

Введение в язык разметки HTML

Андрей Васильев

2019

Архитектура веб-приложения

Веб-приложение - распределённая система, использующая ресурсы двух приложений: веб-браузер и веб-сервер

- Сервер ждёт подключения со стороны клиентов и одновременно обслуживает множество клиентов
- Браузер получает документы с сервера и отображает их пользователю, обрабатывает действия пользователя

Современные браузеры поддерживают следующие документы:

- HTML-документы с текстовой информацией
- CSS-документы с описанием стилей
- Мультимедиа документы: изображения, видео и аудио
- JavaScript документы с исполняемым кодом

Структура веб-приложения

При создании веб-приложения разработчику необходимо выполнить следующие этапы работ:

- Реализовать логику обработки запросов на веб-сервере
- Сформировать структуру HTML-документов для представления информации и выполнения действий
- Сформировать CSS-стили для качественного отображения информации в формируемых документах
- Реализовать логику обработки действий пользователя на стороне браузера с помощью языка JavaScript

В рамках курса мы не будем:

- Формировать собственные CSS-стили оформления
- Использовать язык JavaScript в приложениях

Язык HTML

HTML - Hyper Text Markup Language, язык разметки гипертекстовых документов

Данный язык является стандартом для представления структурированной текстовой и мультимедийной информации

- Документы на языке HTML являются текстовыми
- Ключевая особенность - возможность ссылаться на другие документы удобным для пользователя образом
- Язык достаточно простой, что позволяет его читать людям и обрабатывать программно
- Документ представляет собою иерархическую структуру из узлов, которую можно представить в виде дерева

Описание элементов с помощью тегов

Для описания элементов используется специальная нотация

`<div>Котики очень пушистые</div>`

- Тегом называется текст, состоящий из имени элемента, находящегося между символами `<` и `>`
- Описание элемента начинается с открывающего тега, `<div>`
- Описание элемента завершается с закрывающим тегом, который начинается со строки `</, </div>`
- Текст между открывающим и закрывающим тегом, называется содержимым элемента
- В зависимости от типа элемента способ отображения информации будет различный

Иерархия элементов

HTML-документ представляет собою иерархию элементов, рассмотрим пример

```
<div>
  <p>Песня</p>
  <ul>
    <li>Текст</li>
    <li>Музыка</li>
  </ul>
</div>
```

- Корневым элементом в примере является `div`, всё остальное является его содержимым
- Прямыми детьми элемента `div` выступают элементы `ul` и `p`
- Элемент `p` содержит только текст
- Элемент `ul` содержит 2 тега `li`, причём порядок их появления в тексте важен
- Элементы `li` содержат только текст
- Обычно используется 2 пробела для указания вложенности элементов, хотя отступы не играют роли в языке

Некорректное вложение элементов

При описании HTML-узлов необходимо следить за корректностью вложенности

Корректное вложение:

```
<p>Мой кот <strong>очень</strong> пушистый.</p>
```

Некорректное вложение:

```
<p>Мой кот <strong>очень пушистый.</p></strong>
```

Закрывающий тег `` идёт после тега `</p>`

Большинство браузеров смогут обработать некорректный код, но результат обработки скорее всего будет отличаться между различными браузерами

Комментарии в HTML-документе

Все элементы, которые мы помещаем в HTML-документ, предназначены для отображения информации пользователю

Для добавления информации для других разработчиков можно использовать комментарии

Комментарии начинаются с последовательности `<!--`

Комментарии заканчивается последовательностью `--->`

Пример комментария:

```
<!-- Это просто текстовый комментарий --->  
<!-- <p> Тут тоже комментарий </p> --->
```

Блочные и строчные элементы

Блочные элементы

Формируют видимый блочный элемент на странице

- Показываются с новой строки относительно предыдущего
- Следующие элементы тоже начинаются с новой строки
- Обычно к ним относятся структурные элементы страницы: параграфы, списки, навигационное меню
- Могут быть вложены друг в друга

Строчные элементы

Формируют описание для внутренней части блока

- Описывают *часть* внутреннего содержимого блока
- Обычно служат для описания ссылок на другие страницы, а также выделения текста жирным

Элементы без текстового содержимого

Некоторые элементы HTML-документов не имеют текстового содержимого. Для их описания достаточно указать только лишь открывающий тег и необходимые атрибуты

```

```

- `img` - элемент для отображения изображения на странице
- `src` - атрибут содержит ссылку на источник изображения, которое надо отобразить
- `alt` - атрибут содержит текст, описывающий изображение, который используется если изображение не загрузилось и для обеспечения доступности
- Закрывающий тег не нужен

Атрибуты элементов

У каждого элемента помимо его содержимого можно указать также и атрибуты, которые поясняют его поведение

```
<table>
```

```
  <tr> <!-- Первая строка --->
```

```
    <td colspan="2">Ячейка на 2 столбца</td>
```

```
  </tr>
```

```
  <tr> <!-- Вторая строка --->
```

```
    <td>Первый</td> <td>Второй</td>
```

```
  </tr>
```

```
</table>
```

