



[Главная](#) » [HTML и HTML5](#) » [1.11. Семантические элементы HTML5](#) » [1.11.2. Метаданные документа](#)

1.11.2. Метаданные документа

Опубликовано: 20 февраля 2020

Обновлено: 5 июня 2022

🗨 0



К метаданным относится содержимое, которое устанавливает представление или поведение остального содержимого, отношения документа с другими документами, или передает другую «внешнюю» информацию.

HTML-элементы, семантика которых связана с метаданными

СОДЕРЖАНИЕ:

1. Элемент `<head>`
2. Элемент `<title>`
3. Элемент `<base>`
4. Элемент `<link>`
5. Элемент `<meta>`
6. Элемент `<style>`

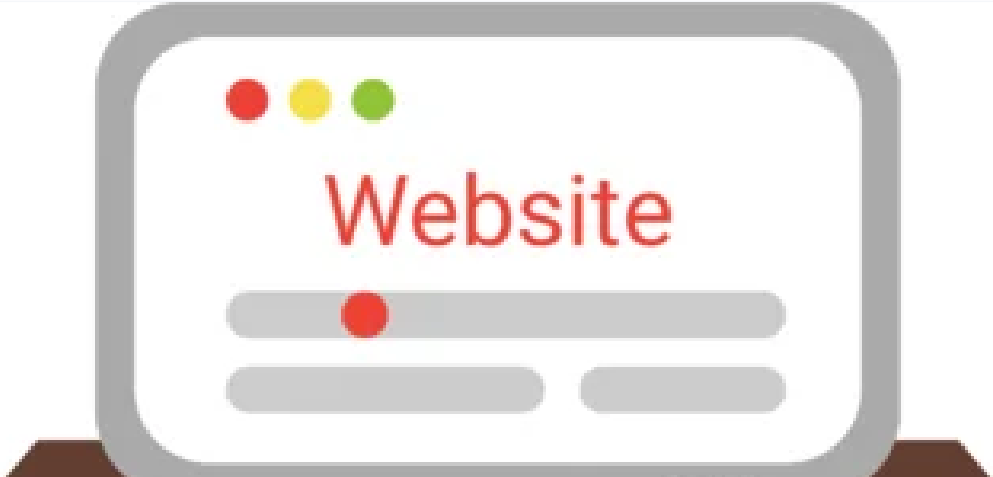


1. Элемент <head>

Категории содержимого: нет.

Контекст, в котором этот элемент может быть использован: как первый элемент в элементе <html> .

РЕКЛАМА



Безлимитный хостинг 2.0

smart

Пропуск тегов: начальный тег <head> может быть пропущен, если элемент <head> пуст, или если сразу после него идет другой HTML-элемент. Закрывающий тег </head> может быть пропущен, если он не следует сразу за пробелом или за комментарием.

HTML

```
<!DOCTYPE html>
<html lang="ru">
  <title>Тест</title>
  <body>
    <h1>Тестовая страница</h1>
  </body>
```

Для элемента доступны [глобальные атрибуты](#).

Раздел <head>...</head> содержит набор технической информации (метаданных) о текущей веб-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.

Набор метаданных может быть как большим, так и маленьким:

HTML

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <title>Документ с небольшим head</title>
  </head>
  <body>
    ...
```

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Документ с большим head</title>
    <link rel="stylesheet" href="default.css">
    <link rel="stylesheet alternate" href="big.css" title="Большой текст">

    <script src="main.js"></script>
  </head>
  <body>
    ...
```

2. Элемент <title>

Категории содержимого: метаданные.

Контекст, в котором этот элемент может быть использован: в элементе <head>, не содержащем других элементов <title>.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны [глобальные атрибуты](#).

Элемент <title> представляет заголовок или название документа (веб-страницы). Авторы должны использовать заголовки, которые дают поисковой системе понять, что содержится на странице, даже если заголовки используются вне контекста, например, в истории, закладках пользователя или в результатах поиска. Заголовок документа может отличаться от заголовка первого уровня, поскольку <h1> не должен стоять отдельно, когда он вырван из контекста.

```
<title>Подробное руководство по Flexbox</title>
...
<h1>Что такое Flexbox</h1>
<p>...</p>
```

Текст внутри <title> отображается браузером в заголовке окна. Также этот текст будет содержать ссылку на ваш сайт на странице результатов поиска. Длина заголовка должна быть не более 60 символов, чтобы поместиться полностью.

