**Структура HTML-документа**

Язык HTML следует правилам, которые содержатся в файле объявления типа документа *(Document Type Definition, или DTD)*. DTD представляет собой XML-документ, определяющий, какие элементы, атрибуты и их значения действительны для конкретного типа HTML. Для каждой версии HTML есть свой DTD.

**DOCTYPE** отвечает за корректное отображение веб-страницы браузером. DOCTYPE определяет не только версию HTML (например, html), но и соответствующий DTD-файл в Интернете.

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head> <!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>...</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

</head>

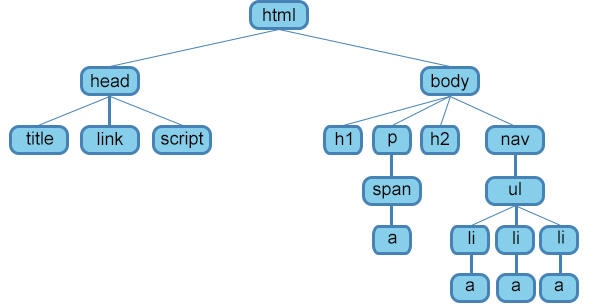
<body> <!-- Основная часть документа -->

</body>

</html>

HTML

Элементы, находящиеся внутри элемента <html>, образуют дерево документа, так называемую **объектную модель документа, DOM (document object model)**. При этом элемент <html> является корневым элементом.

РИС. 1. ПРОСТЕЙШАЯ СТРУКТУРА ВЕБ-СТРАНИЦЫ

Чтобы разобраться во взаимодействии элементов веб-страницы, необходимо рассмотреть так называемые **«родственные отношения»** между элементами. Отношения между множественными вложенными элементами подразделяются на родительские, дочерние и сестринские.

* **Предок** — элемент, который заключает в себе другие элементы. На рисунке 1 предком для всех элементов является <html>. В то же время элемент <body> является предком для всех содержащихся в нем элементов: <h1>, <p>, <span>, <nav> и т.д.
* **Потомок** — элемент, расположенный внутри одного или более типов элементов. Например, <body> является потомком <html>, а элемент <p> является потомком одновременно для <body> и <html>.
* **Родительский элемент** — элемент, связанный с другими элементами более низкого уровня, и находящийся на дереве выше их. На рисунке 1 <html> является родительским только для <head> и <body>. Элемент <p> является родительским только для <span>.
* **Дочерний элемент** — элемент, непосредственно подчиненный другому элементу более высокого уровня. На рисунке 1 только элементы <h1>, <h2>, <p> и <nav> являются дочерними по отношению к <body>.
* **Сестринский элемент** — элемент, имеющий общий родительский элемент с рассматриваемым, так называемые элементы одного уровня. На рисунке 1 <head> и <body> — элементы одного уровня, так же как и элементы <h1>, <h2> и <p> являются между собой сестринскими.

**1.1. Элемент <html>**

Является корневым элементом документа. Все остальные элементы содержатся внутри <html>...</html>. Все, что находится за пределами элемента, не воспринимается браузером как HTML-код и никак им не обрабатывается.

Для элемента доступен атрибут, описанный в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 1. АТРИБУТ ЭЛЕМЕНТА <HTML> | |
| **Атрибут** | **Описание, принимаемое значение** |
| manifest | С помощью значения атрибута указывается путь к документу кэша манифеста, например: <html manifest="about\_company.appcache"> |

### 1.2. Элемент <head>

Раздел <head>...</head> содержит техническую информацию о странице: заголовок, описание, ключевые слова для поисковых машин, кодировку и т.д. Введенная в нем информация не отображается в окне браузера, однако содержит данные, которые указывают браузеру, как следует обрабатывать страницу.

Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 1.2.1. Элемент <title>

Обязательным элементом раздела <head> является <title>. Текст, размещенный внутри элемента <title>, отображается в строке заголовка веб-браузера.

Длина заголовка должна быть не более 60 символов, чтобы полностью поместиться в заголовке. Текст заголовка должен содержать максимально полное описание содержимого веб-страницы.

Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 1.2.2. Элемент <meta>

Необязательным элементом раздела <head> является элемент <meta>. С его помощью можно задать описание содержимого страницы и ключевые слова для поисковых машин, автора HTML-документа и прочие свойства метаданных.

Элемент <head> может содержать несколько элементов <meta>, потому что в зависимости от используемых атрибутов они несут различную информацию.

<meta name="description" content="Описание содержимого страницы">

<meta name="keywords" content="Ключевые слова через запятую">

HTML

Описание содержимого страницы и ключевые слова одновременно можно указывать на нескольких языках, например, на русском и английском:

<meta name="description" lang="ru" content="Описание содержимого страницы">

<meta name="description" lang="en" content="Description">

<meta name="keywords" lang="ru" content="Ключевые слова через запятую">

<meta name="keywords" lang="en" content="Keywords">

HTML

С помощью элемента <meta> можно запретить или разрешить индексацию веб-страницы поисковыми машинами:

Индексация и переход по ссылкам разрешены:

<meta name="robots" content="index, follow">

HTML

Индексация разрешена, переход по ссылкам запрещен:

<meta name="robots" content="index, nofollow">

HTML

Индексация и переход по ссылкам запрещены:

<meta name="robots" content="noindex, nofollow">

HTML

Для автоматической перезагрузки страницы через заданный промежуток времени (например, через 30 секунд) нужно воспользоваться значением refresh:

<meta http-equiv="refresh" content="30">

HTML

Чтобы перебросить посетителя на другую страницу, нужно указать URL-адрес в параметре url:

<meta http-equiv="refresh" content="0; url=http://yandex.ru/">

HTML

Для элемента <meta> доступны атрибуты, описанные в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 2. АТРИБУТЫ ЭЛЕМЕНТА <META> | |
| **Атрибут** | **Описание, принимаемое значение** |
| charset | Указывает кодировку символов для текущего HTML-документа: <meta charset="UTF-8"> |
| content | Содержит произвольный текст, который определяет значение, ассоциируемое с атрибутом http-equiv или name, в зависимости от их значения. |
| http-equiv | Контролирует действия браузера на данной веб-странице (эквивалент HTTP заголовков). При отображении страницы браузер будет следовать инструкциям, заданным в атрибуте: default-style указывает предпочтительный стиль для использования на странице. Атрибут content должен содержать идентификатор элемента <link>, который ссылается на таблицу стилей CSS, или идентификатор элемента <style>, содержащего таблицу стилей. refresh указывает время в секундах до перезагрузки страницы или время до перенаправления на другую страницу, если в атрибуте content после указания времени идет строка "url=адрес\_страницы". Автоматическая перезагрузка страницы через заданный промежуток времени, в данном примере, через 30 секунд: <meta http-equiv="refresh" content="30"> Если необходимо сразу перебросить посетителя на другую страницу, то можно указать URL-адрес в параметре url: <meta http-equiv="refresh" content="0; url=http://mail.ru/"> |
| name | Ассоциируется со значением, содержащемся в атрибуте content. Не должен использоваться в случае, если для элемента уже заданы атрибуты http-equiv, charset или itemprop. application-name указывает название веб-приложения, используемого на странице. author указывает имя автора документа в свободном формате. description определяет краткое описание к содержимому страницы, например: <meta name="description" content="Описание содержимого страницы"> generator указывает один из пакетов программного обеспечения, используемого для создания документа, например: <meta name="generator" content="WordPress 4.0"> . keywords содержит список ключевых слов, разделенных запятыми, соответствующих содержимому страницы, например: <meta name="keywords" content="Ключевые слова через запятую">. Также атрибут *name* может принимать следующие значения из расширенной спецификации, такие как *creator*, *googlebot*, *publisher*, *robots*, *slurp*, *viewport*, хотя ни одно из них еще не было официально принято. |

#### 1.2.3. Элемент <style>

Внутри этого элемента задаются стили, которые используются на странице. Для задания стилей в HTML-документе используется язык CSS. Таких элементов на странице может быть несколько.

