

# Основы HTML

Основы HTML содержат основные правила языка HTML, описание структуры HTML-страницы, отношения в структуре HTML-документа между HTML-элементами.

HTML-документ — это обычный текстовый документ, может быть создан как в обычном текстовом редакторе (**Блокнот**), так и в специализированном, с подсветкой кода (**Notepad++**, **Visual Studio Code** и т.п.). HTML-документ имеет расширение `.html`.

HTML-документ состоит из дерева HTML-элементов и текста. Каждый элемент обозначается в исходном документе начальным (открывающим) и конечным (закрывающим) тегом (за редким исключением).

**Начальный тег** показывает, где начинается элемент, конечный — где заканчивается. **Закрывающий тег** образуется путем добавления слэша `/` перед именем элемента: `<имя элемента> ... </имя элемента>`. Между начальным и закрывающим тегами находится содержимое элемента — контент.

Элементы, представленные одиночными тегами, не могут хранить в себе содержимого напрямую, оно прописывается как значение атрибута, например, элемент `<input type="button" value="Кнопка">` создаст кнопку с текстом **Кнопка** внутри.

Элементы могут вкладываться друг в друга, например, `<p><i>Текст</i></p>`. При вложении следует соблюдать порядок их закрытия (**принцип «матрёшки»**), например, следующая запись будет неверной:

```
<p><i>Текст</p></i>
```

HTML-элементы могут иметь атрибуты (глобальные, применяемые для всех HTML-элементов, и собственные). Атрибуты прописываются в открывающем теге элемента и содержат имя и значение, указываемые в формате `имя атрибута="значение"`. Атрибуты позволяют изменять свойства и поведение элемента, для которого они заданы.

Каждому элементу можно присвоить несколько значений `class` и только одно значение `id`. Множественные значения `class` записываются через пробел, `<div class="nav top">`. Значения `class` и `id` должны состоять только из букв, цифр, дефисов и нижних подчеркиваний и должны начинаться только с букв или цифр.

Браузер просматривает (интерпретирует) HTML-документ, выстраивая его структуру (DOM) и отображая ее в соответствии с инструкциями, включенными в этот файл (таблицы стилей, скрипты). Если разметка правильная,

то в окне браузера будет отображена HTML-страница, содержащая HTML-элементы — заголовки, таблицы, изображения и т.д.

Процесс интерпретации (**парсинг**) начинается прежде, чем веб-страница полностью загружена в браузер. Браузеры обрабатывают HTML-документы последовательно, с самого начала, при этом обрабатывая CSS и соотнося таблицы стилей с элементами страницы.

HTML-документ состоит из двух разделов — заголовка — содержимое элемента `<head>` и содержательной части — содержимое `<body>` .

## 1. Структура HTML-документа

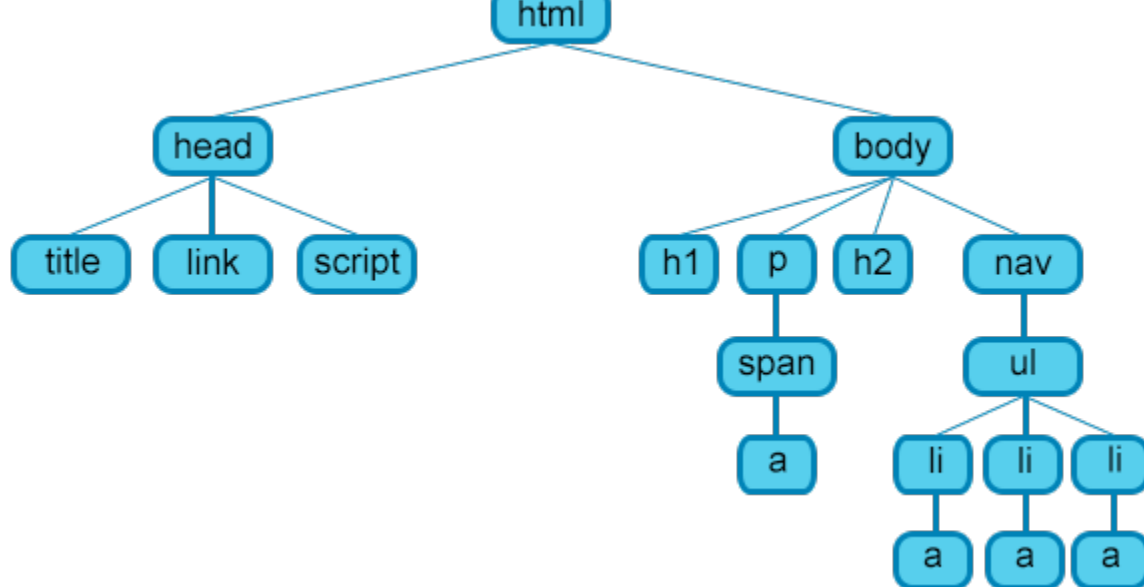
Язык HTML следует правилам, которые содержатся в файле объявления типа документа (*Document Type Definition, или DTD*). DTD представляет собой XML-документ, определяющий, какие элементы, атрибуты и их значения действительны для конкретного типа HTML. Для каждой версии HTML есть свой DTD.


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

HTML

```
<!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>
```

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 WEB-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 включает 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](#).

ТАБЛИЦА 1. АТРИБУТ ЭЛЕМЕНТА <HTML>

Атрибут	Описание, принимаемое значение
manifest	С помощью значения атрибута указывается путь к документу кэша манифеста, например: <code>&lt;html manifest="about_company.appcache"&gt;</code>

## 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](#).

### 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](#).

### 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](#).

ТАБЛИЦА 2. АТРИБУТЫ ЭЛЕМЕНТА `<META>`

Атрибут	Описание, принимаемое значение
<code>charset</code>	Указывает кодировку символов для текущего HTML-документа: <code>&lt;meta charset="UTF-8"&gt;</code>
<code>content</code>	Содержит произвольный текст, который определяет значение, ассоциируемое с атрибутом <code>http-equiv</code> или <code>name</code> , в зависимости от их значения.
<code>http-equiv</code>	<p>Контролирует действия браузера на данной веб-странице (эквивалент HTTP заголовков). При отображении страницы браузер будет следовать инструкциям, заданным в атрибуте:</p> <ul style="list-style-type: none"><li><code>default-style</code> указывает предпочтительный стиль для использования на странице. Атрибут <code>content</code> должен содержать идентификатор элемента <code>&lt;link&gt;</code> , который ссылается на таблицу стилей CSS, или идентификатор элемента <code>&lt;style&gt;</code> , содержащего таблицу стилей.</li><li><code>refresh</code> указывает время в секундах до перезагрузки страницы или время до перенаправления на другую страницу, если в атрибуте <code>content</code> после указания времени идет строка <code>"url=адрес_страницы"</code> . Автоматическая перезагрузка страницы через заданный промежуток времени, в данном примере, через 30 секунд: <code>&lt;meta http-equiv="refresh" content="30"&gt;</code></li></ul> <p>Если необходимо сразу перебросить посетителя на другую страницу, то можно указать URL-адрес в</p>

	параметре <code>url</code> : <code>&lt;meta http-equiv="refresh" content="0; url=http://mail.ru/"&gt;</code>
name	<p>Ассоциируется со значением, содержащемся в атрибуте <code>content</code> . Не должен использоваться в случае, если для элемента уже заданы атрибуты <code>http-equiv</code> , <code>charset</code> или <code>itemprop</code> .</p> <p><code>application-name</code> указывает название веб-приложения, используемого на странице.</p> <p><code>author</code> указывает имя автора документа в свободном формате.</p> <p><code>description</code> определяет краткое описание к содержимому страницы, например:  <code>&lt;meta name="description" content="Описание содержимого страницы"&gt;</code></p> <p><code>generator</code> указывает один из пакетов программного обеспечения, используемого для создания документа, например:  <code>&lt;meta name="generator" content="WordPress 4.0"&gt;</code> .</p> <p><code>keywords</code> содержит список ключевых слов, разделенных запятыми, соответствующих содержимому страницы, например:  <code>&lt;meta name="keywords" content="Ключевые слова через запятую"&gt;</code> .</p> <p><i>Также атрибут <code>name</code> может принимать следующие значения из расширенной спецификации, такие как <code>creator</code> , <code>googlebot</code> , <code>publisher</code> , <code>robots</code> , <code>slurp</code> , <code>viewport</code> , хотя ни одно из них еще не было официально принято.</i></p>

### 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](#).

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 <code>media query</code> .
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 <code>type="text/css"</code> .

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 значения атрибута `style`, например:

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

HTML

#### 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.

HTML

```
<!DOCTYPE html>
<html>
<head>
<style>
@import url(style.css);
</style>
<meta>
<title> </title>
</head>
```

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

HTML

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

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [глобальные атрибуты](#).

ТАБЛИЦА 4. АТРИБУТЫ ЭЛЕМЕНТА `<LINK>`

Атрибут	Описание, принимаемое значение
crossorigin	<p>Указывает, должен ли использоваться CORS (технология браузеров, которая позволяет предоставить веб-странице доступ к ресурсам другого домена) при извлечении изображения с сайта.</p> <p><code>anonymous</code> — в кросс-доменный запрос браузер автоматически добавляет заголовок <code>Origin</code>, содержащий имя домена, с которого осуществлён запрос. Если сервер не ответит CORS-заголовком <code>Access-Control-Allow-Origin: *</code> (или имя домена вместо звёздочки), то загрузка изображения будет заблокирована.</p> <p><code>use-credentials</code> — если сервер не предоставляет учётные данные с помощью <code>Access-Control-Allow-Credentials: true</code>, то загрузка изображения будет заблокирована.</p>

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	<p>Атрибут определяет отношения между текущим документом и документом, на который идет ссылка.</p> <p>alternate — ссылка на тот же документ, но в другом формате (например, страницы для печати, перевод, зеркало, лента в формате RSS или Atom),</p> <pre>&lt;link rel="alternate stylesheet" type="text/css" title="small_font" href="small-font.css"&gt; .</pre> <pre>&lt;link rel="alternate" type="application/rss+xml" title="my_RSS" href="index.xml"&gt;</pre> <pre>&lt;link rel="alternate" type="application/atom+xml" title="News" href="/atom.xml?type=news"&gt;</pre> <p>archives — указывает на то, что документ по ссылке представляет исторический интерес. Ссылка может указывать на коллекцию записей, документов и других материалов.</p> <p>author ссылка на страницу об авторе документа или на страницу с контактными данными автора.</p> <p>bookmark ссылка на ближайшего предка статьи, являющегося связующим звеном, или на раздел статьи, наиболее тесно связанных с элементом, если нет предка.</p> <p>external используется для указания того, что страница на которую ведет ссылка не являются частью данного сайта.</p> <p>first указывает ссылку, ведущую на первый документ из последовательности документов.</p> <p>help ссылка на документ со справкой.</p> <p>icon определяет путь к иконке, которая будет использована для текущего документа.</p> <p>last указывает ссылку, ведущую на последний документ в последовательности документов.</p> <p>license ссылка на сведения об авторских правах для документа.</p> <p>next указывает, что этот документ является частью серии, и что ссылка ведет на следующий документ в этой серии.</p> <pre>&lt;link rel="next" href="/next.html" type="text/html" title="Следующая страница"&gt;</pre> <p>nofollow указывает на то, что ссылка не одобрена автором страницы или что ссылка носит коммерческий характер.</p> <p>noreferrer указывает на то, что заголовок запроса клиента, содержащий url источника запроса, не должен передаваться при переходе по ссылке.</p> <p>pingback указывает адрес пингбэк-сервера, что дает возможность для блога автоматически оповещать сайты, ссылающиеся на него.</p> <p>prefetch указывает, что следует заранее кэшировать файл, на который ведет ссылка.</p> <p>prev указывает, что этот документ является частью серии, и что ссылка ведет на предыдущий документ в этой серии.</p> <pre>&lt;link rel="prev" href="/next.html" type="text/html" title="Предыдущая страница"&gt;</pre> <p>search указывает, что ссылаемый документ содержит интерфейс поиска и связанных с ним ресурсов.</p> <p>sidebar указывает, что ссылаемый документ, если это возможно, будет показан в дополнительном</p>

	<p>контексте браузера, и некоторые браузеры при щелчке по гиперссылке открывают окно для добавления ссылки в панель закладок.</p> <p><code>stylesheet</code> ссылка на внешний файл, который будет использоваться в качестве таблицы стилей для данного документа.</p> <p><code>tag</code> указывает на то, что метка, на которую ведет гиперссылка, относится к данному документу.</p> <p><code>up</code> указывает, что страница является частью иерархической структуры, и что гиперссылка ведет на более высокий уровень ресурса в структуре.</p>
<code>sizes</code>	<p>Указывает размер иконок для визуального отображения. Атрибут <code>sizes</code> используется только совместно с <code>rel="icon"</code>, и может принимать следующие значения:</p> <p><code>ширинаxвысота</code> — определяет список размеров, разделенных пробелами, каждый размер должен быть в формате <code>— ширинаxвысота</code> (размеры иконки задаются в пикселях), например:</p> <pre>&lt;link rel="icon" href="favicon.png" sizes="16x16 32x32" type="image/png"&gt; ;</pre> <p><code>any</code> — иконка может масштабироваться до любого размера.</p>
<code>title</code>	<p>Определяет заголовок ссылки или название набора альтернативных таблиц стилей. Значение атрибута — текст.</p>
<code>type</code>	<p>Определяет MIME-тип документа, на который идет ссылка. В данном случае он принимает значение <code>"text/css"</code>.</p>