В одном документе должно быть не более одного элемента <title>. Элемент <title> является обязательным в большинстве ситуаций, но если протокол более высокого уровня предоставляет информацию о заголовке, например, в строке «Тема» электронного письма, когда HTML используется в качестве формата создания электронного письма, элемент <title> может быть опущен.



3. Элемент <base>

Категории содержимого: метаданные.

Контекст, в котором этот элемент может быть использован: в элементе `<head>`, не содержащем других элементов `<base>`.

Пропуск тегов: отсутствует закрывающий тег.

Для элемента доступны [глобальные атрибуты](#).

Элемент `<base>` с помощью атрибута `href` предоставляет базовый URL документа для парсинга всех относительных URL-адресов на странице, установленных атрибутами `src` и `href`.

Атрибут `target` задает тип окна просмотра по умолчанию при переходе по всем гиперссылкам.

В одном документе может быть только один элемент `<base>` и он должен иметь атрибут `href`, `target` или оба сразу.

Элемент `<base>` должен находиться перед любыми другими элементами в дереве, которые имеют атрибуты, определенные как принимающие URL, кроме элемента `<html>` (его атрибут `manifest` не подвержен влиянию элемента `<base>`).

HTML

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <title>Пример для элемента base</title>
    <base href="https://www.example.com/news/index.html">
  </head>
  <body>
    <p>Посетите страницу <a href="archives.html">архивов</a>.</p>
  </body>
</html>
```

Ссылка в приведенном выше примере будет ссылкой на `https://www.example.com/news/archives.html`.

4. Элемент `<link>`

Категории содержимого: метаданные. Если его использование разрешено в `<body>` — потоковое или текстовое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидаются метаданные. В элементе `<noscript>`, который является дочерним элементом элемента `<head>`. Если элемент разрешен в `<body>` — там, где ожидается текстовое содержимое.

Пропуск тегов: отсутствует закрывающий тег.

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



Атрибут	Описание, принимаемое значение
href	Задаёт адрес гиперссылки.
crossorigin	Описывает, как элемент обрабатывает CORS-запросы, предназначен для использования со ссылками на внешние ресурсы.
rel	Задаёт тип указанной ссылки, может содержать как одно значение, так и набор разделённых пробелами ключевых слов: alternate , dns-prefetch , icon , next , pingback , preconnect , prefetch , preload , prerender , search , serviceworker .
rev	Описывает обратное отношение текущего документа к связываемому документу, как определено атрибутом href .
media	Указывает, к какому типу медиа относится ресурс. Значение должно быть допустимым списком медиазапросов.
nonce	Используется в проверках политики безопасности контента, представляет криптографический одноразовый номер, который может использоваться политикой безопасности содержимого, чтобы определить, будет ли внешний ресурс, указанный в ссылке, загружаться и применяться к документу.
hreflang	Задаёт язык связанного ресурса.
type	Устанавливает подсказку для типа ссылочного ресурса.
referrerpolicy	Указывает URL источника запроса при переходе с одной страницы на другую.
sizes	Задаёт размеры иконок (для rel="icon") для визуальных медиа, значение носит рекомендательный характер.
title	Устанавливает заголовок ссылки, альтернативное имя таблицы стилей.


Атрибут href элемента <link> позволяет связывать HTML-документ с различным видами ресурсов, например, таблицами стилей, скриптами, альтернативными формами документа и ссылками навигации (оглавление, предыдущие и последующие страницы, уведомления об авторских правах и т.п.).

Тип связанного ресурса задаётся значением обязательного атрибута rel .

С помощью элемента <link> можно создать две категории ссылок: ссылки на внешние ресурсы и гиперссылки. Например, следующий элемент ссылки создаёт две гиперссылки (на одну и ту же страницу):

<pre><link rel="author license" href="/about"></pre>	HTML
--	------

Семантика первой состоит в том, что целевая страница содержит информацию об авторе текущей страницы, семантика второй заключается в том, что целевая страница содержит информацию о лицензии, под которой предоставляется текущая страница.

Гиперссылки, созданные с помощью элемента <link> и его атрибута rel , применяются ко  документу. Это отличается от атрибута rel элементов <a> и <area> , который указывает тип

ссылки, контекст которой определяется местоположением ссылки в документе.

Если значения атрибута `rel` содержат только ключевые слова, разрешенные в `<body>`, элемент `<link>` можно использовать там, где ожидается фразовое содержание, то есть внутри `<body>`.

5. Элемент `<meta>`

Категории содержимого: метаданные.

Контекст, в котором этот элемент может быть использован: если для элемента указаны атрибуты `charset` и `http-equiv`, то в элементе `<head>`. Если значением атрибута не является `content-type`, то внутри элемента `<noscript>`, являющимся дочерним элементом `<head>`. Если присутствует атрибут `name` — там, где ожидаются метаданные.

Пропуск тегов: отсутствует закрывающий тег.

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

ТАБЛИЦА 2. АТРИБУТЫ ЭЛЕМЕНТА `<META>`

Атрибут	Описание, принимаемое значение
<code>charset</code>	Определяет кодировку символов, используемую в документе. В документе должен быть один элемент <code><meta></code> с атрибутом <code>charset</code> . Необходимо использовать <code>utf-8</code> или другую кодировку, совместимую с ASCII.
<code>content</code>	Задаёт значение метаданных документа или прагма директив.
<code>http-equiv</code>	Задаёт прагма директиву.
<code>name</code>	Устанавливает название/имя метаданных документа.

Элемент `<meta>` представляет различные виды метаданных, которые не могут быть выражены с использованием элементов `<title>`, `<base>`, `<link>`, `<style>` и `<script>`.

Для элемента обязательно должен быть определен один из атрибутов — `name`, `http-equiv` или `charset`. Если указан атрибут `name` или `http-equiv`, также должен присутствовать атрибут `content` (или пропущен, если нет соответствующих значений).

Стандартные названия метаданных

Для атрибута `name` доступны следующие значения (чувствительны к регистру):

- `application-name` — значение должно быть короткой строкой произвольной формы, содержащей название веб-приложения, которое представляет страница. Если страница не является веб-приложением, `application-name` не должно использоваться. В одном документе должно быть не более одного названия веб-приложения.

Браузеры могут использовать название веб-приложения в пользовательском интерфейсе вместо `<title>`, поскольку `<title>` может содержать сообщения о состоянии и тому подобное, относящиеся к состоянию страницы в определенный момент времени, а не просто как наз приложения. ^

- **author** — значение должно быть строкой произвольной формы с указанием имени одного из авторов страницы.
- **description** — значение должно быть строкой произвольной формы, описывающей страницу. Значение должно быть подходящим для использования в каталоге страниц, например, в поисковой системе.
- **generator** — значение должно быть строкой произвольной формы, которая идентифицирует один из пакетов программного обеспечения, использованных для создания документа. Это значение не должно использоваться на страницах, разметка которых не создается программным обеспечением, например на страницах, разметка которых была написана пользователем в текстовом редакторе.

```
<meta name="generator" content="WordPress 5.2">
```

HTML

- **keywords** — значение должно быть набором разделенных запятыми ключевых слов, относящихся к странице. Многие поисковые системы не рассматривают такие ключевые слова, потому что эта функция исторически использовалась ненадежно и спамила результаты поиска.

```
<meta name="keywords" content="шрифт, шрифты, типографика">
```

HTML

- **referrer** — необязательное поле заголовка HTTP, которое позволяет отслеживать перемещения пользователей между страницами в инструментах аналитики, а также понять происхождение входящего трафика. Реферер передаётся при переходе с http на любой тип сайта, при переходе с https на https, и не передаётся при переходе с https на http.

В наиболее распространенной ситуации это означает, что когда пользователь щелкает гиперссылку в браузере, на сервер отправляется запрос, содержащий целевую веб-страницу. Запрос может содержать поле `referer`, в котором указана последняя страница, на которой был пользователь (то есть та, на которой он щелкнул ссылку).

Значения атрибута `content`:

- `no-referrer` — не передает никакую информацию о реферере.
- `no-referrer-when-downgrade` — передает реферальные данные только сайтам на HTTPS. Поведение браузера по умолчанию, если не указано иное.
- `unsafe-url` — всегда передает полный URL реферера.
- `origin-when-cross-origin` — отправляет полный URL при переходе на страницы в рамках одного сайта, вне зависимости от протокола, а на все остальные — только базовый домен/поддомен.

```
<meta name="referrer" content="origin-when-cross-origin">
```

HTML

- **viewport** — позволяет определять конкретные характеристики области просмотра, например, ширину макета и коэффициент масштабирования веб-страниц. Можно запретить или ограничить пользователям возможность масштабирования, используя такие значения, как `content="user-scalable=no"` или `content="width=device-width, initial-scale=1.0, maximum-scale=1.0"`.

Могут быть конкретные случаи использования, в которых целесообразно предотвращение масштабирования пользователями, например, приложения-карты — когда пользовательские функции масштабирования обрабатываются с помощью сценариев, но в целом такой практики следует избегать.

Распознаваемые свойства атрибута `content` :

- **width** — определяет ширину области просмотра, значением может быть определенное количество пикселей, например, `width=768` или ключевое слово `device-width` (соответствует 100vw).

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

HTML

- **height** — определяет высоту области просмотра, значением может быть определенное количество пикселей, например, `height=480` или ключевое слово `device-height` (соответствует 100vh)
- **initial-scale** — указывает браузеру соотношение пикселей CSS и устройства.
- **minimum-scale** — определяет наименьший допустимый коэффициент масштабирования.
- **maximum-scale** — определяет максимально допустимый коэффициент масштабирования.
- **user-scalable** — указывает, может ли коэффициент масштабирования быть изменен в результате взаимодействия с пользователем или нет.

Прагма директивы

Когда атрибут `http-equiv` указан в элементе `<meta>`, элемент `<meta>` является прагма директивой, которая предоставляет дополнительную информацию о документе:

- **content-type** — является альтернативной формой установки атрибута `charset`.

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

HTML

- **default-style** — задает имя альтернативной таблицы стилей, используемой по умолчанию.

```
<meta http-equiv="default-style" content="default">
```

HTML

- **refresh** — устанавливает таймер для обновления и перенаправления. Например, главная страница новостного сайта может содержать следующую разметку в элементе `<head>`, чтобы

обеспечить автоматическую перезагрузку страницы с сервера каждые пять минут:

```
<meta http-equiv="refresh" content="300">
```

HTML

Последовательность страниц может использоваться в качестве автоматического слайд-шоу, если каждая страница обновляется до следующей страницы в последовательности с использованием следующей разметки:

```
<meta http-equiv="refresh" content="20; url=page4.html">
```

HTML

- **content-security-policy** — позволяет настроить политику защиты содержимого, с помощью которой можно защищаться, например, от межсайтового скриптинга:

```
<meta http-equiv="content-security-policy" content="script-src 'self'">
```

HTML

6. Элемент <style>

Категории содержимого: метаданные.

Контекст, в котором этот элемент может быть использован: где ожидаются метаданные. Внутри элемента `<noscript>` который является дочерним элементом `<head>` . Внутри `<body>` , где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

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

ТАБЛИЦА 3. АТРИБУТЫ ЭЛЕМЕНТА <STYLE>

Атрибут	Описание, принимаемое значение
media	Указывает, к каким медиа применяются стили. Значение должно быть допустимым списком медиазапросов. Браузер должен применять стили, когда значение атрибута <code>media</code> соответствует среде и применяются другие соответствующие условия. Если атрибут <code>media</code> пропущен, по умолчанию он принимает значение <code>all</code> , то есть стили применяются ко всем видам медиа.
nonce	Представляет криптографический одноразовый номер, который может использоваться политикой безопасности содержимого, чтобы определить, будет ли стиль, указанный элементом, применяться к документу.
type	Устанавливает язык таблиц стилей, значение должно быть допустимым MIME-типом. Значением по умолчанию является <code>text/css</code> .
title	Задаёт альтернативное имя таблиц стилей.



Элемент `<style>` позволяет авторам встраивать информацию о стилях в свои документы. Элемент

не представляет какое-либо содержимое для пользователя.

Элемент `<style>` желательно использовать внутри раздела `<head>` .

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Моя любимая книга</title>
    <style>

      em {
        font-style: normal;
        color: red;
      }
    </style>
  </head>
  <body>
    <p>Моя <em>любимая</em> книга - «Приключения Алисы в Стране чудес».</p>
  </body>
</html>
```

По материалам [Document metadata](#)

< Элемент документа

Разделы документа >

Поделиться:

