

# Лекция №3

## HTML 4

- **ESC-последовательности**
- **подключение нетекстовых ресурсов**
- **фреймы**
- **стандарт MIME-types**
- **meta – теги**
- **маппинг**
- **формы**

# ESC-последовательности

- для того, чтобы представлять спецсимволы в тексте документа, заменять символьные объекты

Некоторые служебные символы иногда необходимо отображать в тексте документа

Для того, чтобы интерпретатор браузера не воспринимал их как команды, эти служебные символы заменяют на ESC-последовательности

Синтаксис ESC последовательности:

&# код символа; (&#035 – значок #)

& служебное слово;

## Символ

## Служебное слово

<

&lt;

>

&gt;

&

amp;

“

&quot;

©

&copy;

@

&#064;

неразрывный пробел

&nbsp;

# Теги для подключения нетекстовых ресурсов

рисунки

объекты

апплеты Java

```

```

*Атрибуты:*

alt="текстовое описание рисунка"

align=top || middle || bottom – *выравнивание*

title = "всплывающее текстовое описание"

**Для вставки аудио и видео файлов:**

```
<a href="URL аудио или видео файла"> текст </A>
```

*картинку можно использовать как маршрутизатор гиперссылки:*

```
<a href="URL" > </a>
```

# Объекты

для встраивания объектов в веб-документы применяются теги:

`<embed>`

`<noembed>`

`<applet>`

`<object>`

`<param>`

# Контейнер <embed>



```
<embed width="..." height="..."> </embed>
```

**height** - высота объекта

**width** - ширина объекта

**align** - выравнивание

**hidden** - указывает, скрывать ли объект на странице

**pluginspace** - адрес страницы, откуда можно скачать и установить плагин к браузеру

**src** - путь к файлу

**type** - MIME-тип объекта

**vspace** - вертикальный отступ от объекта до окружающего контента

**hspace** - горизонтальный отступ от объекта до окружающего контента



# Контейнер <object>

<object> ... </object>

**height** - высота объекта

**width** - ширина объекта

**align** - выравнивание

**classid** - адрес программы (приложения или плагина), которая работает с данным объектом, и будет запускать его

**code** - имя объекта для его выполнения

**codebase** - путь к папке с объектом, который указан атрибутом *code* или *classid*

**codetype** - указывает на тип объекта, который задан атрибутом *classid*

**data** - адрес файла для его отображения в окне браузера

**tabindex** - определяет порядок перехода между элементами с помощью клавиши *Tab*

**type** - MIME-тип объекта

**vspace** - вертикальный отступ от объекта до окружающего контента

**hspace** - горизонтальный отступ от объекта до окружающего контента

Внутри <object> находятся теги

<param name="" value="">

*позволяют передать объекту*

*некоторые значения*

*Пример:* <param name="pole1" value="12345">

## Пример вставки объекта

```
<html><head>  
<title>Ter EMBED</title>  
</head>  
<body>  
  <embed  
    src='heart.swf'  
    width='200'  
    height='200'  
    type='application/x-shockwave-flash'  
    pluginspace='http://www.macromedia.com/go/getflashplayer'  
  </embed>  
</body>  
</html>
```

[просмотр примера](#)

# Фреймы

Позволяют разбивать Web-документ на кадры-подокна

Свойства фреймов:

Каждый фрейм имеет свой собственный URL

Каждый фрейм имеет свое собственное имя (name), что позволяет переходить из одного фрейма в другой

Размер фрейма может изменяться пользователем, если это не запрещено

Фреймы позволяют создавать интерфейсные решения:

- размещение статической информации (кнопки, пункты меню) в одном фрейме, а динамическую – в другом)
- создание окон запросов (в одном окне запрос, в другом – его результаты)
- создание форм для обращения к базам данных

Фрейм-документ является специфическим HTML-файлом:  
не имеет тега <body>

# Структура фреймового документа

```
<HTML>  
<HEAD> ... </HEAD>  
<FRAMESET>  
    КОНТЕНТ  
</ FRAMESET >
```

## Задание фреймовых структур

```
<frameset rows =“ряды”  
          cols =“столбцы”>
```

*Пример:*

```
<frameset cols="100, *">
```

- делит экран на два фрейма, один шириной 100 px.

### Способы задания размеров фрейма:

- пиксельный;
- с использованием заменяющего символа '\*';
- с использованием %

Атрибуты тега **frameset**:

**border** – *толщина окаймляющей линии*

**frameborder** – *толщина разграничительной линии*

Внутри <frameset> можно разместить нужное количество тегов <frame>

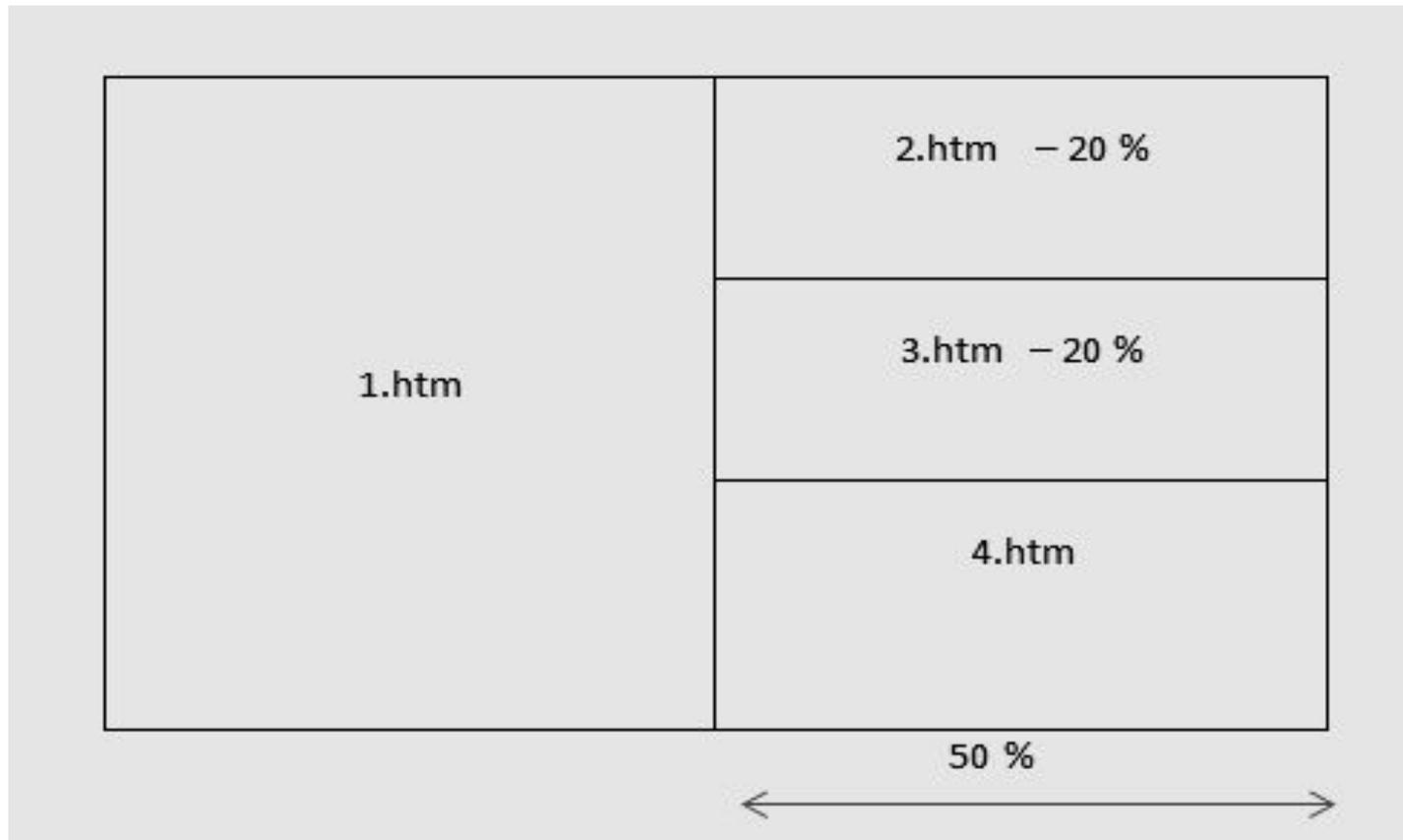
<frame

src='url документа, который должен загружаться в этот фрейм'

name='MyFrame'>

атрибут *name* может принимать собственное имя фрейма, чтобы указать, какому фрейму использовать ссылку

