



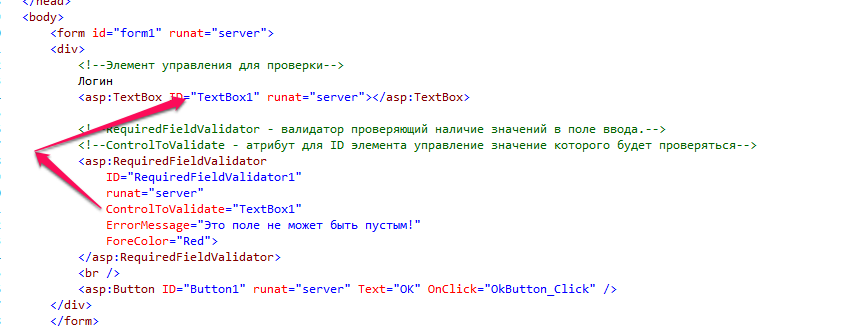


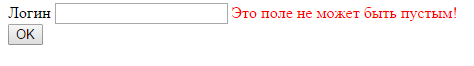
<asp:RequiredFieldValidator

На наличие хотя б одного значения

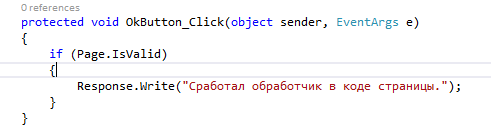
RequiredFieldValidator - валидатор проверяющий наличие значений в поле ввода

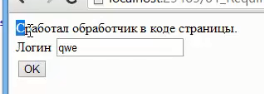
ControlToValidate - атрибут для ID элемента управление значение которого будет проверяться





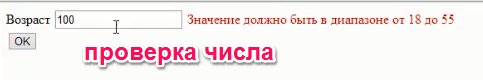
Page.IsValid

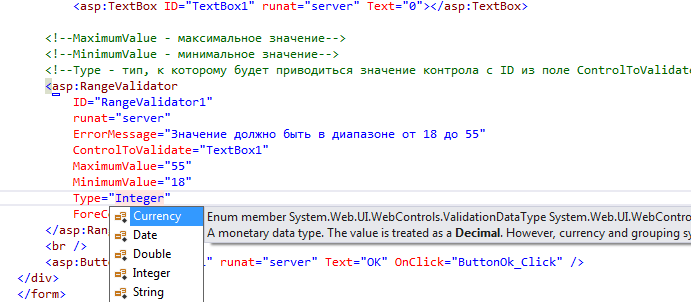




<asp:RangeValidator

Диапазон числа





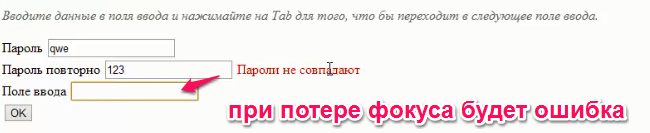
<asp:CompareValidator

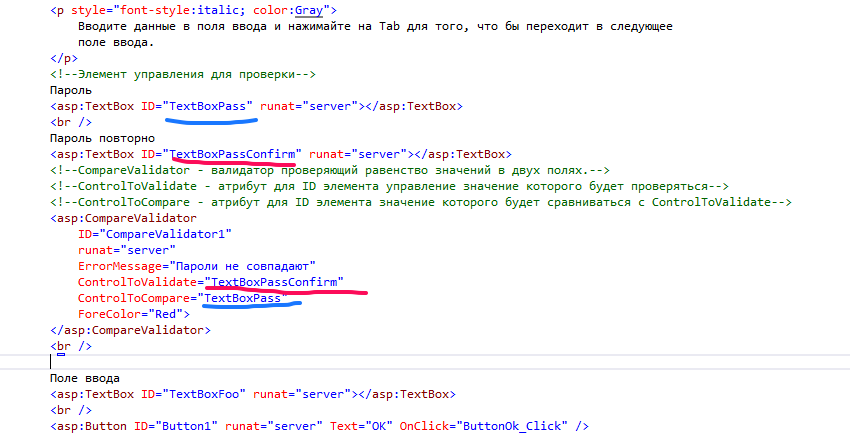
Проверяет равенство значений в двух полях

CompareValidator - валидатор проверяющий равенство значений в двух полях

ControlToValidate - атрибут для ID элемента управление значение которого будет проверяться

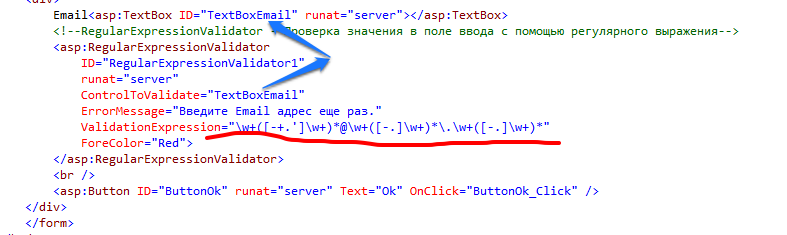
ControlToCompare - атрибут для ID элемента значение которого будет сравниваться с ControlToValidate

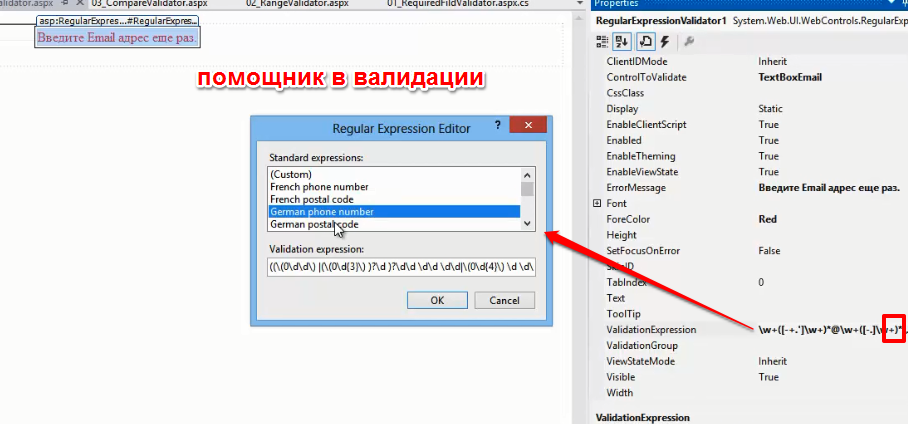




RegularExpressionValidator

Проверка значения в поле ввода с помощью регулярного выражения





<asp:CustomValidator

ClientValidationFunction - функция для проверки данных на стороне клиента

OnServerValidate - метод для проверки данных на стороне сервера.

<head id="Head1" runat="server">

<title>CustomValidator</title>

<script type="text/javascript">

// source - span, в который выводиться сообщение об ошибке. ЭТО Валидатор в который будем записывать значение

// arguments - дополнительные параметры:

// Value - значение поля ввода.

// IsValid - корректность введенных данных в поле ввода.

function validate(source, arguments) {

if (arguments.Value % 2 == 0) {

arguments.IsValid = true;

}

else {

source.innerHTML = "Значение должно быть четным числом";

arguments.IsValid = false;

