

Интернет-технологии



Сторожев Дмитрий Алексеевич,
старший преподаватель
кафедры экономической
информатики БГУИР, магистр
экономических наук,
исследователь технических наук

ПЛАН (лекция 3 в рамках вебинара 1)

1. Некоторые элементы HTML
2. Новинки HTML5
3. Введение в CSS
4. Немного о JavaScript



HTML формы

Формы являются одним из важных элементов любого сайта и предназначены для обмена данными между пользователем и сервером.

Любая форма характеризуется следующими параметрами.

- ❑ Элементы формы, которые представляют собой стандартные поля для ввода информации.
- ❑ Кнопку отправки данных формы на сервер (кнопка SUBMIT).
- ❑ Адрес программы на веб-сервере, которая будет обрабатывать содержимое данных формы.

```
<form name="forma1">  
....  
</form>
```

Параметры тега `<form>`:

`name` - имя формы. Необходимо, если на странице несколько форм

`action` - определяет URL-адрес, по которому будет отправлена информация введенная пользователем

`method` - определяет способ отправки информации

`target` - указывает имя окна, в котором будут отображаться результаты обработки отправленной формы

HTML формы

HTML формы могут содержать такие элементы ввода как:

- A. Текстовые поля
- B. Флажки
- C. Радио-кнопки
- D. Кнопки отправления и др.
- E. Формы также могут содержать списки выбора, многострочные текстовые поля, метки и др.

Для того, чтобы создать форму в HTML используется элемент **<form>**.

Пример 3.1 (3_1.html)

```
<form>  
<p>Введите ваше имя:</p>  
<input type="text" />  
<p>Введите пароль: </p>  
<input type="password" />  
</form>
```

Введите ваше имя:

Введите пароль:

HTML формы

Элементы ввода

Элементы ввода используются для приема пользовательских данных.

Элементы ввода сильно отличаются друг от друга в зависимости от значения атрибута type.

Текстовое поле

`<input type="text" />` определяет однострочное текстовое поле, в которое пользователь может вводить различную информацию.

Пример 3.2 (3_2.html)

```
<form>
```

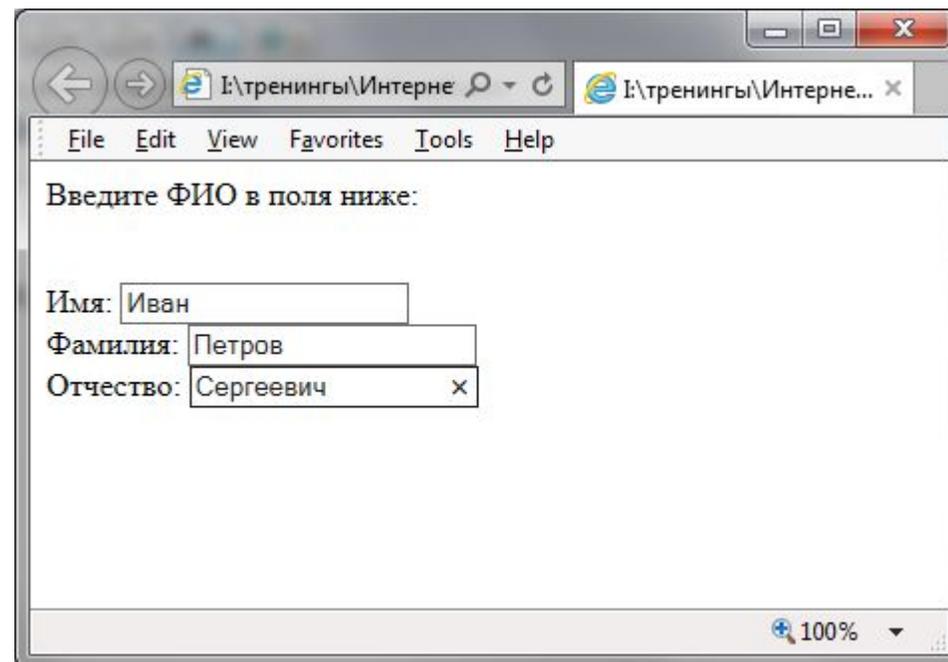
```
<p> Введите ФИО в поля ниже: </p> <br />
```

```
Имя: <input type="text" name="firstname" /><br />
```

```
Фамилия: <input type="text" name="lastname" /><br />
```

```
Отчество: <input type="text" name="lastname" />
```

```
</form>
```



HTML формы

Поле пароля

`<input type="password" />` определяет поле для ввода пароля. Содержимое, вводимое в данное поле, закрывается черными кружками, позволяя вводить пароли даже в присутствии посторонних.

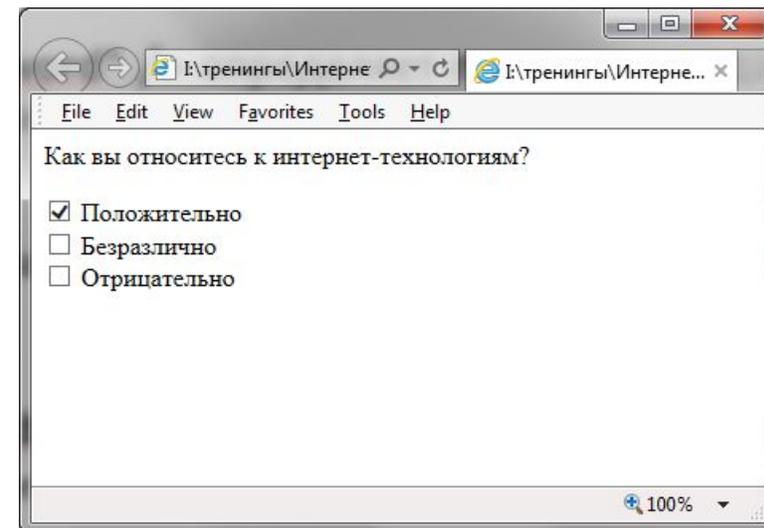
`<form>` Введите пароль: `<input type="password" name="pass" />` `</form>`

Флажок

`<input type="checkbox" />` определяет флажок. Флажки позволяют пользователям выбирать несколько пунктов с предварительно заполненной информацией из группы.

Пример 3.3 (3_3.html)

```
<form>
<p> Как вы относитесь к интернет-технологиям? </p>
<input type="checkbox" name="space" value="1" /> Положительно<br />
<input type="checkbox" name="space" value="2" /> Безразлично <br />
<input type="checkbox" name="space" value="3" /> Отрицательно<br />
</form>
```



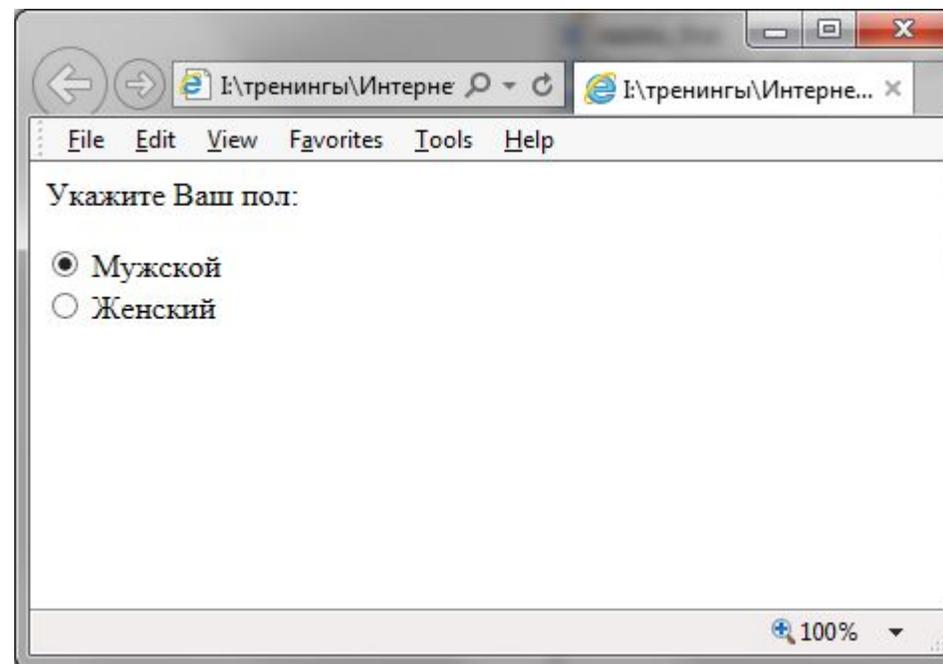
HTML формы

Радио - кнопка

`<input type="radio" />` определяет радио-кнопку. Радио-кнопки позволяют пользователям выбрать только один пункт с предварительно заполненной информацией из группы.

Пример 3.4 (3_4.html)

```
<form>  
<p> Укажите Ваш пол: </p>  
<input type="radio" name="s" value="m" /> Мужской<br />  
<input type="radio" name="s" value="f" /> Женский  
</form>
```



HTML формы

Кнопка отправления

`<input type="submit" />` определяет кнопку отправления. После нажатия на данную кнопку данные, введенные пользователем, будут отправлены на сервер.

Адрес, на который будут пересылаться данные формы, указывается в атрибуте тэга `form` - **action**. Если данный атрибут отсутствует, данные будут отправлены на текущую страницу.

Обратите внимание: обработка данных, полученных сервером в результате отправки форм, будет производиться с помощью серверных языков (таких как PHP, Ruby, Python, .NET, Java).

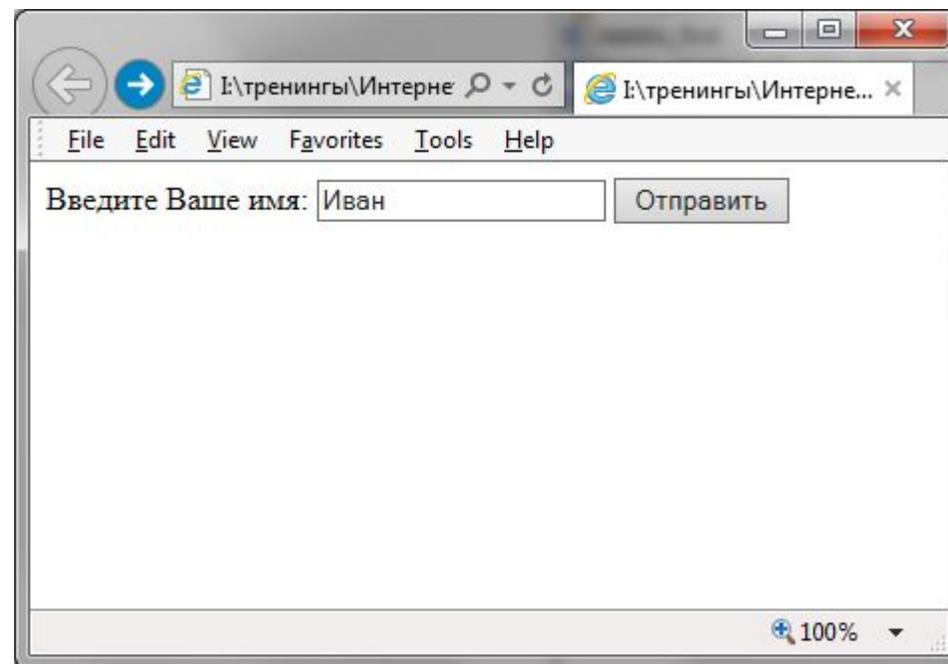
Пример 3.5 (3_5.html)

```
<form name="input" action="form.php" method="get">
```

```
Введите Ваше имя: <input type="text" name="name" />
```

```
<input type="submit" value="Отправить" />
```

```
</form>
```



HTML формы

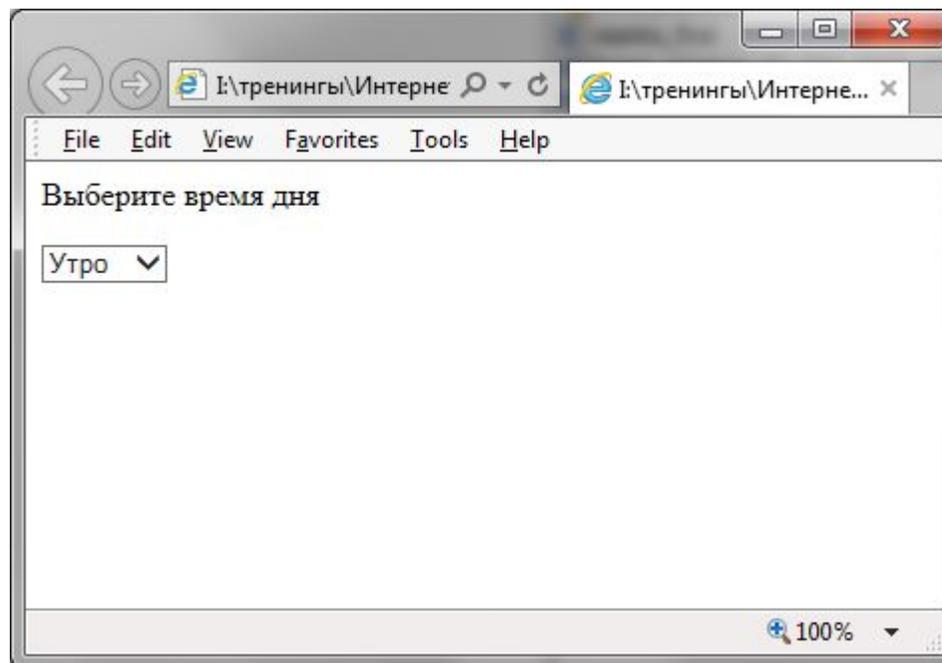
Выпадающий список

Для создания выпадающих списков используется тэг `<select>`, а элементы выпадающего списка определяются с помощью тэга `<option>`.

Обратите внимание: пользователи не любят вводить данные вручную, поэтому если Вы можете заменить обычное текстовое поле выпадающим списком, радио кнопками, или флажками, то обязательно сделайте это.

Пример 3.6 (3_6.html)

```
<p> Выберите время дня</p>
<form>
<select name="time" >
<option value="u"> Утро </option>
<option value="o"> Обед </option>
<option value="v"> Вечер </option>
</select>
</form>
```



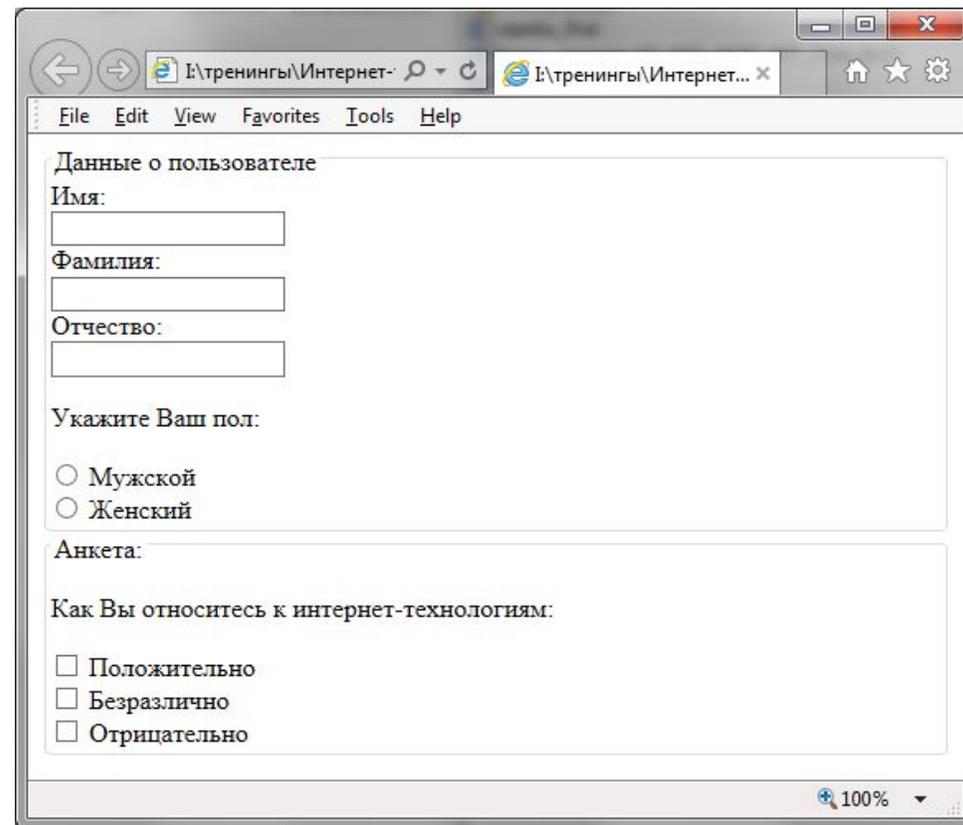
HTML формы

Заголовки в формах

Для того, чтобы озаглавить группу элементов формы, Вы должны с помощью тэга `<fieldset>` сгруппировать желаемую часть формы и затем с помощью тэга `<legend>` установить желаемое заглавие.

Пример 3.7 (3_7.html)

```
<form>
<fieldset>
<legend>Данные о пользователе</legend>
Имя: <br /><input type="text" name="firstname" /><br />
Фамилия:<br /> <input type="text" name="lastname" /><br />
Отчество: <br /><input type="text" name="lastname" /><br />
<p>Укажите Ваш пол:</p>
<input type="radio" name="s" value="m" /> Мужской<br />
<input type="radio" name="s" value="f" /> Женский
</fieldset>
<fieldset>
<legend> Анкета: </legend>
<p>Как Вы относитесь к интернет-технологиям:</p>
<input type="checkbox" name="space" value="1" /> Положительно<br />
<input type="checkbox" name="space" value="2" /> Безразлично<br />
<input type="checkbox" name="space" value="3" /> Отрицательно<br /> </fieldset>
</form>
```

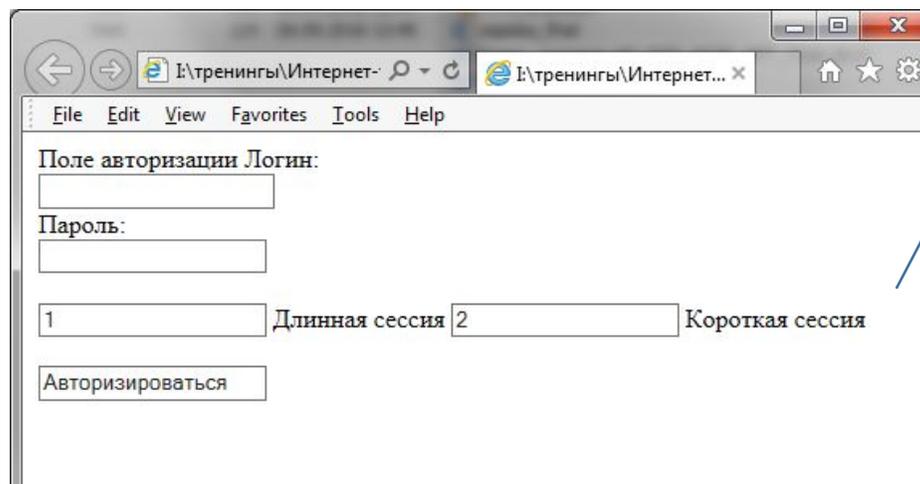


HTML формы

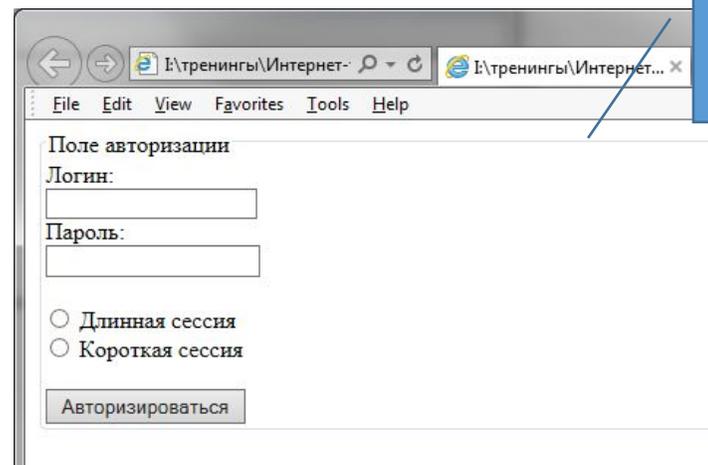
Самостоятельно (код 3_8_error.html)!

Задание. Исправьте ошибки в коде. После исправления ошибок форма должна корректно отображаться в редакторе:

```
<html>
<body>
<form>
<fieldset>
<legend>Поле авторизации
Логин: <br /><input type='password' name='login' /><br />
Пароль:<br /> <input type='text' name='pas' /><br /><br />
<input type='checkbox' name='session' value='1' />
Длинная сессия
<input type='checkbox' name='session' value='2' />
Короткая сессия<br /><br />
<input type='button' value="Авторизироваться" /> </fieldset>
</form>
</body>
</html>
```



Ошибка
а



Правильно
о

HTML фреймы

HTML фреймы

Обратите внимание: фреймы на данный момент считаются устаревшей технологией и не поддерживаются в HTML5 (это относится только к обычным фреймам, строковые фреймы, рассмотренные далее, не считаются устаревшими).

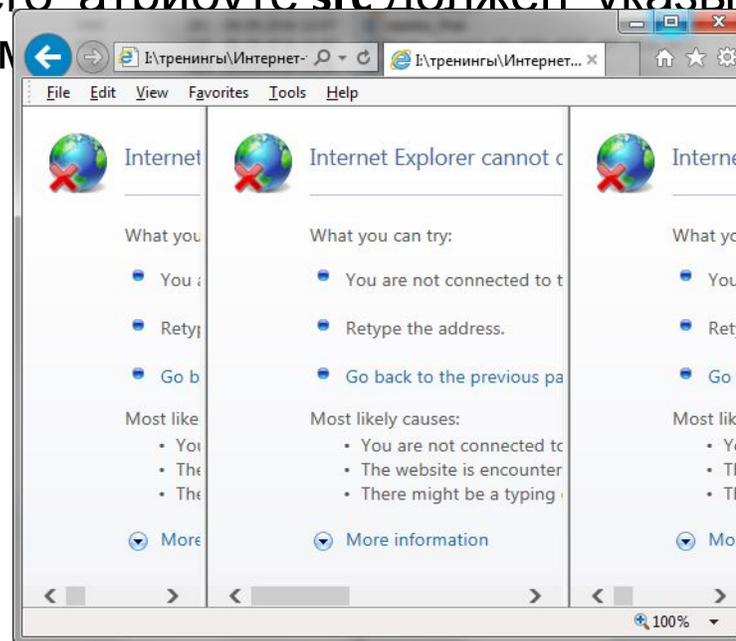
В одном окне браузера может быть отображено несколько веб-страниц, которые в данном случае будут называться **фреймами**.

С помощью HTML тэга `<frameset>` Вы можете описать, как будут располагаться фреймы в окне браузера.

Тэг `<frame>` описывает один отдельный фрейм. В его атрибуте `src` должен указываться адрес документа, который будет отображен в данном фрейме.

Пример 2.0 (2.0.html)

```
<html>
<frameset cols="25%,50%,25%">
<frame src="fr_a.html" />
<frame src="fr_b.html" />
<frame src="fr_c.html" />
</frameset>
</html>
```



HTML фреймы

Размер фреймов

Размеры фреймов устанавливаются с помощью атрибутов тэга <frameset>. Размеры могут задаваться в пикселях (px) и процентах (%).

С помощью атрибута **rows** Вы можете установить высоту фрейма. Размеры для отдельных фреймов должны отделяться запятой.

```
<frameset rows="35%,35%,30%">  
<frame src="fr_c.html" />  
<frame src="fr_a.html" />  
<frame src="fr_b.html" />  
</frameset>
```

С помощью атрибута **cols** Вы можете задать ширину отдельного фрейма.

```
<frameset cols="25%,75%">  
<frame src="fr_a.html" />  
<frame src="fr_b.html" />  
</frameset>
```

Обратите внимание: фрейм по умолчанию имеет видимую границу. Это значит, что пользователь может изменять его размер, перетаскивая границу. Чтобы запретить перетаскивание границ, Вы можете добавить атрибут **noresize** со значением "noresize" к тэгу <frame>.

HTML фреймы

Что делать, если браузер пользователя не поддерживает фреймы?

Если браузер пользователя не поддерживает фреймы, Вы должны использовать тэг `<noframes>`. Значение этого тэга будет отображено только тем пользователям, браузер которых не может отобразить фреймы.

Пример 3.10 (3_10.html)

```
<frameset cols="25%,75%">
<frame src="fr_a.html" />
<frame src="fr_b.html" />
</frameset>
<noframes>
<body>
<p> Ваш браузер не поддерживает фреймы. </p>
</body>
</noframes>
```

HTML фреймы

Строковые фреймы

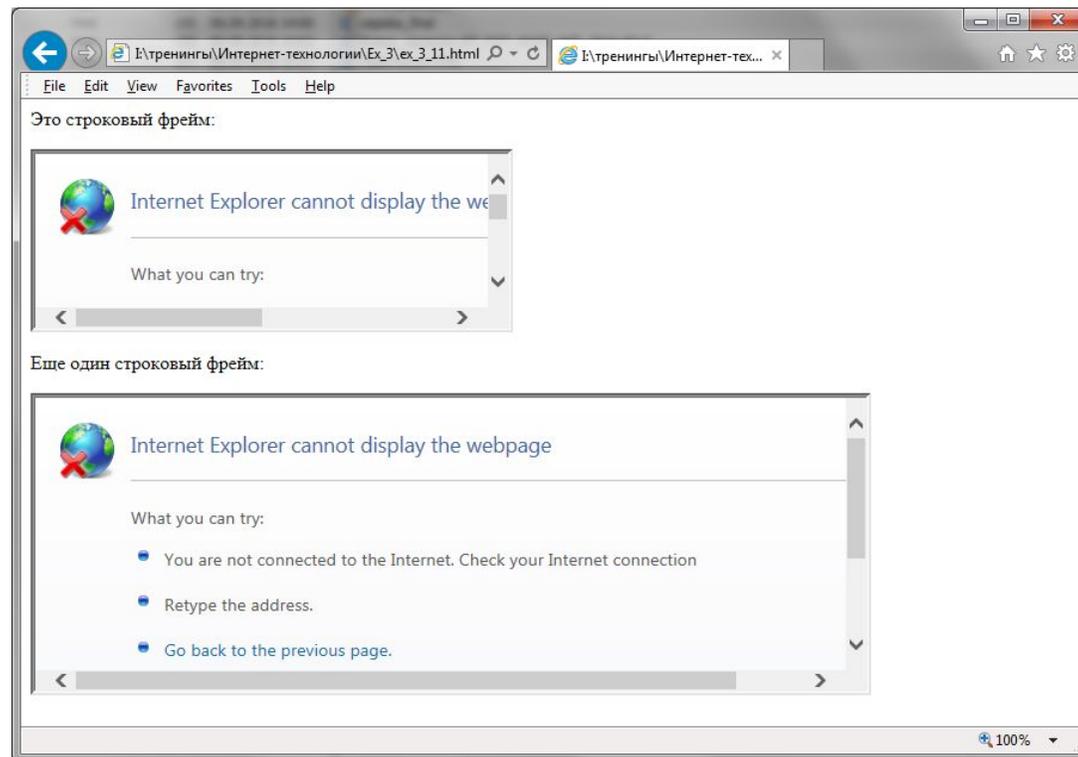
С помощью тэга `<iframe>` Вы можете вставить фрейм в любое место обычного HTML документа.

Данный тэг часто используют для отображения рекламы на сайтах.

Ширина строкового фрейма задается с помощью атрибута **width**, а высоту с помощью атрибута **height**.

Пример 3.11 (3_11.html)

```
<html>  
<body>  
<p> Это строковый фрейм: </p>  
<iframe src='fr_a.html' width='400' height='150'></iframe>  
<p> Еще один строковый фрейм:</p>  
<iframe src='fr_b.html' width='700' height='250'></iframe>  
</body>  
</html>
```



DTD

DTD расшифровывается **D**ocument **T**ype **D**efinition (объявление типа документа).

DTD - это не HTML тэг, а инструкция браузеру о версии языка разметки данной страницы. DTD указывается перед тэгом <html>.

DTD нужен для того, чтобы браузер правильно понимал, как отображать HTML документ. В HTML существует несколько видов DTD:

HTML 4.01 Strict (строгий)

HTML документы со строгим DTD могут содержать все HTML элементы и атрибуты, кроме презентационных и устаревших. Использование фреймов запрещено.

Объявление строгого DTD:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Обратите внимание: вместо презентационных тэгов сейчас рекомендуется использовать CSS.

HTML 4.01 Transitional (переходный)

HTML документы с переходным DTD могут содержать все HTML элементы и атрибуты, включая презентационные и устаревшие. Использование фреймов запрещено.

Объявление переходного DTD:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

DTD

HTML 4.01 Frameset (фреймовый)

HTML документы с фреймовым DTD могут содержать все HTML элементы и атрибуты включая презентационные и устаревшие. Использование фреймов разрешено.

Объявление фреймового DTD:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

HTML 5 Doctype

В HTML5 вместо трех различных Doctype был введен один универсальный.

```
<!DOCTYPE html>
```

Как проверить свои страницы на наличие ошибок разметки?

Обратите внимание: DTD должен быть указан.

<https://validator.w3.org/>

HTML Head

Секция head

Элементы, располагающиеся в секции head, не отображаются явно на странице и используются для служебных функций.

В секции head могут располагаться скрипты, инструкции об оформлении страницы и различная мета информация о данном HTML документе.

Следующие элементы должны обязательно располагаться в секции head: <link>, <title>, <base>, <style>, <script> и <meta>.

Элемент title

С помощью элемента <title> Вы можете задать заголовок HTML документа. Элемент title обязательно должен присутствовать во всех HTML документах.

Элемент title:

Определяет заголовок окна браузера;

Используется как заголовок страницы в результатах выдачи поисковых систем;

Используется как заголовок страницы при добавлении сайта в избранное.

```
<head>
```

```
<title> Интернет-технологии - курс по HTML, CSS, JavaScript</title>
```

```
</head>
```

HTML Head

Элемент meta

Метаданные - это информация о данных, находящихся в HTML документе. Пример метаданных: кодировка страницы, краткое описание содержимого, ключевые слова, имя автора, дата последней модификации.

Метаданные не отображаются явно на странице, но используются браузерами и поисковыми системами.

В HTML метаданные HTML документов определяются с помощью тэга `<meta>`. Тэг `<meta>` всегда должен располагаться в секции `head`.

О кодировке

Для того, чтобы указать браузеру пользователя какая кодировка используется на данной странице, необходимо использовать атрибут `charset` тэга `meta`.

```
<!-- Укажем браузеру, что текст данного HTML документа набран в типичной для windows кодировке windows-1251 -->
```

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
```

```
<!-- Укажем браузеру, что текст данного HTML документа набран во всемирной кодировке UTF-8 -->
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

Обратите внимание: если явно не указать кодировку, браузер при отображении страницы будет определять ее автоматически. Если кодировка при этом будет определена неверно, пользователь увидит страницу, содержащую бессмысленные символы, поэтому кодировка обязательно должна указываться к каждому HTML документу.

Обратите внимание: в HTML5 кодировка может быть задана с помощью нового атрибута `charset`, например, так `<meta charset="UTF-8">`.

HTML Head

Метаэлементы и поисковые системы

Некоторые поисковые системы во время индексации страницы обращаются к мета элементам.

Например, мета элемент ниже определяет описание для HTML документа (данное описание может использоваться поисковыми системами при отображении документа в поисковой выдаче):

```
<meta name="description" content="Бесплатные онлайн учебники по HTML, CSS, JavaScript" />
```

Следующий мета элемент задает ключевые слова для документа:

```
<meta name="keywords" content="HTML, XHTML, CSS, JavaScript, jQuery, HDOM, AJAX" />
```

Обратите внимание: в связи с тем, что многие веб-мастера для поднятия рейтинга выдачи сайта указывали огромное количество ключевых слов, не имеющих отношения к их сайту, но пользующихся популярностью, поисковые системы были вынуждены прекратить использование keywords для определения ключевых слов страницы.

HTML Head

Элемент link

С помощью тэга **<link>** Вы можете связать HTML документ с внешним CSS файлом.

```
<head> <link rel="stylesheet" type="text/css" href="style.css" /> </head>
```

Элемент base

С помощью тэга **<base>** Вы можете задать корневой адрес (адрес, который будет подставляться к относительным ссылкам) и способ открытия ссылок по умолчанию.

Атрибут href используется для задания корневого адреса по умолчанию. По умолчанию корневой адрес равен доменному имени сайта.

```
<html>
<head>
<base href="http://www.bsuir.by/link/" />
</head>
<body>
<p>После щелчка на ссылку ниже Вы будете перемещены на документ, находящийся по адресу
"http://www.bsuir.by/link/example.html"</p>
<p>Если бы тэг base не был определен, мы бы были перемещены на документ
"http://www.bsuir.by/example.html".</p>
<a href="example.html">Ссылка на example.html</a>
</body>
</html>
```

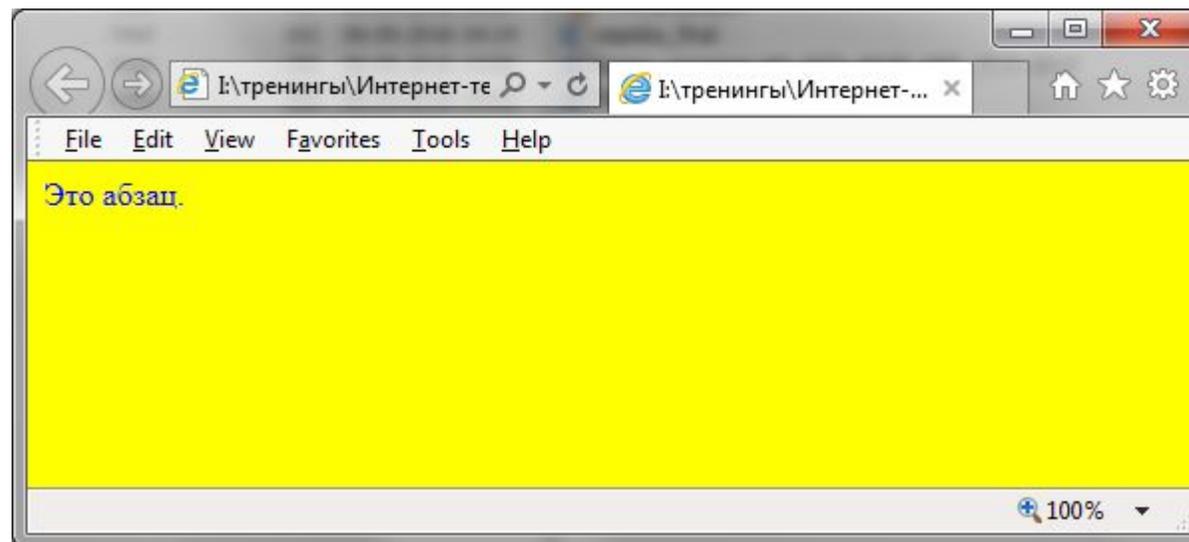
HTML Head

Элемент style

Элемент `<style>` используется для оформления HTML документов.

Пример 3.12 (3_12.html)

```
<html>
<head>
<style type="text/css">
body {background-color:yellow}
p {color:blue}
</style>
</head>
<body>
<p> Это абзац. </p>
</body>
</html>
```



Обратите внимание: в HTML5 атрибут `type` тэга `style` стал необязательным.

HTML скрипты

Элемент script

HTML тэг `<script>` используется для определения скриптов, выполняющихся на стороне клиента.

JavaScript является самым популярным языком программирования этого типа.

JavaScript используется для создания веб-приложений и динамических сайтов, способных взаимодействовать с пользователем.

Тэг `script` может либо содержать непосредственно код скрипта, либо ссылаться на внешний скриптовый файл через атрибут `src`.

Обратите внимание: в HTML5 атрибут `type` стал необязательным.

Пример 3.13 (3_13.html)

`<p>`Нажмите на кнопку ниже, чтобы заданный скрипт выполнился. `</p>`

`<input type='button' value='Нажмите на меня' onclick='showMessage()' />`

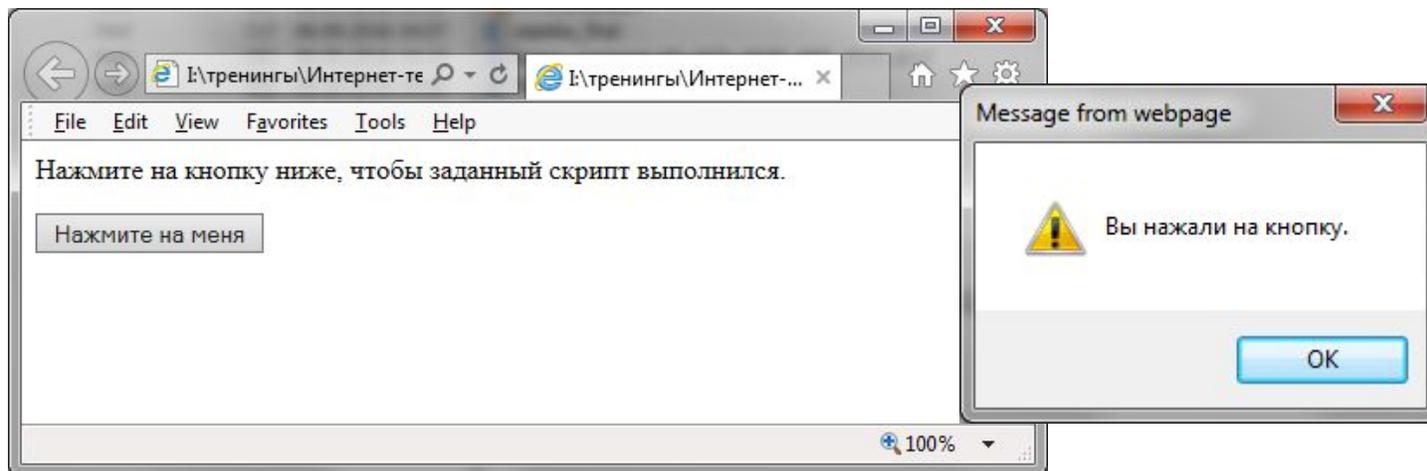
`<script type="text/javascript">`

`function showMessage() {`

`alert('Вы нажали на кнопку.')`

`}`

`</script>`



HTML скрипты

В следующем примере элемент `<script>` ссылается на внешний скриптовый файл, выводящий сообщение "Привет, привет!":

Пример 3.14 (3_14.html)

```
<html>
<body>
<script type='text/javascript' src='hello.js'></script>
</body>
</html>
```

+ смотрим файл hello.js

+ посмотреть "все в одном файле – 3_15.html"