- Атрибуты описываются в открывающем теге
- Для описания атрибута используется формат имя="значение", атрибуты разделяются пробелами
- Значение атрибута надо помещать в двойные кавычки
- Бывают атрибуты без значения, достаточно указать имя

Описание значения атрибутов

По умолчанию следует использовать двойные кавычки для описания значения атрибута

- Можно вообще без кавычек, но это трудно воспринимать
- Пробелы являются разделителями атрибутов
- Потенциально можно использовать одинарные кавычки, но это не принято и может привести к проблеме смешивания кавычек

```
<a href="http://www.example.com">  
  A link to my example.</a>
```

Следите, чтобы внутри значения не встречались кавычки

```
<a href='http://www.example.com'  
  title='Isn't this fun?'>  
  A link to my example.</a>
```


Структура HTML-документа

```
<!DOCTYPE html> <!-- Исторический артефакт --->
<html>
  <head>
    <!-- Метаинформация о документе --->
  </head>
  <body>
    <!-- Содержимое документа --->
  </body>
</html>
```

- `html` - обязательный корневой элемент
- `head` содержит информацию о содержимом: какой язык использован, какая кодировка у документа, откуда подгружать стилевые файлы
- `body` включает собственно содержимое документа, которое отображается пользователю

Содержимое заголовка

Заголовок содержит описание содержимого документа

- Кодировка символов в документе:

```
<meta charset="utf-8">
```

- Заголовок для отображения в браузере:

```
<title>Классные котики</title>
```

- Автор документа:

```
<meta name="author" content="Chris Mills">
```

- Краткое описание документа:

```
<meta name="description"  
  content="This is example">
```

- Иконка для отображения в списке избранного

```
<link rel="shortcut icon" href="favicon.ico"  
  type="image/x-icon">
```

Подключение внешних ресурсов

Заголовок приходит в браузер первым, поэтому важно размещать ссылки на другие важные документы в заголовке, чтобы браузер мог их начать скачивать

- CSS-документы описывают стили, которые можно применять к различным элементам

```
<link rel="stylesheet" href="my-css-file.css">
```

- JavaScript-документы содержат динамический код для выполнения на стороне браузера

```
<script src="my-js-file.js"></script>
```

Первые надо размещать в заголовке, а JS-код в конце

- `rel` описывает тип ссылки, обязательно `stylesheet`
- `href` содержит ссылку на внешний документ

Указание языка документа

Тегу `html` следует указать язык документа, так как это облегчает работу поисковых систем и систем помощи людям с ограниченными возможностями

Язык указывается в качестве значения тега `lang`:

```
<html lang="ru">
```

Если у вас в документе несколько языков, то соответствующие блоки можно разметить с помощью элемента `span`:

```
<p>  
  Пример английского:  
  <span lang="en-US">The cats are cool</span>.  
</p>
```

Формирование содержимого документа

Содержимое документа находится внутри элемента `body`

Язык HTML предлагает набор из различных элементов, которые позволяют структурировать информацию на странице

- Базовым элементом документа является параграф, `p`
- Есть шесть уровней заголовков: `h1`, `h2`, `h3`, `h4`, `h5`, `h6`
- Каждый из этих элементов блочный

```
<h1>Документ о котиках</h1>
```

```
<p>Котики очень важны!</p>
```

```
<h2>Котики: начало</h2>
```

```
<p>Котики пушистые</p>
```

```
<h2>Котики: продолжение</h2>
```

```
<h3>Котолапки</h3>
```

```
<p>У котолапок есть когти</p>
```

Предложения по структурированию

- Желательно использовать 1 заголовок первого уровня
- Далее информацию следует группировать в подзаголовках
- h3 должно следовать за h2, а не наоборот
- Следует использовать не более четырёх видов заголовков на одной странице, а лучше всего три

Кому нужны заголовки

- Пользователям удобнее ориентироваться в структурированной информации, а не в длинных текстах
- Поисковым движкам легче извлекать важные данные
- Обеспечения доступа для людей с ограниченными возможностями
- Заголовки можно красиво стилизовать

Списки

Ненумерованный список

- Список описывается элементом `ul`
- Отдельный элемент списка описывается элементом `li`

```
<h1>Мои покупки</h1>
```

```
<ul>
```

```
  <li>Молоко</li>
```

```
  <li>Яйца</li>
```

```
  <li>Хлеб</li>
```

```
  <li>Сыр</li>
```

```
</ul>
```

Нумерованный список

- Список описывается элементом `ol`
- Отдельный элемент списка описывается элементом `li`

```
<h1>Мои котики</h1>
```

```
<ol>
```

```
  <li>Пёстрый</li>
```

```
  <li>Рыжик</li>
```

```
  <li>Кусак</li>
```

```
  <li>Белый</li>
```

```
</ol>
```

Внутри содержимого элемента списка может находиться другой список, эффективно делая его вложенным

