**Дерево DOM**

Основным инструментом работы и динамических изменений на странице является DOM (Document Object Model) – объектная модель, используемая для XML/HTML-документов.

Согласно DOM-модели, документ является иерархией, деревом. Каждый HTML-тег образует узел дерева с типом «элемент». Вложенные в него теги становятся дочерними узлами. Для представления текста создаются узлы с типом «текст».

DOM – это представление документа в виде дерева объектов, доступное для изменения через JavaScript.

**[Пример DOM](http://learn.javascript.ru/dom-nodes" \l "primer-dom)**

Построим, для начала, дерево DOM для следующего документа.

<!DOCTYPE HTML>

<html>

<head>

<title>О лосях</title>

</head>

<body>

Правда о лосях

</body>

</html>

Его вид:

▾HTML▾HEAD#text ↵␣␣▾TITLE#text О лосях#text ↵#text ↵▾BODY#text Правда о лосях

В этом дереве выделено два типа узлов.

1. Теги образуют *узлы-элементы* (element node). Естественным образом одни узлы вложены в другие. Структура дерева образована исключительно за счет них.
2. Текст внутри элементов образует *текстовые узлы* (text node), обозначенные как #text. Текстовый узел содержит исключительно строку текста и не может иметь потомков, то есть он всегда на самом нижнем уровне.

**На рисунке выше синие узлы-элементы можно кликать, при этом их дети будут скрываться-раскрываться.**

Обратите внимание на специальные символы в текстовых узлах:

* перевод строки: ↵
* пробел: ␣

**Пробелы и переводы строки – это тоже текст, полноправные символы, которые учитываются в DOM.**

В частности, в примере выше тег <html> содержит не только узлы-элементы <head> и <body>, но и #text(пробелы, переводы строки) между ними.

Впрочем, как раз на самом верхнем уровне из этого правила есть исключения: пробелы до <head> по стандарту игнорируются, а любое содержимое после </body> не создаёт узла, браузер переносит его внутрь, в конец body.

В остальных случаях всё честно – если пробелы есть в документе, то они есть и в DOM, а если их убрать, то и в DOM их не будет, получится так:

<!DOCTYPE HTML>

<html><head><title>О лосях</title></head><body>Правда о лосях</body></html>

▾HTML▾HEAD▾TITLE#text О лосях▾BODY#text Правда о лосях

**[Автоисправление](http://learn.javascript.ru/dom-nodes" \l "avtoispravlenie)**

При чтении неверного HTML браузер автоматически корректирует его для показа и при построении DOM.

В частности, всегда будет верхний тег <html>. Даже если в тексте нет – в DOM он будет, браузер создаст его самостоятельно.

То же самое касается и тега <body>.

Например, если файл состоит из одного слова "Привет", то браузер автоматически обернёт его в <html> и <body>.

**При генерации DOM браузер самостоятельно обрабатывает ошибки в документе, закрывает теги и так далее.**

Такой документ:

<p>Привет

<li>Мама

<li>и

<li>Папа

…Превратится вот во вполне респектабельный DOM, браузер сам закроет теги:

▾HTML▾HEAD▾BODY▾P#text Привет▾LI#text Мама▾LI#text и▾LI#text Папа

**Таблицы всегда содержат <tbody>**

Важный «особый случай» при работе с DOM – таблицы. По стандарту DOM они обязаны иметь <tbody>, однако в HTML их можно написать без него. В этом случае браузер добавляет <tbody> самостоятельно.

Например, для такого HTML:

<table id="table">

<tr><td>1</td></tr>

</table>

DOM-структура будет такой:

▾TABLE▾TBODY▾TR▾TD#text 1#text ↵

Вы видите? Появился <tbody>, как будто документ был таким:

<table>

<tbody>

<tr><td>1</td></tr>

</tbody>

</table>

Важно знать об этом, иначе при работе с таблицами возможны сюрпризы.

**[Другие типы узлов](http://learn.javascript.ru/dom-nodes" \l "drugie-tipy-uzlov)**

Дополним страницу новыми тегами и комментарием:

<!DOCTYPE HTML>

<html>

<body>

Правда о лосях

<ol>

<li>Лось — животное хитрое</li>

<!-- комментарий -->

<li>...и коварное!</li>

</ol>

</body>

</html>

▾HTML▾HEAD▾BODY#text Правда о лосях▾OL#text ↵␣␣␣␣▾LI#text Лось — животное хитрое#text ↵␣␣␣␣#comment комментарий#text ↵␣␣␣␣▾LI#text ...и коварное!#text ↵␣␣#text ↵↵

**В этом примере тегов уже больше, и даже появился узел нового типа – *комментарий*.**

Казалось бы, зачем комментарий в DOM? На отображение-то он всё равно не влияет. Но так как он есть в HTML – обязан присутствовать в DOM-дереве.

**Всё, что есть в HTML, находится и в DOM.**

Даже директива <!DOCTYPE...>, которую мы ставим в начале HTML, тоже является DOM-узлом, и находится в дереве DOM непосредственно перед <html>. На иллюстрациях выше этот факт скрыт, поскольку мы с этим узлом работать не будем, он никогда не нужен.

Даже сам объект document, формально, является DOM-узлом, самым-самым корневым.

Всего различают 12 типов узлов, но на практике мы работаем с четырьмя из них:

1. Документ – точка входа в DOM.
2. Элементы – основные строительные блоки.
3. Текстовые узлы – содержат, собственно, текст.
4. Комментарии – иногда в них можно включить информацию, которая не будет показана, но доступна из JS.

**[Возможности, которые дает DOM](http://learn.javascript.ru/dom-nodes" \l "vozmozhnosti-kotorye-daet-dom)**

Зачем, кроме красивых рисунков, нужна иерархическая модель DOM?

**DOM нужен для того, чтобы манипулировать страницей – читать информацию из HTML, создавать и изменять элементы.**

Узел HTML можно получить как document.documentElement, а BODY – как document.body.

Получив узел, мы можем что-то сделать с ним.

Например, можно поменять цвет BODY и вернуть обратно:

document.body.style.backgroundColor = 'red';

alert( 'Поменяли цвет BODY' );

document.body.style.backgroundColor = '';

alert( 'Сбросили цвет BODY' );

DOM предоставляет возможность делать со страницей всё, что угодно.

Позже мы более подробно рассмотрим различные свойства и методы DOM-узлов.

**[Особенности IE8-](http://learn.javascript.ru/dom-nodes" \l "osobennosti-ie8)**

IE8- не генерирует текстовые узлы, если они состоят только из пробелов.

То есть, такие два документа дадут идентичный DOM:

<!DOCTYPE HTML>

<html><head><title>О лосях</title></head><body>Правда о лосях</body></html>

И такой:

<!DOCTYPE HTML>

<html>

<head>

<title>О лосях</title>

</head>

<body>

Правда о лосях

</body>

</html>

Эта, с позволения сказать, «оптимизация» не соответствует стандарту и IE9+ уже работает как нужно, то есть как описано ранее.

Но, по большому счёту, для нас это отличие должно быть без разницы, ведь при работе с DOM/HTML мы в любом случае не должны быть завязаны на то, есть пробел между тегами или его нет. Мало ли, сегодня он есть, а завтра решили переформатировать HTML и его не стало.

К счастью, свойства и методы DOM, которые мы пройдём далее, вполне позволяют писать код, который будет работать корректно во всех версиях браузеров. Так что знать об этом отличии надо, если вы хотите поддерживать старые IE, но проблем оно нам создавать не будет.

**[Итого](http://learn.javascript.ru/dom-nodes" \l "itogo)**

* DOM-модель – это внутреннее представление HTML-страницы в виде дерева.
* Все элементы страницы, включая теги, текст, комментарии, являются узлами DOM.
* У элементов DOM есть свойства и методы, которые позволяют изменять их.
* IE8- не генерирует пробельные узлы.

Кстати, DOM-модель используется не только в JavaScript, это известный способ представления XML-документов.

В следующих главах мы познакомимся с DOM более плотно.