

*Кратко, исчерпывающе,
незаменимо*

5-е издание

HTML5

Карманный справочник



O'REILLY®

Дженнифер Роббинс

ПЯТОЕ ИЗДАНИЕ

HTML5

Карманный справочник

ПЯТОЕ ИЗДАНИЕ

HTML5

Карманный справочник

Дженнифер Роббинс



Москва · Санкт-Петербург · Киев
2015

Издательский дом “Вильямс”

Главный редактор С.Н. Тригуб

Зав. редакцией В.Р. Гинзбург

Перевод с английского Л.М. Ильинчевой

Под редакцией А.П. Сергеева

По общим вопросам обращайтесь в Издательский дом “Вильямс” по адресу:
info@williamspublishing.com, <http://www.williamspublishing.com>

Роббинс, Дженифер.

P58 HTML5: карманный справочник, 5-е издание. : Пер. с англ. — М. :
ООО “И.Д. Вильямс”, 2015. — 192 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1937-3 (рус.)

ББК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства O'Reilly Media, Inc.

Authorized Russian translation of the English edition of *HTML5 Pocket Reference, Fifth Edition* (ISBN 978-1-449-36335-2), © 2013 Littlechair, Inc.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the Publisher.

Научно-популярное издание

Дженифер Роббинс

HTML5: карманный справочник

5-е издание

Литературный редактор **Л.Н. Красножон**

Верстка **Л.В. Чернокозинская**

Художественный редактор **В.Г. Павлютин**

Корректор **Л.А. Гордиенко**

Подписано в печать 20.08.2015. Формат 84x108/32.

Гарнитура Times.

Усл. печ. л. 10,14. Уч.-изд. л. 6,22.

Тираж 1000 экз. Заказ № 4453.

Отпечатано способом ролевой струйной печати

в АО “Первая Образцовая типография”

Филиал “Чеховский Печатный Двор”

142300, Московская область, г. Чехов, ул. Полиграфистов, д. 1

ООО “И. Д. Вильямс”, 127055, г. Москва, ул. Лесная, д. 43, стр. 1

Содержание

Ждем ваших отзывов!	9
HTML5. Карманный справочник	11
Обзор HTML5	12
W3C и WHATWG	12
Новые семантические элементы в HTML5	14
Библиотеки HTML5 API	15
Структура документа HTML5	16
Поддержка HTML5 в браузерах	17
Проверка достоверности документов HTML5	19
Глобальные атрибуты HTML5	19
Обработчики событий HTML5	23
Алфавитный список элементов	24
Элемент a	25
Элемент abbr	27
Элемент address	28
Элемент area	29
Элемент article	31
Элемент aside	32
Элемент audio	33
Элемент b	36
Элемент base	36
Элемент bdi	37
Элемент bdo	38
Элемент blockquote	39
Элемент body	40
Элемент br	42
Элемент button	42
Элемент canvas	45
Элемент caption	47
Элемент cite	48
Элемент code	48
Элемент col	49
Элемент colgroup	50
Элемент command	52

(Комментарии)	54
Элемент data	54
Элемент datalist	55
Элемент dd	56
Элемент del	57
Элемент details	58
Элемент dfn	60
Элемент dialog	60
Элемент div	61
Элемент dl	62
Элемент dt	64
Элемент em	65
Элемент embed	65
Элемент fieldset	67
Элемент figcaption	69
Элемент figure	70
Элемент footer	71
Элемент form	72
Элементы h1, h2, h3, h4, h5, h6	74
Элемент head	75
Элемент header	76
Элемент hgroup	77
Элемент hr	78
Элемент html	79
Элемент i	80
Элемент iframe	81
Элемент img	83
Элемент input	85
Элемент ins	106
Элемент kbd	107
Элемент keygen	108
Элемент label	109
Элемент legend	111
Элемент li	111
Элемент link	112
Элемент main	114
Элемент map	115
Элемент mark	116

Элемент <code>menu</code>	117
Элемент <code>menuitem</code>	118
Элемент <code>meta</code>	120
Элемент <code>meter</code>	122
Элемент <code>nav</code>	123
Элемент <code>noscript</code>	124
Элемент <code>object</code>	126
Элемент <code>ol</code>	128
Элемент <code>optgroup</code>	129
Элемент <code>option</code>	130
Элемент <code>output</code>	132
Элемент <code>p</code>	133
Элемент <code>param</code>	134
Элемент <code>pre</code>	135
Элемент <code>progress</code>	135
Элемент <code>q</code>	136
Элемент <code>rp</code>	137
Элемент <code>rt</code>	138
Элемент <code>ruby</code>	139
Элемент <code>s</code>	140
Элемент <code>samp</code>	141
Элемент <code>script</code>	141
Элемент <code>section</code>	144
Элемент <code>select</code>	145
Элемент <code>small</code>	147
Элемент <code>source</code>	148
Элемент <code>span</code>	149
Элемент <code>strong</code>	150
Элемент <code>style</code>	150
Элемент <code>sub</code>	152
Элемент <code>summary</code>	153
Элемент <code>sup</code>	154
Элемент <code>table</code>	155
Элемент <code>tbody</code>	157
Элемент <code>td</code>	158
Элемент <code>textarea</code>	159
Элемент <code>tfoot</code>	162
Элемент <code>th</code>	163

Элемент <code>thead</code>	165
Элемент <code>time</code>	166
Элемент <code>title</code>	167
Элемент <code>tr</code>	168
Элемент <code>track</code>	169
Элемент <code>u</code>	170
Элемент <code>ul</code>	171
Элемент <code>var</code>	172
Элемент <code>video</code>	173
Элемент <code>wbr</code>	176
Элементы, упорядоченные по функциям	177
Корневой элемент	177
Элементы метаданных	177
Текст: разделы	177
Текст: заголовки	177
Текст: элементы группировки	177
Текст: списки	177
Элементы уровня текста (встроенные)	178
Таблицы	178
Элементы формы	178
Интерактивный контент	178
Внедренный контент	178
Написание сценариев	178
Аннотация Ruby	178
Приложение А. Символьные сущности	179
Именованные XML-сущности	180
Набор символов Latin-1 (ISO-8859-1)	180
Буквы — модификаторы интервалов	183
Общая пунктуация	184
Приложение Б. Обзор синтаксиса XHTML	187
Предметный указатель	189

Ждем ваших отзывов!

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересны любые ваши замечания в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш сайт и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Отправляя письмо или сообщение, не забудьте указать название книги и ее авторов, а также свой обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию новых книг.

Наши электронные адреса:

E-mail: info@williamspublishing.com

WWW: <http://www.williamspublishing.com>

Наши почтовые адреса:

в России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

в Украине: 03150, Киев, а/я 152

HTML5. Карманный справочник

С помощью *HTML* (HyperText Markup Language — язык гипертекстовой разметки) обычные текстовые документы превращаются в веб-страницы и веб-приложения. Основное назначение *HTML* как языка разметки — поддержка *семантического* описания (значения) контента и создание *структур* документа (иерархии элементов).

В этом карманном справочнике приводится краткий, но в то же время исчерпывающий перечень элементов и атрибутов. Эти атрибуты и элементы описаны в стандартах World Wide Web Consortium (W3C): HTML5 Candidate Recommendation от 17 декабря 2012 года и HTML5.1 Editor's Draft от 15 июня 2013 года. Атрибуты и элементы HTML5 также описаны в “живой” спецификации HTML от 15 июня 2013 года, поддерживаемой Web Hypertext Application Technology Working Group (WHATWG).

В книгу не включены атрибуты и элементы HTML 4.01, которые объявлены устаревшими в HTML5. Атрибуты и элементы, описанные в книге, могут использоваться в документах HTML 4.01, XHTML 1.0 и XHTML 1.1 за исключением тех случаев, когда для них явно указано “Не поддерживается в HTML 4.01”. Если вы проигнорируете это указание, то документ будет некорректным.

Документы HTML5 могут быть переписаны с применением синтаксиса XHTML (формально называется “XML-сериализация HTML5”). В этом случае приведены специальные указания, относящиеся к XHTML. Подробнее требования, выдвигаемые при использовании синтаксиса XHTML, описаны в приложении Б.

Книга имеет следующую структуру.

- Обзор HTML5.
- Глобальные атрибуты HTML5.
- Алфавитный список элементов.
- Список элементов, упорядоченных по назначению.
- Приложение А. Символьные сущности.
- Приложение Б. Обзор синтаксиса XHTML.

Обзор HTML5

В HTML5 появились новые средства (элементы, атрибуты, обработчики событий и API), упрощающие разработку веб-приложений и обеспечивающие дополнительные возможности по обработке данных, вводимых в формы.

Спецификация HTML5 основана на стандарте HTML 4.01 Strict, но, в отличие от предыдущих стандартов HTML Recommendation, HTML5 не предусматривает использование определения типа документа (DTD — Document Type Definition). Вместо этого применяется объектная модель документа (DOM — Document Object Model), представляющая собой “дерево”, формируемое структурой документа. Эта модель используется в качестве базиса, а не набора определенных синтаксических правил. В HTML5 также предусмотрены подробные инструкции по обработке браузерами некорректных и устаревших элементов разметки.

W3C и WHATWG

На момент написания книги существовали две спецификации HTML 5, которые немного отличались одна от другой.

Изначально стандарт HTML5 был разработан рабочей группой Web Hypertext Application Technology Working Group (WHATWG). В 2003 году члены группы по разработке браузеров Apple, Mozilla и Opera сформировали группу WHATWG. Назначение новой группы заключалось в дальнейшем развитии HTML на основе современных принципов веб-дизайна и поведения браузеров. Изначально появились документы Web Applications 1.0 и Web Forms 1.0, на основе которых был создан документ HTML5. Работу по дальнейшему развитию последнего документа в настоящее время курирует редактор Иэн Хиксон, входящий в состав группы WHATWG. Со временем члены группы WHATWG отказались от нумерации версий HTML. В настоящее время на сайте whatwg.org доступна “живая” (ненумерованная) спецификация HTML.

В 2006 году консорциум World Wide Web Consortium (W3C) сформировал собственную рабочую группу (Working Group) по разработке стандарта HTML5 на основе стандарта, созданного

WHATWG. В 2009 году эта рабочая группа прекратила дальнейшее развитие стандарта XHTML 2.0 и переключилась на развитие стандарта HTML5. Группа W3C поддерживает “снимок” (нумерованный) версии HTML5 (www.w3.org/TR/html5/), который в 2014 году получил статус рекомендации. Также разрабатывается версия HTML5.1, которая получит статус рекомендации в 2016 году. Документы HTML5.1 Editor’s Draft доступны на сайте www.w3.org/html/wg/drafts/html/master/.

Версии W3C, HTML5 Candidate Recommendation и WHATWG незначительно различаются между собой. На момент написания книги между спецификациями WHATWG и HTML5.1 имели место следующие отличия.

Только WHATWG

- Атрибут img в элементах a и area.
- Атрибут srcset в элементе img.

Только WHATWG и W3C HTML5.1

- Элемент data.
- Элемент menuitem.
- Элемент dialog.
- Элемент main.
- Глобальный атрибут inert.
- Глобальные атрибуты itemid, itemprop, itemref, itemscope и itemtype.
- Глобальные обработчики событий onclose и onsort.
- Атрибут download в элементах a и area.
- Атрибут sortable в элементе table.
- Атрибут sorted в элементе th.

Только W3C HTML5

- Элемент command (заменяется элементом menuitem).
- Атрибут media в элементе a.
- Атрибут pubdate в элементе time.

Новые семантические элементы в HTML5

В HTML5 появились новые семантические элементы, предназначенные для разметки контента страницы. Дополнительные сведения по каждому из них приведены в разделе “Алфавитный список элементов”.

article	figcaption	output
aside	figure	progress
audio	footer	rp
bdi	header	rt
canvas	group*	ruby
command*	keygen	section
data**	main**	source
datalist	mark	time
details	menuitem**	track
dialog**	meter	video
embed	nav	wbr

* Удалены из HTML5.1.

** Только WHATWG и HTML5.1.

Новые типы контроля за вводом данных

Язык HTML5 предоставляет следующие новые типы контроля за вводом данных (отображаются в виде значений атрибута type для элемента input): color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url и week.

Устаревшие элементы HTML 4.01

Следующие элементы HTML 4.01 были признаны устаревшими в HTML5, поскольку они носят исключительно презентационный характер, вызывают затруднения при использовании либо плохо поддерживаются: acronym, applet, basefont, big, center, dir, frame, frameset, font, isindex, noframes, strike и tt.

Библиотеки HTML5 API

В связи с ростом популярности интерактивного контента, отображаемого на веб-страницах, в HTML5 поддерживается много новых *API* (Application Programming Interface — интерфейс программирования приложения), предназначенных для создания веб-приложений. Благодаря применению этих API стандартизируется выполнение задач, которые обычно реализуются с помощью патентованных подключаемых модулей или пользовательских программ. Одни API включают компоненты разметки, такие как `audio`, `video` и `canvas`, другие используют только JavaScript и/или серверные компоненты.

В спецификации W3C HTML5 описаны следующие API.

- **Media API.** Воспроизведение видео- и аудиофайлов с синхронизацией мультимедиа и отображением субтитров. Этот API применяется вместе с новыми элементами `video` и `audio` (bit.ly/17EC8HT и bit.ly/13xEOBr).
- **TextTrack.** Добавление субтитров и заголовков для элементов `video` и `audio` (bit.ly/16ZQm6S).
- **Session History API.** Отображение журнала (истории) браузера (bit.ly/12iDo0Q).
- **Offline Web Applications API.** Использование веб-ресурсов в автономном режиме (bit.ly/108K1mp).
- **Editing API.** Включение нового глобального атрибута `contenteditable` (bit.ly/11Tw0Tv).
- **Drag and Drop API.** Включение нового атрибута `draggable` (bit.ly/18sdPhs).

Другие API, связанные с HTML5, описаны в отдельных спецификациях и перечислены в следующем списке.

- **Canvas API.** Создание рисунков в двумерной плоскости, используется вместе с новым элементом `canvas` (<http://bit.ly/1Vl7vhx>).
- **Web Storage API.** Хранение данных в кеше браузера, обеспечивающем их доступность для приложения в дальнейшем (bit.ly/168vZjW).

- **Geolocation API.** Обеспечение доступа пользователей к информации о географических координатах (широта и долгота), используемой сценариями в веб-приложениях (bit.ly/168w5rQ).
- **Web Workers API.** Обеспечение выполнения сценариев в фоновом режиме для повышения быстродействия (bit.ly/17ECwWI).
- **Web Sockets API.** Установка открытого подключения между клиентом и сервером, обеспечивающего передачу данных в режиме реального времени. Эта технология применяется в многопользовательских играх, чатах и при передаче потоков “живых” данных (bit.ly/13UfWUK).
- **File API.** Обеспечение доступа к файлам, переданным при вводе данных в формы. При этом можно просматривать загружаемый файл, отображать его и выполнять загрузку путем перетаскивания (bit.ly/11sTkro).

Выше были описаны лишь некоторые наиболее популярные API. В следующий перечень включены ресурсы, содержащие описание всех API и других веб-технологий (в простой для восприятия форме).

- *The Web Platform: Browser Technologies.* Поддерживается Майком Смитом из W3C (platform.html5.org).
- *HTML5 Landscape Overview.* От Эрика Уайлда (dret.typepad.com/dretblog/html5-api-overview.html).
- *Web Platform Docs* (docs.webplatform.org/wiki/apis).

Структура документа HTML5

HTML5 существует в виде единственной версии и не использует определение DTD. Документы HTML5 используют упрощенную декларацию DOCTYPE, применяемую для перехода в стандартный режим формирования изображения в окне браузера. Базовый документ HTML5 имеет следующую минимальную структуру.

```
<!DOCTYPE html>
<html>
  <head>
```

```
<title>Заголовок документа</title>
</head>
<body>
    Контент документа...
</body>
</html>
```

Документы HTML5, созданные на основе синтаксиса XML, не требуют объявления DOCTYPE, но могут включать декларацию XML. Эти документы также должны обрабатываться как тип MIME application/xhtml+xml или application/xml. Ниже приводится пример простого документа HTML5, созданного с применением синтаксиса XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Заголовок документа</title>
    </head>
    <body>
        Контент документа...
    </body>
</html>
```

Поддержка HTML5 в браузерах

Большинство новых семантических элементов и атрибутов HTML5 поддерживается современными браузерами, установленными на компьютерах и мобильных устройствах (Chrome, Safari, Opera, Firefox и Internet Explorer 9+). Для браузеров, не поддерживающих новые элементы и API, обычно применяется *полезаполнение JavaScript* (заплата), имитирующее поддержку HTML5. Полный набор существующих на данный момент полезаполнений можно найти в коллекции Modernizr, доступной по следующему адресу:

[www.github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills](https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills)

В качестве простого примера полезаполнения можно рассмотреть сценарий “HTML5 Shiv”, созданный Реми Шарпом. Чтобы

обеспечить совместимость устаревших браузеров (например, Internet Explorer 8 и более ранние версии) с новыми элементами HTML5, применяется сценарий JavaScript, последовательно создающий каждый элемент в модели DOM. При этом браузер будет воспринимать соответствующие стили и сценарии HTML5.

Например, в результате выполнения следующей команды создается элемент section:

```
document.createElement("section");
```

Сценарий HTML5 Shiv, разработанный Реми Шарпом, одновременно создает все новые элементы. Чтобы выполнить этот сценарий, воспользуйтесь следующим кодом, содержащим ссылку на хост Google.

```
<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/
html5-els.js"></script>
<![endif]-->
```

Используйте следующие ресурсы для отслеживания поддержки и применения HTML5.

“When Can I Use...” (<http://caniuse.com>)

Описание браузеров, поддерживающих HTML5, CSS3 и другие веб-технологии, от Алексиса Деверии.

HTML5 Please (<http://html5please.com>)

Включает рекомендации по использованию средств HTML5 и CSS3.

Wikipedia “Comparison of Layout Engines (HTML5)” ([http://en.wikipedia.org/wiki/Comparison_of_layout_engines_\(HTML_5\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML_5)))

Диаграммы, демонстрирующие поддержку HTML5 основными браузерами.

HTML5 Readiness (<http://html5readiness.com/>)

Полезная видеограмма, демонстрирующая растущую поддержку HTML5 и CSS3 с 2008 года до настоящего времени.

Проверка достоверности документов HTML5

Чтобы проверить достоверность документа HTML5, примените интерактивную программу проверки достоверности, доступную по адресу validator.w3.org (на момент написания книги была доступна поддержка HTML5 на стадии бета-тестирования) или по адресу html5.validator.nu.

Глобальные атрибуты HTML5

Атрибуты, используемые всеми элементами HTML5, называются **глобальными**.

`accesskey = "символ"`

Назначает клавишу доступа (сочетание клавиш, выполняющее команду), которая активизирует или выделяет элемент. В качестве значения используется одиночный символ. Пользователи могут получить доступ к элементу, нажав комбинацию клавиш `<Alt+клавиша>` (ПК) или `Ctrl+<клавиша>` (Mac).

`class = "текстовая строка"`

Присваивание одного или нескольких названий, классифицирующих элемент.

`contenteditable = "true | false"`

Не поддерживается в HTML 4.01. Указывает пользователю на возможность изменения элемента. Этот атрибут поддерживается в современных версиях браузеров.

`contextmenu = "идентификатор элемента меню"`

Не поддерживается в HTML 4.01. Указывает контекстное меню, которое применяется по отношению к элементу. Контекстное меню должно запрашиваться пользователем, например, путем щелчка правой кнопкой мыши.

`dir = "ltr | rtl | auto"`

Определяет направление, выбранное для элемента. Значение `ltr` соответствует выбору направления слева направо, `rtl` — справа налево, а `auto` указывает на то, что направление должно выбираться программой.

`draggable = "true | false"`

Не поддерживается в HTML 4.01. Этот атрибут соответствует перетаскиваемому элементу, т.е. для перемещения элемента выполняется его перетаскивание с помощью мыши.

`dropzone = "copy | move | link | string: текстовая строка | file: тип файла"`

Не поддерживается в HTML 4.01. С помощью этого атрибута определяются действия, которые выполняются после перетаскивания данных на элемент, а также задается допустимый тип данных. Атрибут `copy` приводит к созданию копии перетаскиваемых данных, `move` — к перемещению данных в новое местоположение, `a link` — к созданию ссылки на исходные данные. Если используется атрибут `string: text/plain`, то допускается использование произвольной текстовой строки. С помощью атрибута `file:` задается допустимый тип файла (например, `file: image/png`). Перечисленные действия и допустимые данные могут быть заданы, например, с помощью следующей строки: `dropzone = "copy string:text/plain"`.

`hidden (hidden="hidden" в XHTML)`

Не поддерживается в HTML 4.01. Предотвращение отображения элемента и его потомков в окне пользовательского агента (браузера). В этом случае продолжают выполняться произвольные сценарии или элементы управления формой, находящиеся в скрытых разделах, но пользователь не получает никакой информации.

`id "текстовая строка"`

Присваивание элементу уникального идентификатора.

`inert (inert="inert" в XHTML)`

Только WHATWG и HTML5.1. С помощью этого атрибута указывается, что элемент должен быть инертным. Это означает, что в результате выполнения действий пользователя элемент не может выбираться, быть искомым или целевым.

`itemid = "текст"`

Только WHATWG и HTML5.1. Выступая в качестве части системы микроданных, предназначенной для внедрения данных, считываемых компьютером, атрибут `itemid` отображает глобально распознаваемый идентификатор (такой,

как ISBN для книги). Этот атрибут применяется совместно с атрибутом `itemtype` в элементе, содержащем элемент `itemscope`.

`itemprop = "текст"`

Только WHATWG и HTML5.1. Выступая в качестве части системы микроданных, предназначенной для внедрения данных, считываемых компьютером, атрибут `itemprop` поддерживает имя свойства. В качестве значения этого атрибута используется контент элемента. В качестве значения атрибута может также использоваться URL-ссылка, передаваемая с помощью атрибута `href` (в элементах) или атрибута `src` элемента `img`.

`itemref = "список идентификаторов, разделенный пробелами"`

Только WHATWG и HTML5.1. Выступая в качестве части системы микроданных, предназначенной для внедрения данных, считываемых компьютером, атрибут `itemref` задает список сущностей (по значениям ID), которые будут включены в элемент. Атрибут `itemref` должен использоваться в том же элементе, что и атрибут `itemscope`, с помощью которого создан элемент.

`itemscope`

Только WHATWG и HTML5.1. Выступая в качестве части системы микроданных, предназначенной для внедрения данных, считываемых компьютером, атрибут `itemscope` создает новый элемент — группу свойств (пары имя/значение).

`itemtype = "URL-ссылка или метка обратного DNS"`

Только WHATWG и HTML5.1. Выступая в качестве части системы микроданных, предназначенной для внедрения данных, считываемых компьютером, атрибут типа `itemtype` отображает стандартизованный тип элемента, указанный URL-ссылкой (например, `http://vocab.example.net/book`), или метку обратного DNS (например, `com.example.person`). Атрибут `itemtype` используется в составе элемента, включающего атрибут `itemscope`.

`lang`

Выбор языка для элемента в соответствии с кодом языка.

`xml:lang`

Только XHTML. Выбор языка для элементов в документах XHTML.

`spellcheck = "true | false"`

Не поддерживается в HTML 4.01. Отображает элемент, для которого включена проверка орфографии и грамматики.

`style = "стили CSS"`

Связывает информацию стиля с элементом.

`tabindex = "число"`

Определяет позицию текущего элемента в порядке табуляции, принятом для текущего документа. Значение этого атрибута изменяется от 0 до 32 767. Применяется для задания обхода ссылок при табуляции на странице (или полей в форме).

`title = "текстовая строка"`

Поддержка заголовка или справочной информации об элементе.

`translate = "yes | no"`

Не поддерживается в HTML 4.01. С помощью этого атрибута задается перевод текстового контента элемента и значений атрибутов в локализованном документе. По умолчанию используется значение yes, задающее перевод контента элемента.

Следующие атрибуты не включены в список глобальных, но тем не менее разрешены в документах HTML5.

`aria-* = "текстовая строка или число"`

Не поддерживается в HTML 4.01. Допускает применение к элементу произвольных состояний и свойств WAI-ARIA, способствующих улучшению доступности, например `aria-hidden = "true"`. Значение атрибута изменяется в зависимости от свойства. Дополнительные сведения о поддержке состояний и свойств можно найти на сайте www.w3.org/TR/wai-aria/states_and_properties.

`data-* = "текстовая строка или числовые данные"`

Не поддерживается в HTML 4.01. Предоставляет авторам возможность создавать пользовательские атрибуты, связанные с данными, например `data-length`, `dataduration`,

`data-speed` и др. В результате обеспечивает возможность внедрения невидимых данных, которые могут использоваться клиентскими приложениями или сценариями.

`role = "стандартизованная роль WAI-ARIA"`

Не поддерживается в HTML 4.01. Присваивание элементу одной из стандартизованных ролей WAI-ARIA. В результате упрощается использование этого элемента пользователями с особыми потребностями. Роли (`application`, `banner`, `complementary`, `contentinfo`, `form`, `main`, `navigation` и `search`) весьма важны при навигации на устройствах-помощниках. Полный список допустимых значений ролей можно найти на сайте www.w3.org/TR/wai-aria/roles#role_definitions.

Обработчики событий HTML5

Следующие атрибуты контента обработчиков событий по умолчанию используются в произвольном элементе HTML.

<code>onabort</code>	<code>ondragleave</code>	<code>onload*</code>	<code>onratechange</code>
<code>onblur*</code>	<code>ondragover</code>	<code>onloadeddata</code>	<code>onreset</code>
<code>oncancel</code>	<code>ondragstart</code>	<code>onloadedmetadata</code>	<code>onscroll</code>
<code>oncanplay</code>	<code>ondrop</code>	<code>onseeked</code>	<code>onseeked</code>
<code>oncanplaythrough</code>	<code>ondurationchange</code>	<code>onmousedown</code>	<code>onseeking</code>
<code>onchange</code>	<code>onemptied</code>	<code>onmousemove</code>	<code>onselect</code>
<code>onclick</code>	<code>onended</code>	<code>onmouseout</code>	<code>onshow</code>
<code>onclose**</code>	<code>onerror*</code>	<code>onmouseover</code>	<code>onsort**</code>
<code>oncontextmenu</code>	<code>onfocus*</code>	<code>onmouseup</code>	<code>onstalled</code>
<code>oncuechange</code>	<code>oninput</code>	<code>onmousewheel</code>	<code>onsubmit</code>
<code>ondblclick</code>	<code>oninvalid</code>	<code>onpause</code>	<code>onsuspend</code>
<code>ondrag</code>	<code>onkeydown</code>	<code>onplay</code>	<code>ontimeupdate</code>
<code>ondragend</code>	<code>onkeypress</code>	<code>onplaying</code>	<code>onvolumechange</code>
<code>ondragenter</code>	<code>onkeyup</code>	<code>onprogress</code>	<code>onwaiting</code>

* Обработчик событий для объекта `Window`, используемый вместе с элементом `body`.

** Только WHATWG и HTML5.1. Не применяется в HTML5 Candidate Recommendation.

ПРИМЕЧАНИЕ

Если обработчики событий `onblur`, `onerror`, `onfocus` и `onload` применяются к элементу `body`, то ведут себя немного иначе. Это связано с тем, что элемент `body` совместно использует эти обработчики событий с родительским окном.

Алфавитный список элементов

В этом разделе приведен список всех элементов и атрибутов HTML (в соответствии со стандартами W3C и WHATWG). Обратите внимание на значения следующих меток.

Обязательно

Атрибуты, помеченные этой меткой, должны быть включены в элемент для обеспечения корректности разметки.

Только W3C HTML5

Элементы и атрибуты, помеченные этой меткой, описаны лишь в W3C HTML5 Candidate Recommendation, но не в документах HTML5.1 Editor's Draft и WHATWG.

Только WHATWG

Элементы и атрибуты, к которым относится эта метка, описаны лишь в “живой” спецификации HTML, поддерживаемой WHATWG, и не включены в документ W3C Recommendation.

Не поддерживается в HTML 4.01

Элементы и атрибуты, помеченные этой меткой, появились в HTML5 и не могут использоваться в коде, совместимом с HTML 4.01, XHTML 1.0 и XHTML 1.1 DOCTYPE.

Только XHTML

Атрибуты, помеченные этой меткой, применяются только в документах, написанных с применением синтаксиса XHTML (XML).

Элемент a

<a>...

Определяет якорь (anchor), применяемый в качестве гипертекстовой ссылки или именованного фрагмента документа. Если атрибуту href присвоен корректный идентификатор URI, то в качестве якоря выступает гипертекстовая ссылка на веб-страницу, фрагмент страницы или другой ресурс. Атрибут id используется в качестве метки якоря и служит целевой точкой ссылки. Элемент a может включать как атрибут href, так и атрибут id. Если элемент используется в качестве “ссылки заполнителя”, то атрибут href не указывается.

В HTML5 допускается включение в элемент a потокового контента (блочных элементов). Не допускается включение в состав элемента a интерактивного контента (a, audio с элементами управления, button, details, embed, iframe, img с usemap, input, keygen, label, object с usemap, select, textarea и video с элементами управления).

Применение

Категории

Потоковый контент, фразообразующий контент, интерактивный контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Прозрачный, не включающий другие интерактивные элементы.

Начальные/конечные дескрипторы

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

download = "имя файла"

Только WHATWG и HTML5.1. Указывает на использование ссылки для загружаемого ресурса. В качестве значения

применяется имя файла ресурса, которое должно быть присвоено в локальной файловой системе сразу же после загрузки файла.

`href = "URI"`

Указывает местоположение целевого документа или веб-ресурса (такого, как изображение, звуковой файл, PDF-документ или другой мультимедийный файл).

`hreflang = "код языка"`

Указывает основной язык целевого документа.

`media = "all | aural | braille | handheld | print | projection | screen | tty | tv"`

Не поддерживается в HTML 4.01, только HTML5. Описывает носитель целевого документа. По умолчанию принимает значение `all`. Этот атрибут удален в HTML5.1 и WHATWG.

`ping = "URL-ссылки"`

Только WHATWG. Указывает список URL-ссылок, используемых после перехода по ссылке. Может применяться для отслеживания пользователя.

`rel = "ключевое слово типа ссылки"`

Описывает одну или несколько связей между исходным и связанным документами. Для элемента `a` применяются следующие типы ссылочных связей: `alternate`, `author`, `bookmark`, `help`, `license`, `next`, `nofollow`, `noreferrer`, `prefetch`, `prev`, `search` и `tag`.

`target = "текст"`

Указывает имя окна или области `iframe`, где будет отображаться целевой документ.

`type = "тип MIME"`

Указывает носители или тип контента (тип MIME) для связанного контента, например `text/html`.

Примеры

Ссылка на локальный файл:

```
<a href="имя_файла.html">...</a>
```

Ссылка на внешний файл:

```
<a href="http://сервер/путь/файл.html">...</a>
```

Для отправки сообщения электронной почты (браузер может запускать пользовательскую почтовую программу с открытым окном создания сообщения) используется следующий код:

```
<a href="mailto:имя_пользователя@домен">...</a>
```

Ссылка на телефонный номер (браузеры мобильных телефонов могут отображать окно предупреждения, позволяющее пользователю позвонить по номеру, используемому в качестве ссылки):

```
<a href="tel:+15085551212">...</a>
```

Ссылка на файл, находящийся на сервере FTP:

```
<a href="ftp://сервер/путь/имя файла">...</a>
```

Создание именованного якоря в HTML:

```
<a id="фрагмент">...</a>
```

Ссылка на именованный якорь:

```
<a href="http://сервер/путь/файл.html#фрагмент">...</a>
```

Ссылка на именованный якорь в текущем файле:

```
<a href="#фрагмент">...</a>
```

Элемент **abbr**

```
<abbr>...</abbr>
```

Применяется для обозначения акронима (аббревиатуры). Если помимо акронима требуется полное описание, то используется атрибут `title`.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`title = "текст"`

При использовании совместно с элементом abbr атрибут title обеспечивает полное “развертывание” акронима. Это может быть полезным для невизуальных браузеров, речевых синтезаторов, систем перевода и поисковых систем.

Пример

```
<abbr title="Massachusetts">Mass.</abbr>
<abbr title="World Wide Web">WWW</abbr>
```

Элемент address

`<address>...</address>`

Поддержка контактной информации или статьи в составе документа. Если почтовый адрес не обозначен в качестве контактных данных автором документа, то элемент address в этом случае не используется.

Применение

Категории

Потоковый контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент, не может включать следующие элементы: h1-h6, article, aside, nav, section, header, footer и address.

Начальные/конечные теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

<address>

Contributed by Jennifer Robbins, O'Reilly Media
</address>

Элемент area

<area> (XHTML: <area /> или <area />)

Включается в состав элемента map клиентской карты изображений для определения специфической области щелчков (“горячая область”).

Применение

Категории

Потоковый контент, фразообразующий контент.

Допустимые контексты

В области возможного появления фразообразующего контента должен находиться в составе элемента map.

Разрешенный контент

Пустой.

Начальный/конечный теги

Этот элемент пустой, что означает наличие единственного начального тега и возможное отсутствие контекста.

В HTML завершающий тег не допускается. В XHTML применяется завершающий тег (<area/> или <area />).

Атрибуты

Глобальные атрибуты HTML5

alt = "текст"

Обязательный. Задает краткое описание изображения, которое появляется в том случае, когда файл изображения недоступен.

`coords = "значения"`

Указывает разделенный запятыми список пиксельных координат, которые задают “горячую” область карты изображений.

`download = "имя файла"`

Только WHATWG и HTML5.1. Показывает ссылку, используемую для загрузки ресурса. Значение является именем файла ресурса, которое должно быть дано в локальной файловой системе один раз, когда он загружается.

`href = "URI"`

Указывает местоположение документа или ресурса, который становится доступным после щелчка или касания определенной области.

`hreflang = "код языка"`

Не поддерживается в HTML 4.01. Указывает язык целевого документа.

`media = "all | aural | braille | handheld | print | projection | screen | tty | tv"`

Не поддерживается в HTML 4.01, только HTML5. Описывает носитель, для которого был создан целевой документ. Значение по умолчанию — `all`. Этот атрибут удален в HTML5.1 и WHATWG.

`ping = "URL-ссылки"`

Только WHATWG. Определение разделенного пробелами списка URL-ссылок, используемого после перехода по ссылке. Эти ссылки применяются для отслеживания пользователей.

`rel = "ключевое слово типа ссылки"`

Описывает одну или несколько связей между исходным и связанным документами. Для элемента `area` применяются следующие типы ссылочных связей: `alternate`, `author`, `bookmark`, `help`, `license`, `next`, `nofollow`, `noreferrer`, `prefetch`, `prev`, `search` и `tag`.

`shape = "rect | circle | poly | default"`

Определение формы области для щелчков.

`target = "текст"`

Определение имени окна или фрейма `iframe`, используемого для отображения целевого документа.

`type = "тип MIME"`

Определение типа носителя или контента (тип MIME) и связанного контента (например, `text/html`).

Пример (HTML)

(См. также `map`.)

```
<map name="space">
  <area shape="rect" coords="203,23,285,106"
  href=http://
www.nasa.gov alt="">
  <area shape="rect" coords="203,23,285,106"
  href=http://
www.nasa.gov alt="">
</map>
```

Элемент `article`

```
<article>...</article>
```

Не поддерживается в HTML 4.01. Представление автономной части контента, например журнальной статьи, поста блога, комментария читателя или газетной статьи, который предназначен для независимого распространения, повторного использования или синдицирования. Элементы `article` могут быть вложенными, подобно комментариям, связанным с постом блога.

Применение

Категории

Потоковый контент, секционирующий контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент (без потомков элемента `main`).

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<article>
  <header>
    <h1>Further Research</h1>
    <p><time datetime="2010-01-14T03:13">January 14, 2010
       </time></p>
    <p>An introduction to the topic...</p>
  </header>
  <p>Content of the article starts...</p>
  <p>And another paragraph in the article.</p>
  <footer>Copyright © 2013 Jane Author</footer>
</article>
```

Элемент aside

`<aside>...</aside>`

Не поддерживается в HTML 4.01. Применяется для представления контента, который косвенно связан с окружающим контентом (раздел, статья или другой потоковый контент), такого как цитаты, списки ссылок, рекламный текст или другой контент, обычно представленный как текст во врезках.

Применение

Категории

Потоковый контент, секционирующий контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент (без потомков элемента main).

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<article>
  <h1>Important Experiment Findings</h1>
  <p>First paragraph...</p>
  <p>Second paragraph...</p>
  <aside>
    <h1>For Further Reading</h1>
    <ul>
      <li><a href="">Interesting Article</a></li>
      <li><a href="">Another Interesting Article</a></li>
    </ul>
  </aside>
</article>
```

Элемент audio

```
<audio>...</audio>
```

Не поддерживается в HTML 4.01. С помощью этого элемента осуществляется внедрение звукового файла в веб-страницу (при этом подключаемый модуль не применяется). Контент элемента audio может применяться пользовательскими агентами, которые не поддерживают этот элемент. При передаче аудиофайла может использоваться атрибут src. Поскольку разные браузеры поддерживают различные звуковые форматы, элемент audio включает наборы параметров для форматов файлов, поддерживаемых элементом source. Элемент audio также поддерживает альтернативный контент, отображаемый браузерами, которые не поддерживают данный элемент.

Форматы звуковых файлов, поддерживаемые в элементе audio, находятся в стадии обсуждения. В настоящий момент не существует аудиоформата, поддерживаемого всеми браузерами.

На момент написания книги распространенные браузеры поддерживали следующие форматы звуковых файлов.

- **IE 9+** (версии, предшествующие 9, не поддерживают элемент `audio`): MP3, MP4 и WebM.
- **Chrome 5+:** MP3, MP4, WAV, Ogg Vorbis и WebM.
- **Firefox 3.5+:** WAV, Ogg Vorbis и WebM (4+). Поддержка MP3 и MP4 появляется в Firefox только в том случае, если доступен декодер от независимого поставщика.
- **Safari 4+ и Mobile Safari 3+:** MP3, MP4 и WAV.
- **Android (2.0+):** MP3, WAV, Ogg Vorbis и WebM (2.3.3+).

Применение

Категории

Потоковый контент, фразообразующий контент, внедренный контент, интерактивный контент (при наличии атрибута `controls`), явный контент (при наличии атрибута `controls`).

Допустимые контексты

В области возможного появления внедренного контента.

Разрешенный контент

Прозрачный контент, включающий атрибут `src` либо один или более элементов `source`, за которыми следует потоковый или фразообразующий контент. Применение элементов `video` и `audio` не допускаются.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`autoplay` (или `autoplay="autoplay"` в XHTML)

Автоматическое воспроизведение мультимедийного файла.

`controls` (или `controls="controls"` в XHTML)

Указывает на необходимость отображения пользовательским агентом (браузером) набора элементов управления воспроизведением мультимедийного файла.

`crossorigin = "anonymous | use-credentials"`

Указывает на то, должен ли пользовательский агент проверять полномочия для мультимедийного файла,

загруженного по URL-ссылке, отличающейся от адреса исходного документа. По умолчанию используется значение `anonymous` (полномочия не требуются).

`loop` (или `loop="loop"` в XHTML)

Указывает на то, что мультимедийный файл должен воспроизводиться повторно в автоматическом режиме после завершения предыдущего воспроизведения.

`mediagroup = "текст"`

Связывает несколько мультимедийных элементов путем присваивания им одного и того же значения `mediagroup`.

`muted` (или `muted="muted"` в XHTML)

Отключает (приглушает) звук, даже если при этом переопределются пользовательские предпочтения.

`preload = "none | metadata | auto"`

Подсказывает браузеру, должен ли мультимедийный файл загружаться автоматически на основании прогнозируемого поведения пользователя. Значение `none` способствует предотвращению предварительной загрузки, а `metadata` осуществляет выборку метаданных ресурса, но не загружает мультимедийный файл. Если же указано значение `auto`, то пользовательский агент сам принимает решение о предварительной загрузке мультимедийного файла.

`src = "URL-ссылка"`

Указывает местоположение мультимедийного файла.

Примеры

(См. также `source`.)

```
<audio src="soundtrack.webm" autoplay controls>
```

Ваш браузер не поддерживает встроенный звук.

Прослушайте файл здесь.

```
</audio>
```

```
<audio id="soundtrack" controls preload="auto">
```

```
  <source src="soundtrack.mp3" type="audio/mp3">
```

```
  <source src="soundtrack.ogg" type="audio/ogg">
```

```
  <source src="soundtrack.webm" type="audio/webm">
```

```
</audio>
```

Элемент **b**

`...`

Слова и фразы, которые должны выделяться из окружающего текста без подчеркивания особой важности и без использования акцента, например ключевые слова или названия товаров. В более ранних спецификациях HTML элемент **b** определялся как **bold**.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

`Turn left onto Blackstone Blvd.`

Элемент **base**

`<base>` (XHTML: `<base />` или `<base />`)

Указывает базовое имя пути для всех относительных URL-ссылок в документе. Этот элемент находится в области действия элемента `head` в документе.

Применение

Категории

Контент метаданных.

Допустимые контексты

В области действия элемента head, в которой отсутствуют другие элементы base.

Разрешенный контент

Отсутствует.

Начальный/конечный теги

Поскольку этот элемент пустой (void), это означает, что используется только начальный тег и могут отсутствовать какие-либо контенты. В HTML для таких элементов не допускаются конечные теги. В XHTML применяется закрывающая косая черта (<base /> или <base />).

Атрибуты

Глобальные атрибуты HTML5

`href = "URI"`

Задает абсолютный идентификатор URI, используемый в качестве базового при разрешении относительных идентификаторов URI.

`target = "имя"`

Определяет заданное по умолчанию целевое окно или фрейм iframe для всех ссылок, используемых в документе.

Пример (HTML)

```
<head>
  <title>Образец документа</title>
  <base href="http://www.example.com/stories/">
</head>
```

Элемент bdi

```
<bdi>...</bdi>
```

Название этого элемента представляет собой аббревиатуру от “bidirectional isolation” (дву направленная изоляция). Этот элемент применяется для выделения текста, который может читаться в направлении, противоположном направлению чтения окружающего текста. Может применяться для выделения имен и комментариев,

добавленных пользователями, или текста, внедренного из внешнего источника.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`dir = "ltr | rtl | auto"`

С помощью этого атрибута указывается направление чтения текста: `ltr` (слева направо), `rtl` (справа налево) или `auto` (определяется пользовательским агентом). Если значение атрибута `dir` не указывается, то по умолчанию используется значение `auto` и не осуществляется наследование от элемента-предка.

Пример

```
<ul>
  <li>User <bdi>jcranmer</bdi>: 12 posts.
  <li>User <bdi>hober</bdi>: 5 posts.
  <li>User <bdi>!بان</bdi>: 3 posts.
</ul>
```

(Пример используется с разрешения группы разработчиков спецификации WHATWG HTML.)

Элемент `bdo`

```
<bdo>...</bdo>
```

Название этого элемента представляет собой аббревиатуру от “bidirectional override” (дву направленное переопределение). Этот элемент применяется для выделения текста, который читается в обратном направлении (по сравнению с окружающим текстом). Например, в документе, который читается слева направо, элемент может применяться для выделения текста на иврите, который читается справа налево (*rtl*).

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`dir = "ltr | rtl | auto"`

С помощью этого атрибута указывается направление чтения текста: *ltr* (слева направо), *rtl* (справа налево) или *auto* (определяется пользовательским агентом).

Пример

```
<bdo dir="ltr">English phrase in otherwise Arabic text.  
</bdo>
```

Элемент **blockquote**

```
<blockquote>...</blockquote>
```

Применяется для отображения длинной цитаты, заимствованной из другого источника. В качестве содержимого этого элемента могут использоваться элементы блочного уровня, такие как абзацы.

В HTML5 элемент **blockquote** является также *секционирующим корнем*, указывающим на раздел, который может иметь собственную структуру. Это значит, что уровни заголовка, используемые внутри элементов **blockquote**, не входят в состав общей структуры страницы.

Применение

Категории

Потоковый контент, секционирующий корень, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`cite = "URI"`

Создает ссылку, включающую информацию об источнике, из которого была заимствована цитата.

Пример

```
<blockquote cite="http://www.example.com">
    <h1>Fascinating Evidence</h1>
    <p>This is the beginning of a lengthy quoted
       passage (продолжение текста...) </p>
    <p>And it is still going on and on
       (продолжение текста...)</p>
</blockquote>
```

Элемент **body**

```
<body>...</body>
```

В области тела документа находится его контент. Контент может быть представлен визуально (как в окне графического

браузера) или в виде звука (экранный диктор). Каждому документу соответствует единственный элемент `body`. В документах HTML использовать этот элемент необязательно, в документах XHTML — обязательно.

Применение

Категории

Секционирующий корень.

Допустимые контексты:

В качестве второго элемента в элементе `html`.

Разрешенный контент

Потоковый контент.

Начальный/конечный теги

Начальный тег необязателен, если элемент пуст или если в начале области действия элемента `body` не встречается символ пробела или комментария, за исключением случаев, когда область действия элемента `body` начинается с элемента `script` или `style`. Конечный тег необязателен, если за элементом `body` не следует комментарий. (XHTML: обязателен/обязателен.)

Атрибуты

Глобальные атрибуты HTML5, а также `onafterprint`, `onbeforeprint`, `onbeforeunload`, `onblur`, `onerror`, `onfocus`, `onhashchange`, `onload`, `onmessage`, `onoffline`, `ononline`, `onpagehide`, `onpageshow`, `onpopstate`, `onresize`, `onscroll`, `onstorage` и `onunload`.

Пример

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/HTML4.01/strict.dtd">
<html>
  <head>
    <title>Заголовок документа</title>
  </head>
  <body>
    <p>Контент документа...</p>
  </body>
</html>
```

Элемент `br`

`
` (XHTML: `
` или `
`)

С помощью этого элемента формируется разрыв строки в контенте, например в стихотворении или почтовом адресе.

Применение

Категории

Потоковый контент, фразообразующий контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Пустой.

Начальный/конечный теги

Это пустой (`void`) элемент, который включает только начальный тег. Этот элемент не может включать какой-либо контент. В HTML применение завершающего тега не допускается. В XHTML для закрытия этого элемента применяется завершающая косая черта (`
` или `
`).

Атрибуты

Глобальные атрибуты HTML5

Пример (HTML)

```
<p>O'Reilly Media<br>
1005 Gravenstein Highway North<br>
Sebastopol, CA 95472</p>
```

Элемент `button`

`<button>...</button>`

Используется в форме ввода данных, определяя кнопку, на которой можно выполнить щелчок. В данном случае создается кнопка, подобная кнопкам, созданным с помощью элемента `input`. Кнопки могут содержать такой контент, как текст и изображения (но не карты изображений).

Применение

Категории

Потоковый контент, фразообразующий контент, интерактивный контент, “отображаемый в списке, подписываемый и передающий данные формы элемент”, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент, не может включать интерактивный контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`autofocus` (`autofocus="autofocus"` в XHTML)

Не поддерживается в HTML 4.01. Отображает элемент управления с выделенным фокусом (выделен и подготовлен для пользовательских данных) при загрузке документа.

`disabled` (`disabled="disabled"` в XHTML)

Блокирует экранный элемент управления для пользовательского ввода. Состояние блокировки может быть изменено только в результате выполнения сценария. Браузеры могут отображать блокированные экранные элементы управления по разному (например, выделять серым цветом). С помощью этой техники можно, например, скрывать определенные экранные элементы управления до тех пор, пока не будет передана требуемая информация.

`form = "идентификатор владельца формы"`

Не поддерживается в HTML 4.01. Явно связывает экранный элемент управления вводом с ассоциированной формой (владельцем этой формы). При использовании этого метода экранный элемент управления вводом не должен являться потомком применяемого к нему элемента `form`.

`formaction = "URL-ссылка"`

Не поддерживается в HTML 4.01. Указывает приложение, которое обрабатывает форму. Используется только вместе с кнопкой передачи данных (`input type="submit"`) и выполняет те же функции, что и атрибут `action` для элемента `form`.

`formenctype = "тип контента"`

Не поддерживается в HTML 4.01. Указывает, каким образом значения формы кодируются при использовании метода `post`. Используется только совместно с кнопкой передачи данных (`input type="submit"`) и выполняет ту же функцию, что и атрибут `enctype` для элемента `form`. По умолчанию применяется кодирование Internet Media Type (`application/x-www-form-urlencoded`). Значение `multipart/form-data` должно использоваться в комбинации с типом ввода данных `file`. Значение `text/plain` устанавливает тип MIME `text/plain` для передаваемых данных.

`formmethod = "get | post"`

Не поддерживается в HTML 4.01. Указывает, какой метод HTTP используется для передачи данных формы. Используется только вместе с кнопкой передачи данных (`input type="submit"`) и выполняет те же функции, что и атрибут `method` для элемента `form`.

`formnovalidate = "URL-ссылка"`

Не поддерживается в HTML 4.01. Указывает на то, что достоверность формы не проверяется во время передачи данных. Используется только вместе с кнопкой передачи данных (`input type="submit"`) и выполняет ту же функцию, что и атрибут `novalidate` для элемента `form` (появился в HTML5).

