

Мировые информационные ресурсы

Преподаватель:

*Попков Сергей Игоревич,
аспирант,
заведующий лабораторией*

(rsteach0@gmail.com)

Лекция 1. HTML

Понятие HTML

- HTML - “Hyper Text Markup Language”
- HTML описывает структуру веб-страниц с помощью разметки
- Элементы HTML – блоки для создания веб-страниц
- Элементы HTML представляются тэгами
- Тэги HTML размечают элементы содержимого: "heading", "paragraph", "table", и т.д.
- Предназначение веб-браузеров (Chrome, IE, Firefox, Safari) заключается в отображении и интерпретации HTML-документов
- Браузер не отображает тэги HTML, а использует их для определения принципа отображения документа и построения содержимого страницы
- Объявление `<!DOCTYPE html>` представляет собой тип HTML-документа, помогая браузерам отображать веб-страницы корректно. Должен находиться в самом начале страницы (до любых тэгов)

Пример разметки

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Page Title</title>
```

```
  </head>
```

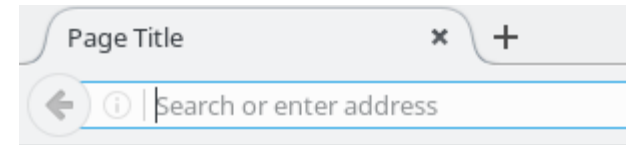
```
  <body>
```

```
    <h1> Heading, level 1 </h1> <!--видимые элементы-->
```

```
    <p> Paragraph </p>
```

```
  </body>
```

```
</html>
```



Heading, level 1

Paragraph

Элементы HTML

- Элемент HTML обычно состоит из открывающего и закрывающего тэга с содержимым между ними:
 - `<tagname>Content goes here...</tagname>`
- Элемент HTML представляет собой все, что находится в документе от открывающего до закрывающего тэга
- Элементы HTML можно располагать друг в друге
- Все документы HTML состоят из элементов HTML, вложенных друг в друга
- Пустыми называются элементы HTML без содержимого
- Примером пустого элемента без закрывающего тэга можно считать `
` (`
` указывает на разрыв строки)
- Пустые элементы можно закрывать внутри открывающего тэга: `
`
- Тэги HTML не чувствительны к регистру: `<P>` значит то же самое, что и `<p>`. Настоятельно рекомендуется стандартом всегда использовать только нижний регистр: `<p>`

Атрибуты элементов HTML

- Все элементы HTML могут обладать атрибутами
- Атрибуты предоставляют дополнительную информацию об элементе
- Атрибуты всегда определяются в открывающем тэге
- Как правило, атрибуты представляются в виде пары имя-значение, например: `name="value"`
- Пример: Значение атрибута `"title"` будет отображено в качестве всплывающей подсказки при наведении курсора мыши на параграф:
 - `<p title="Tooltip">Paragraph</p>`
- Стандарт HTML5 не требует написания атрибутов в нижнем регистре, но рекомендуется всегда использовать именно нижний регистр при указании имени атрибута
- Весьма важно закрывать значения атрибутов в кавычки, одинарные (") либо двойные (")

Paragraph

Tooltip

Примеры распространенных тэгов

- Заголовки определяются тэгами `<h1>` - `<h6>`
- `<h1>` определяет самый крупный заголовок, `<h6>` - наименее важный
- Поисковые движки используют заголовки для индексации структуры и содержимого веб-страниц
- Скользя взглядом по странице, пользователи уделяют внимание именно заголовкам. Поэтому весьма важно использовать заголовки для демонстрации структуры документа
- Тэг `<hr>` определяет тематическое разделение на странице HTML, его наиболее частое графическое отображение – горизонтальная прямая
- Элемент `<hr>` используется для отделения содержимого (или подчеркивания некоторых внесенных изменений) на странице HTML
- Элемент HTML `<p>` определяет параграф
- Элемент HTML `<pre>` определяет текст с заведомо заданным форматированием
- Текст внутри элемента `<pre>` отображается с помощью шрифта фиксированной ширины (обычно Courier) с сохранением разрывов строк, пробелов и табуляций

Тэги для форматирования

- Форматирование элементов предназначается для отображения различных, особых видов текста:

- `` - Bold text (жирный) **Bold text** **Important text** *Italic text* *Emphasized text* **Marked text**
- `` - Important text (контекстно значимый) Small text ~~Deleted text~~ Inserted text _{Subscript text} ^{Superscript text}
- `<i>` - Italic text (курсив) *Italic text*
- `` - Emphasized text (контекстно выделенный) *Emphasized text*
- `<mark>` - Marked text (выделенный [HTML5]) **Marked text**
- `<small>` - Small text (небольшой) Small text
- `` - Deleted text (вычеркнутый) ~~Deleted text~~
- `<ins>` - Inserted text (подчеркнутый) Inserted text
