

Тема-1. Введение. HTML & DOM

Что тут есть?

Вступление про курс. HTML: что это такое и зачем; основные понятия; использование атрибутов; часто используемые теги; использование атрибутов; версии HTML; что такое DOM; полезные ресурсы.

Что будет в семестре

В этом семестре будут рассмотрены основы веб-программирования.

Планируется, что вы научитесь пользоваться:

- базовыми фронтенд-технологиями: HTML, CSS, JavaScript.
- одной продвинутой фронтенд-технологией: React;
- некоторыми бэкенд-технологиями: FastAPI, Alembic, SQLAlchemy;
- некоторыми общими веб-технологиями: NPM, Docker, CI/CD.

В основе нашего стека технологий будет: JavaScript, Python3, PostgreSQL.

Что нужно для работы

Ноутбук с установленными Python3, браузерами, какой-нибудь IDE (например, PyCharm) и папкой для кода.

ОС любая (я всё показываю с Linux).

Вообще, веб-разработка в целом удобнее с Linux или Mac, чем с Windows. Особенно это касается бекенда (проще всё поставить).

Начало работы

Мы начнем работать с фронтенда. Типичная структура файлов у нас будет такой:

```
our-project/  
- index.html  
- images/  
  - - some-image.png  
  - - ...  
- styles/  
  - - some-style.css  
  - - ...  
- scripts/  
  - - some-script.js  
  - - ...
```

Что такое HTML и для чего он?

HTML – это Hyper Text Markup Language – язык гипертекстовой разметки документов для просмотра веб-страниц в браузере.

Презназначен он для того, чтобы создать основную структуру элементов документа и отобразить их в браузере. Это „скелет“ web-приложения.

Основные понятия HTML: документ, элемент, тег, атрибут. О них далее.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <!--comment-->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
    <title>Creative map studio</title>
  </head>
  <body>
    <noscript>
      To run the application, you should enable JavaScript in your
browser settings.
    </noscript>
    <div id="root">
      <p>Page content</p>
    </div>
  </body>
</html>
```

Кто придумал HTML?

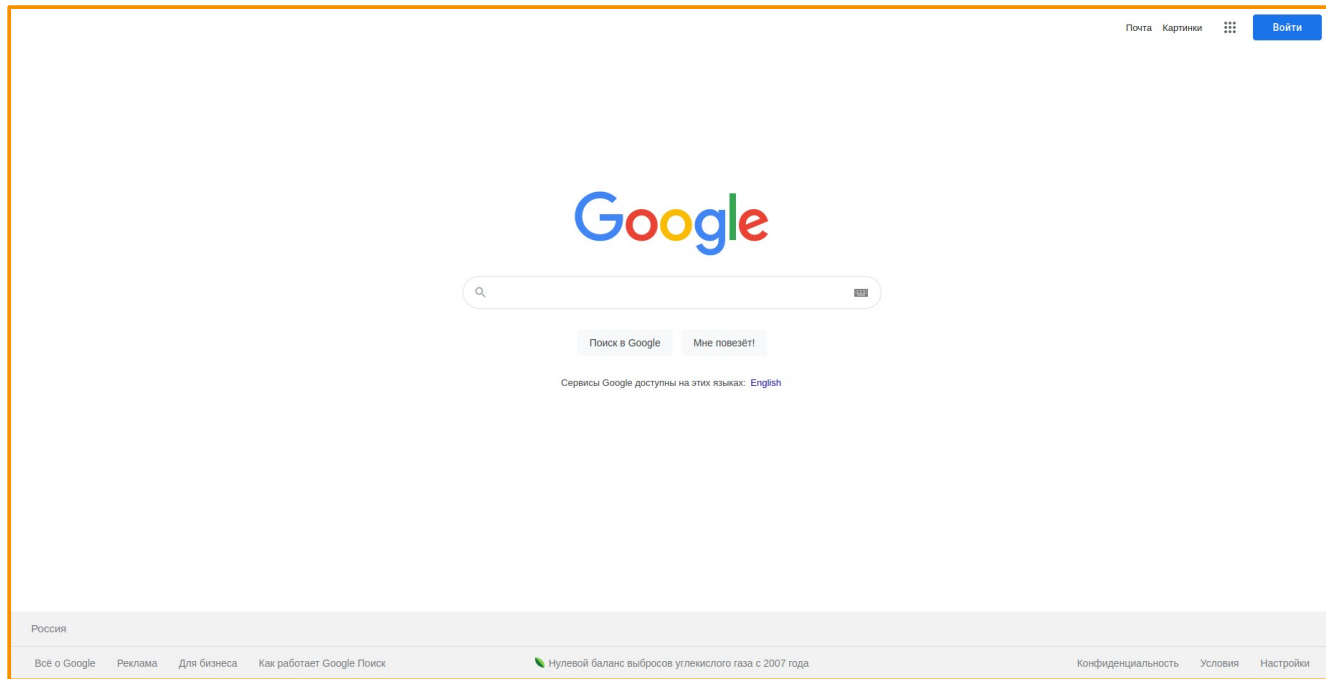


Авторство изображения: Paul Clarke.
Собственная работа, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=53878695>

Тимоти Джон Бернерс-Ли – создатель HTML, HTTP и URI

Основные понятия HTML: документ

Документ (document) – это некоторый отрисованный в браузере HTML-файл. HTML-файл состоит из вложенных друг в друга тегов и обычного текста. Также он может содержать стили (специальный код, описывающий внешний вид документов) или код на JavaScript, или ссылки на такой код.



<!-- Любая страница – это документ -->

<!-- Любой сайт – это один или несколько документов -->

Основные понятия HTML: элемент

Элемент (element) – это часть документа, которая описывает определенный объект документа или свойство документа.

Документ состоит из элементов, и только из них. То, что находится внутри элемента называется контентом (content).

Виды элементов:

`<input type="button" class="run-code-btn" value="run"/>` - самозакрывающий элемент.

`<h1 class="main-page-title">Main page</h1>` - парный элемент (есть и открывающий, и закрывающий тег).

Some text - просто текст.

Основные понятия HTML: тег

Тег (Tag) – это специальное слово, описывающее как именно нужно интерпретировать элемент, для чего он нужен.

Теги обычно используются для:

- добавления элемента в документ;
- группировки существующих элементов;
- описания глобальных свойств документа;
- подключения стилей и кода на JavaScript.