`formtarget = "имя"`

Не поддерживается в HTML 4.01. Указывает целевое окно, в котором отображаются результаты передачи данных формы. Используется только вместе с кнопкой передачи данных (`input type="submit"`) и выполняет ту же функцию, что и атрибут `target` для элемента `form`.

menu = "текст"

Не поддерживается в HTML 4.01. Если атрибуту type кнопки присвоено значение menu, то этот атрибут обязательный для предоставления ID элемента menu на той же странице, на которой атрибуту type присвоено значение button.

name = "текст"

Обязательный. Присваивает элементу название элемента управления.

type = "submit | reset | button | menu"

Идентифицирует поведение кнопки: submit (кнопка передачи данных; тип, выбранный по умолчанию), reset (кнопка сброса), button (клиентская кнопка, управляемая с помощью кода JavaScript) или menu (отображает меню).

value = "текст"

Присваивает значение экранному элементу управления (кнопке). Поведение кнопки определяется атрибутом type.

Пример

```
<button type="reset" name="reset"> Попробуйте снова.</button>
```

Элемент canvas

```
<canvas>...</canvas>
```

Не поддерживается в HTML 4.01. Представляет двумерную область, которая может использоваться для отображения динамической растровой графики, такой как диаграммы, игры, графические редакторы, анимационные экраны и т.п. При выполнении кода, включающего элемент canvas, на странице, используемой в качестве холста, создается прямоугольная область. Изображение на холсте формируется с помощью сценария (обычно JavaScript).

Элемент canvas — одно из лучших средств, предоставляемых HTML5 и имеющих базовую поддержку в Firefox 2.0+, Safari 3.1+, Chrome 1.0+, Internet Explorer 9+ и Opera 9.0. С помощью библиотеки JavaScript FlashCanvas (<http://flashcanvas>.

net) можно добавить в браузер поддержку этого элемента, реализуемую на основе Flash Drawing API.

Применение

Категории

Потоковый контент, фразообразующий контент, внедренный контент, явный контент.

Допустимые контексты

В области возможного появления внедренного контента.

Разрешенный контент

Прозрачный, фразообразующий или потоковый контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`height = "число"`

Высота области холста, выраженная в CSS-пикселях.

`width = "число"`

Ширина области холста, выраженная в CSS-пикселях.

Пример

```
<html>
<head>
    <script type="application/x-javascript">
function draw() {
    var canvas = document.getElementById("box");
    var ctx = canvas.getContext("2d");
    ctx.fillStyle = "rgb(163, 120, 240)";
    ctx.fillRect (55, 50, 75, 100);
}
    </script>
</head>
<body onload="draw()">
    <canvas id="box" width="250" height="250"></canvas>
</body>
</html>
```

Элемент `caption`

`<caption>...</caption>`

Поддержка сводки содержимого таблицы или отображение информации о назначении таблицы. Элемент `caption` должен указываться после начального тега таблицы, предваряя другие элементы таблицы.

Примечание

Если элемент `table` представляет лишь контент элемента `figure`, то вместо элемента `caption` используйте элемент `fig-caption`.

Применение

Категории

Нет.

Допустимые контексты

В качестве первого потомка элемента `table`.

Разрешенный контент

Потоковый контент, не может включать элементы `table`.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<table>
  <caption>Краткое описание данных в таблице.</caption>
  <tr>
    <td>данные</td><td>данные</td>
  </tr>
</table>
```

Элемент `cite`

`<cite>...</cite>`

Помечает заголовок печатной работы (ссылка на другую статью, книги, журналы, статьи, телепередачи и т.п.).

Применение

Категории

Потоковый контент, контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления внедренного контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<p>Recipe from <code>Food & Wine Magazine</code>.</p>
```

Элемент `code`

`<code>...</code>`

С помощью этого элемента выделяется фрагмент компьютерного кода, который отображается как встроенный (фразообразующий) элемент. По умолчанию визуальные браузеры отображают код с помощью моноширинного шрифта.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<p>DOM reference: <code>document.getElementById</code></p>
```

Элемент col

`<col>` (XHTML: `<col/>` или `<col />`)

С помощью этого элемента создается столбец (или столбцы при помощи атрибута `span`) в таблице. При этом свойства атрибута могут применяться ко всем ячейкам в столбце (столбцах). Элемент `col` не выполняет структурное группирование столбцов (т.е. не обрабатывается с помощью элемента `colgroup`), но скорее является пустым элементом, который обеспечивает совместное использование атрибутов. Элемент `col` нужно включать после элемента заголовка и перед произвольными элементами строки (`tr`) или группы строк (`thead`, `tbody`, `tfoot`) вместе с элементом `table`.

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента `colgroup`, для которого не используется атрибут `span`.

Разрешенный контент

Отсутствует.

Начальный/конечный теги

Поскольку этот элемент пустой (`void`), это означает, что используется только начальный тег, а какие-либо контенты могут отсутствовать. В HTML для таких элементов запрещено использовать закрывающие теги. В XHTML применяется закрывающая косая черта (`<col/>` или `<col />`).

Атрибуты

Глобальные атрибуты HTML5

`span = "число"`

Указывает количество столбцов, попадающих в область действия элемента `col`. По умолчанию выбрано значение 1. Все столбцы, на которые распространяется влияние элемента `span`, форматируются в соответствии с выбранными настройками атрибутов элемента `col`.

Пример (HTML)

См. также `colgroup` и `table`.

```
<table>
  <col span="2" width="100" class="name">
  <col span="1" width="50" class="date">
  <thead> ...
  (продолжение разметки)
```

Элемент `colgroup`

`<colgroup>...</colgroup>`

Определение группы столбцов, формирующих структуру таблицы. Элемент `colgroup` должен использоваться после элемента `caption` и перед произвольными элементами строки (`tr`) или группы строк (`thead`, `tbody`, `tfoot`) вместе с элементом `table`.

Таблица может включать несколько групп столбцов. Количество столбцов в группе указывается с помощью значения атрибута `span` либо числа столбцов (`col`) в группе. Группы столбцов могут применяться для ускорения отображения таблицы (например, столбцы могут отображаться последовательно, не ожидая вывода

всего контента). Также с помощью групп столбцов формируется система вывода информации с помощью невизуальных агентов, таких как речевые браузеры и браузеры, использующие шрифт Брайля.

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента `table` после произвольных элементов `caption` и перед любыми элементами `thead`, `tbody`, `tfoot` и `tr`.

Разрешенный контент

Если элемент включает атрибут `span`, то он пуст. Если атрибут `span` отсутствует, то элемент должен содержать один или более элементов `col`.

Начальный/конечный теги

Начальный тег необязателен, если первым потомком является `col` и отсутствует предшествующий начальный тег `colgroup` без конечного тега. Конечный тег необязателен, если за элементом `colgroup` не следует символ пробела или комментарий. (XHTML: обязательен/обязателен.)

Атрибуты

Глобальные атрибуты HTML5

`span = "число"`

Задает количество столбцов, попадающих в область действия элемента `colgroup`. По умолчанию используется значение 1. Все столбцы, “охватываемые” элементом `colgroup`, отформатированы в соответствии со значениями атрибутов элемента `colgroup`.

Пример

(См. также `col`.)

```
<table>
  <colgroup id="employinfo">
```

```
<col span="2" width="100">
<col span="1" width="50" class="date">
</colgroup>
<thead>... (продолжение разметки)
```

Элемент `command`

`<command>` (XHTML: `<command/>` или `<command />`)

Только HTML5. Элемент `command`, используемый вместе с элементом `menu`, является интерактивным. Этот элемент представляет действие, которое может немедленно вызываться пользователем на выполнение (обычно с помощью обработчика прерываний `onclick`).

Примечание

В HTML5.1 и в спецификации WHATWG элемент `command` не рекомендуется к применению, вместо него используется элемент `menuitem`. (См. примеры кода с элементами `menu` и `menuitem`.)

Применение

Категории

Контент метаданных, потоковый контент, фразообразующий контент.

Допустимые контексты

В области возможного появления контента метаданных и фразообразующего контента.

Разрешенный контент

Пустой.

Начальный/конечный теги

Поскольку этот элемент пустой (`void`), это означает, что используется только открывающий тег, а также могут отсутствовать какие-либо контенты. В HTML для таких элементов запрещено использовать закрывающие теги. В XHTML применяется закрывающая косая черта (`<comand/>` или `<comand />`).

Атрибуты

Глобальные атрибуты HTML5

`checked (checked="checked" в XHTML)`

Этот атрибут указывает на то, что команда выбрана.

`command = "идентификатор мастер-элемента"`

Если элемент `command` использует атрибут `command`, то его состояние устанавливается на основании мастер-команды.

`disabled(disabled="disabled" в XHTML)`

Этот атрибут указывает на то, что элемент `command` недоступен в данный момент времени.

`icon = "URL-ссылка"`

Указывает местонахождение изображения, используемого в качестве кнопки для команды.

`label = "текст"`

Поддерживает имя команды, отображаемой для пользователя.

`radiogroup = "текст"`

Определяет название группы команд, если выбран тип команды `radio`.

`title = "текст"`

Создает подсказку команды, отображаемую для пользователя.

`type = "command | checkbox | radio"`

Определяет тип команды. Ключевое слово `command` соответствует обычной команде вместе со связанным с ней действием. Ключевое слово `checkbox` показывает состояние команды, которая может быть “включена” или “выключена”. Ключевое слово `radio` соответствует команде, которая представляет результат выбора одного элемента из списка.

Пример

Примечание: вместо элемента `command` в HTML5.1 и в “живой” спецификации WHATWG используется элемент `menuitem`.

```
<menu>
<command onclick="cut()" label="Cut">
<command onclick="copy()" label="Copy">
```

```
<command onclick="paste()" label="Paste">
<command onclick="delete()" label="Clear">
</menu>
```

(Комментарии)

```
<!-- ... -->
```

Вставляет в документ примечания или сценарии, которые не отображаются браузером. Комментарии могут быть любой длины и не ограничиваются одной строкой.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Не применимы.

Пример

```
<!-- Начало нового раздела -->
<ul> ... (продолжение разметки)
```

Элемент data

```
<data>...</data>
```

Только WHATWG и HTML5.1. Предоставляет распознаваемый компьютером эквивалент для контента элемента (с помощью атрибута value), используемый при обработке данных. Может применяться для всех видов данных, включая дату, время, измерения, веса, микроформаты и т.п.

Применение

Категории

Потоковый контент, контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`value = "значение, распознаваемое компьютером"`

Поддерживает распознаваемый компьютером эквивалент для контента элемента.

Пример

```
<data value="12">Twelve</data>
<data value="2011-11-12">Last Saturday</data>
```

Элемент **datalist**

```
<datalist>...</datalist>
```

Не поддерживается в HTML 4.01. Используемый вместе с элементом управления `input` для создания нового типа элемента управления `list`, элемент `datalist` создает раскрывающееся меню, включающее набор предварительно заданных вариантов (с помощью элемента `option`). При этом поддерживается функция “автозаполнения”, включающаяся, когда пользователь вводит данные в поле (*поле со списком*). Разница между элементом `datalist` и `select` заключается в том, что в первом случае пользователю не нужно выбирать один из заранее заданных вариантов, а можно вводить в поле все что угодно.

Применение

Категории

Потоковый контент, фразообразующий контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент либо несколько элементов `option`.

Начальный/конечный теги
Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<input type="text" list="flavors">
  <datalist id="flavors">
    <option value="Vanilla">
    <option value="Chocolate">
    <option value="Mango">
  </datalist>
```

Элемент dd

<dd>...</dd>

Отмечает описание, определение или значение в паре “термин/описание” в списке описаний (dl). Элемент dd должен использоваться внутри элемента dl и предваряться элементом dt либо элементом dd. Элемент dd может включать другие элементы блочного уровня.

Применение

Категории

Нет.

Допустимые контексты

После элементов dt или dd внутри элементов dl.

Разрешенный контент

Потоковый контент.

Начальный/конечный теги

Требуемый/необязательный (XHTML: требуемый/обязательный).

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<dl>
  <dt><code>em</code></dt>
  <dd>Текст выделен курсивом.</dd>

  <dt><code>strong</code></dt>
  <dd>Текст выделен полужирным.</dd>
</dl>
```

Элемент del

```
<del>...</del>
```

С помощью этого элемента отмечается удаленный в документе текст. Может применяться для выделения правок в юридических документах, а также при редактировании текстов (для отслеживания изменений). В качестве дополнения к этому элементу выступает элемент выделения вставленного текста (*ins*). Элемент *del* используется на уровне отдельных строк и блоков.

Применение

Категории

Потоковый контент, фразообразующий контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Прозрачный (производный от модели контента предка).

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`cite = "URL-ссылка"`

Может применяться для ссылки на исходный документ, в котором указана причина изменения.

`datetime = "ГГГГ-ММ-ДДТЧЧ:ММ:ССTZD"`

С помощью этого атрибута указываются дата и время выполненного изменения. Дата и время записываются в соответствии с показанным здесь форматом, где *ДДДД* обозначает год, *ММ* — месяц, *ДД* — день, *чч* — час (от 00 до 23), *мм* — минуты (от 00 до 59), а *сс* — секунды (от 00 до 59). Аббревиатура “TZD” расшифровывается как “Time Zone Designator” (Обозначение часового пояса). В качестве значения этого параметра может использоваться *Z*, соответствующее часовому поясу UTC (Coordinated Universal Time — всемирное координированное время) либо часовому поясу, опережающему или отстающему от UTC (например, +03:00 или -02:20).

Это стандартный формат для значений даты и времени в HTML. Чтобы получить дополнительные сведения по этой теме, обратитесь по адресу www.w3.org/TR/1998/NOTE-datetime-19980827.

Пример

```
Chief Executive Officer: <del title="retired" datetime="2013-05-01T14:09:00EDT">Peter Pan</del> <ins>Pippi Longstocking</ins>
```

Элемент details

```
<details>...</details>
```

Не поддерживается в HTML 4.01. Представляет виджет замыкания (область, для которой выбирается открытое и замкнутое состояние). В открытом состоянии отображаются дополнительные сведения или элементы управления. Элемент `details` может включать элемент `summary`, с помощью которого формируется резюме или заголовок для деталей, за которым следует дополнительный потоковый контент. Элемент `details` не должен использоваться для подстрочных примечаний.

Примечания

На момент написания книги элементы `details` и `summary` поддерживались лишь браузерами Chrome 12+, Safari 6+, iOS Safari 6+, Android 4.0+ и Blackberry 10+.

Применение

Категории

Потоковый контент, секционирующий корень, интерактивный контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Единственный элемент `summary`, за которым следует потоковый контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`open` (`open="open"` в XHTML)

С помощью этого атрибута указывается на то, что область, описываемая элементом `details`, должна быть в открытом или явном состоянии, когда загружается документ.

Пример

(См. также `summary`.)

`<details>`

```
<summary>Дополнительная информация</summary>
```

```
<ul>
```

```
    <li>Информация об авторе, которая может
```

```
заинтересовать пользователя.</li>
```

```
    <li>Уточняющие сведения.</li>
```

```
</ul>
```

```
</details>
```

Элемент `dfn`

`<dfn>...</dfn>`

Указывает на определяющий экземпляр выделенного термина. Может применяться для привлечения внимания к специальным терминам и фразам либо к ссылкам на них в тексте документа.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент, но не может включать потомки элемента `dfn`.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

`<dfn>Truecolor</dfn>` uses 24 bits per pixel.

Элемент `dialog`

`<dialog>...</dialog>`

Только WHATWG и HTML5.1. Представляет “плавающее” диалоговое окно, такое как окно регистрации, подсказки интерфейса пользователя, лайтбоксы и другой всплывающий или модальный контент, который требует взаимодействия с пользователем. Если элемент `dialog` открыт, то остальная часть страницы неактивна (инертна).

Примечание

На момент написания книги элемент dialog не поддерживался ни одним браузером.

Применение

Категории

Потоковый контент, секционирующий корень.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

open (open="open" в XHTML)

С помощью этого атрибута указывается на то, что элемент dialog активен и пользователь может с ним взаимодействовать.

Пример

```
<dialog id="login" draggable="true">
  <p>You need to be logged in to post a comment</p>
  <button>Log In</button>
  <button>No thanks</button>
<dialog>
```

Элемент div

```
<div>...</div>
```

С помощью этого элемента определяется общая “секция” или контейнер для контента внутри потока документа. Элементы, содержащиеся в области действия элемента div, трактуются как одна семантическая группа. Элементу div обычно придается значение с помощью атрибутов class, id, title и lang.

В результате обеспечивается доступ к этому элементу в сценариях, а также возможность выбора в таблицах стилей.

Применение

Категории

Потоковый контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<div id="summary">
  <h1>In Closing</h1>
  <p>We can summarize as follows...</p>
</div>
```

Элемент dl

```
<dl>...</dl>
```

Этот элемент демонстрирует список описаний (также именуемый списком связей). Каждый элемент в списке является парой “имя–значение”, состоящей из имени (`dt`) и его значения (`dd`). Списки описаний могут применяться для терминов и определений, вопросов и ответов или любых других групп данных “имя–значение”. Могут использоваться несколько элементов `dt` для `dd` или наоборот (см. раздел “Примеры”).

Примечание

Элемент `dl` определен как “список определений” в HTML 4.01.

Применение

Категории

Потоковый контент, явный контент (если потомок элемента включает по крайней мере одну пару “имя–значение”).

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Несколько групп, каждая из которых состоит из одного или нескольких элементов `dt`, за которыми следует один или более элементов `dd`. Также контент не может включать группы.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Примеры

```
<dl>
  <dt>Serif</dt>
  <dd>Times, Times New Roman, Georgia</dd>

  <dt>Sans-serif</dt>
  <dd>Arial, Verdana, Helvetica</dd>

  <dt>Monospace</dt>
  <dd>Courier, Andale Mono, Lucida Console</dd>
</dl>

<dl>
  <dt>Authors</dt>
  <dd>John</dd>
  <dd>Jane</dd>
  ...
</dl>

<dl>
  <dt lang="en-US"><dfn>color</dfn></dt>
```

```
<dt lang="en-GB"><dfn>colour</dfn></dt>
<dd> A sensation which (in humans) derives from the
the fine structure of the eye to distinguish
three differently filtered analyses of a view. </dd>
</dl>
```

Элемент **dt**

<dt>...</dt>

Выделяет имя элемента внутри списка описаний. Элемент **dt** может включать только встроенный (фразообразующий) контент.

Применение

Категории

Нет.

Допустимые контексты

Перед элементами **dd** или **dt** в составе элементов **dl**.

Разрешенный контент

Потоковый контент, но не может включать элементы **header**, **footer**, **article**, **aside**, **nav**, **section** и **h1-h6**.

Начальный/конечный теги

Обязательный/необязательный (XHTML: обязательный/обязательный).

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<dl>
  <dt>Serif</dt>
  <dd>Times, Times New Roman, Georgia</dd>

  <dt>Sans-serif</dt>
  <dd>Arial, Verdana, Helvetica</dd>

  <dt>Monospace</dt>
  <dd>Courier, Andale Mono, Lucida Console</dd>
</dl>
```

Элемент `em`

`...`

С помощью этого элемента выделяется текст, который должен быть подчеркнут или выделен (обычно стилистически выделяется курсивом).

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

В области возможного появления фразообразующего контента.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

`<p>Simon is an excellent craftsman.</p>`

`<p>Simon is an excellent craftsman.</p>`

Элемент `embed`

`<embed>` (XHTML: `<embed/>` или `<embed />`)

Не поддерживается в HTML 4.01. С помощью этого элемента осуществляется внедрение объекта в веб-страницу. В качестве внедряемых объектов чаще всего выступают мультимедийные файлы, для воспроизведения которых используется технология подключаемых модулей (например, Flash-фильмы, QuickTime-фильмы и т.п.). В дополнение к перечисленным ниже атрибутам

некоторые типы мультимедийных файлов и соответствующие им подключаемые модули могут иметь проприетарные атрибуты, используемые для управления воспроизведением файла.

Примечания

Несмотря на то что элемент `embed` обычно поддерживается браузерами, он не был включен в HTML 4.01 и более ранние спецификации (в отличие от элемента `object`, используемого для внедряемых мультимедийных файлов). Документированная поддержка элемента `embed` появилась в HTML5. Поскольку этот элемент ранее поддерживался, но не документировался, он включает множество специфических для браузеров и мультимедийных элементов атрибутов, а его синтаксис непрозрачен. Многие разработчики используют элементы `object` и `embed` для единственного мультимедийного объекта (в целях обратной совместимости), даже если это не соответствует стандарту.

Применение

Категории

Потоковый контент, фразообразующий контент, внедренный контент, интерактивный контент, явный контент.

Допустимые контексты

В области возможного появления внедренного контента.

Разрешенный контент

Пустой.

Начальный/конечный теги

В HTML5 `embed` определяется в виде пустого элемента, без закрывающего тега. Современные браузеры обычно поддерживают вариант элемента `embed` с закрывающим тегом.

Атрибуты HTML5

Глобальные атрибуты HTML5

Примечание. Некоторые подключаемые модули требуют использования собственных проприетарных атрибутов для элемента `embed`. Прежде чем приступить к внедрению выбранного типа мультимедийного файла, прочтите соответствующую документацию.

`height = "число"`

Задает высоту объекта, выраженную в пикселях. Этот атрибут обязателен при внедрении некоторых типов мультимедийных файлов.

`src = "URL-ссылка"`

Задает местонахождение ресурса, помещаемого на странице.

`type = "тип MIME мультимедийного объекта"`

Задает тип MIME мультимедийного объекта, требуемый при загрузке соответствующего подключаемого модуля. Суффикс имени файла, предоставленный в качестве источника, также может использоваться для определения типа подключаемого модуля.

`width = "число"`

Указывает ширину объекта, выраженную в пикселях. Этот атрибут требуется для некоторых типов мультимедийных объектов.

Пример (HTML5)

```
<embed src="movies/vacation.mov" width="240" height="196"  
type="video/quicktime">
```

Нестандартный пример с закрывающим тегом

```
<embed src="movies/vacation.mov" width="240" height="196"  
pluginspage="http://www.apple.com/quicktime/download/">  
<noembed> У вас не установлен  
подключаемый модуль.</noembed>  
</embed>
```

Элемент `fieldset`

```
<fieldset>...</fieldset>
```

С помощью этого элемента формируется группа связанных элементов управления формой и меток. Элементы `fieldset` включаются в состав элемента `form`. Это было сделано для улучшения доступности форм для тех пользователей, которые используют альтернативные браузеры. Название элемента `fieldset` поддерживается первым потомком элемента `legend`. Также элементы `fieldset` могут бытьложенными.

Применение

Категории

Потоковый контент, секционирующий корень, “элемент списка, связанный с формой”, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Необязательный элемент `legend`, за которым следует потоковый контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`disabled` (`disabled="disabled"` в XHTML)

Не поддерживается в HTML 4.01. Отключает элементы управления, находящиеся в области действия элемента `fieldset`. Значение этого атрибута может быть изменено только в результате выполнения сценария. Браузеры могут по-разному отображать отключенные элементы управления (например, выделять серым цветом). Это может быть полезным в том случае, если нужно затемнить определенные элементы управления до тех пор, пока не будет получена требуемая информация.

`form = "идентификатор владельца формы"`

Не поддерживается в HTML 4.01. Явно связывает набор полей с соответствующей формой (владелец формы). При использовании этого метода элемент `fieldset` не должен быть потомком элемента `form`, который применяется к `fieldset`.

`name = "текст"`

Не поддерживается в HTML 4.01. Присваивает имя объекту `fieldset`.

Пример

```
<form>
<fieldset id="customer">
  <legend>Customer contact information</legend>
  <label>Full name <input type="text" name="name">
  </label>
  <label>Email Address <input type="text" name="email">
  </label>
  <label>State <input type="text" name="state"></label>
</fieldset>
</form>
```

Элемент **figcaption**

```
<figcaption>...</figcaption>
```

Не поддерживается в HTML 4.01. Представляет заголовок или легенду, относящуюся к его предку, элементу `figure`.

Применение

Категории

Нет.

Допустимые контексты

В качестве первого или последнего потомка элемента `figure`.

Разрешенный контент

Потоковый контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<figure>
  
  <figcaption>Sales skyrocket in 2014</figcaption>
</figure>
```

Элемент **figure**

<figure>...</figure>

Не поддерживается в HTML 4.01. Обозначает некоторое отдельное подмножество контента, на которое установлены ссылки из основного контента, например иллюстрации, примеры кода, диаграммы или стихотворения. Для контента, выделяемого с помощью элемента **figure**, может назначаться заголовок. При этом используется необязательный элемент **figcaption**, который может находиться в начале или конце разметки, создаваемой с помощью элемента **figure**.

Применение

Категории

Потоковый контент, секционирующий корень, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент; в начале или в конце находится необязательный элемент **figcaption**.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Примеры

```
<figure>
  
  <figcaption>Sales skyrocket in 2014</figcaption>
</figure>
```

```
<figure>
<figcaption>Sample CSS Rule</figcaption>
<pre><code>
body {
  background-color: #369;
```

```
        color: white;  
    }  
    </code></pre>  
</figure>
```

Элемент footer

```
<footer>...</footer>
```

Не поддерживается в HTML 4.01. Представляет информацию, связанную с документом. Эта информация обычно отображается в нижней части документа, статьи или отдельного раздела и представляет собой уведомление о защите авторского права, дату публикации, информацию об авторе или список связанных ссылок. Вовсе не обязательно, чтобы нижний колонтитул находился в конце раздела или документа. Колонтитул может отображаться в начале документа, если это имеет семантический смысл.