### 1.2.5. Элемент <script>

Элемент `<script>` позволяет присоединять к документу различные сценарии. Текст сценария может располагаться либо внутри этого элемента, либо во внешнем файле. Если текст сценария расположен во внешнем файле, то он подключается с помощью атрибутов элемента.

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [глобальные атрибуты](#).

ТАБЛИЦА 5. АТРИБУТЫ ЭЛЕМЕНТА <SCRIPT>

Атрибут	Описание, принимаемое значение
<code>async</code>	Атрибут указывает на то, что сценарий будет выполняться асинхронно с остальной частью страницы (сценарий начнет выполняться одновременно с загрузкой страницы).
<code>charset</code>	Определяет кодировку символов
<code>crossorigin</code>	<p>Определяет, будет ли использоваться CORS при загрузке внешних скриптов (с использованием атрибута <code>src</code>).</p> <p><code>anonymous</code> — перед загрузкой скрипта в кросс-доменный запрос браузер автоматически добавляет</p>

	<p>заголовок Origin, при этом не передаются параметры доступа (cookie, сертификат X.509, логин/пароль для базовой аутентификации по HTTP). Если в ответе сервера отсутствует заголовок Access-Control-Allow-Origin: имя домена , скрипт не будет загружен.</p> <p>use-credentials — перед загрузкой скрипта в кросс-доменный запрос браузер автоматически добавляет заголовок Origin с указанием параметров доступа (cookie, SSL-сертификат или пары логин/пароль). Если в ответе сервера отсутствует заголовок Access-Control-Allow-Credentials: true , скрипт не будет загружен.</p>
defer	Интерпретация сценариев откладывается до окончания отображения документа на устройстве пользователя.
nonce	Обеспечивает безопасность, защищая от атак с внедрением межсайтового скриптинга (XSS, cross site scripting). Устанавливает правила использования встроенных скриптов с помощью nonce-значений и хэшей. Во время рендеринга страницы браузер для каждого инлайн-скрипта вычисляет хэши и сравнивает с перечисленными в CSP. Загрузка с ресурсов, не входящих в «белый список», блокируется.
src	Указывает на месторасположение файла со сценарием, значение атрибута — это url файла, содержащего JavaScript-программу.
type	Используются для объявления языка сценария, использованного при составлении содержимого элемента.

### 1.3. Элемент <body>

В разделе <body> располагается все содержимое документа.

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [глобальные атрибуты](#).

ТАБЛИЦА 5. АТРИБУТЫ ЭЛЕМЕНТА <BODY>

Атрибут	Описание, принимаемое значение
onafterprint	Событие, срабатывающее после отправки страницы на печать или после закрытия окна печати.
onbeforeprint	Событие, срабатывающее перед отправкой страницы на печать.
onbeforeunload	Событие срабатывает, когда посетитель инициировал переход на другую страницу или нажал «закрыть окно». Позволяет отображать сообщение в диалоговом окне подтверждения, чтобы сообщить пользователю, хочет ли он остаться или покинуть текущую страницу.

onhashchange	Событие срабатывает, когда меняется hash-часть URL, например, когда пользователь перейдет с адреса example.domain/test.aspx#page1 на example.domain/test.aspx#page2 .
onmessage	Событие происходит, когда сообщение получено через источник события.
onoffline	Событие вызывается браузером в том случае, когда браузер определит, что соединение с интернет пропало.
ononline	Событие вызывается браузером в том случае, когда соединение с интернет возобновилось.
onpagehide	Событие происходит, когда пользователь покидает страницу посредством навигации, например, нажав на ссылку, обновив страницу, заполнив форму и т.д.
onpageshow	Событие происходит, когда пользователь переходит на веб-страницу, после события onload.
onunload	Событие срабатывает если страница не загрузилась по каким-либо причинам, либо при закрытии окна браузера.