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 3. АТРИБУТЫ ЭЛЕМЕНТА <STYLE> | |
| **Атрибут** | **Описание, принимаемое значение** |
| media | Определяет, для какого типа устройства предназначены те или иные каскадные таблицы стилей. Внутри таблицы стилей CSS специфичные правила для определенных устройств прописываются с помощью media query. |
| nonce | Генерируемая случайным образом на сервере строковая переменная, которая устанавливает правила использования встроенных стилей с целью защиты контента. Значение атрибута — строка текста. |
| type | Определяет MIME-тип подключаемого файла со стилями, в данном случае type="text/css". |

Элемент может содержать код форматирования как самих элементов веб-страницы, так и веб-страницы целиком.

<style>

.paper {

width: 200px;

height: 300px;

background-color: #ef4444;

color: #666666;

}

</style>

HTML

Чтобы подключить к элементу заданный стиль, необходимо через атрибут class (или id) присвоить элементу соответствующее название:

<div class="paper">

...

</div>

HTML

CSS-код можно встраивать непосредственно в элемент разметки в виде значение атрибута style, например:

<p style="color: #666666; background-color: #ef4444; padding: 20px;">

#### 1.2.4. Элемент <link>

Задать стили для документа можно также при помощи другого способа — записать их в отдельный файл с расширением .css, например, style.css.

Подключить файл со стилями к веб-странице можно двумя способами:

* через директиву @import url
* с использованием элемента <link>. Данный способ является предпочтительным.

<!DOCTYPE html>

<html>

<head>

<style>

@import url(style.css);

</style>

<meta>

<title> </title>

</head>

HTML

Элемент <link> определяет отношение между текущей страницей и другими документами. Таких элементов на странице может быть несколько.

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

HTML

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 4. АТРИБУТЫ ЭЛЕМЕНТА <LINK> | |
| **Атрибут** | **Описание, принимаемое значение** |
| crossorigin | Указывает, должен ли использоваться CORS (технология браузеров, которая позволяет предоставить веб-странице доступ к ресурсам другого домена) при извлечении изображения с сайта. anonymous — в кросс-доменный запрос браузер автоматически добавляет заголовок Origin, содержащий имя домена, с которого осуществлён запрос. Если сервер не ответит CORS-заголовком Access-Control-Allow-Origin: \* (или имя домена вместо звёздочки), то загрузка изображения будет заблокирована. use-credentials — если сервер не предоставляет учётные данные с помощью Access-Control-Allow-Credentials: true, то загрузка изображения будет заблокирована. |
| href | Основной атрибут элемента, в качестве значения выступает путь к файлу со стилями. |
| hreflang | Определяет язык текста в документе, на который идет ссылка. |
| media | Определяет тип устройства, к которым должен быть применен ресурс ссылки. |
| nonce | Генерируемая случайным образом на сервере строковая переменная, которая устанавливает правила использования встроенных стилей с целью защиты контента. Значение атрибута — строка текста. |
| rel | Атрибут определяет отношения между текущим документом и документом, на который идет ссылка. alternate — ссылка на тот же документ, но в другом формате (например, страницы для печати, перевод, зеркало, лента в формате RSS или Atom), <link rel="alternate stylesheet" type="text/css" title="small\_font" href="small-font.css">. <link rel="alternate" type="application/rss+xml" title="my\_RSS" href="index.xml"> <link rel="alternate" type="application/atom+xml" title="News" href="/atom.xml?type=news"> archives — указывает на то, что документ по ссылке представляет исторический интерес. Ссылка может указывать на коллекцию записей, документов и других материалов. author ссылка на страницу об авторе документа или на страницу с контактными данными автора. bookmark ссылка на ближайшего предка статьи, являющегося связующим звеном, или на раздел статьи, наиболее тесно связанных с элементом, если нет предка. external используется для указания того, что страница на которую ведет ссылка не являются частью данного сайта. first указывает ссылку, ведущую на первый документ из последовательности документов. help ссылка на документ со справкой. icon определяет путь к иконке, которая будет использована для текущего документа. last указывает ссылку, ведущую на последний документ в последовательности документов. license ссылка на сведения об авторских правах для документа. next указывает, что этот документ является частью серии, и что ссылка ведет на следующий документ в этой серии. <link rel="next" href="/next.html" type="text/html" title="Следующая страница"> nofollow указывает на то, что ссылка не одобрена автором страницы или что ссылка носит коммерческий характер. noreferrer указывает на то, что заголовок запроса клиента, содержащий url источника запроса, не должен передаваться при переходе по ссылке. pingback указывает адрес пингбэк-сервера, что дает возможность для блога автоматически оповещать сайты, ссылающиеся на него. prefetch указывает, что следует заранее кэшировать файл, на который ведет ссылка. prev указывает, что этот документ является частью серии, и что ссылка ведет на предыдущий документ в этой серии. <link rel="prev" href="/next.html" type="text/html" title="Предыдущая страница"> search указывает, что ссылаемый документ содержит интерфейс поиска и связанных с ним ресурсов. sidebar указывает, что ссылаемый документ, если это возможно, будет показан в дополнительном контексте браузера, и некоторые браузеры при щелчке по гиперссылке открывают окно для добавления ссылки в панель закладок. stylesheet ссылка на внешний файл, который будет использоваться в качестве таблицы стилей для данного документа. tag указывает на то, что метка, на которую ведет гиперссылка, относится к данному документу. up указывает, что страница является частью иерархической структуры, и что гиперссылка ведет на более высокий уровень ресурса в структуре. |
| sizes | Указывает размер иконок для визуального отображения. Атрибут sizes используется только совместно с rel="icon", и может принимать следующий значения: ширинахвысота — определяет список размеров, разделенных пробелами, каждый размер должен быть в формате — ширинахвысота (размеры иконки задаются в пикселях), например: <link rel="icon" href="favicon.png" sizes="16x16 32х32" type="image/png">; any — иконка может масштабироваться до любого размера. |
| title | Определяет заголовок ссылки или название набора альтернативных таблиц стилей. Значение атрибута — текст. |
| type | Определяет MIME-тип документа, на который идет ссылка. В данном случае он принимает значение "text/css". |

# DOM-дерево

Основой HTML-документа являются теги.

В соответствии с объектной моделью документа («Document Object Model», коротко DOM), каждый HTML-тег является объектом. Вложенные теги являются «детьми» родительского элемента. Текст, который находится внутри тега, также является объектом.

DOM-узел можно создать двумя методами:

**document.createElement(tag)**

Создаёт новый *элемент* с заданным тегом:

let div = document.createElement('div');

**document.createTextNode(text)**

Создаёт новый *текстовый узел* с заданным текстом:

let textNode = document.createTextNode('А вот и я');

Большую часть времени нам нужно создавать узлы элементов, такие как div для сообщения.

Чтобы наш div появился, нам нужно вставить его где-нибудь в document. Например, в document.body.

Для этого есть метод append, в нашем случае: document.body.append(div).

метод .**append**() позволяет вставить содержимое, указанное в параметре метода в конец каждого выбранного элемента в наборе совпавших элементов. Метод вставляет указанное содержимое как последний дочерний элемент. Методы .**append**() и .appendTo() выполняют одну и ту же задачу, главное отличие этих методов заключается в синтаксисе, а именно, в размещении вставляемого контента и в указании цели, куда это содержимое вставляется.

* document и body - это объекты?

В соответствии с объектной моделью **документа** («**Document** Object Model», коротко DOM), каждый HTML-тег является **объектом**. Вложенные теги являются «детьми» родительского элемента. Текст, который находится внутри тега, также является **объектом**. Все **эти** **объекты** доступны при помощи JavaScript, мы можем использовать их для изменения страницы. Например, **document**.**body**– **объект** для тега< **body**>.

Переменная – **объект JavaScript, предназначенный для хранения задаваемых значений**. Основной элемент, без которого невозможно представить методы, функции, а также команды. Обязательно идентифицируются посредством уникальных имен. Они называются идентификаторами.