Применение

Категории

Потоковый контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент; но не может включать элементы header, footer и main.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<h1>How to Etch Glass</h1>  
<p>Start with clean glass...</p>  
... продолжение разметки...  
<footer>Copyright © 2013 Jennifer Robbins  
</footer>  
</article>
```

Элемент `form`

`<form>...</form>`

Применяется для отображения интерактивной формы, включающей элементы управления, которые предназначены для сбора вводимых пользователем данных, а также другого контента страницы. В документе может находиться несколько форм, причем формы не могут вкладываться одна в другую и не могут перекрываться.

Применение

Категории

Потоковый контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент; не может включать другие элементы `form`.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`accept-charset = "список кодировок символов"`

Задает список кодировок символов входящих данных, которые должны приниматься сервером для обработки текущей формы. В качестве значения используется список названий наборов символов ISO, разделенных пробелами и/или запятыми. В качестве значения по умолчанию используется `unknown`. Этот атрибут широко не поддерживается.

`action = "URL-ссылка"`

Обязательный. Указывает URL-ссылку приложения, которое обрабатывает форму. По умолчанию используется текущая URL-ссылка.

`autocomplete = "on|off"`

Не поддерживается в HTML 4.01. Позволяет пользовательскому агенту (браузеру) заполнять поле автоматически

(on) или требует, чтобы пользователь каждый раз сам вводил информацию (off).

enctype = "тип контента"

Указывает, как кодируются значения элементов управления формой, передаваемые на сервер при выборе метода post. По умолчанию используется кодирование Internet Media Type (application/x-www-form-urlencoded). Значение multipart/form-data должно применяться в комбинации с элементом ввода данных file. Новое значение text/plain используется для выбора текста/простого текста в качестве типа MIME.

method = "GET | POST"

Указывает метод HTTP, применяемый для передачи данных формы. В случае выбора метода get (значение, заданное по умолчанию) информация добавляется и отправляется вместе с URL-ссылкой.

name = "текст"

Присваивает имя форме.

novalidate (novalidate="novalidate" в XHTML)

Не поддерживается в HTML 4.01. Указывает на то, что форма не была верифицирована при передаче данных.

target = "имя"

Указывает цель для результатов передачи данных формы, которые должны быть загружены. Это приведет к тому, что эти результаты могут отображаться в другом окне или в фрейме iframe. Могут быть использованы специальные целевые значения: _bottom, _top, _parent и _self.

Пример

```
<form action="/mailinglist.php" method="POST">
  <fieldset>
    <legend>Join Our Mailing List</legend>
    <ol>
      <li><label>Name: <input type="text" name="username">
    </label></li>
      <li><label>Password: <input type="text" name="pswd">
    </label></li>
```

```
</ol>
</fieldset>
</form>
```

Элементы h1, h2, h3, h4, h5, h6

```
<hn>...</hn>
```

Определяет заголовок, который кратко описывает соответствующий раздел. Используются шесть уровней заголовков, от h1 (наиболее важный) до h6 (наименее важный). В соответствии с синтаксисом HTML заголовки должны следовать по порядку (например, уровень заголовка h2 не должен превышать уровень заголовка h1), чтобы сохранить корректную структуру документа. Это не только способствует улучшению доступности документа, но и способствует оптимизации работы поисковых механизмов (данные, относящиеся к заголовкам более высоких уровней, должны обладать большим весом, чем информация, соответствующая заголовкам более низких уровней).

Примечание

В HTML5 порядок следования заголовков может повторяться внутри разделов одного и того же документа, обеспечивая большую степень гибкости при использовании уровней заголовков. Это облегчает создание структуры документа и позволяет разделам документа встраиваться корректно в его структуру независимо от документов, в которых находится тот или иной раздел. На момент написания книги в HTML5 отсутствовала хорошая поддержка алгоритма создания структуры.

Применение

Категории

Потоковый контент, контент заголовков, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<h1>Story Title</h1>
<p>In the beginning ... </p>
<h2>Subsection Title</h2>
<p>And so on ... </p>
<p>And so on ... </p>
```

Элемент head

`<head>...</head>`

Содержит информацию о документе, организованную в виде коллекции метаданных. Каждый элемент `head` должен содержать элемент `title`, включающий описание документа. Элемент `head` также может включать любой из следующих элементов, находящихся в произвольном порядке: `base`, `link`, `meta`, `noscript`, `script` и `style`. В данном случае `head` выступает в качестве контейнера этих элементов и не включает собственный контент.

Применение

Категории

Нет.

Допустимые контексты

В качестве первого элемента в элементе `html`.

Разрешенный контент

Элементы контента метаданных (`base`, `command`, `link`, `meta`, `noscript`, `script`, `style`, `title`), используемые вместе с единственным элементом `title` (элементы контента метаданных также могут отсутствовать). Если документ задается с помощью атрибута `srcdoc` элемента `iframe`, то элемент `title` не требуется.

Начальный/конечный теги

Необязательный/необязательный (XHTML: обязательный/обязательный).

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<!DOCTYPE html>
<html>
  <head>
    <title>Название документа</title>
    <style type="text/css">h1 {color: #333; }</style>
  </head>
  <body>
    <p>Контент документа...</p>
  </body>
</html>
```

Элемент **header**

`<header>...</header>`

Не поддерживается в HTML 4.01. Представляет информацию, которая отображается в начале раздела (чаще всего это заголовок). Также элемент `header` может включать навигационные ссылки, рекламу, вводные положения и т.п. Этот элемент может содержать произвольный текущий контент за исключением элементов `header` и `footer`.

Применение

Категории

Потоковый контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент, но не может включать элементы `header` и `footer`.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Примеры

```
<body>
<header>
  <nav><ul><li>About</li><li>Home</li></ul></nav>
  <h1>White Rabbits</h1>
  <p>Welcome to the White Rabbits fan site.</p>
</header>
  <h2>Rabbit Sightings</h2>... продолжение разметки...
</body>
```

Элемент hgroup

<hgroup>...</hgroup>

Не поддерживается в HTML 4.01 либо в HTML5.1. Используется для группирования набора заголовков h1–h6 таким образом, что последующие уровни заголовков обрабатываются как подзаголовки или строки тегов и не участвуют в формировании структуры страницы.

Примечание

Элемент hgroup был помечен как подлежащий удалению из HTML5. Этот элемент остался лишь для дублирования состояния спецификации HTML5 (на момент написания книги), но он трактуется как устаревший и по причине недостаточной реализации не рекомендуется к применению.

Применение

Категории

Потоковый контент, контент заголовка, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Один или несколько элементов h1, h2, h3, h4, h5 и /или h6.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<hgroup>
  <h1>Web Design in a Nutshell</h1>
  <h2>A Desktop Quick Reference</h2>
</hgroup>
```

Элемент hr

`<hr>` (XHTML: `<hr/>` либо `<hr />`)

Применяется для отображения тематического прерывания на уровне абзаца, т.е. завершение изложения одной мысли и начало рассмотрения другой мысли.

Применение

Категории

Потоковый контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Пустой.

Начальный/конечный теги

Пустой (`void`) элемент; это означает, что элементу соответствует только начальный тег, а контент может отсутствовать. В HTML закрывающий тег запрещен. В XHTML элемент завершается обратной косой чертой (`<hr/>` или `<hr />`).

Атрибуты

Глобальные атрибуты HTML5

Пример (HTML)

```
<p>These are notes from Thursday.</p>
<hr>
<p>These are notes from Friday.</p>
```

Элемент `html`

`<html>...</html>`

Это корневой элемент документов HTML (и XHTML), т.е. этот элемент включает все остальные элементы. Элемент `html` не имеет предков. Открывающий тег `<html>` помещается в начале документа, сразу после описания типа документа. Закрывающий тег находится в конце документа. Если теги не указываются, то все равно подразумевается использование элемента `html` в качестве корневого элемента.

Примечание

Рекомендуется с помощью атрибута `lang` выбрать язык документа.

Применение

Категории

Нет.

Допустимые контексты

В качестве корня документа.

Разрешенный контент

Элемент `head`, за которым следует элемент `body`.

Начальный/конечный теги

Необязательный/необязательный (XHTML: обязательный/обязательный).

Атрибуты

Глобальные атрибуты HTML5

`manifest = "URL-ссылка"`

Не поддерживается в HTML 4.01. Указывает на кеш-память, используемую API веб-приложения в автономном режиме.

`xmlns = "http://www.w3.org/1999/xhtml"`

Обязателен только в XHTML. В документе XHTML объявляет пространство имен XML для документа.

Пример (HTML5)

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Название документа</title>
    </head>
    <body>
        <p>Контент документа...</p>
    </body>
</html>
```

Пример (XHTML)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
    <head>
        <title>Название документа</title>
    </head>
    <body>
        <p>Контент документа...</p>
    </body>
</html>
```

Элемент i

```
<i>...</i>
```

При чтении абзаца, выделенного с помощью этого элемента, используется другая интонация или выражается другое настроение (по сравнению с окружающим текстом). Обычно с помощью этого элемента выделяется фраза, написанная на другом языке, или технический термин либо подчеркивается какая-то мысль. В устаревших спецификациях с помощью элемента i выделялся курсивный текст.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

The Western Black Widow Spider, `<i>Latrodectus hesperus</i>`, is commonly found...

Элемент `iframe`

`<iframe>...</iframe>`

Определяет встроенный фрейм, используемый для внедрения документа HTML в отдельный просматриваемый контекст (окно), который вложен в родительский документ. Во встроенном фрейме отображается содержимое внешнего документа, а также могут выводиться устройства прокрутки, если контент не помещается в выбранную область окна. Встроенные фреймы могут позиционироваться подобно изображениям.

Применение

Категории

Потоковый контент, фразообразующий контент, внедренный контент, интерактивный контент, явный контент.

Допустимые контексты

В области возможного появления внедренного контента.

Разрешенный контент

Текстовый контент. Если используется алгоритм, то генерируемый вывод должен включать список узлов, в который не входят ошибки, возникающие в процессе синтаксического анализа, либо сценарии. Этот список может включать только фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`allowfullscreen`

Не поддерживается в HTML 4.01. Показывает, что объекты, находящиеся в `iframe`, могут использовать `requestFullScreen()`.

`height = "число"`

Указывает высоту встроенного фрейма (в пикселях или в виде процента от размера окна).

`name = текст"`

Назначает имя встроенному фрейму, используемое для ссылки на него с помощью целевых ссылок.

`sandbox = "allow-forms | allow-pointer-lock | allow-popups | allow-same-origin | allow-scripts | allow-top-navigation"`

Не поддерживается в HTML 4.01. Используется для активизации сценариев, всплывающих окон, подключаемых модулей и передачи данных формы во внедренных документах.

`seamless (seamless="seamless" в XHTML)`

Не поддерживается в HTML 4.01. При использовании этого атрибута браузер трактует внедренный документ таким образом, как будто он был частью документа предка для целевых ссылок, структуры документа и наследования CSS.

`src = "URL-ссылка"`

Указывает URL-адрес документа HTML, который изначально отображается во встроенном фрейме.

`srcdoc = "контент HTML"`

Не поддерживается в HTML 4.01. Указывает фактический контент HTML, отображаемый во фрейме, который задается с помощью `iframe`.

`width = "число"`

Указывает ширину встроенного фрейма, выраженную в пикселях (как процентное соотношение от размера окна). В Internet Explorer и Navigator используется заданное по умолчанию значение ширины, равное 300 пикселям.

Пример

```
<h1>Inline (Floating) Frames</h1>
<iframe src="ads.html" width="300" height="200"></iframe>
```

Элемент `img`

`` (XHTML: `` или ``)

Представляет изображение в потоковом контенте. Атрибуты `src` и `alt` обязательны для применения.

Применение

Категории

Потоковый контент, фразообразующий контент, внедренный контент, осозаемый контент, интерактивный контент (при наличии атрибута `usemap`).

Допустимые контексты

В области возможного появления внедренного контента.

Разрешенный контент

Пустой.

Начальный/конечный теги

Это пустой (`void`) элемент, который имеет лишь начальный тег и не может включать какой-либо контент. В HTML использование закрывающего тега запрещено. В XHTML элемент закрывается с помощью обратной косой черты (`` или ``).

Атрибуты

Глобальные атрибуты HTML5

`alt = "текст"`

Обязательный. Поддерживает строку альтернативного текста, которая выводится в том случае, когда изображение не отображается или не может быть отображено.

`crossorigin = "anonymous |use-credentials"`

Не поддерживается в HTML 4.01. На сайте, поддерживающем кросс-доменний доступ к файлам, этот атрибут указывает на то, будут ли запросы из других доменов представлять полномочия, требуемые для доступа к изображению.

`height = "число"`

Указывает высоту изображения, заданную в пикселях. Этот атрибут не обязательен, но рекомендуется для ускорения визуализации веб-страницы.

`ismap (ismap="ismap" в XHTML)`

Указывает на то, что изображение является основой серверной карты изображений (изображение, содержащее несколько гипертекстовых ссылок).

`src = "URL-ссылка"`

Обязательный. Указывает местоположение отображаемого графического файла.

`srcset = "строка-кандидат в изображении"`

Только WHATWG. С помощью этого атрибута создатели веб-страниц могут выбирать альтернативные изображения, выбираемые для окон просмотра, имеющих разные размеры и большие разрешения (в пикселях). В качестве значения используется список местоположений изображений, разделенных запятыми, с дескриптором, который определяет требования, соблюдение которых нужно для использования устройства. Например, инструкция `srcset = "image-HD.jpg 2x' image-phone.jpg 240w"` передает пользовательским агентам инструкции, в которых указывается на необходимость вывода отображения `image-HD.jpg`, если дисплей имеет двойную пиксельную плотность

(например, Apple Retina), либо `image-phone.png`, если разрешение экрана устройства составляет 240 пикселей.

`usemap = "URL-ссылка"`

Указывает карту изображений, содержащую координаты и ссылки для клиентской карты изображений (изображение, содержащее несколько гипертекстовых ссылок).

`width = "число"`

Указывает ширину изображения, заданную в пикселях. Этот атрибут необязателен, но желателен для ускоренного отображения веб-страницы.

Пример

```
<p>Your ideal pet: </p>
```

Элемент `input`

`<input>` (XHTML: `<input />` либо `<input />`)

Применяется для создания разнообразных элементов управления для данных, вводимых в поля формы. Тип элемента управления задается с помощью атрибута `type`. Затем следует полный список атрибутов (с описаниями), которые могут быть использованы совместно с этим элементом. Обратите внимание, что далеко не все атрибуты могут быть использованы со всеми типами элементов управления. Список атрибутов, относящихся к определенному типу элемента управления, приводится ниже.

Применение

Категории

Потоковый контент, фразообразующий контент. Если атрибут `type` не находится в состоянии `Hidden`, то используются интерактивный контент, явный контент, а также “контент, отображаемый в виде списка, снабженный подписями, передаваемый форме и связанный с формами, для которых выполняется сброс данных”. Если для атрибута `type` выбрано состояние `Hidden`, то используется “контент, отображаемый в виде списка, передаваемый форме и связанный с формами, для которых выполняется сброс данных”.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Пустой.

Начальный/конечный теги

Поскольку этот элемент пустой (`void`), он может иметь только открывающий тег и не включать какой-либо контент. В HTML использование закрывающего тега запрещено. В XHTML элемент должен закрываться с помощью обратной косой черты (`<input/>` или `<input />`).

Атрибуты

Глобальные атрибуты HTML5

`accept = "тип MIME"`

Указывает список разделенных запятыми типов контента, которые будут корректно восприниматься сервером, выполняющим обработку данных формы. Может применяться для фильтрации несовместимых файлов при отображении запроса пользователю, в котором пользователю предлагается выбрать файлы для отправки на сервер. Применяется только к типу вводимых данных `file`.

`alt = "текст"`

Указывает альтернативный текст, отображаемый вместо изображения, используемого в качестве кнопки. Применяется по отношению к типу ввода данных `image`.

`autocomplete = "on | off"`

Не поддерживается в HTML 4.01. Позволяет пользовательскому агенту (браузеру) заполнять поле автоматически (`on`) или требует, чтобы пользователь вводил информацию каждый раз (`off`). Если этот атрибут не установлен, то элемент управления наследует настройку `autocomplete` для связанного с ним элемента `form`. Применяется к типам ввода данных `text`, `password`, `email`, `tel`, `search`, `url`, `number`, `range`, `date`, `time`, `datetime`, `datetimelocal`, `month`, `week` и `color`.

`autofocus` (`autofocus="autofocus" в XHTML`)

Не поддерживается в HTML 4.01. Указывает на то, что элемент управления должен принимать фокус (выделен и готов к вводу данных пользователем) при загрузке документа. Применяется ко всем типам вводимых данных.

`checked` (`checked="checked" в XHTML`)

Если этот атрибут добавляется к переключателю или к флажку, то соответствующий тип ввода данных выбирается при загрузке страницы. Применяется к типам ввода данных `checkbox` и `radio`.

`dirname = "текстовая строка"`

Не поддерживается в HTML 4.01. Активизирует изменение направления элемента (`ltr` или `rtl`) путем указания имени поля, содержащего соответствующее значение направления. Это позволяет поддерживать в форме настройку выбираемого пользователем направления при вводе данных. Применяется к типам ввода данных `text` и `search`.

`disabled` (`disabled="disabled" в XHTML`)

Отключает контроль ввода данных пользователем. Может изменяться только путем выполнения сценария. Браузеры отображают отключенные элементы управления по-разному (например, выделив их серым цветом), что может использоваться для скрытия определенных элементов управления до тех пор, пока не будут переданы требуемые данные. Применяется по отношению ко всем типам ввода данных.

`form = "идентификатор владельца формы"`

Не поддерживается в HTML 4.01. Явно связывает элемент управления вводом данных со связанной с ним формой (с соответствующим владельцем формы). При использовании этого метода не нужно, чтобы элемент управления вводом данных был потомком применяемого элемента `form`. Применяется ко всем типам ввода данных.

`formaction = "URL-ссылка"`

Не поддерживается в HTML 4.01. Указывает приложение, которое обрабатывает форму. Используется только вместе с кнопкой передачи данных (`type="submit"` или

"image") и выполняет ту же функцию, что и атрибут action для элемента form.

formenctype = "тип контента"

Не поддерживается в HTML 4.01. Указывает, каким образом значения формы кодируются вместе с типом метода post. Применяется только совместно с кнопкой передачи данных (type="submit" или "image") и выполняет ту же функцию, что и атрибут enctype для элемента form. В качестве значения, заданного по умолчанию, используется Internet Media Type (application/x-www-form-urlencoded). Значение multipart/form-data должно использоваться в комбинации с типом ввода file. Значение text/plain применяется для присваивания типу MIME значения text/plain.

formmethod = "get | post"

Не поддерживается в HTML 4.01. Указывает тип метода HTTP, применяемый для передачи данных формы. Используется только вместе с кнопкой передачи данных (type="submit" или "image") и выполняет ту же функцию, что и атрибут method элемента form.

formnovalidate (formnovalidate="formnovalidate" в XHTML)

Не поддерживается в HTML 4.01. Указывает на то, что передаваемые данные формы не проверяются на достоверность. Используется только вместе с кнопкой передачи данных (type="submit" или "image") и выполняет ту же функцию, что и атрибут novalidate для элемента form.

formtarget = "имя"

Не поддерживается в HTML 4.01. Указывает целевое окно для отображения результатов обработки данных формы. Используется только вместе с кнопкой передачи данных формы (type="submit" или "image") и выполняет ту же функцию, что и атрибут target для элемента form.

height = "количество пикселей"

Не поддерживается в HTML 4.01. Указывает высоту изображения кнопки в том случае, если в качестве типа ввода данных выбран image.

`inputmode = "verbatim | latin | latin-name | latin-prose | full-width-latin | kana | katakana | numeric | tel | email | url"`

Не поддерживается в HTML 4.01. Показывает, какой тип механизма ввода наиболее полезен для пользователей, которые вводят контент с помощью элемента управления формой. Применяется по отношению к типам ввода данных `text` и `search`.

`list = "идентификатор списка данных"`

Не поддерживается в HTML 4.01. Указывает на то, что элемент управления включает список предопределенных предложений для пользователя. Этот список отображается с помощью элемента `datalist`. В качестве значения атрибута `list` используется `id` связанного элемента `datalist`. Применяется к типам ввода данных `text`, `email`, `tel`, `search`, `url`, `number`, `range`, `date`, `time`, `datetime-local`, `datetime`, `month`, `week` и `color`.

`max = "число или строка"`

Не поддерживается в HTML 4.01. Указывает верхнюю границу приемлемого диапазона значений для элемента. Значение `max` не должно быть меньшим значения `min`. Применяется к типам ввода данных `number`, `range`, `date`, `time`, `datetime`, `datetime-local`, `month` и `week`.

`maxlength = "число"`

Указывает максимальное количество символов, которые пользователь может ввести для элементов ввода данных `text`, `password`, `email`, `search`, `tel` или `url`.

`min = "число или строка"`

Не поддерживается в HTML 4.01. Указывает нижнюю границу приемлемого диапазона значений для элемента. Значение `min` определяет базу для операций, выполняемых с помощью инструкции `step`. Применяется к типам ввода данных `number`, `range`, `date`, `time`, `datetime`, `datetime-local`, `month` и `week`.

`multiple (multiple="multiple" в XHTML)`

Не поддерживается в HTML 4.01. Указывает на то, что пользователю можно выбирать несколько значений. Применяется к типам ввода данных `file` и `email`.

`name = "текст"`

Присваивает имя элементу управления; это имя используется сценарием для создания ссылки на элемент управления.

`pattern = "регулярное выражение JavaScript"`

Не поддерживается в HTML 4.01. Задает регулярное выражение, применяемое для проверки значения элемента управления. Это полезно, если нужно убедиться в том, что данные, вводимые пользователем, согласуются с форматом ожидаемого значения, например с форматом телефонного номера или адреса электронной почты. Атрибут `title` может быть использован вместе с атрибутом `pattern` для поддержки описания ожидаемого шаблона/формата ввода. Применяется к типам ввода данных `text`, `password`, `email`, `tel`, `search` и `url`.

`placeholder = "число"`

Не поддерживается в HTML 4.01. Применяется для отображения краткой (одно слово или краткая фраза) подсказки или примера, призванного помочь пользователю ввести корректные данные или выбрать нужный формат. Если нужно более длинное описание, то используйте атрибут `title`. Применяется к типам ввода данных `text`, `password`, `email`, `tel`, `search` и `url`.

`readonly (readonly="readonly" в XHTML)`

С помощью этого атрибута указывается, что введенные в форму данные не могут изменяться пользователем. Применяется к типу ввода данных `text`, `password`, `email`, `tel`, `search`, `url`, `number`, `date`, `time`, `datetime`, `datetime-local`, `month` и `week`.

`required (required="required" в XHTML)`

Не поддерживается в HTML 4.01. Этот атрибут (в случае его наличия) указывает на обязательный характер ввода данных. Применяется к типам ввода данных `text`, `password`, `checkbox`, `radio`, `file`, `email`, `tel`, `search`, `url`,

`number, date, time, datetime, datetime-local, month и week.`

`size = "число"`

Задает ширину элемента управления ввода текста, измеряемого в количестве символов. Пользователи могут вводить данные, длина которых превышает ширину поля, выполняя прокрутку поля вправо. Применяется к типам ввода данных `text, password, email, tel, search и url`.

`src = "URL-ссылка"`

Если в качестве типа ввода данных используется `image`, то этот атрибут указывает на местоположение изображения, применяемого в качестве кнопки, воспринимающей давление.

`step = "any | число"`

Не поддерживается в HTML 4.01. Задает степень структурирования, которая ожидается и требуется от значения, путем ограничения разрешенного значения допускаемыми единицами. Значение этого атрибута зависит от типа элемента управления вводом данных. Это может быть число, большее нуля, или ключевое слово `any`, которое обеспечивает возможность использования произвольных единиц. Применяется по отношению к типам ввода данных `number, range, date, time, datetime, datetime-local, month и week`.

`type = "button | checkbox | color | date | time | datetime | datetime-local | email | file | hidden | image | month | number | password | radio | range | reset | search | submit | tel | text | url | week"`

Задает тип данных и связанный элемент управления формой. Описание всех типов данных и связанных с ними атрибутов приведено ниже.

`value = "текст"`

Указывает начальное значение для данного элемента управления. Применяется ко всем типам ввода данных, за исключением `image`.

`width = "пиксельное значение"`

Не поддерживается в HTML 4.01. Указывает ширину нижнего изображения при выборе типа ввода данных `image`.

input type = "button"

Создает настраиваемую кнопку, которую можно “нажать”. Настраиваемым кнопкам не присуще определенное поведение, они могут применяться для вызова функций, созданных с элементами управления JavaScript. Данные из элементов управления type="button" никогда не отправляются вместе с формой, когда ее данные передаются серверу. Эти элементы управления предназначены для использования только вместе со сценариями в браузере.

```
<input type="button" value="Push Me!">
```

Глобальные атрибуты HTML5

`autofocus` (`autofocus="autofocus"` в XHTML) (не поддерживается в HTML 4.01)

`disabled` (`disabled="disabled"` в XHTML)

`form` = “идентификатор владельца формы” (не поддерживается в HTML 4.01)

`name` = “текст”

`value` = “текст”

input type = "checkbox"

Создает элемент ввода типа “флажок” в области действия элемента `form`. Флажки подобны переключателям (“Вкл./выкл.”), которые может переключать пользователь. Несколько флажков, относящихся к одной группе, могут быть выбраны одновременно. После передачи данных, введенных в форму, фактически серверу передаются данные только в том случае, если установлены соответствующие флажки.

```
<p>Which of the following operating systems have you  
used?</p>  
<ul>  
<li><input type="checkbox" name="os" value="Win">Windows</li>  
<li><input type="checkbox" name="os" value="Linux"  
checked>Linux</li>  
<li><input type="checkbox" name="os" value="OSX"  
checked>Macintosh OSX</li>  
<li><input type="checkbox" name="os" value="DOS">DOS</li>  
</ul>
```

Глобальные атрибуты HTML5

`autofocus (autofocus="autofocus" в XHTML) (не поддерживается в HTML 4.01)`

`checked (checked="checked" в XHTML)`

`disabled (disabled="disabled" в XHTML)`

`form = "идентификатор владельца формы" (не поддерживается в HTML 4.01)`

`name = "текст"`

`required (required="required" в XHTML) (не поддерживается в HTML 4.01)`

`value = "текст"`

input type = "color"

Этот тип ввода данных недопустим в HTML 4.01.

Создает элемент управления окна выбора цвета, применяемый для выбора цвета.

```
<input type="color" name="background" value="#3D458A">
```

Глобальные атрибуты HTML5

`autocomplete = "on | off"`

`autofocus (autofocus="autofocus" в XHTML)`

`disabled (disabled="disabled" в XHTML)`

`form = "идентификатор владельца формы"`

`list = "идентификатор списка данных"`

`name = "имя"`

`value = "текст"`

input type = "date"

Этот тип ввода данных недопустим в HTML 4.01.

Создает элемент управления ввода даты, такой как всплывающий календарь, для выбора даты (год, месяц, день), не учитывая часовой пояс. Первоначальное значение должно быть представлено в формате даты ISO.

```
<input type="date" name="birthday" value="2004-01-14">
```

Глобальные атрибуты HTML5

`autocomplete = "on | off"`

`autofocus (autofocus="autofocus" в XHTML)`

`disabled (disabled="disabled" в XHTML)`

```
form = "идентификатор владельца формы"
list = "идентификатор списка данных"
max = "число или строка"
min = "число или строка"
name = "имя"
readonly(readonly="readonly" в XHTML)
required(required="required" в XHTML)
step = "any | число"
value = "ГГГГ-ММ-ДД"
```

input type = "datetime"

Этот тип ввода данных недопустим в HTML 4.01.

Создает объединенный элемент управления ввода данных типа “дата/время”. В качестве значения используются отформатированные в соответствии со стандартом ISO дата и время, которые определяются и передаются как время UTC (эквивалент часового пояса GMT).

```
<input type="datetime" name="post" value=
"2004-01-14T03:13:00-5:00">
```

Глобальные атрибуты HTML5

```
autocomplete = "on | off"
autofocus (autofocus="autofocus" в XHTML)
disabled(disabled="disabled" в XHTML)
form = "идентификатор владельца формы"
list = "идентификатор списка данных"
max = "число или строка"
min = "число или строка"
name = "имя"
readonly(readonly="readonly" в XHTML)
required(required="required" в XHTML)
step = "any | число"
value = "ГГГГ-ММ-ДДТЧч:мм:cctZD"
```

input type = "datetime-local"

Этот тип ввода данных недопустим в HTML 4.01.

Создает элемент управления ввода комбинации “дата/время”, принимая время, соответствующее локальному часовому поясу.

Первоначальные значения должны быть представлены в формате ISO “дата/время”.

```
<input type="datetime-local" name="post" value="2009-06-23T13:44:16:00">
```

Глобальные атрибуты HTML5

autocomplete = "on | off"

autofocus (autofocus="autofocus" в XHTML)

disabled (disabled="disabled" в XHTML)

form = "идентификатор владельца формы"

list = "идентификатор списка данных"

max = "число или строка"

min = "число или строка"

name = "имя"

readonly (readonly="readonly" в XHTML)

required (required="required" в XHTML)

step = "any | число"

value = "ГГГГ-ММ-ДДТчч:мм:сс"

input type = "email"

Этот тип ввода данных недопустим в HTML 4.01.

Создает текстовое поле ввода, предназначенное для ввода одного или более адресов электронной почты. Пользовательский агент может применять шаблоны для проверки соответствия вводимых данных формату адреса электронной почты.

```
<input type="email" name="address" value="jn@example.com">
```

Глобальные атрибуты HTML5

autocomplete = "on | off"

autofocus (autofocus="autofocus" в XHTML)

disabled (disabled="disabled" в XHTML)

form = "идентификатор владельца формы"

list = "идентификатор списка данных"

maxlength = "число"

multiple (multiple="multiple" в XHTML)

pattern = "регулярное выражение JavaScript"

placeholder = "текст"

name = "имя"

```
readonly (readonly="readonly" в XHTML)
required (required="required" в XHTML)
size = "число"
value = "текст"
```

input type = "file"

С помощью механизма передачи данных формы обеспечивает пользователям возможность передачи внешних файлов вместе с данными формы.

```
<form enctype="multipart/form-data">
<p>Send this file with my form information:<br>
  <input type="file" name="attachment" size="28">
</p>
</form>
```

Глобальные атрибуты HTML5

accept = "тип MIME"
autofocus (autofocus="autofocus" в XHTML) (не поддерживается в HTML 4.01)
disabled (disabled="disabled" в XHTML)
form = "идентификатор владельца формы" (не поддерживается в HTML 4.01)
multiple (multiple="multiple" в XHTML) (не поддерживается в HTML 4.01)
name = "текст"
required (required="required" в XHTML) (не поддерживается в HTML 4.01)

input type = "hidden"

Создает элемент управления, который не отображается в окне браузера. Скрытые элементы управления могут применяться для передачи серверу специальной информации по обработке формы, которую пользователь не может просмотреть или изменить.

```
<input type="hidden" name="productID" value="12-XL">
```

Глобальные атрибуты HTML5

disabled (disabled="disabled" в XHTML) (не поддерживается в HTML 4.01)

`form = "идентификатор владельца формы" (не поддерживается в HTML 4.01)`

`name = "текст" (обязательный)`

`value = "текст" (обязательный)`

`input type = "image"`

Обеспечивает возможность применения изображения в качестве заменителя кнопки `submit`. Если кнопка, заданная с помощью выражения `type="image"`, нажата, то передаются данные формы.

`<input type="image" src="graphics/send.gif" alt="Send me">`

Глобальные атрибуты HTML5

`alt = "текст"`

`autofocus (autofocus="autofocus" в XHTML) (не поддерживается в HTML 4.01)`

`disabled(disabled="disabled" в XHTML)`

`form = "идентификатор владельца формы" (не поддерживается в HTML 4.01)`

`formaction = "URL-ссылка" (не поддерживается в HTML 4.01)`

`formenctype = "тип контента" (не поддерживается в HTML 4.01)`

`formmethod = "get | post" (не поддерживается в HTML 4.01)`

`formnovalidate (formnovalidate="formnovalidate" в XHTML) (не поддерживается в HTML 4.01)`

`formtarget = "имя" (не поддерживается в HTML 4.01)`

`height = "количество пикселей" (не поддерживается в HTML 4.01)`

`name = "текст"`

`src = "URL-ссылка"`

`width = "количество пикселей" (не поддерживается в HTML 4.01)`

`input type = "month"`

Этот тип ввода данных недопустим в HTML 4.01.

Создает элемент управления для ввода даты (всплывающий календарь), позволяющий выбрать определенный месяц года.

```
<input type="month" name="expires" value="2009-09">
```

Глобальные атрибуты HTML5

autocomplete = "on | off"

autofocus (autofocus="autofocus" в XHTML)

disabled(disabled="disabled" в XHTML)

form = "идентификатор владельца формы"

list = "идентификатор списка данных"

max = "число или строка"

min = "число или строка"

name = "имя"

readonly (readonly="readonly" в XHTML)

required (required="required" в XHTML)

step = "any | число"

value = "ГГГГ-ММ"

input type = "number"

Этот тип ввода данных недопустим в HTML 4.01.

Создает элемент управления (текстовое поле или счетчик), применяемый для ввода числового значения.

```
<input type="number" name="price" min="100000"
max="300000" step="10000">
```

Глобальные атрибуты HTML5

autocomplete = "on | off"

autofocus (autofocus="autofocus" в XHTML)

disabled(disabled="disabled" в XHTML)

form = "идентификатор владельца формы"

list = "идентификатор списка данных"

max = "число"

min = "число"

name = "имя"

placeholder = "текстовая строка"

readonly (readonly="readonly" в XHTML)

required (required="required" в XHTML)

step = "any | число"

value = "текстовая строка или числовая строка"

input type = "password"

Создает элемент ввода текста (подобно `<input type="text">`), обеспечивающий скрытие символов при вводе (например, вместо символов отображаются звездочки или кружочки). Учтите, что вводимая информация не шифруется, поэтому не следует рассматривать этот элемент как подлинно безопасный.

```
<input type="password" name="password" size="8"  
maxlength="8" value="abcdefg">
```

Глобальные атрибуты HTML5

`autocomplete = "on | off"` (**не поддерживается в HTML 4.01**)

`autofocus (autofocus="autofocus" в XHTML)` (**не поддерживается в HTML 4.01**)

`disabled(disabled="disabled" в XHTML)`

`form = "идентификатор владельца формы"` (**не поддерживается в HTML 4.01**)

`maxlength = "число"`

`name = "текст"`

`pattern = "регулярное выражение JavaScript"` (**не поддерживается в HTML 4.01**)

`placeholder = "текст"` (**не поддерживается в HTML 4.01**)

`readonly(readonly="readonly" в XHTML)`

`required(required="required" в XHTML)` (**не поддерживается в HTML 4.01**)

`size = "число"`

`value = "текст"`

input type = "radio"

Создает переключатель, который может находиться во включенном либо выключенном положении. Если несколько переключателей используют одно и то же имя элемента управления, то во включенном состоянии может находиться лишь один переключатель в группе, а остальные являются выключенными. Этим переключатели отличаются от флагков, которые могут устанавливаться одновременно. При передаче данных формы передаются лишь те данные, которые соответствуют включенному переключателю.

```
<p>Which of the following operating systems do you like best?</p>
<ul>
<li><input type="radio" name="os" value="Win">Windows</li>
<li><input type="radio" name="os" value="Linux">Linux</li>
<li><input type="radio" name="os" value="OSX" checked> Macintosh OSX</li>
<li><input type="radio" name="os" value="DOS">DOS</li>
</ul>
```

Глобальные атрибуты HTML5

checked (checked="checked" в XHTML)

autofocus (autofocus="autofocus" в XHTML) (не поддерживается в HTML 4.01)

disabled (disabled="disabled" в XHTML)

form = "идентификатор владельца формы" (не поддерживается в HTML 4.01)

name = "текст" (обязательный)

required (required="required" в XHTML) (не поддерживается в HTML 4.01)

value = "текст" (обязательный)

input type = "range"

Этот тип ввода данных недопустим в HTML 4.01.

Создает элемент управления в виде ползунка, с помощью которого можно быстро вводить значение (необязательно точно). Диапазон вводимых значений начинается величиной, задаваемой с помощью атрибута min (0 по умолчанию), и завершается величиной, задаваемой с помощью атрибута max (100 по умолчанию).

```
<input type="range" name="satisfaction" min="0" max="10">
```

Глобальные атрибуты HTML5

autocomplete = "on | off"

autofocus (autofocus="autofocus" в XHTML)

disabled (disabled="disabled" в XHTML)

form = "идентификатор владельца формы"

list = "идентификатор списка данных"

max = "число"

min = "число"

```
name = "имя"  
step = "any | число"  
value = "текст"
```

input type = "reset"

Создает кнопку сброса данных, после щелчка на которой очищается контент элементов в форме (или возвращает значения этих элементов, заданные по умолчанию).

```
<input type="reset" value="Start Over">
```

Глобальные атрибуты HTML5

autofocus (autofocus="autofocus" в XHTML) (**не поддерживается в HTML 4.01**)

disabled (disabled="disabled" в XHTML)

form = "идентификатор владельца формы" (**не поддерживается в HTML 4.01**)

name = "текст"

value = "текст"

input type = "search"

Этот тип ввода данных недопустим в HTML 4.01.

Создает односторочный элемент управления, предназначенный для ввода поисковых запросов.

```
<input type="search" name="srch" size="25"  
value="поисковый запрос">
```

Глобальные атрибуты HTML5

autocomplete = "on | off"

autofocus (autofocus="autofocus" в XHTML)

disabled (disabled="disabled" в XHTML)

dirname = "текстовая строка"

form = "идентификатор владельца формы"

inputmode= "verbatim | latin | latin-name | latin-prose | full-width-latin | kana | katakana | numeric | tel | email | url"

list = "идентификатор списка данных"

maxlength = "число"

name = "имя"

```
pattern = "регулярное выражение JavaScript"
placeholder = "текст"
readonly (readonly="readonly" в XHTML)
required (required="required" в XHTML)
size = "число"
value = "текст"
```

input type = "submit"

Создает элемент управления в виде кнопки передачи данных. После нажатия этой кнопки находящаяся в полях формы информация немедленно отправляется серверу для дальнейшей обработки.

```
<p>You have completed the form.</p>
<p><input type="submit"></p>
```

Глобальные атрибуты HTML5

autofocus (autofocus="autofocus" в XHTML) (не поддерживается в HTML 4.01)

disabled (disabled="disabled" в XHTML)

form = "идентификатор владельца формы" (не поддерживается в HTML 4.01)

formaction = "URL-ссылка" (не поддерживается в HTML 4.01)

formenctype = "тип контента" (не поддерживается в HTML 4.01)

formmethod = "get | post" (не поддерживается в HTML 4.01)

formnovalidate (formnovalidate="formnovalidate" в XHTML) (не поддерживается в HTML 4.01)

formtarget = "имя" (не поддерживается в HTML 4.01)

name = "текст"

value = "текст"

input type = "tel"

Этот тип ввода данных недопустим в HTML 4.01.

Создает элемент управления, предназначенный для ввода и редактирования телефонного номера.

```
<input type="tel" name="homeno" value="000-000-0000">
```

Глобальные атрибуты HTML5

autocomplete = "on | off"
autofocus (autofocus="autofocus" в XHTML)
disabled(disabled="disabled" в XHTML)
form = "идентификатор владельца формы"
list = "идентификатор списка данных"
maxlength = "число"
name = "имя"
pattern = "регулярное выражение JavaScript"
placeholder = "текст"
readonly(readonly="readonly" в XHTML)
required(required="required" в XHTML)
size = "число"
value = "текст"

input type = "text"

Создает элемент управления, предназначенный для ввода текста. Этот элемент управления задан по умолчанию и является одним из самых полезных и чаще всего применяемых. Текст, предназначенный для атрибута value, отображается в элементе управления текстом при загрузке формы.

```
<input type="text" name="username" size="15"  
  
maxlength="50" value="enter your name">
```

Глобальные атрибуты HTML5

autocomplete = "on | off" (не поддерживается в HTML 4.01)

autofocus (autofocus="autofocus" в XHTML) (не поддерживается в HTML 4.01)

disabled(disabled="disabled" в XHTML)

dirname = "текстовая строка" (не поддерживается в HTML 4.01)

form = "идентификатор владельца формы" (не поддерживается в HTML 4.01)

inputmode = "verbatim | latin | latin-name | latin-prose | full-width-latin | kana | katakana |

`numeric | tel | email | url`" (**не поддерживается в HTML 4.01**)

`list = "идентификатор списка данных"` (**не поддерживается в HTML 4.01**)

`maxlength = "число"`

`name = "имя"`

`pattern = "регулярное выражение JavaScript"` (**не поддерживается в HTML 4.01**)

`placeholder = "текст"` (**не поддерживается в HTML 4.01**)

`readonly (readonly="readonly" в XHTML)`

`required (required="required" в XHTML)` (**не поддерживается в HTML 4.01**)

`size = "число"`

`value = "текст"`

input type = "time"

Этот тип ввода данных недопустим в HTML 4.01.

Создает элемент управления, предназначенный для отображения показаний времени (часы, минуты, секунды и миллисекунды) без вывода сведений о часовом поясе.

```
<input type="time" name="currenttime" value="23:15:00">
```

Глобальные атрибуты HTML5

`autocomplete = "on | off"`

`autofocus (autofocus="autofocus" в XHTML)`

`disabled (disabled="disabled" в XHTML)`

`form = "идентификатор владельца формы"`

`list = "идентификатор списка данных"`

`max = "число или строка"`

`min = "число или строка"`

`name = "имя"`

`readonly (readonly="readonly" в XHTML)`

`required (required="required" в XHTML)`

`step = "any | число"`

`value = "чч:мм:сс"`

input type = "url"

Этот тип ввода данных недопустим в HTML 4.01.

Создает текстовый элемент управления, предназначенный для ввода единственной абсолютной URL-ссылки. Пользовательский агент может верифицировать достоверность вводимых данных, чтобы убедиться в том, что выбран корректный формат записи URL-ссылки. Если же формат некорректен, то возвращается сообщение об ошибке.

```
<input type="url" name="blog" size="25"  
value="http://www.example.com">
```

Глобальные атрибуты HTML5

autocomplete = "on | off"
autofocus (autofocus="autofocus" в XHTML)
disabled (disabled="disabled" в XHTML)
form = "идентификатор владельца формы"
list = "идентификатор списка данных"
maxlength = "число"
name = "имя"
pattern = "регулярное выражение JavaScript"
placeholder = "текст"
readonly (readonly="readonly" в XHTML)
required (required="required" в XHTML)
size = "число"
value = "текст"

input type = "week"

Этот тип ввода данных недопустим в HTML 4.01.

Создает элемент ввода даты, такой как всплывающий календарь, предназначенный для выбора определенной недели в году. Значения предоставляются в формате ISO для нумерации недель.

```
<input type="week" name="thisweek" value="2009-W34">
```

Глобальные атрибуты HTML5

autocomplete = "on | off"
autofocus (autofocus="autofocus" в XHTML)
disabled (disabled="disabled" в XHTML)
form = "идентификатор владельца формы"

```
list = "идентификатор списка данных"  
max = "число или строка"  
min = "число или строка"  
name = "имя"  
readonly (readonly="readonly" в XHTML)  
required (required="required" в XHTML)  
step = "any | число"  
value = "ГГГГ-#недели"
```

Элемент `ins`

<ins>...</ins>

С помощью этого элемента выделяется текст, который был вставлен в документ. Используется для разметки юридических документов и других текстов, в которых нужно отслеживать изменения. Этот элемент дополняется элементом, помечающим удаленный текст (`del`). Элемент `ins` может применяться для выделения встроенных либо блочных элементов. Но если этот элемент используется в качестве встроенного (как внутри области, выделенной с помощью элемента `p`), то он не может вставлять элементы блочного уровня, поскольку это нарушило бы правила вложения элементов.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Прозрачный (производный от модели контента родительского элемента).

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`cite = "URL-ссылка"`

С помощью этого атрибута указывается исходный документ, в котором объясняются причины изменения текущего документа.

`datetime = "ГГГГ-ММ-ДДТЧЧ:ММ:ССTZD"`

Указывает дату и время внесения изменений в документ.

Чтобы получить дополнительные сведения о формате даты/времени, обратитесь к разделу об элементе `del`.

Пример

```
<li>Chief Executive Officer: <del title="retired">  
Peter Pan</del> <ins>Pippi Longstocking</ins></li>
```

Элемент `kbd`

`<kbd>...</kbd>`

Название этого элемента представляет собой сокращение от слова “keyboard” (клавиатура); указывает на текст (или голосовой ввод), задаваемый пользователем.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

< p > Enter your coupon code. Example: < kbd > AX4003 </ kbd > </ p >

Элемент keygen

<keygen> (XHTML: <keygen/> или <keygen />)

Не поддерживается в HTML 4.01. Используется как часть элемента form для генерирования пар ключей, применяемых в системах управления сертификатами, развернутыми в Интернете (для выполнения безопасных транзакций).

Применение

Категории

Потоковый контент, фразообразующий контент, интерактивный контент, явный контент и “выводимый в виде списка, подписанный, допускаемый передачу и сброс данных элемент, связанный с формой”.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Пустой.

Начальный/конечный теги

Это пустой (void) элемент, который может включать начальный тег и не содержит какой-либо контент. В HTML использование закрывающего тега не допускается. Элементы XHTML должны закрываться символами обратной косой черты (<keygen/> или <keygen />).

Атрибуты

Глобальные атрибуты HTML5

autofocus (autofocus="autofocus" в XHTML)

Указывает на то, что элемент управления будет активным и подготовленным для получения пользовательских данных при загрузке документа.

`challenge = "строка вызова"`

Поддерживает строку вызова, передаваемую вместе с ключом.

`disabled(disabled="disabled" в XHTML)`

Блокирует интерактивный статус элемента управления и препятствует передаче его значения.

`form = "идентификатор владельца формы"`

Связывает элемент с именованным элементом `form` на странице.

`keytype = "ключевое слово"`

Идентифицирует тип генерируемого ключа, например `rsa` или `ec`.

`name = "текст"`

Присваивает элементу управления идентифицирующее имя, применяемое при передаче данных формы.

Пример

Этот пример кода основан на коде, опубликованном на сайте developer.mozilla.org, и применяется с разрешения Creative Commons “Attribution-Share Alike” License.

```
<form method="post" action="http://www.example.com/
cgi-bin/decode.cgi">
  <keygen name="RSA public key" challenge="123456789"
    keytype="RSA">
  <input type="submit" name="createcertificate"
    value="Make Key">
</form>
```

Элемент `label`

`<label>...</label>`

Используется для связывания информации с элементами управления. Каждый элемент `label` связан ровно с одним элементом управления `form`. Элемент `label` может включать элемент управления `form` либо использовать атрибут `for` для идентификации элемента управления по его значению `id`.

Применение

Категории

Потоковый контент, фразообразующий контент, интерактивный контент, элемент, связанный с формой, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент, но не может включать подписываемые элементы, если только это не элемент управления с подписью, относящийся к элементу, или не потомок одного из элементов `label`.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`for = "текст"`

Явно связывает метку с элементом управления путем согласования значения атрибута `for` со значением атрибута `id` в области действия элемента управления.

`form = "идентификатор владельца формы"`

Явно связывает элемент `label` с соответствующей формой (с соответствующим владельцем формы). При использовании этого метода не требуется, чтобы метка была потомком применяемого элемента `form`.

Примеры

Элемент управления формы и относящийся к нему текст надписи содержатся внутри элемента `label`.

```
<label>Last Name: <input type="text" size="32"></label>
```

Использование метода `for/id` для связывания элемента управления формы с текстом подписи.

```
<label for="lastname">Last Name:</label>
<input type="text" id="lastname" size="32">
```

Элемент legend

```
<legend>...</legend>
```

С помощью этого элемента присваивается заголовок элементу `fieldset` (должен быть первым потомком элемента `fieldset`). Благодаря этому улучшается доступность при невизуальном отображении `fieldset`.

Применение

Категории

Нет.

Допустимые контексты

В качестве первого потомка элемента `fieldset`.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<fieldset>
  <legend>Mailing List Sign-up</legend>
  <ul>
    <li><label>Add me to your mailing list
      <input type="radio" name="list"></label></li>
    <li><label>No thanks <input name="list" value="no">
      </label></li>
    </ul>
</fieldset>
```

Элемент li

```
<li>...</li>
```

С помощью этого элемента определяется позиция в списке. Используется внутри элементов списка `ol` и `ul`.

Применение

Категории

Нет.

Допустимые контексты

Внутри элементов ol, ul и menu.

Разрешенный контент

Потоковый контент.

Начальный/конечный теги

Обязательный/необязательный (XHTML: обязательный/обязательный).

Атрибуты

Глобальные атрибуты HTML5

`value = "число"`

Указывает номер элемента, если li является частью упорядоченного списка (ol). При этом нумерация последующих элементов списка возрастает, начиная с указанного числа.

Пример

```
<ol>
  <li>Preheat oven to 300.</li>
  <li>Wrap garlic in foil.</li>
  <li>Bake for 2 hours.</li>
</ol>
```

Элемент link

`<link>` (XHTML: `<link/>` или `<link />`)

Задает связь между текущим и другими документами. В качестве подобной связи может выступать индекс, следующий и предыдущий элементы (чаще всего применяется для создания ссылки на внешнюю таблицу стилей).

Применение

Категории

Контент метаданных (при наличии атрибута itemprop там, где ожидается фразообразующий контент).

Допустимые контексты

В области возможного появления контента метаданных; в элементе `noscript`, который является потомком элемента `head`; при наличии атрибута `itemprop` в области появления фразообразующего контента.

Разрешенный контент

Пустой.

Начальный/конечный теги

Это пустой (`void`) элемент, т.е. имеется только начальный тег, а также не может включать какие-нибудь контенты. В HTML применение закрывающего тега запрещено. В XHTML элемент должен завершаться с помощью обратной косой черты (`<link/>` или `<link />`).

Атрибуты

Глобальные атрибуты HTML5

`crossorigin = "anonymous | use-credentials"`

Не поддерживается в HTML 4.01. На сайте, обеспечивающем возможность кросс-доменного доступа, этот атрибут применяется, чтобы указать на то, должны ли запросы из других доменов представлять удостоверения.

`href = "URL-ссылка"`

Идентифицирует связанный документ.

`hreflang = "код языка"`

Задает основной язык для связанного документа.

`media = "all | screen | print | handheld | projection | tty | tv | braille | aural"`

Идентифицирует мультимедийный объект, по отношению к которому применяется связанный ресурс. Чаще всего используется для сопоставления таблиц стилей с соответствующим мультимедийным объектом.

`rel = "ключевое слово типа ссылки"`

Описывает одну или несколько связей, установленных между текущим исходным документом и связанным документом. Применяются следующие типы взаимосвязей для элемента `link`: `alternate, author, help, icon, license, next, prefetch, prev, search, stylesheet` и `tag`.

`sizes = "any | два пиксельных размера"`

Только HTML5. Указывает размеры пиктограмм, если атрибуту `rel` элемента `link` присвоено значение `icon`.

`type = "ресурс"`

Указывает мультимедийный тип или тип контента для связанного ресурса. Значение `text/css` указывает на то, что связанный документ является внешней таблицей каскадных стилей.

Пример

```
<head>
<link rel="stylesheet" href="/ pathname/stylesheet.css">
</head>
```

Элемент `main`

```
<main>...</main>
```

Только WHATWG и HTML5.1. Представляет главную область контента в теле документа. В документе может быть только один элемент `main`, который не является потомком элементов `article`, `aside`, `footer`, `header` и `nav`.

Применение

Категории

Потоковый контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента (не может содержаться в элементах `article`, `aside`, `footer`, `header` и `nav`).

Разрешенный контент

Потоковый контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<body>
<header>
    <h1>White Rabbits Fan Site</h1>
    <nav><ul> ... </ul></nav>
</header>
<main>
    <h2>Tour Dates</h2>
    <ul>
        <li>Providence, RI</li>
        <li>Brooklyn, NY</li>
    </ul>
    ...продолжение контента...
</main>
</body>
```

Элемент map

<map>...</map>

С помощью этого элемента задается клиентская карта изображений. Включает несколько элементов area, с помощью которых определяются области щелчков внутри карты изображений.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Прозрачный (производный от модели контента предка).

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

id = "текст"

Обязательный в XHTML. Присваивает карте уникальное имя, в результате на нее можно ссылаться из ссылки, сценария или таблицы стиля.

`name = "текст"`

Обязательный. Присваивает карте изображений имя, используемое для ссылки внутри элемента `img`.

Пример

```
<map name="space" id="space">
  <area shape="rect" coords="203,23,285,106"
    href=http://www.nasa.gov alt="">
  <area shape="circle" coords="372,64,40"
    href="mypage.html" alt="">
</map>
```

Элемент `mark`

`<mark>...</mark>`

Не поддерживается в HTML 4.01. Представляет выделенный текст, который отмечен или выделен для установки ссылки или для привлечения внимания читателя. Отмеченный текст рассматривается как наиболее релевантный для пользователя.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного проявления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

В этом примере поисковый запрос пользователя (“estate tax”) помечен в возвращаемом документе.

```
<p> ... PART I. ADMINISTRATION OF THE GOVERNMENT. TITLE IX.  
TAXATION. CHAPTER 65C. MASS. <mark>ESTATE TAX</mark>. Chapter 65C: Sect. 2. Computation of <mark>estate  
tax</mark>. </p>
```

Элемент menu

```
<menu>...</menu>
```

Представляет список интерактивных параметров или команд, таких как пункты меню в веб-приложении. Если элемент `menu` создает всплывающее меню, то в качестве его контента используются один или несколько элементов `menuitem`. В стандарте HTML5 Candidate Recommendation предусмотрено, что пункты меню для элемента `menu` создаются с помощью элементов `command`. В более современных версиях HTML5.1 draft и в текущей спецификации WHATWG вместо элемента `command` используется элемент `menuitem`.

Примечание

В HTML 4.01 не рекомендуемый к применению элемент `menu` указывает список пунктов меню, который состоит из одного или более элементов списка (`li`). Эти меню предназначались для создания списка коротких пунктов, таких как меню ссылок на другие документы.

Применение

Категории

Потоковый контент; если элемент `type` находится в режиме панели инструментов, тогда явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Если же `type=popup`, то в качестве потомка элемента `menu`, который находится в состоянии всплывающего меню.

Разрешенный контент

Если type=toolbar, то потоковый контент или ни одно-го либо несколько элементов li. Если type=popup, то ни одного или нескольких элементов menuitem, ни одного или нескольких элементов hr и ни одного или нескольких элементов menu, для которых type=popup.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

label = "текст"

Не поддерживается в HTML 4.01. Указывает метку для меню, которая может отображаться во вложенных меню.

type = "popup | toolbar"

Не поддерживается в HTML 4.01. Идентифицирует тип меню, которое будет объявлено. Значение popup идентифицирует всплывающее меню или меню для элемента button, в качестве типа которого в форме выбрано значение menu. Значение toolbar означает, что меню используется в качестве панели инструментов, с которой можно тут же взаимодействовать. Если атрибут type не используется, то по умолчанию отображается список команд.

Пример (HTML5.1 и WHATWG)

```
<menu type="popup">
  <menuitem onclick="cut()" label="Cut">
  <menuitem onclick="copy()" label="Copy">
  <menuitem onclick="paste()" label="Paste">
  <menuitem onclick="delete()" label="Clear">
</menu>
```

Элемент menuitem

<menuitem> (XHTML: <menuitem/> или <menuitem />)

Не поддерживается в HTML 4.01. Только WHATWG и HTML5.1. Представляет команду, которая может быть выбрана из всплывающего меню.

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента menu, атрибуту типа которого назначен режим всплывающего меню.

Разрешенный контент

Пустой.

Начальный/конечный теги

Это пустой (void) элемент, поэтому может иметь только начальный тег и не включает какой-либо контент. В HTML использовать закрывающий тег запрещено. В XHTML элемент должен закрываться обратной косой чертой (<menuitem/> или <menuitem />).

Атрибуты

Глобальные атрибуты HTML5

`checked (checked="checked" в XHTML)`

Указывает на то, что команда выбрана. Может быть использована только в том случае, если в качестве типа выбрано значение checkbox или radio.

`command = "идентификатор мастер-команды"`

Ссылается на команду menuitem, которая была определена и установлена где-нибудь в документе или приложении.

`default (default="default" в XHTML)`

Указывает на то, что команда является функцией, выбранной по умолчанию для меню.

`disabled (disabled="disabled" в XHTML)`

Указывает на то, что команда недоступна.

`icon = "URL-ссылка"`

Указывает местоположение изображения, которое представляет команду.

`label = "текстовая строка"`

Поддерживает имя для команды, показанной в меню.

`radiogroup = "текстовая строка"`

Поддерживает название для группы команд, для которых выбран тип `radio`. Не может использоваться вместе с другими типами `menuitem`.

`type = "command | checkbox | radio"`

Показывает тип команды: либо обычная команда вместе со связанным действием (`command`), либо параметр, который может быть выбран (`checkbox`), либо единственный параметр, выбираемый в списке (`radio`).

Пример

```
<menu>
  <menuitem onclick="cut()" label="Cut">
  <menuitem onclick="copy()" label="Copy">
  <menuitem onclick="paste()" label="Paste">
  <menuitem onclick="delete()" label="Clear">
</menu>
```

Элемент `meta`

`<meta>` (XHTML: `<meta/>` или `<meta />`)

Поддерживает дополнительную информацию (метаданные) о документе. Должен находиться внутри заголовка документа. Обычно используется для идентификации типа мультимедиа и наборов символов. Также может применяться для передачи ключевых слов, информации об авторе, описания и других метаданных. Элемент `head` может включать несколько элементов `meta`.

Применение

Категории

Контент метаданных; если присутствует `itemprop`, то потоковый и фразообразующий контенты.

Допустимые контексты

В элементе `head` или в элементе `noscript`, находящемся в элементе `head`. Если присутствует атрибут `name` (<http://www.w3.org/TR/html5/document-metadata.html#attrr-meta-name>), то в области возможного

появления контента метаданных (<http://www.w3.org/TR/html5/dom.html#metadata-content-0>). Если существует атрибут `itemprop`, то в области возможного появления контента метаданных или фразообразующего контента.

Разрешенный контент

Пустой.

Начальный/конечный теги

Это пустой (`void`) элемент, поэтому имеет только начальный тег и не может включать какие-либо контенты. В HTML применение закрывающего тега запрещено. В XHTML элемент должен быть закрыт с помощью обратной косой черты (`<meta/>` или `<meta />`).

Атрибуты

Глобальные атрибуты HTML5

`charset = "набор символов"`

Не поддерживается в HTML 4.01. Может использоваться вместе с элементом `meta` в качестве заменителя метода `http-equiv` для описания набора символов документа.

`content = "текст"`

Обязательный. Указывает значение свойства элемента `meta` и всегда используется вместе с атрибутами `name`, `http-equiv` и `itemprop`.

`http-equiv = "content-language | content-type | default-style | refresh | set-cookie"`

Указанная информация трактуется так, как будто она была включена в заголовок HTTP, который отсылается сервером перед документом. Используется вместе с атрибутом `content` (вместо атрибута `name`).

`name = "текст"`

Указывает имя для свойства элемента мета. Некоторые определенные имена включают `application-name`, `author`, `description`, `generator` и `keywords`.

Пример (HTML)

```
<meta charset="UTF-8">

<meta name="author" content="Jennifer Robbins">

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

<meta http-equiv="refresh" content= "15">
```

Элемент meter

<meter>...</meter>

Не поддерживается в HTML 4.01. Представляет дробное значение или скалярный результат выполнения измерения, находящийся внутри известного диапазона (также известен как *мера*). Не должен использоваться для отображения выполнения процесса (например, в качестве индикатора выполнения) или в случае, когда неизвестно максимальное значение.

Применение

Категории

Потоковый контент, фразообразующий контент, подписываемый элемент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент (не может включать другие элементы *meter*).

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`high = "число"`

Указывает диапазон, рассматриваемый как “высший” для меры.

`low = "число"`

Указывает диапазон, который рассматривается как “нижний” для меры.

`max = "число"`

Задает максимум, или наибольшее значение диапазона.

`min = "число"`

Задает минимум, или наименьшее значение диапазона.

`optimum = "число"`

Показывает диапазон, который рассматривается как “оптimum” для меры.

`value = "число"`

Задает действительное, или “измеренное”, значение для меры.

Примеры

Следующие примеры демонстрируют три способа отображения результата измерения 50%.

```
<meter>50%</meter>
<meter min="0" max="200">100</meter>
<meter min="0" max="200" value="100"></meter>
```

Элемент nav

```
<nav>...</nav>
```

Не поддерживается в HTML 4.01. Представляет раздел документа, предназначенный для навигации. Не все списки или ссылки годятся для формирования раздела, заданного элементом `nav`; подходят только те, которые представляют главные навигационные блоки на странице или внутри раздела. В качестве ссылок, находящихся внутри элемента `nav`, могут выступать ссылки на другие документы или другие области внутри текущего документа.

Применение

Категории

Потоковый контент, секционирующий контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент (не может включать элементы main).

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<nav>
  <ul>
    <li><a href="">About us</a></li>
    <li><a href="">Contact</a></li>
    <li><a href="">Home</a></li>
  </ul>
</nav>
```

Элемент noscript

```
<noscript>...</noscript>
```

Поддерживает альтернативный контент, используемый в случае невозможности выполнения сценария. Контент этого элемента может отображаться в том случае, если пользовательский агент не поддерживает сценарии, если поддержка сценариев отключена или если браузер не распознает язык написания сценариев. Если noscript отображается в области head документа, а поддержка сценариев отключена, то могут использоваться лишь элементы link, style и meta.

Примечания

Элемент noscript не используется в синтаксисе XML HTML5, поскольку в этом случае используется синтаксический анализатор HTML.

Применение

Категории

Контент метаданных, потоковый контент, фразообразующий контент.

Допустимые контексты

В элементе `head`, если отсутствует предок элементов `script`. В области предполагаемого появления фразообразующего контента, если отсутствуют предки элементов `noscript`.

Разрешенный контент

Если поддержка сценариев отключена и элемент находится внутри элемента `head`, то он может содержать следующие в произвольном порядке нуль или более элементов `link`, элементы `style` и/или элементы `meta`. Если поддержка сценариев отключена и элемент не находится в элементе `head`, то используется прозрачный контент (модель контента производна от родительского элемента), но он не должен включать какие-либо элементы `noscript`. Иначе может включать текстовый контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<script type="text/JavaScript">
... здесь находится сценарий ...
</script>
<noscript>
    <p>Для работы этой функции нужно включить
    JavaScript.</p>
</noscript>
```

Элемент `object`

`<object>...</object>`

Обобщенный элемент, используемый для внедрения в веб-страницу внешнего ресурса (такого, как изображение, аплет, фильм или звуковой файл). Атрибуты, требуемые для элемента `object`, варьируются в зависимости от внедряемого контента. Внедренный контент может трактоваться как изображение либо вложенный контекст просмотра или же обрабатываться с помощью подключаемого модуля. Элемент `object` может содержать контент, который будет отображаться в том случае, если объект не может быть внедрен. Элемент `object` также может включать несколько элементов `param`, которые передают важную информацию объекту, когда он отображается или проигрывается. Не все объекты требуют дополнительных параметров. Элементы `object` и `param` используются совместно, чтобы позволить авторам указывать три типа информации.

- Реализация объекта, т.е. исполняемый код, который запускается с целью отображения объекта.
- Данные для воспроизведения. Атрибут `data` указывает местоположение ресурса; в большинстве случаев идет речь о внешнем файле, таком как файл фильма или PDF.
- Дополнительные настройки, требуемые объектом во время выполнения. Некоторые внедренные мультимедийные объекты требуют дополнительных настроек, которые используются при воспроизведении или отображении объекта.

Применение

Категории

Потоковый контент, фразообразующий контент, внедренный контент, интерактивный контент (при наличии атрибута `usemap`), явный контент, “выводимый в виде списка, элемент, передающий данные, элемент, связанный с формой”.

Допустимые контексты

В области возможного появления внедренного контента.

Разрешенный контент

Нуль или более элементов param, затем текущий контент и/или интерактивный контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`data = "URI-адрес"`

Указывает адрес ресурса, а применяемый синтаксис зависит от объекта.

`form = "идентификатор формы"`

Не поддерживается в HTML 4.01. Связывает объект с элементом form на странице.

`height = "число"`

Указывает высоту объекта, выраженную в пикселях.

`name = "текст"`

Указывает название объекта, на который ссылаются сцены на странице.

`type = "тип мультимедийного объекта"`

Указывает тип мультимедийного ресурса.

`typemustmatch (typemustmatch="typemustmatch" в XHTML)`

Не поддерживается в HTML 4.01. Показывает, что ресурс, выбранный с помощью атрибута data, используется только в том случае, когда совпадают значение атрибута type и тип контента ресурса.

`usemap = "URL-ссылка"`

Указывает карту изображений, используемую вместе с объектом.

`width = "число"`

Указывает ширину объекта, выраженную в пикселях.

Пример

```
<object width="640" height="360"
type="application/x-shockwave-flash"
data="flash_player.swf">
<param name="movie" value="flash_player.swf">
<param name="flashvars" value="controlbar=over&amp;
image=poster.jpg&amp;
file=yourmovie-main.mp4">

</object>
```

Элемент ol

```
<ol>...</ol>
```

Определяет упорядоченный (нумерованный) список, который включает один или несколько элементов списка (li). Пользовательский агент вставляет номера элементов автоматически.

Применение

Категории

Потоковый контент; если его потомок включает по крайней мере один элемент li, то явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Элементы li отсутствуют либо имеется несколько элементов li.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

reversed (reversed="reversed" в XHTML)

Не поддерживается в HTML 4.01. Изменяет направление числовой последовательности на обратное, от наибольшего значения к наименьшему.

`start = "число"`

Начинает нумерацию списка с какого-либо числа (вместо 1).

`type = "1 | a | A | i | I"`

Определяет систему нумерации, как указано в следующей таблице.

Значение типа	Генерируемый стиль	Пример нумерации
1	Арабские цифры (по умолчанию)	1, 2, 3, 4
A	Буквы верхнего регистра	A, B, C, D
a	Буквы нижнего регистра	a, b, c, d
I	Римские цифры верхнего регистра	I, II, III, IV
i	Римские цифры нижнего регистра	i, ii, iii, iv

Пример

```
<ol>
  <li>Get out of bed</li>
  <li>Take a shower</li>
  <li>Walk the dog</li>
</ol>
```

Элемент optgroup

`<optgroup>...</optgroup>`

Определяет логическую группу элементов `option`, находящуюся внутри элемента управления меню формы `select`. Элемент `optgroup` не может содержать другие элементы `optgroup` (они не могут быть вложенными).

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента `select`.

Разрешенный контент

Элементы option отсутствуют либо несколько элементов option.

Начальный/конечный теги

Обязательный/необязательный (обязательный/обязательный в XHTML).

Атрибуты

Глобальные атрибуты HTML5

`disabled (disabled="disabled" в XHTML)`

Указывает на то, что группа параметров нефункциональна.

Может быть повторно активизирована с помощью сценария.

`label = "текст"`

Обязательный. Указывает подпись для группы параметров.

Пример

```
<p>What are your favorite ice cream flavors?</p>
<select name="ice_cream" size="6" multiple="multiple">
  <optgroup label="traditional">
    <option>Vanilla</option>
    <option>Chocolate</option>
  </optgroup>
  <optgroup label="specialty">
    <option>Inside-out Rocky Road</option>
    <option>Praline Pecan Smashup</option>
  </optgroup>
</select>
```

Элемент option

`<option>...</option>`

Определяет параметр внутри элемента select (меню с некоторыми вариантами выбора или список с прокруткой) либо элемент datalist (предварительно определенный список текстовых параметров для элемента input). В качестве контента элемента option используется значение, которое передается приложению обработки формы (если не указано альтернативное значение с помощью атрибута value).

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элементов select, datalist или optgroup.

Разрешенный контент

Текст.

Начальный/конечный теги

Обязательный/необязательный (XHTML: обязательный/обязательный).

Атрибуты

Глобальные атрибуты HTML5

disabled(disabled="disabled" в XHTML)

Указывает на то, что выбранный вариант изначально нефункционален. Может быть повторно активизирован с помощью сценария.

label = "текст"

Позволяет автору использовать более краткую метку, чем контент параметра.

selected(selected="selected" в XHTML)

Придает элементу статус выбранного при первоначальном отображении формы.

value = "текст"

Определяет значение, которое присваивается элементу option, находящемуся внутри элемента option, для использования вместо контента элемента option.

Пример

```
<p>What are your favorite ice cream flavors?</p>
<select name="ice_cream" size="4" multiple="multiple">
  <option>Vanilla</option>
  <option>Chocolate</option>
  <option>Inside-out Rocky Road</option>
  <option value="pecan">Praline Pecan Smashup</option>
```

```
<option>Mint Chocolate Chip</option>
<option>Pistachio</option>
</select>
```

Элемент output

```
<output>...</output>
```

Не поддерживается в HTML 4.01. Представляет результат вычисления, обычно результат выполнения сценария.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент и “связанный с формой, отображаемый в виде списка, подписываемый и отправляемый контент”.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`for = "текст"`

Создает явную связь между результатом вычисления и именованным элементом или элементами на странице, которые связаны между собой или подвержены влиянию вычисления.

`form = "идентификатор владельца формы"`

Явно связывает элемент управления вводом со связанной формой (соответствующим владельцем формы). При использовании этого метода элемент управления выводом не должен быть потомком применяемого элемента `form`.

`name = "текст"`

Присваивает идентифицирующее имя элементу управления формы, используемому при передаче данных формы.

Пример

```
<form oninput="myCalc.value = parseInt(a.value) +  
parseInt(b.value)">  
<input type="number" id="a"> + <input type="number"  
id="b">  
= <output name="myCalc" for="a b">[отображает  
значение  
результата вычисления a+b]</output>  
</form>
```

Элемент p

`<p>...</p>`

С помощью этого элемента выделяется абзац. Абзацы могут содержать текст и встроенные элементы, но не могут включать другие блочные элементы, в том числе другие абзацы. Браузеры проинструктированы таким образом, чтобы игнорировать множественные пустые элементы p.

Применение

Категории

Потоковый контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/необязательный (XHTML: обязательный/обязательный).

Атрибуты

Глобальные атрибуты HTML5

Пример

<**p**> Paragraphs are the most rudimentary elements of a text document.</**p**>
<**p**>They are indicated by the <code>p</code> element.</**p**>

Элемент **param**

<**param**> (XHTML: <**param**/> или <**param** />)

Поддерживает параметр внутри элемента **object**. В качестве параметра выступает информация, требуемая для объекта **media** во время выполнения.

Применение

Категории

Нет.

Допустимые контексты

В виде потомка элемента **object**, перед любым потоковым контентом.

Разрешенный контент

Пустой.

Начальный/конечный теги

Это пустой (**void**) элемент, поэтому он имеет только начальный тег и не может включать какие-либо контенты. В HTML использование закрывающего тега запрещено. В XHTML элемент должен быть закрыт с помощью обратной косой черты (<**param**/> или <**param** />).

Атрибуты

Глобальные атрибуты HTML5

name = "текст"

Обязательный. Определяет имя параметра.

value = "текст"

Определяет значение параметра.

Пример

(См. также **object**.)

<**param** name="autoStart" value="false">

Элемент `pre`

`<pre>...</pre>`

Создает “предварительно отформатированный” текст: его строки отображаются в том виде, в котором вводятся. При этом используются такие пробельные символы, как множественные пробелы и прерывания строк. По умолчанию текст внутри элемента `pre` отображается с помощью моноширинного шрифта, поэтому сохраняются интервалы и выравнивание.

Применение

Категории

Потоковый контент, явный контент.

Допустимые контексты

В области возможного появления текущего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`xml:space = "preserve"`

Только XHTML. Инструктирует процессоры XML сохранять пробельные символы в элементе.

Пример

```
<pre>
This      is          an example of
           text with a    lot of
                           curious
                           whitespace.
</pre>
```

Элемент `progress`

`<progress>...</progress>`

Не поддерживается в HTML 4.01. Визуально представляет ход выполнения той или иной задачи, например загрузки файла

(индикатор выполнения). Значение, оценивающее ход выполнения задачи, может быть предоставлено сценарием и включено в качестве контента элемента `progress` или генерироваться с помощью атрибута `value`. Элемент `progress` может использоваться даже в том случае, когда максимальное значение времени ожидания неизвестно, например, чтобы отобразить состояние ожидания ответа удаленного хоста при выполнении задачи.

Применение

Категории

Потоковый контент, фразообразующий контент, подписываемый элемент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент, но не может включать элементы `progress`.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`max = "число"`

Отображает оценку общего количества работы, необходимой для выполнения задачи. По умолчанию 1.

`value = "число"`

Указывает, какая часть задачи выполнена.

Пример

```
<p>Percent downloaded: <progress max="100"><span id="completed">0</span>%</progress></p>
```

Элемент `q`

```
<q>...</q>
```

Применяется для выделения короткой цитаты, которая может включаться в строку, например “быть или не быть”. Отличается от

элемента `blockquote`, который используется на блочном уровне для выделения более длинных цитат. В соответствии со спецификацией пользовательский агент должен автоматически вставлять кавычки перед цитатой и после нее. При использовании вместе с атрибутом `lang` браузер может вставлять специфические для языка кавычки.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`cite = "URL-ссылка"`

Обозначает исходный документ, из которого заимствована цитата.

Пример

```
<p>In that famous speech beginning, <q>Four score and  
seven years ago,</q>...</p>
```

Элемент `rp`

```
<rp>...</rp>
```

Не поддерживается в HTML 4.01. Используется внутри элемента `ruby` для отображения круглых скобок, в которые заключен текст Ruby. Элемент `rp` предназначен для пользовательских агентов, не поддерживающих аннотации Ruby.

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента ruby, непосредственно перед элементом rt или после него.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

(См. также ruby.)