- `<sub>` - Subscript text (в нижнем индексе) _{Subscript text}
- `<sup>` - Superscript text (в верхнем индексе) ^{Superscript text}

Тэги для цитирования и ссылок

- Элемент `<q>` определяет короткую цитату.
 - Как правило, браузеры вставляют кавычки вокруг содержимого элемента `<q>`
- Элемент `<blockquote>` определяет раздел, процитированный из другого источника
 - Браузеры обычно отделяют отступом содержимое элемента `<blockquote>`
- Иногда вместе с цитированием может быть полезно использовать элемент для перенаправления порядка вывода символов:
 - `<bdo dir="rtl">This text will be written from right to left</bdo> <!-- bi-directional override -->`
tfel ot thqir morf nettirw eb lliw txet sihT
- Ссылки HTML называются гиперссылками. В HTML ссылки определяются тэгом `<a>`
 - Можно воспользоваться ссылкой для перехода к другому HTML-документу
 - При движении над ссылкой курсор меняется на указательный палец (“`cursor:pointer`” в нотации CSS)
 - Ссылка не обязательно задается текстом. Это может быть изображение или любой другой элемент HTML

Атрибут href тэга <a>

- Атрибут href определяет адрес места назначения (Например, “http://www.yandex.ru/”) ссылки
 - `Yandex`
- Текст ссылки – видимая часть тэга <a> (“Yandex”)
- Щелчок по тексту ссылки отправит пользователя по указанному адресу назначения
- В примере выше использовался абсолютный URL-адрес (полный веб-адрес). Локальная ссылка (ссылка на тот же веб-сайт) определяется с помощью относительного URL (без “http://www...”)
- По умолчанию, ссылка имеет следующий вид (во всех браузерах):
 - Ненавешенная ссылка подчеркнута и голуба [Yandex](#)
 - Навешенная ссылка подчеркнута и пурпурна [Yandex](#)
 - Активированная ссылка подчеркнута и красна [Yandex](#)
- Обычная практика – использовать изображения в качестве ссылок (помещая их внутрь тэга <a>)

Атрибут “target” тэга <a>.

Ссылки-закладки

- Атрибут “target” указывает на место открытия документа по ссылке
- Атрибут “target” может иметь одно из следующих значений:
 - `_blank` – открыть в новом окне (или вкладке)
 - `_self` – открыть в том же окне (или вкладке), что и документ, из которого осуществлен переход (значение по умолчанию)
 - `_parent` – открыть в родительском фрейме
 - `_top` – открыть в полном окне браузера
 - `framename` – открыть в фрейме с именем “framename”
- Закладки HTML позволяют пользователям перейти к определенной части веб-страницы. Закладки особенно полезны для длинных страниц (например, лендинговых страниц)
- Чтобы создать закладку, для начала следует создать элемент, к которому она прикреплена (по id). После этого следует создать ссылку для этого элемента. По щелчку на ссылку будет осуществлен переход к местоположению закладки
 - `Yandex Top News Bookmark`
- Сначала создайте элемент:
 - `<h1 id="bookmark1">Bookmarked Header</h1>`
- Затем добавьте ссылку на него на этой же странице:
 - `Visit the Bookmarked Header`

Изображения в HTML

- В HTML изображения определяются тэгом ``
- Тэг `` пустой, содержит только атрибуты, и не имеет закрывающего тэга
- Атрибут `“src”` определяет URL (веб-адрес) изображения
- Атрибут `“alt”` предоставляет альтернативный текст для изображения, если оно, по каким-либо причинам, не может быть отображено (возможные причины: низкая скорость подключения к сети, ошибка в атрибуте `“src”` и др.)
- В случае, если браузер не может найти изображение, пользователю вместо него будет показано значение атрибута `“alt”`
- Атрибут `alt` полагается обязательным. Без него веб-страница не пройдет валидацию успешно
- Вы можете указать ширину и длину изображения в пикселях с помощью атрибутов `“width”` и `“height”` соответственно
- Рекомендуется всегда указывать длину и ширину изображения, чтобы избежать возможного мерцания изображения во время его подгрузки

Карта активных областей на изображении

- Тэг `<map>` служит для определения карты активных областей на изображении, с которыми можно вступить во взаимодействие
- Атрибут `"name"` (имя) тэга `<map>` взаимосвязано с атрибутом `"usemap"` тэга `` для создания связи между соответствующими элементами `` и `<map>`
- Тэг `<map>` содержит несколько тэгов `<area>`, определяющих соответствующие зоны активных областей на связанном изображении ``, с которыми можно вступить во взаимодействие
- Пример:

```

```

```
<map name="areamap1">
```

```
  <area shape="rect" coords="0,0,1,1" alt="Rectangle" href="1.htm">
```

```
  <area shape="circle" coords="0,0,10" alt="Circle" href="2.htm">
```

```
  <area shape="poly" coords="0,0,1,1,2,2" alt="Triangle" href="3.htm">
```

```
</map>
```

Понятие таблицы в HTML

- Таблица HTML определяется тэгом `<table>`
- Каждая строка таблицы определяется тэгом `<tr>`
- Заголовочная строка таблицы задается тэгом `<th>`. По умолчанию, такая строка выделена жирным и выровнена по центру
- Ячейка таблицы и данные в ней определены тэгом `<td>`
- Элементы `<td>` представляют собой хранилища данных для таблицы. Они могут хранить любые элементы HTML, в том числе текст, изображения, списки, другие таблицы и т.д.
- Чтобы добавить заголовок к самой таблице, используется тэг `<caption>` (если используется, должен стоять непосредственно после объявления тэга `<table>`)
- Чтобы ячейка занимала более одного столбца таблицы, следует задать атрибут `“colspan”` со значением, равным количеству занимаемых столбцов
- Чтобы ячейка занимала более одной строки таблицы, следует задать атрибут `“rowspan”` со значением, равным количеству занимаемых строк

Пример обычной таблицы (без рамки)

