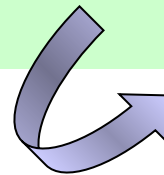
Свойство	Описание
ScriptTimeout	Задаёт период времени, в течение которого скрипт может исполняться на сервере перед тем, как он будет остановлен.
MachineName	Возвращает машинное имя сервера.

Методы

- **Execute и Transfer**
- **HTMLEncode**
- **URLEncode**
- **MapPath**

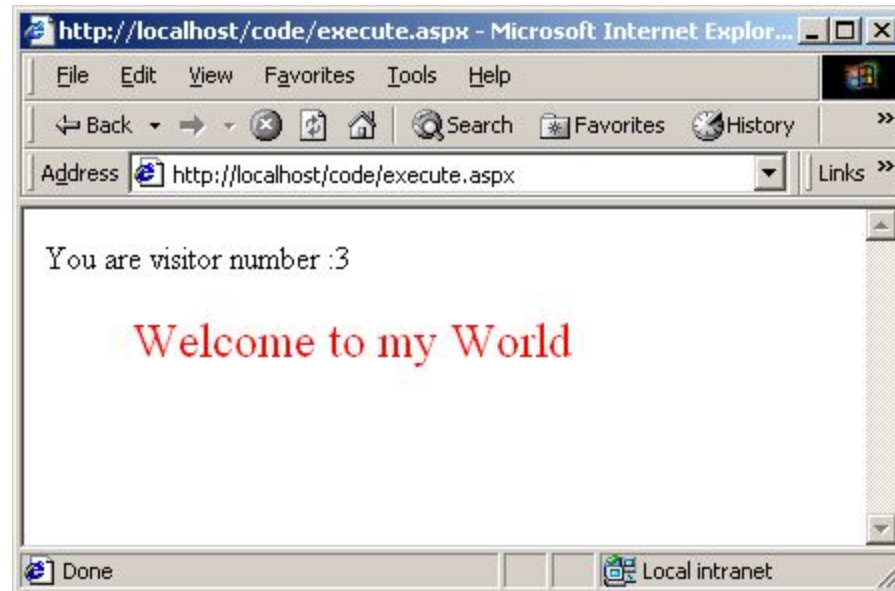
# Метод Execute

```
<%@ Page Debug = "true"%><html>  
<script language="C#" runat="server">  
void clicked (Object Src, EventArgs E){  
Server.Execute("/test/ses6ex1.aspx");  
</script>  
<form runat = "server">  
<asp:button id = "btn1" onclick = "clicked" Text = " Click me to  
transfer execution" runat = "server" />  
</form></html>
```



# Метод Transfer

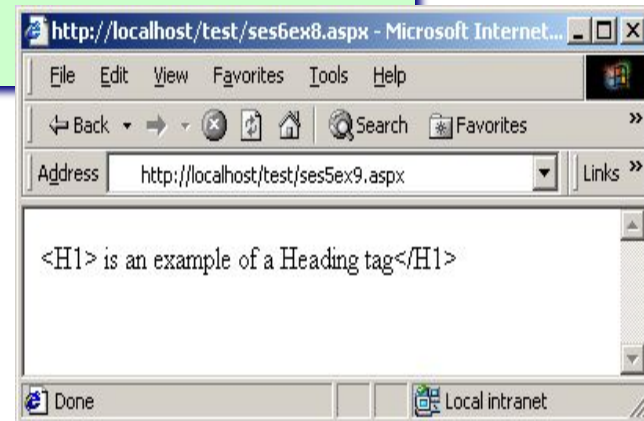
```
Server.Transfer("/test/transfer.aspx");
```



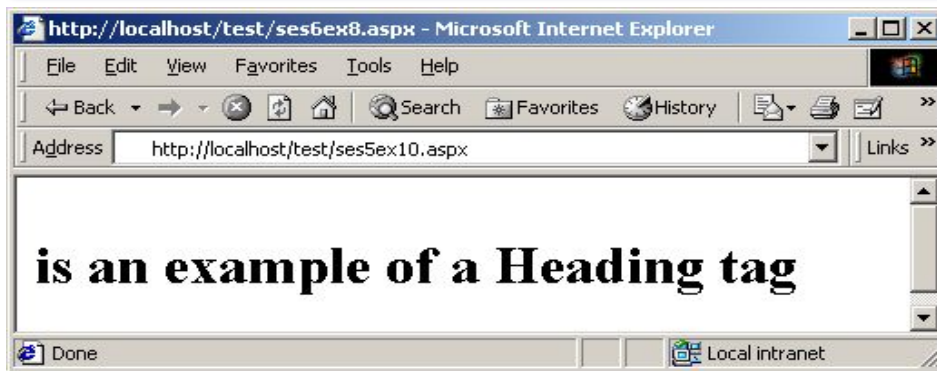
# Метод HTMLEncode

СИНТАКСИС → `Server.HTMLEncode (string)`

```
Response.Write( Server.HtmlEncode("<H1> is an  
example of a Heading tag</H1>"));
```



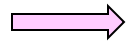
```
Response.Write("<br><H1> is an  
example of a Heading tag</H1>");
```





# Метод URLEncode - 1

СИНТАКСИС



`Server.URLEncode (string)`

```
Response.Write(Server.URLEncode("http://localhost/code/map.aspx"));
```

A pink arrow pointing from the code snippet above to the encoded string below.

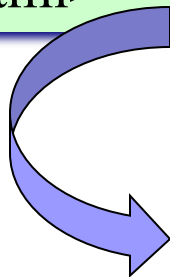
`http%3a%2f%2flocalhost%2fcode%2fmap.aspx`

# Метод URLEncode - 2

```
<html>
<script language="C#" runat="server">
void clicked (Object Src, EventArgs E)
{
String name = Server.UrlEncode("John Saunders");
Response.Redirect
("http://localhost/code/calendar.aspx?name=" + name);
}
</script>
<form runat = "server">
<asp:button id = "btn1" onclick = "clicked" Text = " Click me
to send values" runat = "server" />
</form>
</html>
```

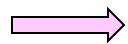
# Метод URLEncode - 3

```
<html><script language="C#" runat="server">
void clicked (Object Src, EventArgs E){
String name = Server.UrlEncode("John Saunders");
String password = Server.UrlEncode("king");
Response.Redirect("http://localhost/code/error.aspx?name=" + name
+ " &password=" + password);
}</script>
<form runat = "server">
<asp:button id = "btn1" onclick = "clicked" Text = " Click me to send
values" runat = "server" />
</form></html>
```



# Метод MapPath

СИНТАКСИС



`Server.MapPath (path)`



```
Response.Write( Server.MapPath("/encode.aspx"));
```

`c:\inetpub\wwwroot\encode.aspx`



```
Response.Write( Server.MapPath("encode.aspx"));
```

`C:\Tested Code\encode.aspx`



```
Response.Write(Server.MapPath(Request.ServerVariables.  
Get("PATH_INFO")));
```

`C:\Tested Code\map.aspx`

# Закладки - 1

**Закладка (cookie) – это небольшой пакет информации, который используется для сохранения информации на компьютере пользователя. Закладки могут быть временными или постоянными.**

**Временные закладки**

**Истекают,  
когда  
пользователь  
покидает веб-  
сайт**



**Постоянные закладки**

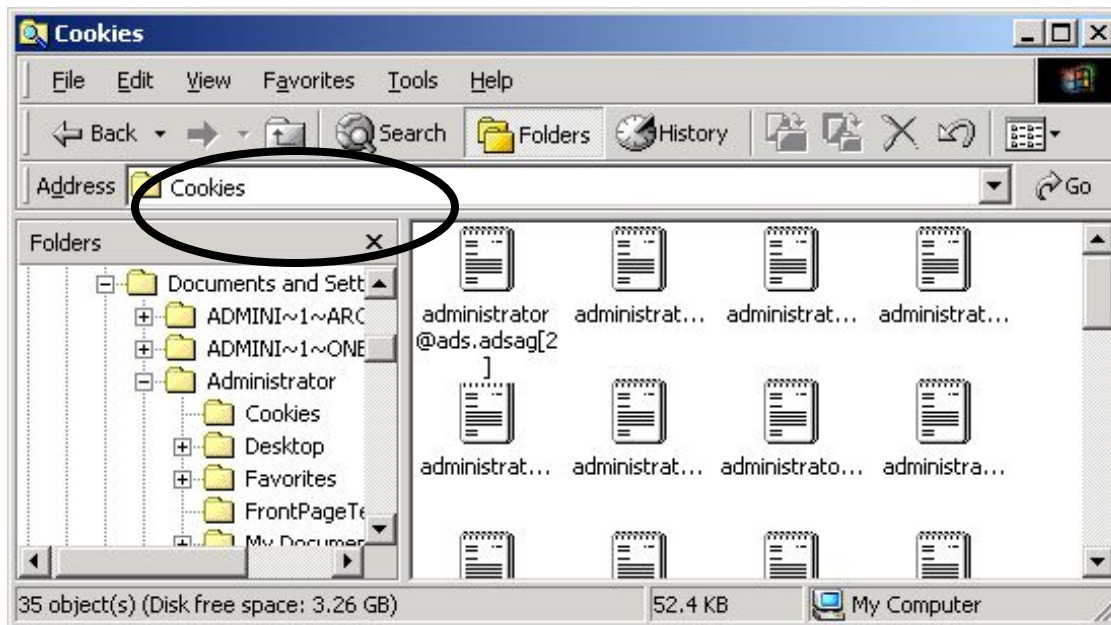
**Остаются на клиентской  
машине и могут быть  
прочитаны веб-сервером,  
когда пользователь снова  
обратится к этому веб-  
сайту.**