В следующем примере браузер, который не может отобразить аннотацию Ruby, должен после идеограммы вывести на экран контент rt, заключенный в круглые скобки.

```
<ruby>
  汉 <rp>(</rp><rt>hàn</rt><rp>) </rp>
  字 <rp>(</rp><rt>zì</rt><rp>) </rp>
</ruby>
```

Пример, демонстрирующий использование элементов rp, rt и ruby, был взят из документа HTML5 Working Draft (на сайте whatwg.com) в соответствии с разрешением в рамках лицензии MIT License.

Элемент rt

```
<rt>...</rt>
```

Не поддерживается в HTML 4.01. Используемый внутри элемента ruby элемент rt поддерживает текст Ruby в аннотациях Ruby. Подсказки обычно отображаются более мелким шрифтом возле исходной идеограммы.

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента `ruby`.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<ruby>
    汉 <rt>hàn</rt>
    字 <rt>zì</rt>
</ruby>
```

Элемент `ruby`

```
<ruby>...</ruby>
```

Не поддерживается в HTML 4.01. Предоставляет подсказки по чтению текста, помеченного аннотациями Ruby, краткие руководства по произношению и другие примечания, используемые преимущественно в восточно-азиатских языках.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент, не может включать элементы `ruby`. Может также содержать один или несколько

элементов `rt`, в том числе с элементами `rp` перед элементом `rt` и после него.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

В следующем примере советы по произношению, выделенные с помощью элементов `rt`, отображаются над идеограммами в визуальных браузерах.

```
<ruby>
  汉 <rt>hàn</rt>
  字 <rt>zì</rt>
</ruby>
```

Элемент `s`

```
<s>...</s>
```

Представляет контент, который не является точным или релевантным. Такой контент обычно отображается как перечеркнутый текст.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<p>All winter gear is <s>20%</s> 40% off.</p>
```

Элемент samp

```
<samp>...</samp>
```

С помощью элемента samp выделяется вывод, формируемый программами, сценариями и т.п.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<p>Provide alternative error messages to <samp>404 Not Found</samp>.</p>
```

Элемент script

```
<script>...</script>
```

Применяется для включения сценария в документ (обычно сценарий JavaScript внедряется в веб-документы). Может

встречаться произвольное количество раз в области `head` или `body` документа. Сценарий может задаваться с помощью элемента `script` или находиться во внешнем файле (в этом случае задается с помощью атрибута `src`).

Примечания

В XHTML, в случае, если сценарий поддерживается как контент элемента `script` (т.е. не в виде внешнего файла), он должен находиться в разделе CDATA, как показано в рассмотренном ниже примере.

Применение

Категории

Контент метаданных, потоковый контент, фразообразующий контент.

Допустимые контексты

В области возможного появления контента метаданных или фразообразующего контента.

Разрешенный контент

Если атрибут `src` не используется, то модель контента зависит от атрибута `type`. При наличии атрибута `src` элемент должен либо быть пустым, либо содержать только документацию сценария.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`async` (`async="async"` в XHTML)

Не поддерживается в HTML 4.01. Указывает на то, что внешний сценарий должен выполняться асинхронно (как только станет доступен).

`charset = "набор символов"`

Задает символьную кодировку документа, в котором находится внешний сценарий (не релевантен по отношению к контенту элемента `script`).

```
crossorigin = "anonymous | use-credentials"
```

Не поддерживается в HTML 4.01. Для сайтов, которые обеспечивают возможность выполнения сценариев из других доменов, этот атрибут определяет, отображается ли информация об ошибках сценариев, полученных из других источников.

```
defer (defer="defer" в XHTML)
```

Указывает пользовательскому агенту на то, что внешний сценарий должен быть выполнен один раз после завершения анализа страницы.

```
src = "URL-ссылка"
```

Указывает местоположение внешнего сценария.

```
type = "тип контента"
```

Обязателен в HTML 4.01. Необязателен в HTML5, если используется JavaScript. Указывает язык написания сценариев, используемый для текущего сценария. Значением value является тип контента, чаще всего text/javascript.

```
xml:space = "preserve"
```

Только XHTML. Предоставляет процессорам XML инструкции по сохранению пробельных символов в элементе.

Примеры

```
<script>  
var foo = 52;  
alert(foo);  
</script>
```

```
<script src="my_script.js"></script>
```

Пример (HTML 4.01)

```
<script type="text/javascript">  
    // Код JavaScript  
</script>
```

Пример (XHTML)

```
<script>
// <![CDATA[
// Код JavaScript
// ]]>
</script>
```

Элемент section

```
<section>...</section>
```

Не поддерживается в HTML 4.01. Представляет раздел (тематическую группу контента) документа или приложения с собственной внутренней структурой и (необязательно) заголовком и нижним колонтитулом. Элемент `section` не является обобщенным контейнером; он должен использоваться только в том случае, когда контенты элемента должны появляться в структуре документа.

Применение

Категории

Потоковый контент, секционирующий контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Потоковый контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<body>
<article>
    <h1>Common Birds</h1>
    <section>
```

```
<h1>Chapter 1: Hummingbirds</h1>
<p>A little something on hummingbirds.</p>
</section>
<section>
  <h1>Chapter 2: Turkeys</h1>
  <p>This is about turkeys</p>
</section>
</article>
</body>
```

Элемент **select**

```
<select>...</select>
```

Определяет меню с несколькими вариантами выбора или список с прокруткой. Элемент **select** является контейнером для одного или нескольких элементов **option** и **optgroup**.

Применение

Категории

Потоковый контент, фразообразующий контент, интерактивный контент, явный контент и “связанный с формой, отображаемый в виде списка, подписываемый и отправляемый контент”.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Нуль или более элементов **option** и **optgroup**.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

autofocus (**autofocus="autofocus"** в XHTML)

Не поддерживается в HTML 4.01. Показывает, что элемент управления должен получать фокус (быть выделенным и готовым для пользовательского ввода), когда загружается документ.

`disabled(disabled="disabled" в XHTML)`

Показывает, что элемент `select` изначально не функционален. Этот элемент можно повторно активизировать с помощью сценария.

`form = "идентификатор владельца формы"`

Не поддерживается в HTML 4.01. Явно ассоциирует элемент управления `select` со связанной формой (с владельцем формы). При использовании этого метода элементу управления не нужно быть потомком элемента `form`, который к нему применяется.

`inputmode = "verbatim | latin | latin-name | latin-prose | full-width-latin | kana | katakana | numeric | tel | email | url"`

Не поддерживается в HTML 4.01. Задает наиболее полезный для пользователей, вводящих контент в элемент управления формы, механизм ввода данных. Применяется к тексту и типам ввода `search`.

`multiple(multiple="multiple" в XHTML)`

Предоставляет пользователю возможность выбора нескольких параметров из списка. Если этот атрибут не указан, то можно выбрать только один параметр.

`name = "текст"`

Обязательный. Определяет имя для элемента управления `select`. Когда форма передает данные приложению обработки формы, это имя отправляется вместе со всеми выбранными значениями `option`.

`required(required="required" в XHTML)`

Не поддерживается в HTML 4.01. В случае указания этого атрибута пользователь должен выбрать параметр до передачи данных формы.

`size = "число"`

Указывает количество строк, которые отображаются в списке параметров (опций). Для значений больше 1 параметры отображаются в виде списка с прокруткой с указанным количеством видимых параметров. Когда указано `size="1"`, список отображается в виде всплывающего меню.

Пример

```
<p>What are your favorite ice cream flavors?</p>
<select name="ice_cream" multiple="multiple">
  <option>Vanilla</option>
  <option>Chocolate</option>
  <option>Mint Chocolate Chip</option>
  <option>Pistachio</option>
</select>
```

Элемент small

```
<small>...</small>
```

Применяется для отображения врезки или заметки, находящейся в стороне от текста основного документа, например замечание о юридической ответственности, отображаемое в нижней части документа.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<p><small>Copyright 2013, O'Reilly Media</small></p>
```

Элемент source

<source> (XHTML: <source/> или <source />)

Не поддерживается в HTML 4.01. Элемент source, используемый внутри элементов audio и video, позволяет авторам указывать несколько версий мультимедийного файла. Если используется элемент source, то атрибут src должен исключаться из элементов audio и video. Пользовательский агент просматривает список элементов source до тех пор, пока не найдет формат, который может воспроизводить.

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента audio или video, перед любым потоковым контентом или элементами track.

Разрешенный контент

Пустой.

Начальный/конечный теги

Это пустой (void) элемент, поэтому включает только начальный тег и не может содержать какие-либо контенты. В HTML использование закрывающего тега не допускается. В XHTML элемент должен закрываться с помощью обратной косой черты (<source/> или <source />).

Атрибуты

Глобальные атрибуты HTML5

media = "all | aural | braille | handheld | print | projection | screen | tty | tv"

Задает целевую среду, применяемую для отображения аудио- или видеофайлов.

src = "URL-ссылка"

Указывает местоположение аудио- или видеофайла.

`type = "тип MIME"`

Указывает тип мультимедийного файла, также может включать MIME-параметр `codecs=`, который задает кодек, применяемый для кодирования мультимедийного файла.

Пример

```
<video>
  <source src="media/vacation.ogv" type="video/ogg;
  codecs='theora, vorbis'">
  <source src="media/vacation.mp4" type="video/mp4">
Ваш браузер не поддерживает элемент <code>video</code>.
</video>
```

Элемент span

```
<span>...</span>
```

Идентифицирует обобщенный встроенный элемент. Элементу `span` обычно присваивается значение с помощью атрибутов `class` и `id`, которые также позволяют ему быть доступным в сценариях и таблицах стилей.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<li>Jenny: <span class="tel">867.5309</span></li>
```

Элемент **strong**

```
<strong>...</strong>
```

С помощью элемента **strong** слово или фраза подчеркивается как важная или требующая дополнительного внимания (обычно отображается полужирным шрифтом).

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<p>All applications are due <strong>no later than February 14</strong>. </p>
```

Элемент **style**

```
<style>...</style>
```

Включает в документ одно или несколько стилевых правил (обычно — каскадные таблицы стилей). Элемент **style** не следует путать с глобальным атрибутом **style**, позволяющим применять стили к отдельному элементу.

Примечания

В HTML5 элемент `style` может быть использован в контенте документа. В HTML 4.01 этот элемент должен находиться в области `head` документа.

Применение

Категории

Контент метаданных, потоковый контент (при наличии атрибута `scoped`).

Допустимые контексты

Там, где ожидаются метаданные или в элементе `noscript`, который является потомком элемента `head`. Если присутствует атрибут `scoped`, то там, где ожидается потоковый контент, но перед любым другим потоковым контентом, отличающимся от пробелов внутри элемента, и не в качестве потомка элемента с прозрачной моделью контента.

Разрешенный контент

Стилевые правила, зависимые от значения атрибута `type`. HTML5 допускает, что язык стилей имеет синтаксис CSS.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`media = "all | aural | braille | handheld | print | projection | screen | tty | tv"`

Указывает целевую среду, предназначенную для стилевой информации. Это может быть единственное ключевое слово или список, разделенный запятыми. По умолчанию в спецификации HTML 4.01 используется значение `screen`. В HTML5 значение по умолчанию — `all`.

`scoped (scoped="scoped" в XHTML)`

Не поддерживается в HTML 4.01. Применяет стили только к локальному дереву документов (т.е. к потомкам предка элемента `style`), а не к целому документу.

`type = "тип контента" (text/css)`

Обязателен только в HTML 4.01. Указывает язык таблицы стилей. Для каскадных таблиц стилей (в настоящее время единственный вариант выбора стиля) выбирается значение `text/css`. В HTML5 атрибут `type` необязателен, если используется CSS.

`xml:space = "preserve"`

Только XHTML. Дает процессорам XML инструкции по сохранению пробельных символов в элементе.

Пример (HTML5 и CSS)

```
<head>
  <style>
    h1 {color: #666; }
    p {line-height: 2; }
  </style>
  <title>Scientific Presentation</title>
</head>
```

Пример (HTML 4.01 и CSS)

```
<head>
  <style type="text/css">
    h1 {color: #666; }
    p {line-height: 2; }
  </style>
  <title>Scientific Presentation</title>
</head>
```

Элемент `sub`

`_{...}`

Форматирует текст в виде нижнего индекса. Элемент `sub` должен быть использован только тогда, когда нижний индекс имеет специфическое значение, но не для создания типографских эффектов.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<p>H<sub>2</sub>0</p>
```

Элемент summary

```
<summary>...</summary>
```

Не поддерживается в HTML 4.01. С помощью элемента `summary` создается резюме, заголовок или легенда для соответствующего предка элемента `details`.

Применение

Категории

Нет.

Допустимые контексты

В качестве первого потомка элемента `details`.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

(См. также `details`.)

```
  
<details>  
  <summary>More about this photo</summary>  
  <p>This photo is of a daylily taken at the....</p>  
</details>
```

Элемент `sup`

```
<sup>...</sup>
```

Представляет текст в виде верхнего индекса. Используйте элемент `sup` только в том случае, когда верхний индекс имеет специфическое значение, а не просто ради типографских эффектов.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<p>E=MC2</sup></p>
```

Элемент `table`

`<table>...</table>`

Формирует таблицу, используемую для отображения строк и столбцов данных либо другой информации. Чтобы создать таблицу, потребуется следующий минимальный набор элементов: `table` — для создания самой таблицы, `tr` — для описания строк таблицы и `td` — для создания ячеек таблицы внутри строки. Завершенная модель таблицы приведена далее.

Применение

Категории

Потоковый контент, явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Разрешенный контент следует в таком порядке: необязательный элемент `caption`, затем элементы `colgroup` (могут отсутствовать), за которыми следует элемент `thead`, за которым необязательно следует элемент `tfoot`, за которым следует несколько элементов `tbody` (могут отсутствовать) или несколько элементов `tr`, за которыми необязательно следует элемент `tfoot` (может быть только единственный итог на основе элемента `tfoot`).

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`border` = "количество пикселей"

Только W3C. Указывает ширину (в пикселях) рамки, обрамляющей таблицу и ее ячейки. Если этому значению присвоить значение 0 (нуль), то рамка не будет отображаться. По умолчанию задается значение 1. Если добавить слово `border` (без значения), то появится однопиксельная рамка, хотя это некорректно в XHTML.

`sortable (sortable="sortable" в XHTML)`

Только WHATWG и HTML5.1. При наличии этого атрибута содержимое таблицы может сортироваться по столбцу. Используется вместе с атрибутом `sorted` элемента `th`, значение которого может присваиваться вручную или пользовательским агентом, когда пользователь щелкает на столбце.

Примеры

Простая таблица с двумя строками (двумя элементами `tr`) и тремя столбцами (тремя элементами `td` в каждом элементе `tr`).

```
<table>
  <tr>
    <td>ячейка 1</td>
    <td>ячейка 2</td>
    <td>ячейка 3</td>
  </tr>
  <tr>
    <td>ячейка 4</td>
    <td>ячейка 5</td>
    <td>ячейка 6</td>
  </tr>
</table>
```

Ниже приводится код, формирующий модель полной таблицы (подробности — в описании элементов `caption`, `tbody`, `thead`, `tfoot`, `colgroup` и `col`).

```
<table>
  <caption>Employee salaries and start dates</caption>
  <colgroup id="employeeinfo">
    <col span="2">
    <col span="1">
  </colgroup>
  <thead>
    <tr>
      <th>Employee</th><th>Salary</th><th>Start date</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td colspan="3">Compiled by Betty D. Boss</td>
```

```
</tr>
</tfoot>
<tbody>
<tr>
<td>Wilma</td><td>50,000</td><td>April 6</td>

</tr>
<tr>...дополнительные ячейки данных...</tr>
<tr>...дополнительные ячейки данных...</tr>
</tbody>
</table>
```

Элемент **tbody**

```
<tbody>...</tbody>
```

Определяет строку или группу строк как тело таблицы. Должен включать по крайней мере один элемент строки (`tr`). Элементы “группы строк” (`tbody`, `thead` и `tfoot`) могут ускорять отображение таблицы и обеспечивать механизм для прокрутки тела таблицы независимо от ее заголовка и нижнего колонтитула или повторять заголовок и нижний колонтитул, когда длинная таблица печатается на нескольких страницах.

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента `table` после произвольных элементов `caption`, `colgroup` и `thead`, но только тогда, когда отсутствуют элементы `tr`, которые являются потомками элемента `table`.

Разрешенный контент

Нуль или более элементов `tr`.

Начальный/конечный теги

HTML: начальный тег необязателен, если первым элементом в таблице является `tr` и перед ним не находится элемент `tbody`, `thead`, `tfoot` или закрывающий тег.

Закрывающий тег необязателен, если сразу за ним следует элемент `tbody` или `tfoot` (отсутствует контент в таблице-предке).

XHTML: обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

См. `table`.

Элемент `td`

`<td>...</td>`

С помощью элемента `td` определяется ячейка данных таблицы. Ячейка таблицы может содержать произвольный контент, включая другую таблицу.

Применение

Категории

Секционирующий корень.

Допустимые контексты

В качестве потомка элемента `tr`.

Разрешенный контент

Потоковый контент.

Начальный/конечный теги

HTML: начальный тег обязателен. Конечный тег необязателен, если за ним следует элемент `<td>` или `<th>` либо конечный тег родительского элемента.

XHTML: обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`colspan = "число"`

Указывает количество столбцов, которые должна занять текущая ячейка. Значение по умолчанию — 1.

`headers = "идентификатор ссылки"`

Выводит ячейки заголовка (по значению `id`), которые включают информацию заголовка для текущей ячейки данных. Этот атрибут позволяет повысить удобство отображения таблиц при использовании невизуальных браузеров.

`rowspan = "число"`

Указывает количество строк, занятых текущей ячейкой. Значение по умолчанию — 1.

Пример

```
<table>
  <tr>
    <td colspan="2">Cell 1</td>
  </tr>
  <tr>
    <td>Cell 3</td><td>Cell 4</td>
  </tr>
</table>
```

Элемент `textarea`

```
<textarea>...</textarea>
```

Определяет многострочный элемент управления вводом текста. Контент элемента `textarea` выводится в поле ввода текста при первом отображении формы и передается на сервер. Среди разработчиков принято ничего не помещать в элемент `textarea` и использовать атрибут `placeholder` либо `title` для создания подсказки, содержащей необходимую информацию.

Применение

Категории

Потоковый контент, фразообразующий контент, интерактивный контент, явный контент и “выводимый в виде списка, подписываемый и передающий данные элемент, связанный с формой”.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Текст.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`autocomplete = "on | off"`

Не поддерживается в HTML 4.01. Позволяет пользовательскому агенту (браузеру) заполнять поле автоматически (`on`) или требует, чтобы пользователь каждый раз вводил информацию (`off`). Если этот атрибут не используется, то элемент управления наследует настройку `autocomplete` для связанного элемента `form`.

`autofocus (autofocus="autofocus" в XHTML)`

Не поддерживается в HTML 4.01. Показывает, что элемент управления должен получать фокус (быть выделенным и готовым для пользовательского ввода), когда загружается документ.

`cols = "число"`

Указывает ожидаемое максимальное количество символов в строке. Визуальные браузеры автоматически выполняют обертывание вводимых пользователем данных таким образом, чтобы каждая строка была меньше значения, заданного с помощью атрибута `cols`. Если атрибуту `wrap` присвоено значение `hard`, то атрибут `cols` обязателен для использования.

`dirname = "имя поля ввода"`

Не поддерживается в HTML 4.01. Указывает имя поля `input`, которое включает инструкции по направлению чтения текста (`ltr` или `rtl`) для области текста.

`disabled (disabled="disabled" в XHTML)`

Деактивирует элемент управления пользовательским вводом. Может изменяться только с помощью сценария. Браузеры могут отображать отключенные элементы управления по-разному (например, выделять серым цветом), что может использоваться для скрытия определенных элементов

управления до тех пор, пока не будет предоставлена нужная информация.

`form = "идентификатор владельца формы"`

Не поддерживается в HTML 4.01. Явно связывает элемент управления `textarea` с относящейся к нему формой (*владельцем формы*). При использовании этого метода не нужно, чтобы элемент `textarea` был потомком элемента `form`, который к нему применяется.

`inputmode = "verbatim | latin | latin-name | latin-prose | full-width-latin | kana | katakana | numeric | tel | email | url"`

Не поддерживается в HTML 4.01. Указывает на то, какой механизм ввода наиболее полезен для пользователей, вводящих контент в элемент управления формой. Применяется к типам ввода `text` и `search`.

`maxlength = "число"`

Не поддерживается в HTML 4.01. Указывает максимальное количество символов, которые может ввести пользователь в элемент `textarea`.

`name = "текст"`

Обязательный. Указывает имя элемента управления. Это имя передается вместе с контентом элемента управления приложению, применяемому для обработки данных формы.

`placeholder = "текст"`

Не поддерживается в HTML 4.01. Поддерживает короткую (одно слово или короткая фраза) подсказку или пример, который поможет пользователю ввести корректные данные. Если нужно более длинное описание, то используйте атрибут `title`.

`readonly(readonly="readonly" в XHTML)`

Указывает на то, что элемент управления формой не может быть изменен.

`required(required="required" в XHTML)`

Не поддерживается в HTML 4.01. При наличии этого атрибута пользователь должен ввести значение в элемент управления перед передачей данных формы.

```
rows = "число"
```

Указывает высоту поля ввода текста (в виде количества строк текста). Если пользователь ввел больше строк, чем отображается, то текстовое поле прокручивается вниз, чтобы принять дополнительные строки.