Выделение текста

При написании очень длинного текста, *зачастую* необходимо выделять различные части текста, чтобы **читающий** мог понять что в тексте *важно*, а что нет

- Для выделения текста можно использовать строчный элемент `em`, сокращение от *emphasis*:

```
<p>Я <em>рад</em>, что вы не  
  <em>опоздали</em></p>
```

- Для выделения текста жирным используется строчный элемент `strong`:

```
<p>Данная житкость <strong>сильно  
  токсичная</strong></p>
```

Форматирование текста

В WYSWYG текстовых редакторах есть возможность указывать атрибуты текста: наклон, жирность, подчёркивание

В старых HTML-стандартах определены элементы `i`, `b`, `u`, которые по названию позволяют форматировать текст

В настоящий момент для оформления текста надо использовать CSS-стили

- `i` следует использовать для оформления иностранных слов, терминов, цитату и т.д.
- `b` следует использовать для оформления ключевых слов, названий продуктов, вводных предложений
- `u` зачастую использовать вообще не стоит, даже в печатных изданиях

Ссылки на другие ресурсы

При описании HTML-документа необходимо ссылаться на другие документы по их единому указателю ресурса, URL

- При формировании ссылок для пользователя
- Для указания местоположения CSS таблиц
- Для подключения исполняемых JS-файлов
- Для указания пути к мультимедиа объектам

Элемент a

Для создания гиперссылки для пользователя используется элемент a: `Поиск`

- href содержит в себе ссылку на другой ресурс
- Текстовое содержимое показывается пользователю

Дополнительные возможности а

title предоставляет дополнительную информацию, которая будет показана пользователю при наведении на ссылку

<p>Создаём ссылку

```
<a href="https://developer.mozilla.org"  
  title="Информация для веб-разработчиков"  
>Mozilla developer network</a>
```

</p>

- По возможности размещайте данную информацию в тексте, чтобы она была доступна на всех устройствах
- Элемент a может включать в себя блочные элементы, в том числе и изображения

```
<a href="https://www.mozilla.org/ru/">  
    
</a>
```

Формирование путей к документам

Как и при работе с файловой системой можно формировать либо полные пути, либо относительные пути

- Полный путь начинается либо с указания схемы доступа, `https:/`, `file:/`, либо с `/`
- Относительный путь высчитывается с «каталога», в котором находится текущий ресурс

Рассмотрим следующую структуру файлов

- `index.html`
- `data.html`
- `img`
 - ▶ `cat.png`
 - ▶ `dog.gif`
- `css`
 - ▶ `style.css`

- Ссылка из `index.html` на `data.html`:

```
<a href="data.html">...
```

- Отображаем картинки

```

```

- Подключаем стили

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

Практики по формированию путей

- По возможности используйте относительные пути
 - ▶ При развёртывании на файловой системе доступны только относительные ссылки
 - ▶ При работе с веб-сервером браузеру не нужно выполнять DNS-запросы
- Текст ссылки должен хорошо описывать ссылку
 - ``
Поискать интересного в интернете``
 - ``
Нажмите на ссылку
``, чтобы поискать в интернете
- Не добавляйте URL в текст ссылки
- Ссылки должны состоять из пары-тройки слов
- Используйте уникальный текст для каждой ссылки

Пути к специальным документам

Если по нажатию на ссылку пользователю необходимо скачать большой объём данных или требуется специальная технология, то пользователю необходимо сообщить об этом

```
<p><a href="http://www.example.com/report.pdf">  
    Скачать большой отчёт (PDF, 10MB)  
</a></p>
```

```
<p><a href="http://www.example.com/video-stream/"  
    target="_blank">  
    Посмотреть видео (поток, отдельная вкладка)  
</a></p>
```

```
<p><a href="http://www.example.com/car-game">  
    Играть в гонки (требует WASM)  
</a></p>
```

Семантическая разметка документа

Ранее рассмотренные элементы структуры могут оказаться в различных частях страницы и нести разную смысловую нагрузку

- Заглавие, обычно изображение или общие данные для всех страниц, принадлежащих одному сайту
- Навигационная панель, также общий элемент между страницами, позволяющий передвигаться между ними
- Содержимое страницы, обычно центральная часть страницы, предоставляющая ключевую информацию
- Боковая панель, содержит дополнительную информацию, рекламу и т.д.
- Нижний колонтитул, обычно общий для всех страниц, содержит информацию о лицензии, дополнительные ссылки

Элементы семантической разметки

<body>

<header>Заголовок страницы**</header>**

<nav>Навигация внутри страницы**</nav>**

<main>Основное содержимое

<article>Статья №1

<header>Заголовок статьи**</header>**

<section>Секция 1 статьи 1

<header>Заголовок секции**</header>**

<p>Содержимое**</p>**

</section>

</article>

<aside>Боковая панель**</aside>**

</main>

<footer>Содержимое нижнего колонтитула**</footer>**

</body>