# Закладки - 2

```
HttpCookie Object name = new HttpCookie("Cookie  
Name","Cookie Value");
```

Имя закладки

Свойство закладки



**Domain**      **Expires**  
**Path**        **Value**  
**Secure**      **HasKeys**

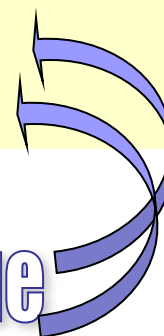
# Закладки - 3

```
HttpCookie cookie1 = new HttpCookie("UserName","John Cruise");  
DateTime dt = DateTime.Now;  
TimeSpan ts = new TimeSpan(30,0,0,0);  
cookie1.Expires = dt.Add(ts);
```

↑  
Одно значение

```
HttpCookie cookie1 = new HttpCookie("User");  
cookie1.Values.Add("name","John Cruise");  
cookie1.Values.Add ("password","smallgun");
```

пары имя-значение



# Извлечение данных - 1

```
HttpCookie <Cookiename1> =  
Request.Cookies[“<Cookiename2>”];
```

```
HttpCookie cookie1 = new HttpCookie("UserName");  
cookie1.Values.Add("name","John Cruise");  
cookie1.Values.Add ("password","smallgun");
```

```
<html><script language="C#" runat="server">  
void Page_Load(Object Sender, EventArgs E){  
HttpCookie cookie1 = new HttpCookie("UserName");  
DateTime dt = DateTime.Now;  
TimeSpan ts = new TimeSpan(0,0,1,0);  
cookie1.Expires = dt;  
cookie1.Expires = dt.Add(ts);  
Response.Cookies.Add(cookie1);
```

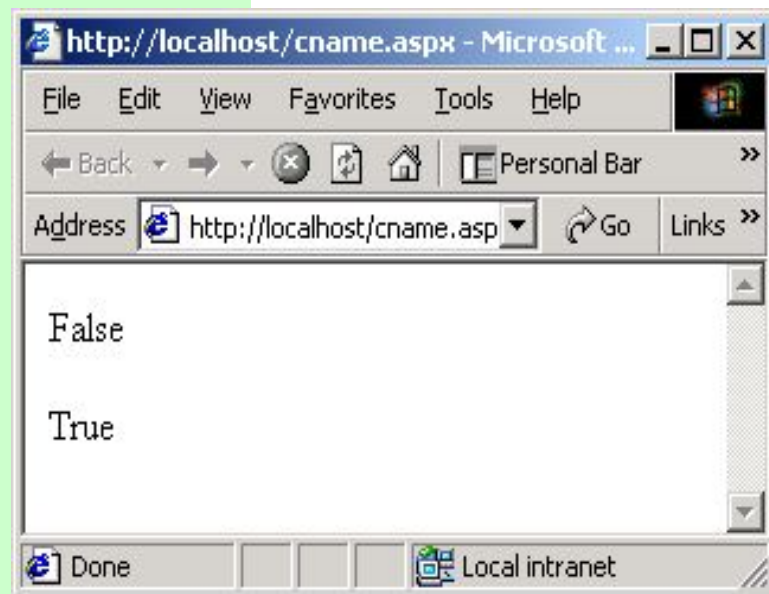


# Извлечение данных - 2

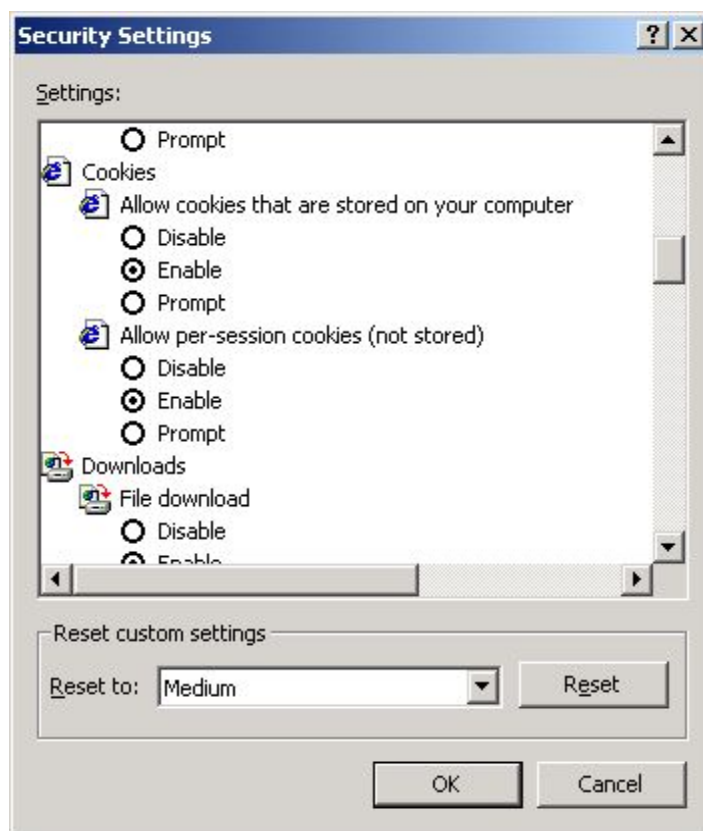
A

```
HttpCookie cookie5;  
cookie5 = Request.Cookies["UserName"];  
bool b = cookie5.HasKeys;  
Display1.Text = b.ToString();  
HttpCookie cookie2 = new HttpCookie("UserName1");  
cookie2.Values.Add("name","John Cruise");  
cookie2.Values.Add ("password","smallgun");  
cookie2.Expires = dt;  
cookie2.Expires = dt.Add(ts);  
Response.Cookies.Add(cookie2);  
HttpCookie cookie6;  
cookie6 = Request.Cookies["UserName1"];  
b = cookie6.HasKeys;  
Display2.Text = b.ToString();}  
</script>
```

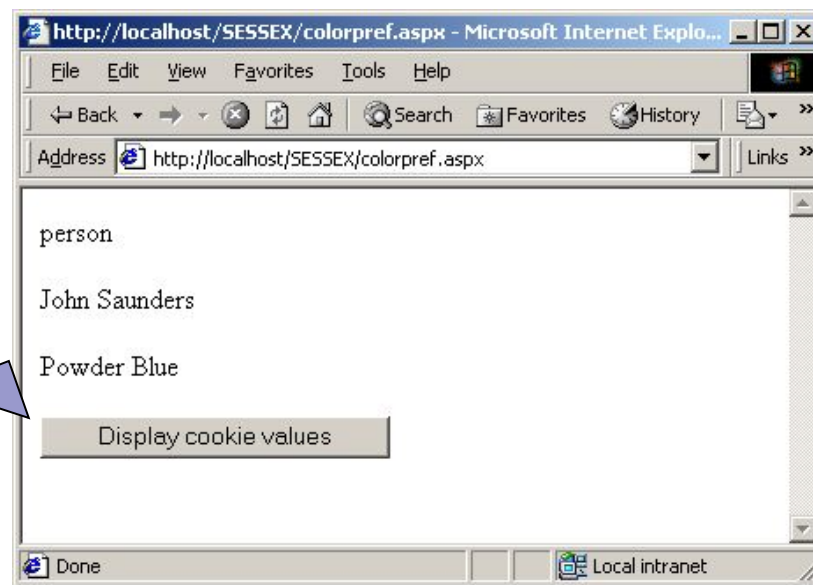
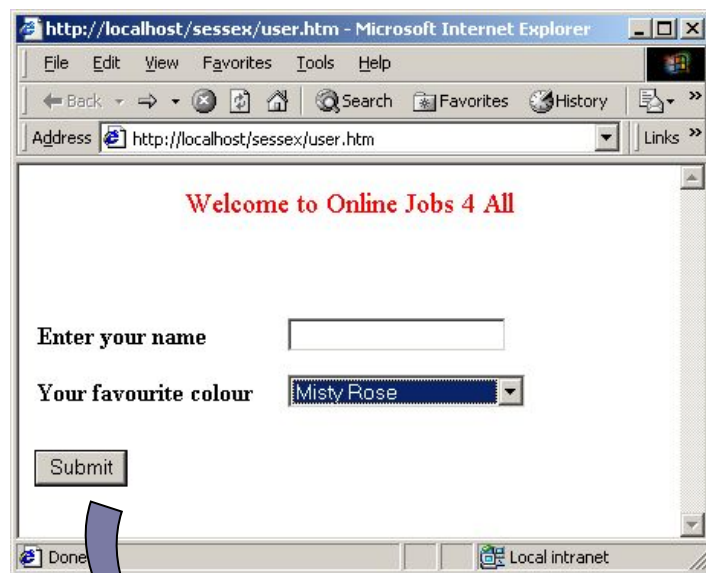
```
<asp:Label id = "Display1" runat="server" /><br><br>  
<asp:Label id = "Display2" runat="server" />  
</html>
```



# Установки для закладок в IE



# Использование закладок



# Объект Session

- Хранит информацию о пользователе

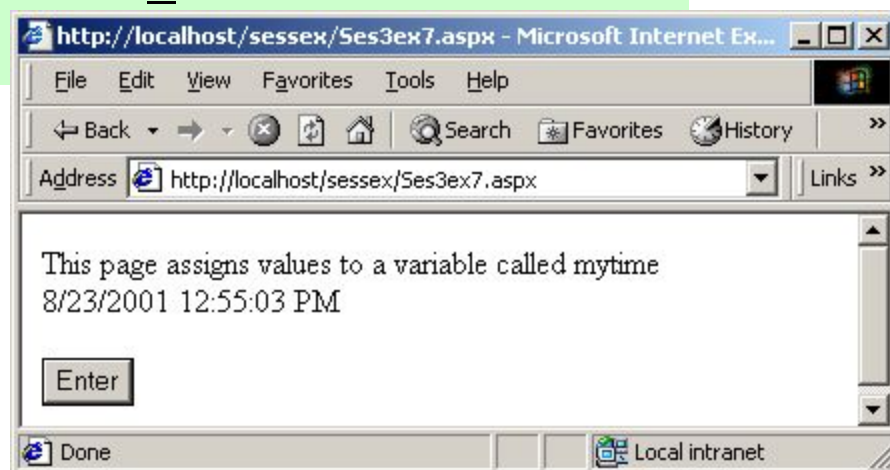
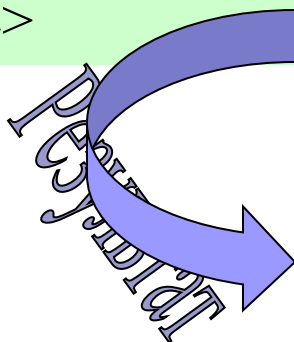
## Достоинства

- Содержит информацию, специфическую для данного пользователя
- Отслеживает и поддерживает информацию о пользователе на протяжении сессии
- Уничтожает объект после завершения сессии

# Использование переменных

## - 1

```
<HTML><script language="C#" runat="server">
void Btn_Click(Object Src, EventArgs E) {
Session ["mytime"] = DateTime.Now;
Message.Text = Session ["mytime"].ToString();}
</script>
<form runat="server">
This page assigns values to a variable called mytime <BR>
<asp:label id="Message" runat=server/><br> <br>
<asp:button text="Enter" Onclick="Btn_Click" runat=server/>
</form></HTML>
```



# Использование переменных

## - 2

```
<HTML><script language="C#" runat="server">
void Page_Load (Object Src, EventArgs E) {
    Message.Text = Session ["mytime"].ToString();
}
</script>
```

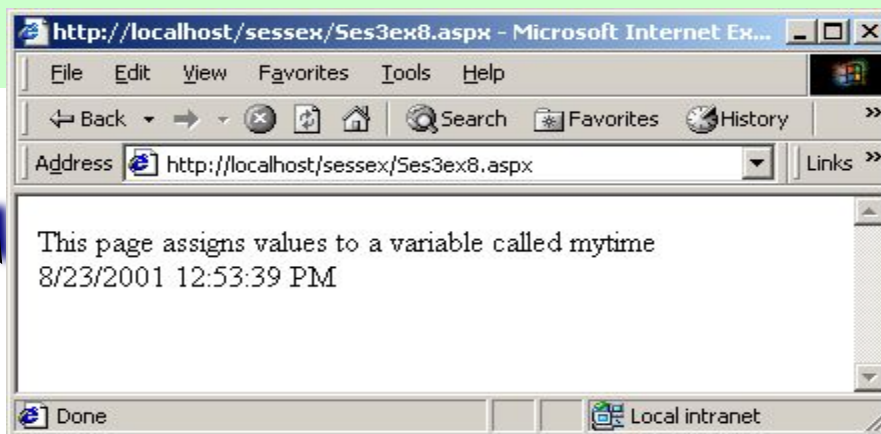
```
<form runat="server">
```

This page assigns values to a variable called mytime <BR>

```
<asp:label id="Message" runat=server/><br> <br>
```

```
</form></HTML>
```

Результат



# События и свойства Session

События



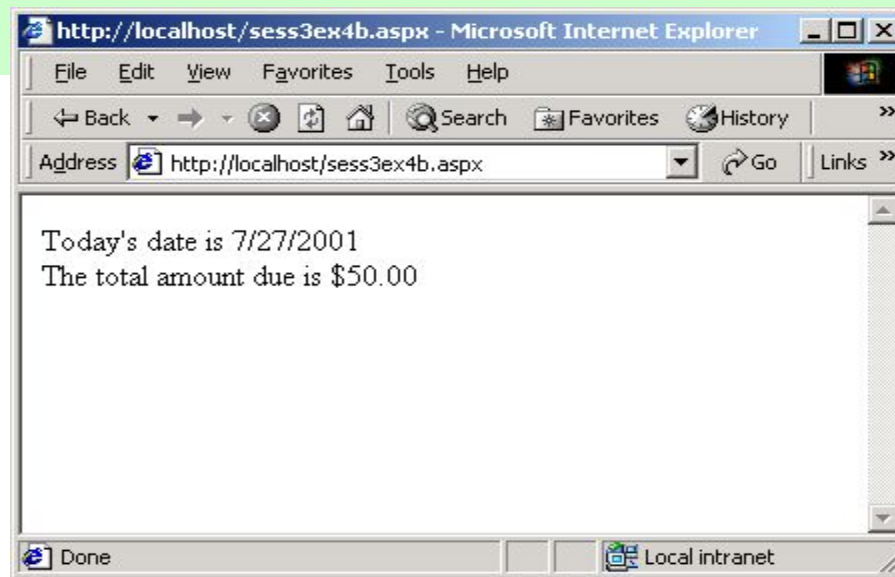
Session.collection | property | method

Свойство	Описание
SessionID	Уникальный идентификатор сессии пользователя
TimeOut	Время истечения сессии пользователя
LCID	Локальный идентификатор
IsNewSession	Возвращает TRUE если в текущем запросе была создана данная сессия

# Пример объекта Session-1

```
<HTML><script language="C#" runat="server">
void Page_Load (Object Src, EventArgs E){
DateTime dt;
dt = DateTime.Now;
int due = 50;
Response.Write("Today's date is " + dt.ToShortDateString() + "<br>");
Response.Write("The total amount due is " + due.ToString("C") ); }
</script> </HTML>
```

Результат





# Пример объекта Session - 2

```
<HTML><script language="C#" runat="server">
void Page_Load (Object Src, EventArgs E){
Session.LCID =0x040C; }
void Btn_Click(Object Src, EventArgs E){
DateTime dt;
dt = DateTime.Now;
int due = 500;
label1.Text = "Today's date is"+dt.ToShortDateString()+ "<br>";
label2.Text = "The total amount due is " + due.ToString("C") ; }
</script> <form runat="server">
<asp:button id=but text="Enter" onclick="Btn_Click" runat="server"/>
<asp:label id=label1 runat="server" />
<asp:label id=label2 runat="server" />
</form> </HTML>
```

Результат