## Пример задания фрейма



```
<frameset cols="*, 50%">  
<frame src="1.htm" name="my1">  
  <frameset rows="20%, 20%, *">  
<frame src="2.htm" name="my2">  
<frame src="3.htm" name="my3">  
<frame src="4.htm" name="my4">  
  </frameset>  
</frameset>
```

если браузер не поддерживает фреймы,  
то используется контейнер  
<noframes>текст документа</noframes>

## Дополнительные атрибуты тега <frame>

<frame src="1.htm"

**scrolling**="yes | no | auto" – управление скроллингом

**noresize** – запрещает изменение размера

**marginwidth** – ширина боковых разделительных полос между фреймами

**marginheight** – ширина верхних и нижних разделительных полос

## Особенности применения атрибута *target*

Первый фрейм – главный, каждая ссылка должна содержать атрибут, который позволяет вернуться к главной странице

Если собственные имена фреймов начинаются с символа '\_' – это целевые фреймы:

**\_blank** – документ будет появляться в новом пустом окне

**\_self** – новая страница будет появляться в том же окне

**\_parent** – все документы, вызываемые по этой ссылке, будут появляться в родительской фреймовой структуре

**\_top** – документы появляются в отдельном нефреймовом окне, для просмотра будет сгенерировано новое окно

*Вложенность фреймов затрудняет навигацию*

# <iframe>

плавающий фрейм, находится внутри документа,  
позволяет загружать в область заданных размеров другие документы.

## Свойства:

содержание игнорируется браузерами, не поддерживающими данный тег.  
Для таких браузеров - альтернативный текст внутри <iframe> и </iframe>

## А т р и б у т ы

**align** - определяет как фрейм будет выравниваться по краю, и способ обтекания его текстом

### **allowtransparency**

устанавливает прозрачный фон фрейма

**frameborder** - устанавливает, отображать границу вокруг фрейма или нет

**height** - высота фрейма

**width** - ширина фрейма

# <iframe> (2)

**hspace** - горизонтальный отступ от фрейма до окружающего контента

**marginheight** - отступ сверху и снизу от содержания до границы фрейма

**marginwidth** - отступ слева и справа от содержания до границы фрейма

**name** - имя фрейма

**sandbox** - ограничения на контент, загружаемый во фрейме

**scrolling** - способ отображения полосы прокрутки во фрейме

**seamless** - определяет, что содержимое фрейма должно отображаться так, будто является частью документа

**src** - путь к файлу, содержимое которого будет загружаться во фрейм

**srcdoc** - хранит содержимое фрейма непосредственно в атрибуте

**vspace** - вертикальный отступ от фрейма до окружающего контента

# Пример

```
<html>  
<head><title>пример с IFRAME</title>  
</head>  
  
<body>  
<iframe src= " id.htm"  
  width="468" height="400"  
  align="right" frameborder="0">  
  Ваш браузер не поддерживает IFRAME  
</iframe>  
  
<div style='background:gray  
....  
</body>  
</html>
```

[просмотр примера](#)

## Мета теги

– это служебные теги,  
которые предназначены не для обработки  
конкретного контента,  
а предназначены для управления документом  
в целом

Мета должны находиться в заголовочной части

## 2 типа мета-тегов

```
<meta name='имя' content='значение'>
```

```
<meta http-equiv='имя_http_заголовка'  
content='значение'>
```

*Пример:*

```
<meta name="description" content="This is my document">
```

Мета теги полезны, если в документе мало текста,  
главная страница сделана в виде скрипта,  
главная страница является фреймсетом и т.п.

## *Дата устаревания документа (для управления кешированием)*

```
<meta http-equiv="expires"  
      content="Mon, 02 Jan 2012 00:00:01GMT">
```

Если content="0", то обновлять нужно немедленно

## *Тип содержимого страницы*

```
<meta http-equiv="content_type"  
      content="MIME_TYPE; charset=кодировка">
```

# Стандарт MIME

«многоцелевое интернет-почтовые расширения»

Формируется пользовательскими почтовыми агентами

## Синтаксис MIME

тип/подтип; параметр=значение;...

