**Спецификации**

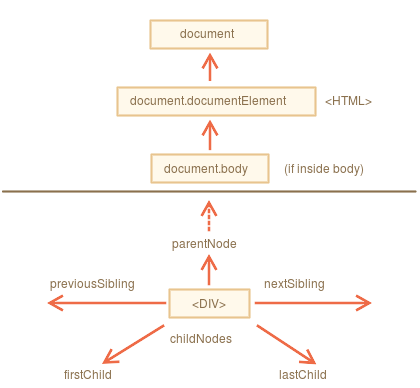
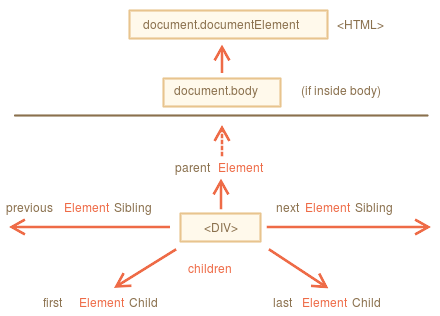
**DOM –** объектная модель документа, которая представляет все содержимое страницы в виде объектов, которые можно менять. Объект document – основная «входная точка». С его помощью мы можем что-то создавать или менять на странице. [Спецификация](https://dom.spec.whatwg.org)

Объектная модель браузера (**BOM**) – это дополнительные объекты, предоставляемые браузером (окружением), чтобы работать со всем, кроме документа. BOM является частью общей [спецификации HTML](https://html.spec.whatwg.org).

**CSSOM** - Описывает файлы стилей, правила написания стилей и манипуляций с ними, а также то, как это всё связано со страницей, подробнее на <https://www.w3.org/TR/cssom-1/>

**Навигация по DOM-элементам**

DOM – узлы: DOM – элементы



Все операции с DOM начинаются с объекта **document**. Это главная «точка входа» в DOM. Из него мы можем получить доступ к любому узлу.

Самые верхние элементы дерева доступны как свойства объекта document:

**<html> = document.documentElement -** cамый верхний узел документа. В DOM он соответствует тегу <html>.

**<body> = document.body** - другой часто используемый DOM-узел – узел тега <body>. Есть одна тонкость: document.body может быть равен null.Нельзя получить доступ к элементу, которого ещё не существует в момент выполнения скрипта. В частности, если скрипт находится в <head>, document.body в нём недоступен, потому что браузер его ещё не прочитал.

В DOM значение null значит «не существует» или «нет такого узла».

**<head> = document.head** - тег <head>

Коллекция **childNodes** содержит список всех детей, включая текстовые узлы.

Для проверки наличия дочерних узлов существует также специальная функция **elem.hasChildNodes().**

**childNodes** это *коллекция*– особый перебираемый объект-псевдомассив.

1. Для перебора *коллекции* мы можем использовать for..of:  
   for (let node of document.body.childNodes) {

alert(node); // покажет все узлы из коллекции

}

1. Можно применить методы массивов:

alert( **Array.from**(document.body.childNodes).filter ); - создаем массив

DOM-коллекции – **только для чтения**

Почти все DOM-коллекции, за небольшим исключением, **живые**. Другими словами, они отражают текущее состояние DOM. Если мы сохраним ссылку на elem.childNodes и добавим/удалим узлы в DOM, то они появятся в сохранённой коллекции автоматически.

**Таблицы**

Элемент <table>, в дополнение к свойствам, о которых речь шла выше, поддерживает следующие:

table.rows – коллекция строк <tr> таблицы.

table.caption/tHead/tFoot – ссылки на элементы таблицы <caption>, <thead>, <tfoot>.

table.tBodies – коллекция элементов таблицы <tbody> (по спецификации их может быть больше одного).

<thead>, <tfoot>, <tbody> предоставляют свойство rows:

tbody.rows – коллекция строк <tr> секции.

<tr>:

tr.cells – коллекция <td> и <th> ячеек, находящихся внутри строки <tr>.

tr.sectionRowIndex – номер строки <tr> в текущей секции <thead>/<tbody>/<tfoot>.

tr.rowIndex – номер строки <tr> в таблице (включая все строки таблицы).

<td> and <th>:

td.cellIndex – номер ячейки в строке <tr>.