```
<input/>, <p></p>,  
<div></div>,  
<head></head>,  
<i></i>,  
<footer></footer>,  
<h2></h2>, <option/>,  
<image/>, <a></a>,  
<meta/>,  
<title></title>,  
<style></style>,  
<body></body>
```

Основные понятия HTML: атрибут

Атрибут – это параметр тега, который позволяет его настраивать.

Есть глобальные атрибуты, которые есть у всех тегов, например:

- class;
- id.

Есть специальные, которые можно установить только у некоторых тегов, например:

- type в `<input type="number"/>`;
- selected в `<option value="some value" selected/>`.

Использование общих атрибутов

Есть несколько таких атрибутов, которые могут быть использованы фактически с любым тегом. Это атрибуты, отвечающие за глобальные характеристики элементов.

Некоторые популярные глобальные атрибуты:

- id – задает уникальный идентификатор элемента в документе;
- class – задает класс(ы) (обычно для стилей) элемента;
- style – задает персональные стили для элемента напрямую;
- все обработчики событий – некоторые из них мы рассмотрим, когда будем работать с JavaScript.

Использование специальных атрибутов

Специальных атрибутов много и их лучше рассматривать непосредственно вместе с тегами, для которых они предназначены. Далее некоторые из таких атрибутов будут рассмотрены.

Часто используемые теги: базовые

<!DOCTYPE html> -- вообще говоря, это не тег, а просто пометка для браузера о том, что данный файл формата HTML. Эта пометка должна быть в начале всякого HTML-файла.

Всякий документ содержит базовый корневой элемент <html>. Внутри него находятся все элементы документа (кроме указанного выше).

Пространство внутри <html> делится на две части – заголовочную часть и тело документа.

Часто используемые теги: базовые

Заголовочная часть `<head>...</head>`

Здесь указываются параметры документа и различная информация, которая не показывается пользователям внутри документа, но нужна для правильного отображения страницы.

Например:

`<title>Мой сайт</title>` – элемент указывает название страницы, которое будет видно на вкладке в браузере.

`<meta charset="utf-8">` – показывает браузеру, что кодировка документа – utf-8.

Также здесь указываются ссылки на стили с помощью `<link>`. Также можно прописать стили напрямую, внутри тега `<style>`.

Например:

`<link rel="stylesheet" href="styles.css"/>` - подключает файл со стилями.

`<style>p {color: red; font-size: 1.5rem}</style>` - подключает стили напрямую.

Часто используемые теги: базовые

Тело документа `<body>...</body>`

Здесь содержатся все элементы, которые будут отображаться в документе (прямо или косвенно). О них далее.

Также здесь можно указать скрипты (с помощью тега `<script>`), которые необходимо запустить.

Например:

`<script type="text/javascript" src="script.js"></script>` - подключает скрипт из файла.

`<script type="text/javascript">...</script>` - подключает код скрипта напрямую.

Часто используемые теги: Тег-комментарий

Есть специальный тег-комментарий. Это работает так же, как и в других языках программирования.

Обозначается он так:

```
<!-- any text -->
```


Часто используемые теги: форматирование текста

Заголовки и подзаголовки в тексте – `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` (h – header).

Отличаются они только размером шрифта. В остальном разделение по уровням чисто семантическое.

32px, `<h1>` **Lorem Ipsum**

24px, `<h2>` **Lorem Ipsum**

19px, `<h3>` **Lorem Ipsum**

16px, `<h4>` **Lorem Ipsum**

13px, `<h5>` **Lorem Ipsum**

11px, `<h6>` **Lorem Ipsum**

Часто используемые теги: форматирование текста

Абзацы текста – <p> (p – paragraph).

Есть также тег span. Фактически этот тег не имеет каких-то особых стилей или назначения. Он просто позволяет выделить как-то отдельно часть текста.

Также части текста можно делать полужирными (), курсивными (<i>) или подчеркнутыми (<u>). Хотя того же самого можно добиться и с помощью CSS, который мы рассмотрим далее. Чтобы добавить в текст перенос строки используется тег
.

Just text Another text

Suspendisse egestas, magna sit amet scelerisque mollis, arcu leo congue tortor, quis venenatis diam dui et mi. Vestibulum mattis felis eu, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis molestie lacinia lacinia. Nunc quam quam, fringilla quis porta eu, feugiat ac

```
<body>
  <div style="margin: 10px;">
    Just text
    Another text
  </div>
  <div style="margin: 10px;">
    <p>Suspendisse egestas, magna <span
style="color: red">sit amet scelerisque mollis</span>,
arcu leo congue tortor, quis venenatis diam dui et mi.
Vestibulum mattis felis eu justo rhoncus, et auctor
augue tempus. </p>
    <p>Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Duis molestie lacinia lacinia.
Nunc quam quam, fringilla quis porta eu, feugiat ac
tortor. Sed laoreet nisi varius nisi convallis, quis faucibus
risus consequat.</p>
  </div>
</body>
```

Часто используемые теги: форматирование текста

Текст может размещаться в маркированных или нумерованных списках. Сам список обозначается тегом `` (маркированный) или `` (нумерованный). Элементы списка обозначаются `` (независимо от того, какой список, LI – List Item).

- Первый вариант списка
- Этот список маркированный
- Называется ul, потому что элементы не упорядочены
- **Unordered List**

```
<div style="display: flex;">
  <ul>
    <li>Первый вариант списка</li>
    <li>Этот список маркированный</li>
    <li>Называется <u>ul</u>, потому что элементы не
упорядочены</li>
    <li><b>U</b>nordered <b>L</b>ist</li>
  </ul>
  <ol>
    <li>Второй вариант списка</li>
    <li>Этот список нумерованный</li>
    <li>Называется <u>ol</u>, потому что элементы
упорядочены</li>
    <li><b>O</b>rdered <b>L</b>ist</li>
  </ol>
</div>
```

1. Второй вариант списка
2. Этот список нумерованный
3. Называется ol, потому что элементы упорядочены
4. **Ordered List**

Часто используемые теги: форматирование текста

Для форматирования текста может быть также использована таблица.

Для создания таблицы используется несколько тегов: `<table>`, `<tbody>`, `<tr>`, `<td>`, `<thead>`, `<th>`.

Корневой элемент таблицы -- `<table></table>`.

Таблица, подобно HTML-документу делится на шапку и тело (head и body). Соответственно, шапка таблицы находится в теге `<thead>` (Table HEAD), тело находится в `<tbody>` (Table BODY).

Внутри таблицы ряды обозначаются тегам `<tr>` (Table Row). А ячейки тегам `<td>` (Table Data cell) (ячейки вкладываются в ряды).

Также есть другие опциональные элементы, такие как `<caption>`, `<tfoot>`, `<colgroup>` и т.д., но в рамках курса мы не будем их рассматривать.

Часто используемые теги: форматирование текста

Пример таблицы

Топ 5 городов России по населению

Название	Население	Год основания
Москва	~12.6 млн	<1096
Санкт-Петербург	~5.4 млн	1703
Новосибирск	~1.6 млн	1893
Екатеринбург	~1.5 млн	1723
Казань	~1.3 млн	<1005