Например: text/html  
image/jpeg  
text/plain  
application/msword

Типы и подтипы, которые начинаются с  
X-

не входят в стандарт и называются  
определяемыми пользователем

Например: audio/x-realaudio

*Пример применения типов MIME:*

```
<script type = " text/javascript ">
```

...

```
</script>
```

## *Принудительная перезагрузка документа или автоматическая загрузка любого документа*

```
<meta http-equiv="refresh" content=" t; url=адрес ">
```

## *Определение окна текущей страницы*

```
<meta http-equiv ="window-target" content="top / bottom">
```

## *Управление кешированием*

```
<meta http-equiv ="cache-control"  
content="public / private / no-cache / no-store">
```

## *Контроль уровня взрослости сайта (платформонезависимая схема рейтинга)*

```
<meta http-equiv ="pics-label" content="violence / sex">
```

```
<meta name="keywords"  
      content="слово, слово, слово...">
```

*для поисковых систем (индексирования)  
Максимально – 1000 символов*

```
<meta name="generator" content="MS Notepad">
```

```
<meta name="copyright"  
      content="авторские_права">
```

```
<meta name="resource-type" content="document">
```

# Пример web-документа с мета-тегами

```
<html>  
<head>  
<title> Это заголовок документа</title>  
<meta http-equiv="refresh"  
      content="3; url=2.htm">  
</head>  
<body>  
<h1><center>Через Три секунды произойдет загрузка  
      документа 2.htm  
</center></h1>  
</body>  
</html>
```

РЕЗУЛЬТАТ РАБОРЕЗУЛЬТАТ РАБОТЫ  
ДОКУМЕНТА

# Изображения-карты (map)

Изображения-карты –

это графические объекты в web-документе, отдельные части которых являются маршрутизаторами гиперссылок (чувствительные области)

```
<map name="myname"> // рекомендуется
```

```
помещать в HEAD
```

```
<area
```

```
    shape=
```

```
    coords=
```

```
    href="url" | nohref>
```

```
</area ...>
```

```
</map>
```

**shape=circle | rect | poly**

**coords="x1, y1, x2, y2"** – для прямоугольника (*rect*)

"x, y, r" – для окружности (*circle*)

"x1, y1, x2, y2, ..., xn, yn, x1, y1" – для

*произвольной ломаной линии*

В том месте, где помещается карта:

```

```

## Пример маппинга

```
<body ><center>  
 </center>  
  
<map name="мумап">  
  <area shape="rect" coords="352,77,455,208" href="center.html"  
    target="_blank" alt="Центральная часть Украины">  
  
  <area shape="rect" coords="98,77,337,198" href="west.html"  
    target="_blank" alt="Западная часть Украины">  
  
  <area shape="rect" coords="597,128,745,215" href="ost.html"  
    target="_blank" alt="Восточная часть Украины">  
  
  <area shape="rect" coords="485,280,580,377" href="south.html"  
    target="_blank" alt="Южная часть - Крым">  
</map></body></html>
```

[просмотр примера](#)

## Формуляры

Формуляр – это специальный объект, с помощью которого клиент может передавать информацию с web-страницы к какому – либо обработчику сервера

Для обработки данных, которые передаются с формуляра, используются CGI- скрипты

## Структура формуляра

`<form attribute=value...> </form>`

`<form action = "url" //местонахождение обработчика формы  
CGI-скрипта)`

`method=get | post //метод отсылки данных из формы в  
CGI-скрипт (по умолчанию - get)`

`enctype= //указывает способ кодировки данных`

`name="MyForm1" //произвольное имя формы`

## Поля формы

поля формы размещаются внутри `<form>`

элементы ввода предназначены для забора данных

```
<input type="тип" value='значение'  
        name='имя' size='размер'  
        checked='checked'>
```

## 1) Кнопка Submit

`<input type="submit">` – создает объект "кнопка"

Нажатие на кнопку (событие `onClick`) приводит к отсылке данных из формуляра

## 2) Текстовое поле

`<input type="text"`

`name="имя переменной"`  
`умолчанию"`

`size="10"`

`maxlength="100">`

`value="значение по`

`size` – длина видимой части строки

### 3) Поле для ввода пароля

```
<input type="password"           name=  
    "fio"                         =  
    value="Иванов">
```

*небезопасно!*

### 4) Многострочный текст

```
<input type="textarea"  
    rows= "20"  
    cols= "100">
```

## 5) Радиокнопка

```
<input type="radio" name="color" value="red"
checked='checked'>
```

Пример:

```
<p>Выберите цвет:<br>
```

```
<input type="radio" name="color" value="red">
```

Красный<br>

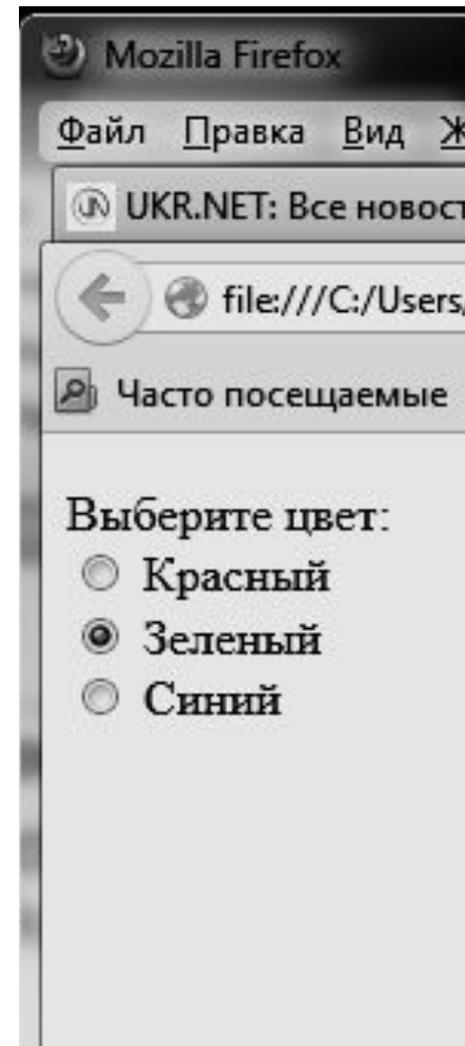
```
<input type="radio" name="color" value="green"
checked='checked'>
```

Зеленый<br>

```
<input type="radio" name="color" value="blue">
```

Синий

```
</p>
```



## 6) Чекбокс

```
< input type="checkbox" name="color" value="red"  
checked='checked'>
```

## 7) Скрытый элемент ввода

```
< input type="hidden" name="predmet"  
value="mathematic">
```

## 8) Кнопка сброса всех полей формы

```
< input type="reset" value="надпись на кнопке">
```

## 9) Объект SELECT (выпадающий список)

```
<select name = "myname">
```

```
<option value = "значение 1">название опции
```

```
<option value = "значение 2">название опции
```

```
...
```

```
</select>
```

Тег `<fieldset>` для группировки элементов в форме  
– создает рамку вокруг элементов формы

Тег `<legend>` - создает заголовок для элементов  
формы, объединенных в группу тегом `<fieldset>`

```
<fieldset>
```

```
<legend>Заполните форму:</legend>
```

```
<form action="mailto: имя @ адрес"  
method= "post"  
enctype="text/plain">
```

```
...
```

```
</form>
```

```
</fieldset>
```



# Пример формы

```
<fieldset> <legend>Заполните форму:</legend>
```

```
<form name =“demoform“
```

```
    action="mailto:someaddress@kture.kharkov.ua"
```

```
    method="post" encoded="url-encoded">
```

```
<h2>Расскажите немного о себе...</h2>
```

```
<p>Указывать подлинные данные совсем не обязательно </p>
```

```
<p>  Имя: < input type="text" size="40" name="fn"><br>
```

```
  Фамилия: < input type="text" size="40" name="ln"><br>
```

```
  Пол: < input type="radio" name="gender: value="male" checked>мужской
```

```
<input type="radio" name="gender" value="female">женский<br>
```

```
  Возраст: <input type="text" size="5" name="age"> лет<br> </p>
```

```
<h3>Выберите род <br> Ваших занятий:</h3>
```

```
<select multiple name="application">
```

```
<option value="Информационные технологии1" checked>Информационные технологии
```

```
<option value="Программирование" checked>Программирование
```

```
<option value="Спор" checked> Спорт
```

```
</select> <br>
```

```
< input type="hidden" name="author" value= "somebody" ><hr>
```

```
< input type="button" value="Test" onClick="formTest()">
```

```
< input type="reset" value="Очистить поля формы">
```

```
< input type="submit" value="Отправить заполненную форму" onClick="return SendForm()">
```

```
</form>
```

```
</fieldset>
```



## Просмотр работы формы