

Содержание

1. Основы HTML
2. Структура HTML-документа
 - Элемент `<html>`
 - Элемент `<head>`
 - Элемент `<title>`
 - Элемент `<meta>`
 - Элемент `<style>`
 - Элемент `<link>`

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 тега: `<имя тега>...</имя тега>`. Между начальным и закрывающим тегами находится содержимое тега — контент.

Одиночные теги не могут хранить в себе содержимого напрямую, оно прописывается как значение атрибута, например, тег `<input type="button" value="Кнопка">` создаст кнопку с текстом **Кнопка** внутри.

Теги 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 — между тегами `<head>...</head>` и содержательной части — между тегами `<body>...</body>`.

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 теги, 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>
```

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

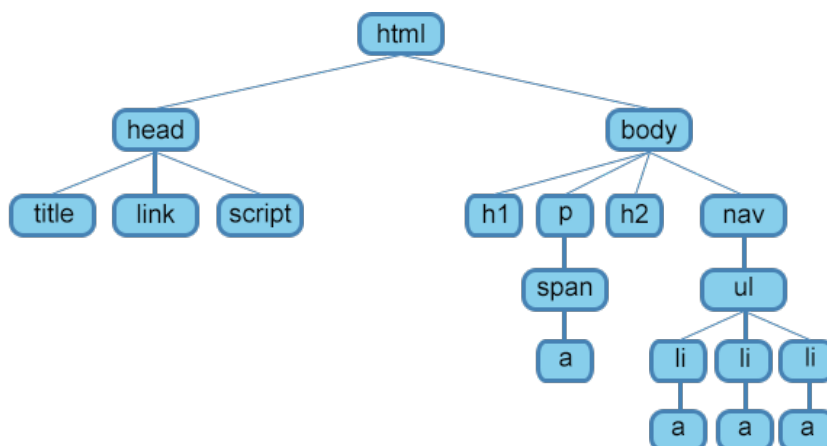


РИС. 1. ПРОСТЕЙШАЯ СТРУКТУРА ВЕБ-СТРАНИЦЫ

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 тегов: `<h1>`, `<p>`, ``, `<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>`. Тег `<p>` является родительским только для ``.

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. Элемент `<html>`

Является корневым элементом документа. Все остальные элементы содержатся внутри тегов `<html>...</html>`. Все, что находится за пределами тегов, не воспринимается браузером как код HTML и никак им не обрабатывается. Для элемента доступны атрибуты `manifest` и `xmlns`, а также глобальные атрибуты.

ТАБЛИЦА 1. АТТРИБУТЫ ТЕГА `<HTML>`

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

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.

3. Элемент `<title>`

Обязательным тегом раздела `<head>` является тег `<title>`. Текст, размещенный внутри этого тега, 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.

4. Элемент `<meta>`

Необязательным тегом раздела `<head>` является одинарный тег `<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">
```

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">
```

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

Индексация и переход по ссылкам разрешены:

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

Индексация разрешена, переход по ссылкам запрещен:

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

Индексация и переход по ссылкам запрещены:

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

Для автоматической перезагрузки страницы через заданный промежуток времени нужно воспользоваться значением `refresh`:

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

Страница будет перезагружена через 30 секунд. Чтобы перебросить посетителя на другую страницу, нужно указать URL-адрес в параметре `url`:

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

Для элемента `<meta>` доступны атрибуты `charset`, `content`, `http-equiv`, `name`, а также глобальные атрибуты.

ТАБЛИЦА 2. АТРИБУТЫ ТЕГА `<META>`

Атрибут	Описание, принимаемое значение
<code>charset</code>	Указывает кодировку символов для текущего HTML-документа: <code><meta charset="UTF-8"></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>	Контролирует действия браузера на данной веб-странице (эквивалент HTTP заголовков). При отображении страницы браузер будет следовать инструкциям, заданным в атрибуте: <code>default-style</code> указывает предпочтительный стиль для использования на странице. Атрибут <code>content</code> должен содержать идентификатор элемента <code><link></code> , который ссылается на таблицу стилей CSS, или идентификатор элемента <code><style></code> , содержащего таблицу стилей. <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><meta http-equiv="refresh" content="30"></code> Если необходимо сразу перебросить посетителя на другую страницу, то можно указать URL-адрес в параметре <code>url</code> : <code><meta http-equiv="refresh" content="0; url=http://mail.ru/"></code>
<code>name</code>	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> . <code>application-name</code> указывает название веб-приложения, используемого на странице. <code>author</code> указывает имя автора документа в свободном формате. <code>description</code> определяет краткое описание к содержимому страницы, например: <code><meta name="description" content="Описание содержимого страницы"></code> <code>generator</code> указывает один из пакетов программного обеспечения, используемого для создания документа, например: <code><meta name="generator" content="WordPress 4.0"></code> . <code>keywords</code> содержит список ключевых слов, разделенных запятыми, соответствующих содержимому страницы, например: <code><meta name="keywords" content="Ключевые слова через запятую"></code> . 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.

5. 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 `media`, `scoped`, `type`, а также глобальные атрибуты.

Внутри этого элемента можно записывать код форматирования как самих элементов веб-страницы, так и веб-страницы целиком.

```
<style type="text/css">
.paper {
width: 200px;
height: 300px;
background-color: #ef4444;
color: #666666;
}
</style>
```

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

```
<div class="paper">
...
</div>
```

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

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

ТАБЛИЦА 3. АТТРИБУТЫ ТЕГА `<STYLE>`

Атрибут	Описание, принимаемое значение
<code>media</code>	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> .
<code>nonce</code>	Генерируемая случайным образом на сервере строковая переменная, которая устанавливает правила использования встроенных стилей с целью защиты контента. Значение атрибута — строка текста.
<code>type</code>	Определяет MIME-тип подключаемого файла со стилями, в данном случае <code><style type="text/css"></code> .

6. 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`

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

с использованием элемента `<link>`. Элемент не требует закрывающего тега. Данный элемент определяет отношение между текущей страницей и другими документами. Таких элементов на странице может быть несколько. Запись будет иметь следующий вид:

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

Для элемента доступны атрибуты `href`, `hreflang`, `media`, `rel`, `type`, а также глобальные атрибуты.

ТАБЛИЦА 4. АТТРИБУТЫ ТЕГА <LINK>

Атрибут	Описание, принимаемое значение
<code>crossorigin</code>	Указывает, должен ли использоваться CORS (технология браузеров, которая позволяет предоставить веб-странице доступ к ресурсам другого домена) при извлечении изображения с сайта. <code>anonymous</code> — в кросс-доменный запрос браузер автоматически добавляет заголовок Origin, 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. <code>use-credentials</code> — если сервер не предоставляет учётные данные с помощью <code>Access-Control-Allow-Credentials: true</code> , то загрузка изображения будет заблокирована.
<code>href</code>	Основной атрибут тега, в качестве значения выступает путь к файлу со стилями.
<code>hreflang</code>	Определяет язык текста в документе, на который идет ссылка.
<code>media</code>	Определяет тип устройства, к которым должен быть применен ресурс ссылки.
<code>nonce</code>	Генерируемая случайным образом на сервере строковая переменная, которая устанавливает правила использования встроенных стилей с целью защиты контента. Значение атрибута — строка текста.
<code>rel</code>	Атрибут определяет отношения между текущим документом и документом, на который идет ссылка. <code>alternate</code> — ссылка на тот же документ, но в другом формате (например, страницы для печати, перевод, зеркало, лента в формате RSS или Atom). <code><link rel="alternate stylesheet" type="text/css" title="small_font" href="small-font.css"></code> . <code><link rel="alternate" type="application/rss+xml" title="my RSS" href="index.xml"></code> . <code><link rel="alternate" type="application/atom+xml" title="News" href="/atom.xml?type=news"></code> <code>archives</code> — указывает на то, что документ по ссылке представляет исторический интерес. Ссылка может указывать на коллекцию записей, документов и других материалов. <code>author</code> ссылка на страницу об авторе документа или на страницу с контактными данными автора. <code>bookmark</code> ссылка на ближайшего предка статьи, являющегося связующим звеном, или на раздел статьи, наиболее тесно связанных с элементом, если нет предка. <code>external</code> используется для указания того, что страница на которую ведет ссылка не являются частью данного сайта. <code>first</code> указывает ссылку, ведущую на первый документ из последовательности документов. <code>help</code> ссылка на документ со справкой. <code>icon</code> определяет путь к иконке, которая будет использована для текущего документа. <code>last</code> указывает ссылку, ведущую на последний документ в последовательности документов. <code>license</code> ссылка на сведения об авторских правах для документа. <code>next</code> указывает, что этот документ является частью серии, и что ссылка ведет на следующий документ в этой серии. <code><link rel="next" href="/next.html" type="text/html" title="Следующая страница"></code> <code>nofollow</code> указывает на то, что ссылка не одобрена автором страницы или что ссылка носит коммерческий характер. <code>noreferrer</code> указывает на то, что заголовок запроса клиента, содержащий url источника запроса, не должен передаваться при переходе по ссылке. <code>pingback</code> указывает адрес пингбэк-сервера, что дает возможность для блога автоматически оповещать сайты, ссылающиеся на него. <code>prefetch</code> указывает, что следует заранее кэшировать файл, на который ведет ссылка. <code>prev</code> указывает, что этот документ является частью серии, и что ссылка ведет на предыдущий документ в этой серии. <code><link rel="prev" href="/next.html" type="text/html" title="Предыдущая страница"></code> <code>search</code> указывает, что ссылаемый документ содержит интерфейс поиска и связанных с ним ресурсов. <code>sidebar</code> 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. <code>stylesheet</code> ссылка на внешний файл, который будет использоваться в качестве таблицы стилей для данного документа. <code>tag</code> указывает на то, что метка, на которую ведет гиперссылка, относится к данному документу. <code>up</code> указывает, что страница является частью иерархической структуры, и что гиперссылка ведет на более высокий уровень ресурса в структуре.
<code>sizes</code>	Указывает размер иконок для визуального отображения. Атрибут <code>sizes</code> используется только совместно с <code>rel="icon"</code> , и может принимать следующие значения: <code>ширинахвысота</code> — определяет список размеров, разделенных пробелами, каждый размер должен быть в формате — <code>ширинахвысота (размеры иконки задаются в пикселях)</code> , например: <code><link rel="icon" href="favicon.png" sizes="16x16 32x32" type="image/png"></code> ; <code>any</code> — иконка может масштабироваться до любого размера.
<code>title</code>	Определяет заголовок ссылки или название набора альтернативных таблиц стилей. Значение атрибута — текст.
<code>type</code>	Определяет MIME-тип документа, на который идет ссылка. В данном случае он принимает значение <code>"text/css"</code> .