```
<table>
  <caption>Caption</caption>
  <tr>
    <th>Header1</th>
    <th>Header2</th>
  </tr>
  <tr>
    <td>Value1</td>
    <td>Value2</td>
  </tr>
  <tr>
    <td>Value3</td>
    <td>Value4</td>
  </tr>
</table>
```

Header1	Header2
Value1	Value2
Value3	Value4

Элемент <table> не был разработан для верстки веб-страниц! Его цель – отображать табличные данные. Не используйте эти элементы для верстки страницы, так как внутренняя структура сайта может стать довольно непрозрачной и трудно поддерживаемой спустя несколько месяцев разработки

Основные принципы отображения элементов в HTML

- Каждый элемент HTML содержит значение отображения по умолчанию в зависимости от типа этого элемента. Для большинства элементов значением по умолчанию считается “блочный” (block) и “строчный” (inline)
- Элемент блочного типа всегда начинается с новой строки и занимает всю предоставляемую ширину (максимально растягивается вправо и влево)
- Примеры блочных элементов:
 - `<div>`
 - `<h1>` - `<h6>`
 - `<p>`
 - `<form>`
- Строчный элемент не начинается с новой строки и занимает лишь необходимую для его отображения ширину
- Примеры строчных элементов:
 - ``
 - `<a>`
 - ``
- Элемент `<div>` часто используется как хранилище для других элементов HTML
- Элемент `` часто используется в качестве хранилища для произвольного текста

Формы HTML

- Элемент HTML `<form>` определяет форму для сбора данных от конечного пользователя
- Форма HTML содержит элементы формы, которые представляют собой разнообразные элементы ввода, такие как текстовые поля, флажки, переключатели, кнопки для отправки данных на обработку и т.п.
- Элемент `<input>` может быть отображен по-разному в зависимости от значения атрибута “type” (тип):
 - `<input type="text">` однострочное текстовое поле ввода
 - `<input type="password">` поле ввода пароля
 - `<input type="radio">` переключатель (для выбора одного варианта из нескольких)
 - `<input type="checkbox">` флажок (для выбора нескольких вариантов из возможных)
 - `<input type="button">` произвольная кнопка
 - `<input type="submit">` кнопка отправки (для отправки данных, введенных в форму, на обработку)
 - `<input type="reset">` кнопка сброса введенных значений (с возвращением к значениям по умолчанию)
 - `<input type="hidden">` невидимый элемент со значением, отправляемым вместе с другими данными формы
- Атрибут “action” определяет род действий, которые следует предпринять во время отправки формы. Как правило, данные отправляются на веб-страницу сервера по нажатию на кнопку отправки. Если параметр опущен, за такую страницу принимается текущая
- Атрибут “method” указывает на метод HTTP (GET или POST), используемый для отправки данных формы
 - Если используется GET (по умолчанию), отправляемые данные будут отображены в адресной строке страницы, на которую совершается отправка
 - Метод POST не показывает пересылаемые данные в адресной строке, а также не имеет ограничений по размеру передаваемых данных, поэтому может быть использован для отправки больших объемов данных

Роль имен полей ввода.

Другие элементы формы

- Каждое поле ввода должно иметь атрибут “name” для отправки значения. Если атрибут опущен, то данные из такого поля ввода не будут отправлены
- Элемент `<fieldset>` позволяет группировать связанные логически данные на форме
- Элемент `<legend>` задает заголовок для элемента `<fieldset>`
- Элемент `<select>` определяет выпадающий список
- Элемент `<option>` определяет входящий в выпадающий список пункт
 - По умолчанию выбирается первый пункт выпадающего списка. Чтобы определить другое заведомо выбираемое значение достаточно задать атрибут “selected” для соответствующего элемента “option”
- Элемент `<textarea>` представляет собой многострочное поле ввода
 - Атрибут “rows” задает количество видимых строк в поле
 - Атрибут “cols” задает ширину области видимости поля
- Элемент `<button>` определяет обычную кнопку, на которую можно нажать
 - Элемент `<button>` предоставляет более богатые возможности отображения, чем элемент `<input type="button">`, поскольку `<button>` может задавать произвольное содержимое, включая изображение. При этом, однако, запрещено связывать карту активных областей с элементом изображения внутри элемента `<button>`

Основные атрибуты элементов форм

- Атрибут “value” задает изначальное значение поля ввода
- Атрибут “readonly” указывает на то, что поле ввода доступно только для чтения (запрещено внесение изменений)
- Атрибут “disabled” определяет поле ввода как недоступное. Такое поле ввода не реагирует на действия пользователя, а значение такого поля не будет учитываться при отправке формы
- Атрибут “size” задает размер (в символах) текстового поля ввода
- Атрибут “maxlength” определяет максимальную длину, допустимую для данного поля ввода. Таким образом, если этот атрибут задан, поле ввода не примет большее число символов, чем указано в качестве значения этого атрибута
 - Атрибут “maxlength” не предупреждает пользователя о наложенном запрете. Для того, чтобы реализовать такое предупреждение, необходимо написать отдельную функцию на JavaScript
 - Ограничения на ввод не достаточно прочны, а JavaScript предоставляет множество способов их обхода. Для обеспечения действительно безопасного ограничения длины необходимо проверять данные также и на стороне сервера

Пример формы

- Фрагмент листинга:

```
<form>
  <pre>
Company: <input type="text" name="Company" value="MGPPU" readonly><br>
Name:   <input type="text" name="User" required><br>
Password:<input type="password" name="Pass" required><br>
Marriage:<select name="Married">
          <option>Yes</option>
          <option>No</option>
        </select>
  </pre>
  <fieldset>
    <input type="radio" name="RadioName1" value="0">Radio1
    <input type="radio" name="RadioName1" value="1">Radio2
  </fieldset><br>
  <fieldset>
    <legend>Fieldset with Legend</legend>
    <input type="radio" name="RadioName2" value="0">Radio3
    <input type="radio" name="RadioName2" value="1">Radio4
  </fieldset><br>
  <input type="checkbox" name="CheckBoxName">Checkbox<br>
  <input type="checkbox" name="CheckBoxBroken" disabled checked>Broken checkbox<br>
Letter:<textarea name="Letter">Some text</textarea><br>
  <input type="button" value="Input button">
  <button><b>Another</b> <i>button</i></button>
  <input type="submit">
  <input type="reset">
</form>
```

Пример формы

- Результат:

Company:

Name:

Password:

Marriage:

☐ Radio1 ☐ Radio2

Fieldset with Legend

☐ Radio3 ☐ Radio4

☐ Checkbox

☒ Broken checkbox

Letter: