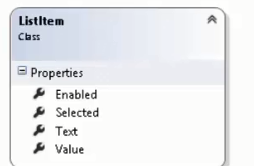
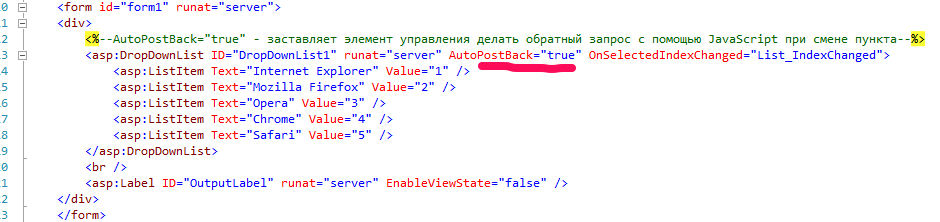


Все наследуются от

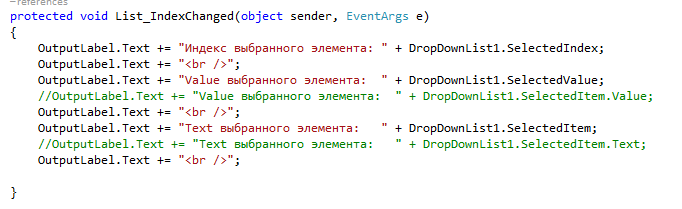


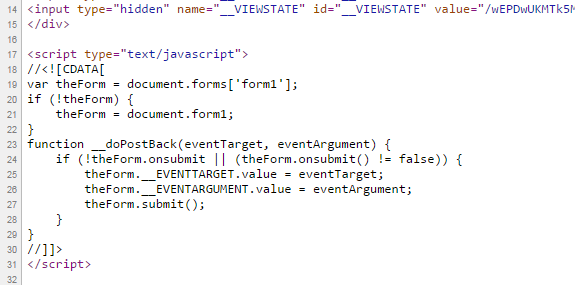
<asp:DropDownList

AutoPostBack="true" - заставляет элемент управления делать обратный запрос с помощью JavaScript при смене пункта



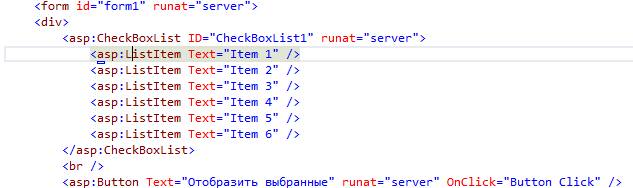
Получение значений



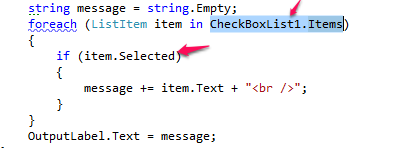


При добавлении новых элементов в элементы они сохраняются так как хранятся во VIEWSTATE

<asp:CheckBoxList



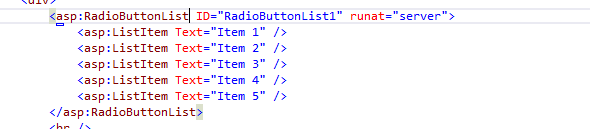
Получаем все элементы, проверяем выделены ли они и выводим на экран



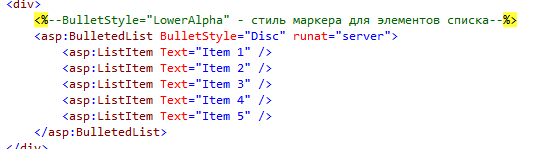
asp:RadioButtonList

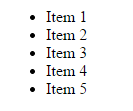
выбранный элемент

RadioButtonList1.SelectedItem.Text;

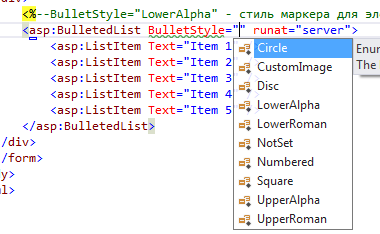


<asp:BulletedList





Разные виды



<asp:AdRotator-

банерная рулетка, случайный показ рекламы

<!--

Все настройки элемента управления находятся в файле AdvertismentFile.xml

Основные тэги:

ImageUrl - путь к изображению, которое будет отображаться

NavigateUrl - путь по которому будет происходить перенаправление в случае если пользователь кликнет баннер

AlternateText - альтернативный текст который будет отображаться если изображение не удалось загрузить или в

браузере пользователя отключены изображения.