```
<table>
  <thead>
    <tr><th colspan="3">Топ 5 городов России по
населению</th></tr>
    <tr><th>Название</th><th>Население</th><th>Год
основания</th></tr>
  </thead>
  <tbody>
    <tr><td>Москва</td><td>~12.6 млн</td><td><1096</td></tr>
    <tr><td>Санкт-Петербур</td><td>~5.4
млн</td><td>1703</td></tr>
    <tr><td>Новосибирск</td><td>~1.6
млн</td><td>1893</td></tr>
    <tr><td>Екатеринбур</td><td>~1.5
млн</td><td>1723</td></tr>
    <tr><td>Казань</td><td>~1.3 млн</td><td><1005</td></tr>
  </tbody>
</table>
```

Часто используемые теги: ссылки

Самое важное в интернете – это его связность. За это отвечает тег <a>, с помощью которого создаются ссылки.

a значит anchor (якорь).

Этот тег всегда используется с атрибутом href (Hypertext-REFerence, гипер-ссылка). В этом атрибуте указывается, куда ведет ссылка.

Пример ссылки: Кликните здесь

Часто используемые теги: ссылки

Ссылка может быть внешней (на другой сайт) или внутренней (на другую страницу сайта или на эту же (на определенное место, с помощью якоря)).

Ссылка считается внутренней, если она не содержит домена (тогда при переходе по ссылке домен текущей страницы автоматически подставится).

Примеры:

[](https://vk.com) -- внешняя ссылка

[](/store/product/2) -- внутренняя ссылка

[](https://developer.mozilla.org/ru/docs/#ссылки) --
внутренняя ссылка с якорем

Часто используемые теги: формы

Формы в HTML предназначены для того, чтобы получать от пользователя определённые данные.

Вся форма изначально содержится в теге <form>. Внутри формы размещаются элементы, которые принимают ввод от пользователя. Обобщенно такие элементы называются control (по-русски можно говорить поле ввода). Помимо собственно полей ввода форма может содержать также вспомогательные элементы (например, подписи к полям ввода, инструкции, подсказки и т.д.).

Дальше рассмотрим разные элементы, часто присутствующие в формах по-отдельности.

Часто используемые теги: формы

В самом простом виде элемент `<form>` должен иметь 2 атрибута: action и method. Эти атрибуты позволяют указать, куда отправить данные и каким образом они будут передаваться.

Например:

`<form action="/form-handling-endpoint" method="post">`

...

`</form>`

Адрес эндпоинта может быть любым (но об эндпоинтах позже). О видах HTTP запросов на следующем слайде.

Часто используемые теги: формы

Виды HTTP запросов:

(которые нужно знать)

- GET – запрос для получения данных
- POST – запрос для добавления новых данных.
- PUT – запрос для обновления данных (любое изменение кроме удаления).
- DELETE – запрос для удаления данных.

(которые пока можно и не знать)

- HEAD – запрос для получения информации об ответе (как GET, только не содержит тела ответа).
- CONNECT – устанавливает туннель с сервером.
- PATCH – используется для мелкого изменения данных (но обычно просто всегда PUT используется).
- OPTIONS – используется, чтобы узнать, какие запросы можно отправлять к этому эндпоинту.
- TRACE – используется в отладочных целях.

Часто используемые теги: формы

Перейдем к наполнению форм. Самый простой элемент формы – <label> (метка). Это просто надпись к полю ввода.

Пример: <label for="control-id">Фамилия:</label>

control-id – это id поля ввода. Это нужно для ассоциирования надписи с полем ввода.

Или можно ещё просто вложить поле ввода внутрь надписи. Тогда атрибут for не нужен.

Часто используемые теги: формы

Основной тег, используемый для полей ввода – это <input>. Этот тег имеет специальный атрибут type, с помощью которого определяется, данные какого формата ожидаются от пользователя.

По умолчанию, значение атрибута type – text. О типах далее.

Часто используемые теги: формы

Некоторые типы полей ввода <input>:

- button

- checkbox

☐

- date

- email

- file

- hidden

- number

- password

- radio

☐

- range

- submit

- text

- time

Часто используемые теги: формы

У полей ввода каждого из типов есть свои определённые атрибуты, которые определяют конкретные значения, которые могут быть в них введены, различные надписи (например, названия кнопок, плейсхолдеры и т.д.)

Посмотреть список атрибутов и с какими полями их можно применять можно здесь:

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input#attributes>

Часто используемые теги: формы

Есть также специальное многострочное поле для ввода больших текстов (например, описаний чего-либо) – <textarea>.

Часто используемые атрибуты: placeholder (небольшой текст-подсказка в поле ввода), cols и rows (позволяет задавать размер в строках и символах в строке), wrap (задает правила переноса текста на следующую строку).

Непосредственно вводимый текст будет внутри элемента <textarea>.

Часто используемые теги: формы

С помощью тегов <select> и <option> можно реализовать поле выбора варианта из списка.

<select> – это своего рода контейнер для будущего поля ввода. Важные атрибуты – name. Подобно другим полям ввода атрибут используется для того, чтобы назвать данные.

<option> – это вариант выбора значения. Важный атрибут value – значение, которое будет записано в форму при выборе. Текст, который увидит пользователь, помещается внутрь элемента.

Часто используемые теги: формы

Пример поля ввода с вариантами выбора:

```
<select name="sex">  
  <option value="m">Мужчина</option>  
  <option value="f">Женщина</option>  
</select>
```

Часто используемые теги: формы

Ещё несколько важных атрибутов:

- size – показывает, сколько элементов одновременно нужно показывать (по умолчанию равен 1 (комбобокс), при других значениях будет поле, которое можно прокручивать);
- multiple – при указании разрешает выбрать несколько вариантов одновременно.

Тип кузова:

Седан ▾

Седан
Хэтчбек
Лифтбек
Лимузин
Внедорожник
Кроссовер
Стретч
Купе
Кабриолет
Родстер
Тарга
Пикап
Фургон
Минивэн

Тип кузова:

Пикап
Фургон
Минивэн

Часто используемые теги: формы

Есть также специальный тег – <button>. Он позволяет создавать кнопки подобно тегу `<input>`.

Здесь также как в `<input>` используется атрибут type с одним из значений: reset, button и submit.

Отличие `<button>` от `<input>` в том, что в последнем случае в качестве „названия“ кнопки можно использовать не только текст, но и любой другой контент (например, картинку).

Часто используемые теги: изображения

Тег для размещения изображения: ``

Это пустой элемент, т. е. внутрь него нельзя положить ещё какой-то элемент.

Самые важные атрибуты:

- src – источник изображения (путь к нему);
- alt – текст, который покажется при наведении на изображение или когда изображение не получилось загрузить (пояснение);
- decoding – позволяет отложить загрузку изображения, чтобы другая информация быстрее загрузилась;
- width и height – задает размеры для изображения.

Часто используемые теги: изображения

Тег <canvas> позволяет динамически рисовать растровые изображения с помощью JavaScript.

JavaScript будет рассмотрен позже, пока же достаточно знать, что для `<canvas>` необходимо указать атрибуты: id (чтобы обращаться из JS), height и width (размеры в пикселях).

Часто используемые теги: SVG-изображения

Тег <svg> позволяет встраивать svg-изображения непосредственно в код HTML (не в качестве картинки, а напрямую, то есть можно обращаться к частям изображения из JavaScript или применять к ним стили).

Однако это достаточно большая тема и она будет рассмотрена отдельно позже (если успеем).

Часто используемые теги: аудио

Тег <audio> позволяет встраивать звуковые файлы в документ. Он используется вместе с тегом <source> (источник), который реализует непосредственно загрузку файла (необязательно звукового).

Для отображения элементов управления воспроизведением нужно указать атрибут controls у тега <audio>. Для того, чтобы загрузить файл в <source> необходимо указать атрибут src с путем к файлу и атрибут type, который указывает MIME-type файла.

Пример:

<audio controls>

<source src="./sound.mp3" type="audio/mpeg"/>

</audio>

Часто используемые теги: видео

Тег <video> позволяет встраивать видео в документ. Он используется точно также как <audio>.

Пример:

<video controls>

<source src="./video.mp4" type="video/mpeg"/>

</video>

Часто используемые теги: скрипты

Тег <script> используется для того, чтобы связать HTML-файл с кодом на JavaScript.

Можно написать код как напрямую внутри элемента `<script>`, так и указать в атрибуте src (SouRce) ссылку на файл с кодом.

Код из скрипта выполняется при загрузке HTML-файла.

Есть ещё несколько полезных атрибутов у элемента `<script>`.

- defer – откладывает загрузку скрипта до тех пор, пока страница не загрузится.

- async – позволяет загружать скрипт асинхронно.

Часто используемые теги: группировка

Самый распространенный и часто используемый тег для группирования элементов и структурирования страницы – <div>.

Этот тег не имеет определённой специализации и используется в самых разных ситуациях.

Он также не имеет каких-то особенных атрибутов, а только использует общие.

Часто используемые теги: группировка

Однако, в некоторых случаях вместо тега `<div>` можно использовать другие специальные теги, такие как: `<aside>` (боковое меню), `<header>` (шапка страницы, или, хедер), `<footer>` (подвал сайта, или, футер), `<nav>` (блок навигации).

Эти специальные теги нужно только лишь для того, чтобы окрасить семантически разные части страницы (фактически, они ничем не отличаются от `<div>` кроме названия).

HTML 4 и HTML5

HTML 4 вышел 18 декабря 1997 года.

XHTML – версия HTML 4, совместимая с форматом XML.

HTML 5 вышел 28 октября 2014 года. Эта версия:

- улучшает разметку документов;
- добавляет специальные семантические элементы (это помогает при индексации и открывает некоторые новые возможности, Пример новых элементов: <nav>, <footer>, <video>, <canvas>);
- добавляет поддержку SVG и MathML (но браузеры не очень-то его поддерживают);
- включает в себя Web Forms 2.0 (всякие удобные фишки для создания форм);
- D'n'D (Drag and Drop) – механизм „перетаскивания“ объектов на странице;
- можно не ставить слеш в конце самозакрывающих тегов.

Что такое DOM?

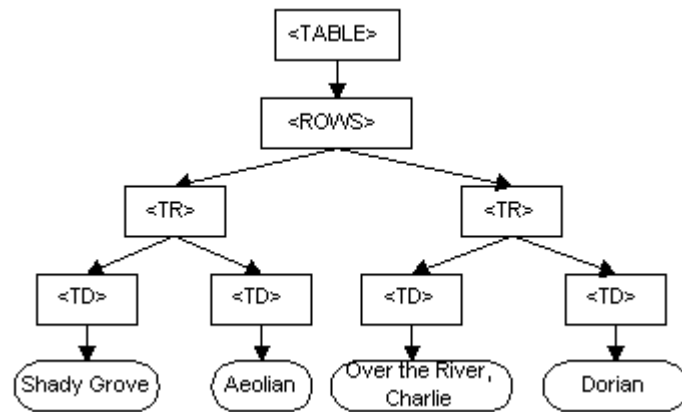
DOM (Document Object Model) – это программное API для HTML.

Применительно к web-приложениям, DOM позволяет получить доступ на чтение и запись в HTML-документ из JavaScript.

Что такое DOM?

DOM имеет древовидную структуру. Соответственно, навигация по DOM-дереву осуществляется в посредством перемещения между потомками и родителями.

Использовать DOM мы будем далее при работе с JavaScript.



Пример структуры
(взято на w3.org)

Полезные ресурсы по HTML

- https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics – **обучалка HTML на MDN;**
- <https://www.w3.org/TR/html5-diff/> - **полный список отличий HTML 5 по сравнению с HTML 4.**
- <https://www.w3.org/TR/WD-DOM/introduction.html> – **подробнее про DOM (английский).**

Следующая тема CSS!