Что же такое DOM?

> **HTML5 в Мире JavaScript с Венгры**
>
> Практическое руководство как наладить общение и сделать веб-страницы такими, как вам нужно.

Эта цитата из книги Эрика Фримена и Элизабет Робсон «Изучаем программирование на JavaScript» как нельзя лучше описывает, насколько отличаются *HTML* и *JavaScript*. Если заглянуть в любой справочник по веб-технологиям, то там мы найдём, что *HTML* (*HyperText Markup Language*) — это код, который используется для структурирования и отображения веб-страницы и её контента, а *JavaScript* — это язык программирования, который описывает алгоритмы и логику приложения.

Но есть механизм, по которому *HTML* и *JavaScript* взаимодействуют между собой. Это особое представление веб-страницы в так называемой «объектной модели документа» или *DOM*.

В справочниках *DOM* описывается так:

> DOM (от англ. Document Object Model — «объектная модель документа») — это независящий от платформы и языка программный интерфейс (API), который позволяет программам и скриптам получить доступ к содержимому HTML-, XHTML- и XML-документов, а также изменять содержимое, структуру и оформление таких документов.
>
> *Любой документ известной структуры с помощью DOM может быть представлен в виде дерева узлов, каждый узел которого представляет собой элемент, атрибут, текстовый, графический или любой другой объект. Узлы связаны между собой отношениями «родительский-дочерний».*
>
> *Wikipedia*

DOM — это универсальный механизм, разные языки программирования его используют, чтобы получить доступ к разным типам документов.

Мы же будем рассматривать *DOM* как представление веб-страницы, которое доступно для изменения через *JavaScript*.

Используя *DOM*, *JavaScript* может изменять атрибуты элементов, создавать новые или удалять существующие элементы с веб-страницы и т. п.

Важно, что все изменения в *DOM* немедленно появляются на веб-странице. Фактически, через *JavaScript*-код вы изменяете содержимое веб-страницы «на ходу».

Но прежде чем выяснить, как же управлять веб-страницей при помощи *JavaScript*, разберёмся, кто и как формирует *DOM*.

Откуда же берётся DOM?

DOM создаёт браузер при загрузке веб-страницы. Он не только разбирает и выводит на экран *HTML*, но и создаёт набор объектов, которые представляют эту разметку. Созданные объекты скрываются в виде узлов Объектной модели документа.

DOM-узлы — это обычные *JavaScript*-объекты. Для наследования они используют классы, основанные на прототипах.

Чтобы проверить свойства DOM-узла, можно использовать команду `console.dir()`. Эта команда выводит в консоль *JSON*-представление переданного объекта, что удобно для анализа его свойств. Подробно команда `console.dir()`, описывается в справочнике по API браузера.

Например, вот как выглядит тело документа — тег `<body>` в виде объекта. В консоли мы ввели `console.dir(document.body)`.

Тег `<body>` в виде объекта

Каждый *DOM*-узел принадлежит определённому встроенному классу. Есть базовый класс для всех узлов — класс *Node*. От него наследуются другие типы узлов. Есть узлы-элементы (класс *Element*), текстовые узлы (класс *Text*), узлы-атрибуты (класс *Attr*), узлы-комментарии (класс *Comment*) и другие.

Схема взаимных связей классов DOM-узлов изображена на рисунке ниже:

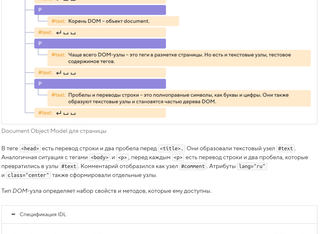


Схема взаимных связей классов DOM-узлов

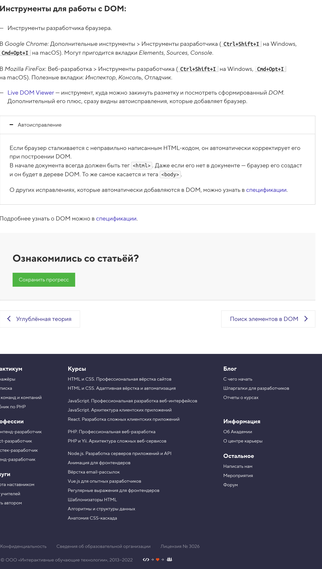
Теги образуют узлы-элементы (*Element*), текст внутри тегов формирует текстовые узлы (*Text*), атрибуты тегов — узлы-атрибуты (*Attr*), комментарии в разметке преобразуются в узлы-комментарии (*Comment*).

Пробелы `␣` и переводы строки `↵` (в JavaScript он обозначается как `\n`) — это полноправные символы, как буквы и цифры. Они также образуют текстовые узлы и становятся частью дерева *DOM*.

Рассмотрим на примере. У нас есть страница:

```
<!DOCTYPE HTML>
<html lang="ru">
<head>
  <title>Вся правда о DOM</title>
</head>
<body>
  <!-- Содержимое страницы -->
  <h1 class="center">Document Object Model</h1>
  <p>Структура DOM похода на перевёрнутое дерево: корень расположен наверху, а листья - внизу.</p>
  <p>Корень DOM – объект document.</p>
  <p>Чаще всего DOM-узлы – это теги в разметке страницы. Но есть и текстовые узлы, тестовое содержимое тегов.</p>
  <p>Пробелы и переводы строки – это полноправные символы, как буквы и цифры. Они также образуют текстовые узлы и становятся частью дерева DOM.</p>
</body>
</html>
```

DOM для этой страницы выглядит так:



Document Object Model для страницы

В теге `<head>` есть перевод строки и два пробела перед `<title>`. Они образовали текстовый узел `#text`. Аналогичные сущности с тегами `<body>` и `<p>`, перед каждым `<p>` есть перевод строки и два пробела, которые превратились в узлы `#text`. Комментарий отобразился как узел `#comment`. Атрибуты `lang="ru"` и `class="center"`, также сформировали отдельные узлы.

Тип *DOM*-узла определяет набор свойств и методов, которые ему доступны.

Спецификация IDL

Подробности обо всех типах *DOM*-узлов можно найти в спецификации. Для описания классов *DOM* в ней используется специальный язык *Interface description language (IDL)*.

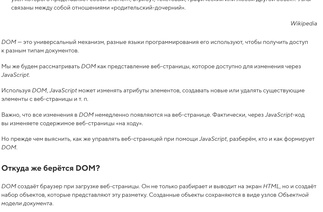
Всё, что есть в HTML: теги, атрибуты тегов, текст, комментарии — есть в DOM. Более того, в DOM содержится масса полезных для разработчика вещей, которые упрощают работу с веб-страницей. О них мы расскажем немного позднее.

Объект `document`

На верхнем уровне *DOM* всегда находится объект `document`. Этот объект используется в JavaScript, чтобы получить доступ ко всей модели *DOM*.

Часто в учебниках по JavaScript можно увидеть упрощённую схему *DOM* — это дерево, которое состоит только из узлов-элементов. Этот вариант, во-первых, прощё для восприятия, а во-вторых, JavaScript взаимодействует как правило, именно с ними. Чтобы получить и поменять содержимое тегов или атрибутов, используют методы и свойства узлов-элементов.

Перерисуем *DOM* для нашего примера на упрощённый вариант и добавим корневой объект `document`:



Упрощённая модель *DOM* с корневым элементом `document`

Отметим важные моменты для *DOM*:

- *DOM* — это древовидная структура, в которой узлы связаны отношениями «родительский-дочерний»;
- корень в *DOM*-дереве может быть только один — объект `document`;
- у каждого *DOM*-узла может быть только один родитель;
- *DOM*-узлы — это *JavaScript*-объекты разных типов (тип узла определяет набор свойств и методов, которые ему доступны).

Инструменты для работы с DOM:

- Инструменты разработчика браузера.

В *Google Chrome*. Дополнительные инструменты > Инструменты разработчика (`Ctrl+Shift+I` на Windows, `Cmd+Opt+I` на macOS). Могут пригодится вкладки *Elements*, *Sources*, *Console*.

В *Mozilla FireFox*. Веб-разработка > Инструменты разработчика (`Ctrl+Shift+I` на Windows, `Cmd+Opt+I` на macOS). Полезные вкладки: *Инспектор*, *Консоль*, *Отладчик*.

- Live DOM Viewer — инструмент, куда можно закинуть разметку и посмотреть сформированный *DOM*. Дополнительный его плюс, сразу видны автоисправления, которые добавляет браузер.

Автоисправление

Если браузер сталкивается с неправильно написанным HTML-кодом, он автоматически корректирует его при построении DOM.

В начале документа всегда должен быть тег `<html>`. Даже если его нет в документе — браузер его создаст и он будет в дереве DOM. То же самое касается и тега `<body>`.

О других *исправлениях*, которые автокатически добавляются в DOM, можно узнать в спецификации.

Подробнее узнать о DOM можно в спецификации.

Ознакомились со статьёй?

Сохранить прогресс