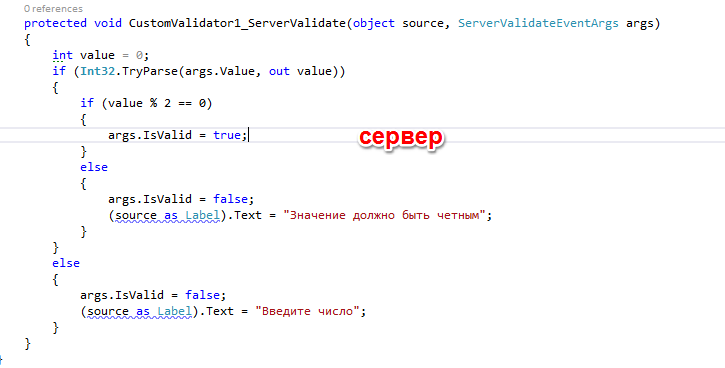
}

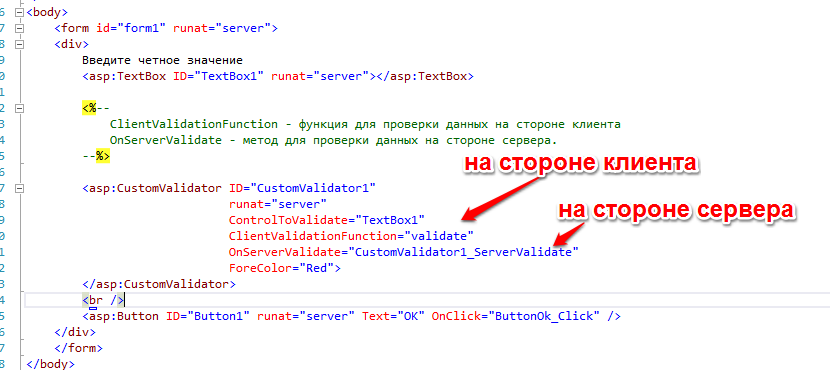
}

</script>

</head>

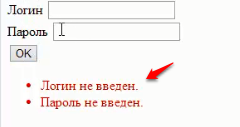
Серверный код пишем если на клиенте выключен javascript





ValidationSummary

серверный элемент управления, который позволяет вывести все ошибки при заполнении формы в одном месте



<asp:ValidationSummary

ID="ValidationSummary1"

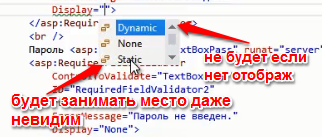
runat="server"

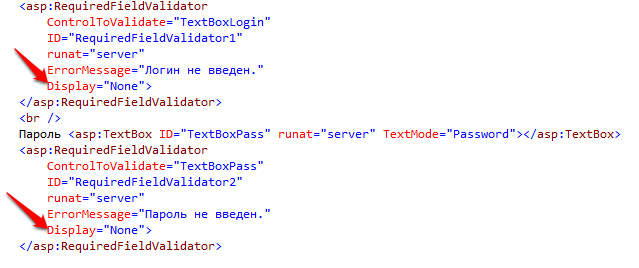
DisplayMode="BulletList"

ForeColor="Red" />

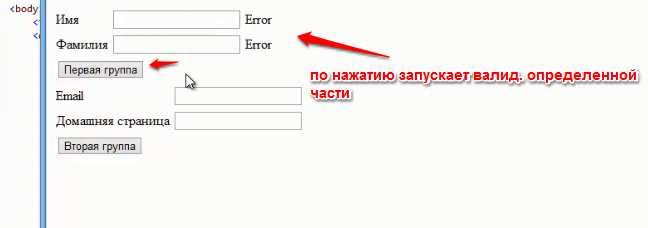
<br />

Валидаторы



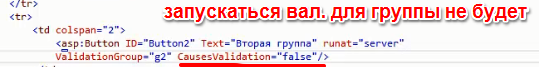


ValidationGroup



Как для первой группы так и для второй, только менять ее имя

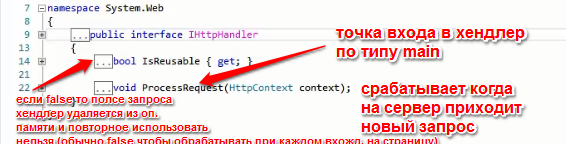






Создать класс унаследовав два интерфейса

IHttpHandler



IRequiresSessionState

Для работы хендлера с сессией

public class Captcha : IHttpHandler, IRequiresSessionState

{

Random rand = new Random();

public void ProcessRequest(HttpContext context)

{

// Указываем тип ответа данного хэндлера.

context.Response.ContentType = "image/jpeg";

// Высота и ширина картинки

int height = 60;

int width = 100;

// Объект изображения.

Bitmap captchaImage = new Bitmap(width, height);

// Объект для изменения пикелей изображения.

Graphics gr = Graphics.FromImage(captchaImage);

// Создание строки для вывода на изображении.

string captchaString = GenerateCaptchaString();

// Сохраняем текст, который будет выведен на изображении в Session

HttpContext.Current.Session["captcha"] = captchaString;

// Заполнение изображения цветом.

gr.FillRectangle(GenerateBrush(), new **Rectangle**(0, 0, width, height));

// Создание кисти для рисования текста.

Brush textBrush = GenerateBrush();

// Добавление в содержимое страницы текста.

gr.DrawString(captchaString,

new Font("Arial", 10.0f),

textBrush,

new **Point**(20, 20));

// Сохранение изображения в ответ пользователю.

captchaImage.Save(context.Response.OutputStream, ImageFormat.Jpeg);

// Завершение формирования ответа и отправка изображения пользователю.

context.Response.End();

}

// Метод для создания текста каптчи.

private string GenerateCaptchaString()

{

string symbols = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

string captchaString = string.Empty;

for (int i = 0; i < 5; i++)

{

int pos = rand.Next(0, symbols.Length);

captchaString += symbols[pos];

}

return captchaString;

}

// Метод для создания случайного цвета.

private Brush GenerateBrush()

{

int r = rand.Next(255);

int g = rand.Next(255);

int b = rand.Next(255);

return new SolidBrush(**Color**.FromArgb(r, g, b));

}

public bool IsReusable

{

get

{

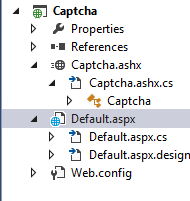
return false;

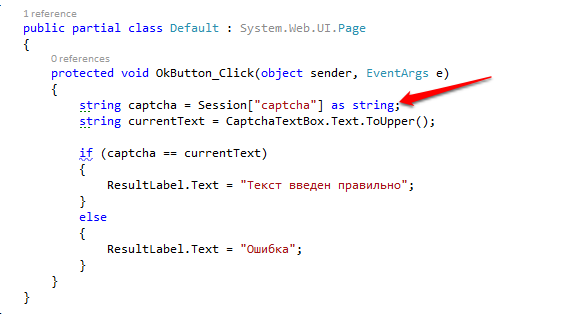
}

}

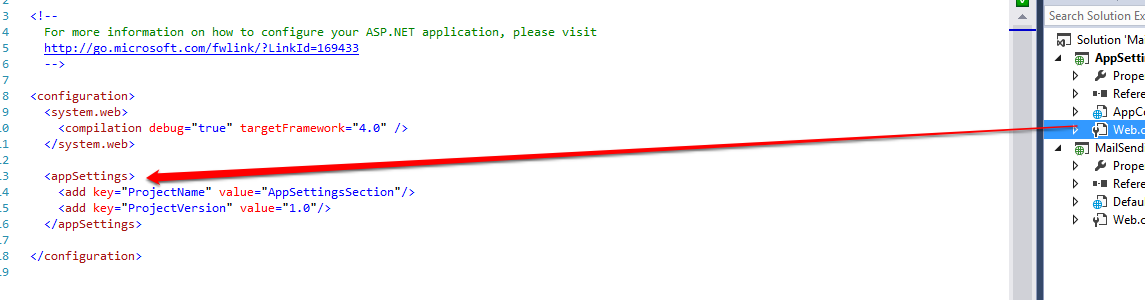
}





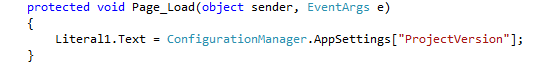


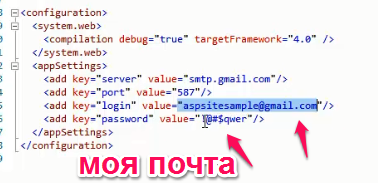
Отправка сообщения пользователя на емаил при регистрации



Обращаемся как со странице так и с кода к web config файлу









MailMessage и SmtpClient



<asp:Menu

<div>

<asp:Menu runat="server"

Orientation="Horizontal"

BackColor="#F7F6F3"

DynamicHorizontalOffset="2"

Font-Names="Verdana"

Font-Size="0.8em"

ForeColor="#7C6F57"

StaticSubMenuIndent="10px">

<%-- Структура пунктов меню --%>

<Items>

<asp:MenuItem Text="Item 1">

<asp:MenuItem Text="Sub Menu 1"></asp:MenuItem>

<asp:MenuItem Text="Sub Menu 2"></asp:MenuItem>

</asp:MenuItem>

<asp:MenuItem Text="Item 2">

<asp:MenuItem Text="Sub Menu 1"></asp:MenuItem>

<asp:MenuItem Text="Sub Menu 2"></asp:MenuItem>

<asp:MenuItem Text="Sub Menu 3">

<asp:MenuItem Text="Sub Sub Menu 1"></asp:MenuItem>

<asp:MenuItem Text="Sub Sub Menu 2"></asp:MenuItem>

</asp:MenuItem>

</asp:MenuItem>

<asp:MenuItem Text="Item 3"></asp:MenuItem>

<asp:MenuItem Text="Item 4"></asp:MenuItem>

</Items>

<%--Стили для выпадающих пунктов меню--%>

<DynamicHoverStyle BackColor="#7C6F57" ForeColor="White" />

<DynamicMenuItemStyle HorizontalPadding="5px" VerticalPadding="2px" />

<DynamicMenuStyle BackColor="#F7F6F3" />

<DynamicSelectedStyle BackColor="#5D7B9D" />

<%--Стили для статических(постоянно видимых) пунктов меню--%>

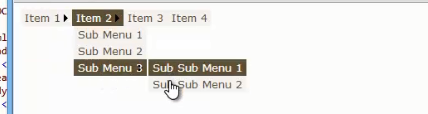
<StaticHoverStyle BackColor="#7C6F57" ForeColor="White" />

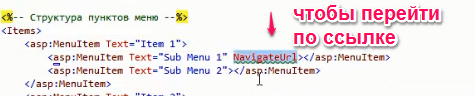
<StaticMenuItemStyle HorizontalPadding="5px" VerticalPadding="2px" />

<StaticSelectedStyle BackColor="#5D7B9D" />

</asp:Menu>

</div>





<asp:TreeView

<div>

<asp:TreeView runat="server" ImageSet="XPFileExplorer" NodeIndent="15">

<HoverNodeStyle Font-Underline="True" ForeColor="#6666AA" />

<Nodes>

<asp:TreeNode Text="Поисковики">

<asp:TreeNode Text="Google" NavigateUrl="http://google.com"></asp:TreeNode>

<asp:TreeNode Text="Yandex" NavigateUrl="http://yandex.ru"></asp:TreeNode>

<asp:TreeNode Text="Yahoo" NavigateUrl="http://yahoo.com"></asp:TreeNode>

</asp:TreeNode>

<asp:TreeNode Text="Регистраторы">

<asp:TreeNode Text="Name.com" NavigateUrl="http://name.com"></asp:TreeNode>

<asp:TreeNode Text="Godaddy" NavigateUrl="http://godaddy.com"></asp:TreeNode>

<asp:TreeNode Text="Имена UA" NavigateUrl="http://imena.ua"></asp:TreeNode>

</asp:TreeNode>

</Nodes>

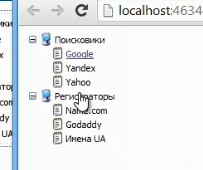
<NodeStyle Font-Names="Tahoma" Font-Size="8pt" ForeColor="Black" HorizontalPadding="2px" NodeSpacing="0px" VerticalPadding="2px" />

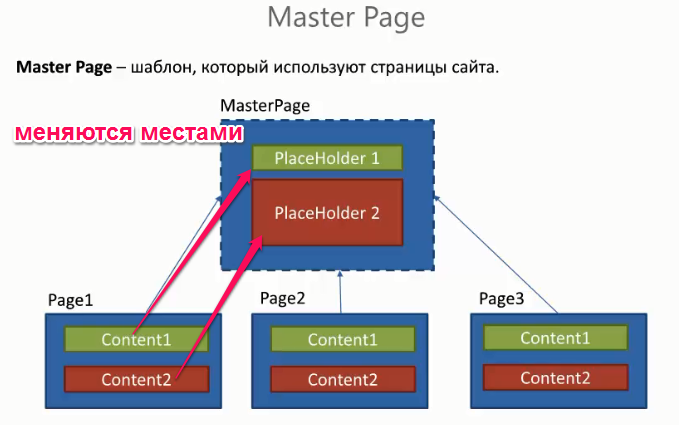
<ParentNodeStyle Font-Bold="False" />

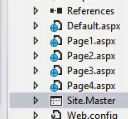
<SelectedNodeStyle BackColor="#B5B5B5" Font-Underline="False" HorizontalPadding="0px" VerticalPadding="0px" />

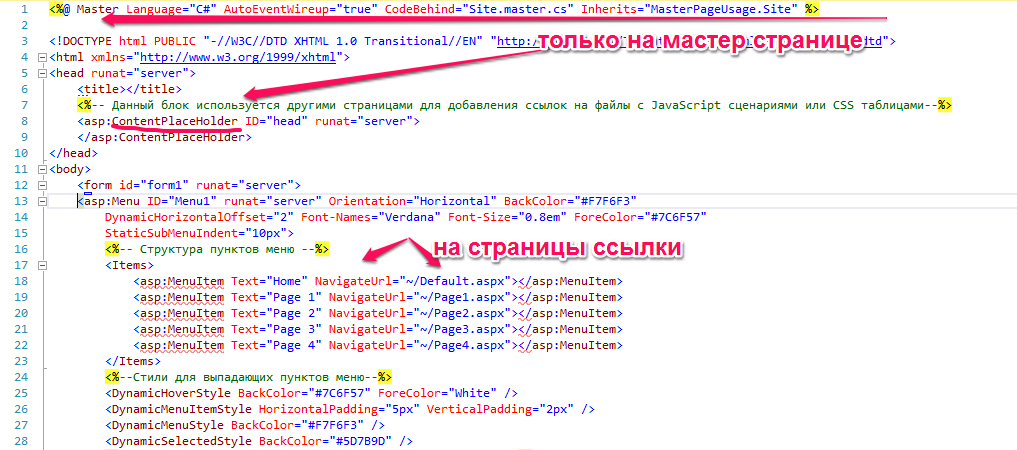
</asp:TreeView>

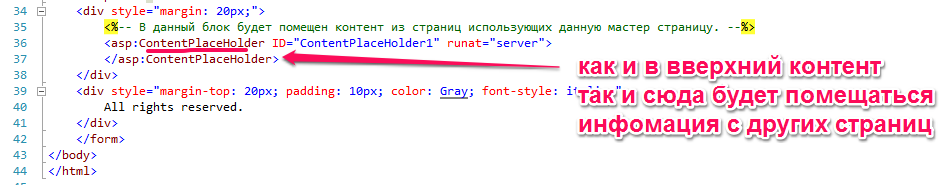
</div>



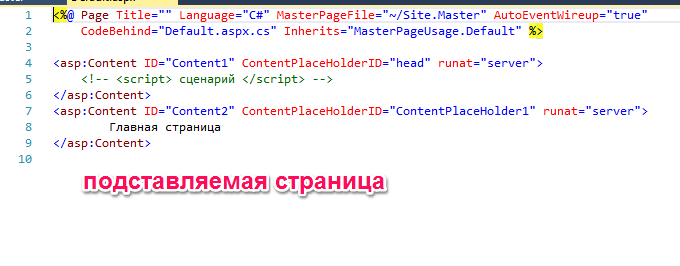






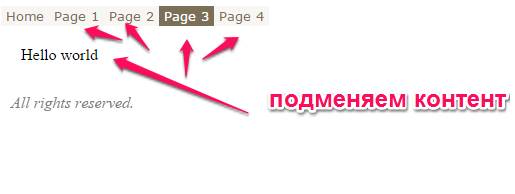


Подставляемая страница



Перегружается полностью документ





Создание страницы привязанной к мастер страницы