Impressions - Значимость объявления по сравнению с остальными объявлениями

Keyword - Категория рекламы. (чтобы потом можно было сделать фильтрацию)

-->

<!--При установке атрибута KeywordFilter объявления из файла будут отображаться только в том случае,

если содержит то же клюяевое слово в теге Keyword в файле AdvertismentFile.xml

Target="\_blank" - открытие ссылки в новой вкладке -->

<asp:AdRotator ID="AdRotator1" runat="server" AdvertisementFile="~/01\_AdRotator/AdvertismentFile.xml"

Target="\_blank" />

<Advertisements>

<Ad>

<ImageUrl>Image1.png</ImageUrl>

<NavigateUrl>http://example.com?param=1</NavigateUrl>

<AlternateText>Advertisment Block 1</AlternateText>

<Impressions>50</Impressions>

<Keyword>Category 1</Keyword>

</Ad>

<Ad>

<ImageUrl>Image2.png</ImageUrl>

<NavigateUrl>http://example.com?param=2</NavigateUrl>

<AlternateText>Advertisment Block 2</AlternateText>

<Impressions>40</Impressions>

<Keyword>Category 1</Keyword>

</Ad>

<Ad>

<ImageUrl>Image3.png</ImageUrl>

<NavigateUrl>http://example.com?param=3</NavigateUrl>

<AlternateText>Advertisment Block 3</AlternateText>

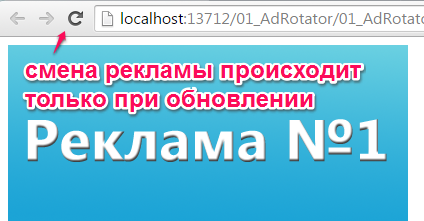
<Impressions>10</Impressions>

<Keyword>Category 2</Keyword>

</Ad>

</Advertisements>

На сколько часто будет показываться реклама зависит от Impressions, т.е. суммируются все значения это и есть процент



<asp:Calendar

<asp:Calendar SelectionMode="DayWeekMonth" runat="server" ID="Calendar1" BackColor="#FFFFCC" BorderColor="#FFCC66"

BorderWidth="1px" DayNameFormat="Shortest" Font-Names="Verdana" Font-Size="8pt"

ForeColor="#663399" Height="200px" Width="220px" ShowGridLines="True">

<DayHeaderStyle BackColor="#FFCC66" Font-Bold="True" Height="1px" />

<NextPrevStyle Font-Size="9pt" ForeColor="#FFFFCC" />

<OtherMonthDayStyle ForeColor="#CC9966" />

<SelectedDayStyle BackColor="#CCCCFF" Font-Bold="True" />

<SelectorStyle BackColor="#FFCC66" />

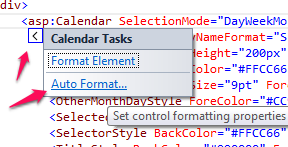
<TitleStyle BackColor="#990000" Font-Bold="True" Font-Size="9pt" ForeColor="#FFFFCC" />

<TodayDayStyle BackColor="#FFCC66" ForeColor="White" />

</asp:Calendar>



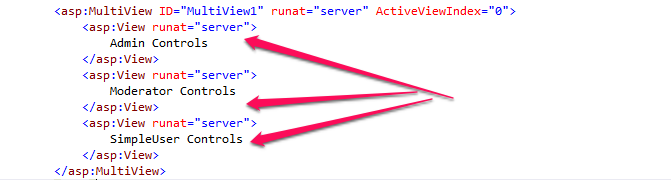
Чтобы не запоминать все настройки нужно нажать и выбрать необходимый вид



<asp:MultiView

Скрывает ссылки в зависимости от прав пользователя (как пример)

Отображаться будет только один элемент



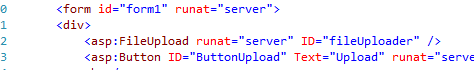


Какой именно выбираем самостоятельно

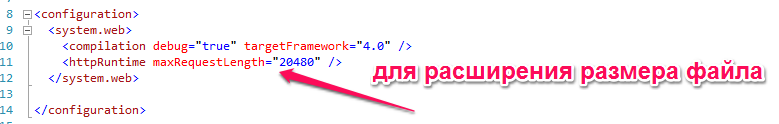
MultiView1.ActiveViewIndex = 0;

<asp:FileUpload

Страница позволяющая загружать на сервер какие то файлы



Web.config



public partial class \_04\_FileUploaderSample : System.Web.UI.Page

{

// Допустимые расширения.

List<string> \_extensions = new List<string>() { ".rar", ".txt", ".doc", ".pdf" };

// В ASP стоит ограничение на загружаймый файл равное 4096 Kb

// Для того что бы увеличить ограничение в web.config нужно добавить строку <httpRuntime maxRequestLength="20480" />

int \_maxLength = 10485760; // Максимальный размер файла. 10 Mb (10 \* 1024 \* 1024)

protected void UploadButton\_Click(object sender, EventArgs e)

{

try

{

// Получение расширения файла.

string fileExtension = Path.GetExtension(fileUploader.PostedFile.FileName);

// Получения размера файла.

int fileLenght = fileUploader.PostedFile.ContentLength;

if (!\_extensions.Contains(fileExtension))

{

throw new Exception("Файлы с таким расширением не принимаются для загрузки.");

}

if (fileLenght > \_maxLength)

{

throw new Exception("Размер файла не более 10 Mb.");

}

// Создание уникального имени для файла.

string newFileName = **Guid**.NewGuid().ToString();

// Создание пути для сохранения файла на сервере.

string newPathForSave = Path.Combine(Server.MapPath("Uploads"), newFileName + fileExtension);

// Сохранение файла.

fileUploader.SaveAs(newPathForSave); - по сути нужен только лишь этот метод

LabelForMessage.Text = "Файл успешно загружен.";

}

catch (Exception ex)

{

LabelForMessage.Text = ex.Message;

}

}

}