```
wrap = "hard | soft"
```

Не поддерживается в HTML 4.01. Если этому атрибуту присвоено значение `hard`, то в конце строк по мере их появления в поле текстового ввода будут включаться жесткие разрывы (символы возврата каретки и перевода строки). В случае выбора значения `hard` также нужно присвоить значение атрибуту `cols`, определяющему длину строки. Если же атрибуту присвоено значение `soft` (значение, заданное по умолчанию), то символы разрыва строки в поле ввода текста не сохраняются в возвращаемых данных.

Примеры

```
<p><label for="loveband">Official Contest Entry</label></p>
<textarea name="band" id="loveband" rows="4" cols="45">Tell
us why you love the band.</textarea>

<p><label for="loveband">Official Contest Entry</label></p>
<textarea name="band" id="loveband" rows="4" cols="45"
placeholder="Tell us why you love the band."></textarea>
```

Элемент `tfoot`

```
<tfoot>...</tfoot>
```

Определяет нижний колонтитул таблицы. Является одним из элементов “группы строк”. Элемент `tfoot` должен включаться после элементов `caption`, `colgroup` и `thead`. Может появляться перед элементом `tbody` или после него. В таблице может быть только один элемент `tfoot`.

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента `table` после любых элементов `caption`, `colgroup` и `thead` и перед любыми элементами `tbody` и `tr`, но только при отсутствии других элементов `tfoot` в родительском элементе `table`. В качестве потомка элемента `table` после произвольных элементов `caption`, `colgroup`, `thead`, `tbody` и `tr`, но только при отсутствии других элементов `tfoot` в родительском элементе `table`.

Разрешенный контекст

Нуль или более элементов `tr`.

Начальный/конечный теги

HTML: начальный тег обязателен. Конечный тег необязателен, если за ним следует элемент `tbody` или если в родительском элементе отсутствует контент.

XHTML: обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

См. `table`.

Элемент `th`

`<th>...</th>`

Элемент `th` определяет ячейку заголовка таблицы. Такие ячейки содержат важную информацию и определяют контекст ячеек таблицы в строке или столбце, которым они предшествуют. Благодаря этому облегчается восприятие информации в таблицах. С точки зрения языка разметки эти ячейки функционируют так же, как и ячейки данных таблицы (элемент `td`).

Применение

Категории

Нет, если только элемент `th` не имеет атрибута `sorted` и не является интерфейсом сортировки, иначе интерактивный контент.

Допустимые контексты

В качестве потомка элемента `tr`.

Разрешенный контент

Потоковый контент, но не может включать элементы `header` и `footer`, секционирующий контент или контент заголовка. Если элемент `th` является интерфейсом сортировки, то не должен содержать интерактивный контент.

Начальный/конечный теги

HTML: обязательный/необязательный (если за элементом следует `td` или `th` или если завершается родительский элемент).

XHTML: обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`colspan = "число"`

Указывает количество столбцов, которые должна занимать текущая ячейка. Значение по умолчанию — 1.

`headers = "идентификатор ссылки"`

Перечисляет в списке ячейки заголовка (по значению `id`), которые включают информацию заголовка для текущей ячейки данных. Это нужно для того, чтобы сделать таблицы более доступными невизуальным браузерам.

`rowspan = "число"`

Указывает количество строк, занятых текущей ячейкой. Значение по умолчанию — 1.

`scope = "row | col | rowgroup | colgroup"`

Показывает, к какой группе ячеек применяется заголовок. Например, если выполняется условие `scope=row`, то ячейка заголовка применяет последовательные ячейки в одной и той же строке.

`sorted = "reversed | число больше нуля" или оба в любом порядке`

Только WHATWG и HTML5.1. Обеспечивает средства, позволяющие сортировать столбцы таблицы помостью атрибута `sortable`.

Пример

```
<table>
<tr><th>Planet</th><th>Distance from Earth</th></tr>
<tr><td>Venus</td><td>pretty darn far</td></tr>
<tr><td>Neptune</td><td>ridiculously far</td></tr>
</table>
```

Элемент **thead**

```
<thead>...</thead>
```

Определяет блок строк, которые состоят из меток столбцов (заголовков) для таблицы. Элемент **thead** является одним из элементов “группы строки”. Может применяться для дублирования заголовков в том случае, когда полная таблица прерывается на страницах, или для создания статического заголовка, который отображается над основной частью прокручивающейся таблицы. Должен включать хотя бы одну строку (**tr**) с элементами **th** и **td**.

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента **table** после произвольных элементов **caption** и **colgroup** и перед любыми элементами **tbody**, **tfoot** и **tr**, но только если нет других элементов **thead** в родительском элементе **table**.

Разрешенный контент

Нуль или более элементов **tr**.

Начальный/конечный теги

HTML: необязательный/необязательный (если за элементом следует **tbody** или **tfoot**).
XHTML: обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

См. **table**.

Элемент `time`

`<time>...</time>`

Не поддерживается в HTML 4.01. Представляет время в 24-часовом формате или дату по григорианскому календарю (дополнительно со временем часового пояса). Элемент `time` может быть использован для передачи распознаваемой компьютером информации о времени и дате в другие приложения (например, чтобы сохранить события в личном календаре) или для преобразования информации о времени в другие форматы (например, 18:00 преобразуется в 6 p.m.). Элемент `time` не предназначен для использования в тех случаях, когда не может быть задано точное время или дата, например “конец прошлого года” или “начало века”.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`pubdate (pubdate="pubdate" в XHTML)`

Только W3C (не в WHATWG). Показывает, что дата и время, передаваемые с помощью атрибута `datetime`, являются датой и временем публикации родительского элемента (контента элемента `article` или целого документа).

`datetime = "ГГГГ-ММ-ДДЧЧ:мм:ссTZD"`

Идентифицирует указанную дату или время. Если используется атрибут `datetime`, то элемент `time` может быть пустым.

Примеры

The deadline for entries is <**time** **datetime**="2013-02-15T20:00-05:00">February 15, 2013, 8pm EST</**time**>

Hours: <**timetime**> to <**time**>9pm</**time**>

Элемент **title**

<**title**>...</**title**>

Обязательный. Определяет заголовок документа. Все документы должны включать содержательный элемент **title**, находящийся в элементе **head**. Заголовки должны содержать только символы ASCII (буквы, цифры и стандартную пунктуацию). Специальные символы (такие, как &) должны задаваться с помощью ссылок на соответствующие символьные сущности внутри заголовка.

Заголовки должны быть четкими и описательными. Заголовок обычно отображается на верхней панели браузера, вне обычного окна контента, а также в списке закладок или в списке избранного. Заголовки документов также используются поисковыми системами.

Применение

Категории

Контент метаданных.

Допустимые контексты

В составе элемента **head**. Допускается не более одного элемента **title**, и он не должен включать другие элементы.

Разрешенный контент

Текст.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<head>
  <title>White Rabbits: Milk Famous</title>
</head>
```

Элемент **tr**

```
<tr>...</tr>
```

Определяет строку ячеек внутри таблицы. Элемент **tr** может содержать лишь несколько элементов **td** и/или **th**. Может использоваться только внутри элемента **table**, **thead**, **tfoot** или **tbody**.

Применение

Категории

Нет.

Допустимые контексты

В качестве элемента **thead**, **tbody** или **tfoot**. В качестве потомка элемента **table**, после произвольных элементов **caption**, **colgroup** и **thead**, но только если отсутствуют элементы **tbody**.

Разрешенный контент

Нуль или более элементов **tr** и **th**.

Начальный/конечный теги

HTML: обязательный/необязательный (если за элементом следует другой элемент **tr** или закрывается родительский элемент).

XHTML: обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<table>
  <tr>
    <td>ячейка 1</td><td>ячейка 2</td>
  </tr>
```

```
<tr>
  <td>ячейка 3</td><td>ячейка 4</td>
</tr>
</table>
```

Элемент track

<track> (XHTML:<track/> или <track />)

Не поддерживается в HTML 4.01. С помощью элемента track задается внешний ресурс (текст или звуковой файл), синхронизированный с воспроизведением аудио- или видеофайла, что улучшает доступность или упрощает навигацию.

Применение

Категории

Нет.

Допустимые контексты

В качестве потомка элемента audio или video, перед произвольным потоковым контентом.

Разрешенный контент

Пустой.

Начальный/конечный теги

Это пустой (void) элемент, поэтому он может включать только открывающий тег и не включает какие-либо контенты. В HTML применение закрывающего тега запрещено. В XHTML элемент должен быть закрыт обратной косой чертой (<track/> или <track />).

Атрибуты

Глобальные атрибуты HTML5

default (default="default" в XHTML)

Показывает, что внешний ресурс должен быть использован по умолчанию, если это не противоречит предпочтениям пользователя.

kind = "subtitles | captions | descriptions | chapters | metadata"

Указывает назначение внешнего ресурса. Значение subtitles обеспечивает транскрипцию или перевод диалога,

отображаемого в окне воспроизведения видео. Значение `captions` обеспечивает полное описание аудиофайла (включая звуковые эффекты и другие звуки) для тех случаев, когда звук недоступен, или для людей со слабым слухом. Значение `descriptions` указывает на то, что ресурс является аудиодорожкой, описывающей видеосюжеты, которые в силу каких-то причин не могут быть просмотрены. Значение `chapters` включает заголовки глав, используемых для навигации по мультимедийному ресурсу. Значение `metadata` представляет дорожку, предназначенную для использования сценариями (не отображается). По умолчанию выбирается значение `subtitles`.

`label = "текстовая строка"`

Указывает заголовок для дорожки, который может быть отображен браузером.

`src = "URL-ссылка"`

Обязательный. Указывает местоположение данных, относящихся к текстовой дорожке.

`srclang = "двухбуквенный код языка"`

Обязателен для субтитров. Указывает язык данных текстовой дорожки.

Пример

```
<video src="blockbuster.webm">
  <track kind="subtitles" src="blockbuster.en.vtt"
    srclang="en" label="English">
  <track kind="subtitles" src="blockbuster.de.vtt"
    srclang="de" label="Deutsch">
</video>
```

Элемент и

`<u>...</u>`

С помощью элемента `u` путем подчеркивания выполняется семантическое выделение текста, например выделяются ключевые слова или слова с орфографическими ошибками (в процессе проверки орфографии). Учтите, что подчеркнутый текст легко

спутать с гиперссылкой, поэтому использовать этот элемент не рекомендуется (за исключением некоторых случаев). Для выделения слов или фраз, требующих особого внимания, применяйте другие элементы разметки или свойства CSS.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<p><u>Подчеркнутый</u> текст легко спутать с  
гиперссылкой.</p>
```

Элемент ul

```
<ul>...</ul>
```

С помощью элемента `ul` задается неупорядоченный список, в котором элементы списка (`li`) не упорядочиваются. По умолчанию визуальные браузеры отображают элементы в неупорядоченном списке с маркерами в виде кружочков. Списки можно форматировать произвольно (даже включать горизонтальные элементы навигации), используя каскадные таблицы стилей.

Применение

Категории

Потоковый контент; если имеется как минимум один элемент `li`, то явный контент.

Допустимые контексты

В области возможного появления потокового контента.

Разрешенный контент

Нуль или более элементов `li`.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<ul>
  <li>About</li>
  <li>Portfolio</li>
  <li>Blog</li>
  <li>Contact</li>
</ul>
```

Элемент `var`

```
<var>...</var>
```

С помощью элемента `var` указывается на наличие переменной или аргумента программы.

Применение

Категории

Потоковый контент, фразообразующий контент, явный контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Фразообразующий контент.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<code><var>myString</var> = 'hello world'; </code>
```

Элемент video

```
<video>...</video>
```

Не поддерживается в HTML 4.01. С помощью элемента `video` осуществляется внедрение видеофайла в веб-страницу. Адрес видеоресурса указывается с помощью атрибута `src`. С помощью одного или нескольких элементов `source`, находящихся внутри элемента `video`, можно задать несколько параметров видеоформата. Также поддерживается альтернативный контент для элемента `video`, отображаемый пользовательскими агентами, которые не поддерживают элемент `video`.

Примечания

До сих пор обсуждаются поддерживаемые видеоформаты для элемента `video`. В настоящее время далеко не все форматы видеофайлов поддерживаются всеми браузерами. На момент написания книги следующие браузеры поддерживают перечисленные ниже форматы видеофайлов.

- **IE 9+** (версии до 9 не поддерживают элемент `video`): MP4/H.264 (тип MIME нужно настроить на сервере, чтобы корректно воспроизводить видео) и WebM (путем установки компонентов проекта The WebM; www.webmproject.org).
- **Chrome 5+:** MP4 (H.264 + AAC), Ogg Theora/Vorbis и WebM (6+).
- **Firefox 3.5+:** Ogg Theora/Vorbis и WebM (4+).
- **Safari 4+ и Mobile Safari 3+:** MP4 (H.264 + AAC).
- **Android (2.1+):** MP4 (H.264 + AAC) и WebM (2.3.3+).

Применение

Категории

Потоковый контент, фразообразующий контент, внедренный контент, интерактивный контент (если элемент имеет атрибут `controls`), явный контент.

Допустимые контексты

В области возможного появления внедренного контента.

Разрешенный контент

Если элемент имеет атрибут `src`, то нуль или более элементов `track`, прозрачный контент, но без потомков элемента `audio` или `video`. Если элемент не имеет атрибута `src`, то нуль или более элементов `source`, нуль или более элементов `track`, прозрачный контент, но без потомков `audio` или `video`.

Начальный/конечный теги

Обязательный/обязательный.

Атрибуты

Глобальные атрибуты HTML5

`autoplay (autoplay="autoplay" в XHTML)`

Автоматическое воспроизведение мультимедийного файла.

`controls (controls="controls" в XHTML)`

Указывает на то, что пользовательский агент (браузер) должен отображать набор элементов управления воспроизведением мультимедийного файла, обычно кнопку “Play/Pause” (воспроизведение/пауза), ползунок, который позволяет перемещаться к выбранной позиции видеофайла, и элементы управления громкостью.

`crossorigin = "anonymous | use-credentials"`

Указывает на то, должен ли пользовательский агент проверять полномочия мультимедийного файла, поступающего из источника (URL-ссылки), отличающегося от источника документа. Значение по умолчанию — `anonymous` (полномочия не требуются).

`height = "число"`

Задает высоту окна видеоплеера в пикселях.

`loop (loop="loop" в XHTML)`

Указывает на то, что мультимедийный файл должен воспроизводиться с начала после завершения предыдущего цикла воспроизведения.

`mediagroup = "текст"`

Устанавливает ссылку на несколько мультимедийных элементов путем присваивания им одного и того же значения `mediagroup`.

`muted(muted="muted" в XHTML)`

Приглушает звучание, даже если при этом переопределяются настройки пользователя.

`poster = "URL-ссылка"`

Задает местоположение файла изображения, которое выводится как заполнитель перед началом воспроизведения видео.

`preload = "none | metadata | auto"`

Подсказывает браузеру, должен ли мультимедийный файл загружаться автоматически. Значение `none` препятствует предварительной загрузке. Значение `metadata` не приводит к загрузке мультимедийного файла, но приводит к выборке метаданных ресурса. Значение `auto` приводит к тому, что решение о предварительной загрузке остается за пользовательским агентом.

`src = "URL-ссылка"`

Указывает на местоположение мультимедийного файла.

`width = "число"`

Задает ширину окна видеоплеера в пикселях.

Примеры

```
<video src="movies/nantucket.ogv" width="640"
height="480"
poster="bay.jpg" type="video/ogg; codecs='theora,
vorbis'">
    Ваш браузер не поддерживает элемент <code>video</
code>.
</video>
```

```
<video id="yourmovieid" width="640" height="360"
poster="yourmovie_still.jpg" controls preload="auto">
    <source src="yourmovie-baseline.mp4" type='video/mp4;
codecs="avc1.42E01E, mp4a.40.2"'>
    <source src="yourmovie.ogv" type='video/ogg; codecs=
"theora, vorbis"'>
    <!--Flash fallback -->
    <object width="640" height="360" type="application/
x-shockwave-flash" data="your_flash_player.swf">
        <param name="movie" value="your_flash_player.swf">
```

```
<param name="flashvars" value="controlbar=over&amp;
image=poster.jpg&amp; file=yourmovie-main.mp4">

</object>
</video>
<p>Download the Highlights Reel:</p>
<ul>
  <li><a href="yourmovie.mp4">MPEG-4 format</a></li>
  <li><a href="yourmovie.ogv">Ogg Theora format</a></li>
</ul>
```

Элемент `wbr`

`<wbr>` (XHTML: `<wbr/>` или `<wbr />`)

Не поддерживается в HTML 4.01. Позволяет задать точку разрыва в длинном слове. В случае необходимости строка будет разорвана только в том месте, где находится элемент `wbr`.

Применение

Категории

Потоковый контент, фразообразующий контент.

Допустимые контексты

В области возможного появления фразообразующего контента.

Разрешенный контент

Пустой.

Начальный/конечный теги

Это пустой (`void`) элемент, поэтому используется только открывающий тег, а также отсутствуют какие-либо контенты. В HTML использование закрывающего тега запрещено. В XHTML элемент должен закрываться с помощью обратной косой черты (`<wbr/>` или `<wbr />`).

Атрибуты

Глобальные атрибуты HTML5

Пример

```
<p>The biggest word you've ever heard & this is how it goes:  
<em>supercalifragilisticexpialidocious</em>!</p>
```

Элементы, упорядоченные по функциям

В этом разделе рассматриваются элементы HTML5, упорядоченные по группам в соответствии с выполняемыми ими функциями. Элементы, которые появились в HTML5 и не являются частью спецификации HTML 4.01, отмечены как (5).

Корневой элемент

html

Элементы метаданных

base, head, link, meta, title, style

Текст: разделы

address, article (5), aside (5), body, footer (5), header (5), hr, main (5), nav (5), section (5)

Текст: заголовки

h1, h2, h3, h4, h5, h6, hgroup (5, удален из HTML5.1)

Текст: элементы группировки

blockquote, div, figcaption (5), figure (5), hr, main (5), p, pre

Текст: списки

dd, dl, dt, li, ol, ul

Элементы уровня текста (встроенные)

abbr, b, bdi (5), bdo, br, cite, code, del, dfn, em, i, ins, kbd, mark (5), q, s, samp, small (5), span, strong, sub, sup, time (5), u, var, wbr

Таблицы

caption, col, colgroup, table, tbody, td, tfoot, th, thead, tr

Элементы формы

button, datalist (5), fieldset, form, input, keygen (5), label, legend, meter (5), optgroup, option, output (5), progress (5), select, textarea

Интерактивный контент

a, details (5), dialog (5), menu (5), menuitem (5), summary (5)

Внедренный контент

area, audio (5), canvas (5), embed (5), iframe, img, map, object, param, source (5), track (5), video (5)

Написание сценариев

script, noscript

Аннотация Ruby

rp (5), rt (5), ruby (5)

Символьные сущности

Символы, отсутствующие в обычных алфавитно-цифровых наборах, такие как < и & ', могут добавляться в документы HTML и XHTML с помощью символьных сущностей. При этом используется процесс, называемый *экранированием* символа. Экранированные символы представлены символьными ссылками, которые начинаются с символа & и завершаются символом ;. Чтобы ссылаться на определенный символ, нужно использовать числовую ссылку на символ (NCR — Numeric Character Reference) либо предопределеннное название символьной сущности.

Числовая ссылка на символ представляет собой код Unicode, который имеет десятичный или шестнадцатеричный формат. Шестнадцатеричные значения указываются с помощью символа "x": &#xhhhh;. Значения в десятичном формате записываются в виде &#nnnn; (символ "x" не используется). Например, длинному тире (—) соответствует код Unicode U+02014, который в HTML-документах может идентифицироваться в виде значения — (шестнадцатеричное) или — (десятичное).

Символьные сущности, или *именованные символьные ссылки* (NCR), представляют собой сокращенные названия символов. Например, сущность < означает символ “меньше или равно” (less-than). Символьные сущности предопределены в языках разметки, таких как HTML и XHTML. Это сделано для удобства разработчиков, поскольку такие сущности легче запомнить, чем числовые ссылки. В HTML 4.01 определены 252 символьные сущности. В HTML5 это число выросло до более чем 2000.

В приложении будут приведены лишь наиболее часто используемые символьные ссылки и сущности. Чтобы получить сведения о дополнительных символьных сущностях, обратитесь к следующим ресурсам.

W3C Character Entity Reference Chart (<http://dev.w3.org/html5/html-author/charref>)

Эта визуальная диаграмма отображает 488 символов, отсортированных по числовым кодам Unicode. Используя эту диаграмму, можно легко сканировать очень большие

символы. Также здесь приведены числовые и именованные символьные ссылки.

HTML5 Named Character Reference List (<http://www.w3.org/TR/2011/WD-html5-20110113/named-character-references.html>)

В этом исчерпывающем списке представлены более 2000 ссылок, рассортированных в алфавитном порядке по именам сущностей, и только коды Unicode.

W3Schools.com HTMLCharsets (<http://www.w3schools.com/charsets/default.asp>)

Полный список номеров и имен сущностей, поддерживаемых в HTML 4.01, включая полный список ASCII-символов, полный список символов ISO-8859-1, математические символы, греческие буквы и другие символы.

Именованные XML-сущности

В XHTML и других языках XML символы кавычек, амперсанда, апострофа, “меньше чем” и “больше чем” должны всегда экранироваться в контенте и в области метаданных документа.

Десятичная запись	Сущность	Символ	Описание
"	"	"	Кавычка
'	'	'	Апостроф
&	&	&	Амперсанд
<	<	<	Меньше чем
>	>	>	Больше чем

Набор символов Latin-1 (ISO-8859-1)

Символ	Десятичная	Сущность	Символ	Описание
Unicode	запись			
U+000A0	 	 		Неразрывный пробел
U+000A1	¡	!	!	Перевернутый восклицательный знак
U+000A2	¢	¢	¢	Символ цента
U+000A3	£	£	£	Символ фунта

Символ Unicode	Десятичная запись	Сущность	Символ	Описание
U+000A4	¤	¤	¤	Обобщенный символ валюты
U+000A5	¥	¥	¥	Символ йены
U+000A6	¦	¦	¦	Разорванная вертикальная черта
U+000A7	§	§	₪	Знак параграфа
U+000A8	¨	¨	„	Умляут
U+000A9	©	©	©	Знак охраны авторского права
U+000AA	ª	ª	ª	Женский порядковый индикатор
U+000AB	«	«	«	Открывающая угловая кавычка
U+000AC	¬	¬	¬	Не равно
U+000AD	­	­	—	Мягкий перенос
U+000AE	®	®	®	Зарегистрированный товарный знак
U+000AF	¯	¯	—	Знак долготы звука
U+000B0	°	°	°	Знак градуса
U+000B1	±	±	±	Плюс-минус
U+000B2	²	²	²	Верхний индекс 2
U+000B3	³	³	³	Верхний индекс 3
U+000B4	´	´	‘	Акут (не буква)
U+000B5	µ	µ	µ	Микрон (греческая буква "мю")
U+000B6	¶	¶	¶	Знак абзаца
U+000B7	·	·	·	Средняя точка
U+000B8	¸	¸	¸	Седиль
U+000B9	¹	¹	¹	Верхний индекс 1
U+000BA	º	º	º	Мужской порядковый индикатор
U+000BB	»	»	»	Закрывающая угловая кавычка
U+000BC	¼	¼	¼	Одна четвертая
U+000BD	½	½	½	Одна вторая
U+000BE	¾	¾	¾	Три четверти
U+000BF	¿	¿	¿	Перевернутый вопросительный знак
U+000C0	À	À	À	Прописная "A" с грависом
U+000C1	Á	Á	Á	Прописная "A" с акутом
U+000C2	Â	Â	Â	Прописная "A" с циркумфлексом
U+000C3	Ã	Ã	Ã	Прописная "A" с тильдой
U+000C4	Ä	Ä	Ä	Прописная "A" с умляутом
U+000C5	Å	Å	Å	Прописная "A" с кольцом

Символ Unicode	Десятичная запись	Сущность	Символ	Описание
U+000C6	Æ	Æ	Æ	Прописная лигатура "AE"
U+000C7	Ç	Ç	Ҫ	Прописная "C" с седилем
U+000C8	È	È	È	Прописная "E" с грависом
U+000C9	É	É	É	Прописная "E" с акутом
U+000CA	Ê	Ê	Ê	Прописная "E" с циркумфлексом
U+000CB	Ë	Ë	Ë	Прописная "E" с умляутом
U+000CC	Ì	Ì	Ì	Прописная "I" с грависом
U+000CD	Í	Í	Í	Прописная "I" с акутом
U+000CE	Î	Î	Î	Прописная "I" с циркумфлексом
U+000CF	Ï	Ï	Ï	Прописная "I" с умляутом
U+000D0	Ð	Ð	Ð	Прописная "eth", исландский алфавит
U+000D1	Ñ	Ñ	Ñ	Прописная "N" с тильдой
U+000D2	Ò	Ò	Ӱ	Прописная "O" с грависом
U+000D3	Ó	Ó	Ӳ	Прописная "O" с акутом
U+000D4	Ô	Ô	Ӵ	Прописная "O" с циркумфлексом
U+000D5	Õ	Õ	Ӷ	Прописная "O" с тильдой
U+000D6	Ö	Ö	Ӹ	Прописная "O" с умляутом
U+000D7	×	×	×	Знак умножения
U+000D8	Ø	Ø	Ӯ	Прописная "O", перечеркнутая
U+000D9	Ù	Ù	ӻ	Прописная "U" с грависом
U+000DA	Ú	Ú	ӻ	Прописная "U" с акутом
U+000DB	Û	Û	ӻ	Прописная "U" с циркумфлексом
U+000DC	Ü	Ü	ӻ	Прописная "U" с умляутом
U+000DD	Ý	Ý	Ӻ	Прописная "Y" с акутом
U+000DE	Þ	Þ	ӷ	Прописная "Thorn", исландский алфавит
U+000DF	ß	ß	ӷ	Строчная лигатура "sz", немецкий алфавит
U+000E0	à	à	à	Строчная "a" с грависом
U+000E1	á	á	á	Строчная "a" с акутом
U+000E2	â	â	â	Строчная "a" с циркумфлексом
U+000E3	ã	ã	ã	Строчная "a" с тильдой
U+000E4	ä	ä	ä	Строчная "a" с умляутом
U+000E5	å	å	å	Строчная "a" с кольцом
U+000E6	æ	æ	æ	Строчная лигатура "ae"

Символ Unicode	Десятичная запись	Сущность	Символ	Описание
U+000E7	ç	ç	ç	Строчная “с” с седилем
U+000E8	è	è	è	Строчная “е” с грависом
U+000E9	é	é	é	Строчная “е” с акутом
U+000EA	ê	ê	ê	Строчная “е” с циркумфлексом
U+000EB	ë	ë	ë	Строчная “е” с умляутом
U+000EC	ì	ì	í	Строчная “и” с грависом
U+000ED	í	í	í	Строчная “и” с акутом
U+000EE	î	î	í	Строчная “и” с циркумфлексом
U+000EF	ï	ï	ï	Строчная “и” с умляутом
U+000F0	ð	ð	ð	Строчная “eth”, исландский алфавит
U+000F1	ñ	ñ	ñ	Строчная “п” с тильдой
U+000F2	ò	ò	ò	Строчная “о” с грависом
U+000F3	ó	ó	ó	Строчная “о” с акутом
U+000F4	ô	ô	ô	Строчная “о” с циркумфлексом
U+000F5	õ	õ	õ	Строчная “о” с тильдой
U+000F6	ö	ö	ö	Строчная “о” с умляутом
U+000F7	÷	÷	÷	Знак деления
U+000F8	ø	ø	ø	Строчная “о” перечеркнутая
U+000F9	ù	ù	ù	Строчная “и” с грависом
U+000FA	ú	ú	ú	Строчная “и” с акутом
U+000FB	û	û	û	Строчная “и” с циркумфлексом
U+000FC	ü	ü	ü	Строчная “и” с умляутом
U+000FD	ý	ý	ý	Строчная “у” с акутом
U+000FE	þ	þ	þ	Строчная “thorn”, исландский алфавит
U+000FF	ÿ	ÿ	ÿ	Строчная “у” с умляутом

Буквы — модификаторы интервалов

Код Unicode	Десятичное значение	Сущность	Символ	Описание
U+002C6	ˆ	ˆ	^	Циркумфлекс
U+002DC	˜	˜	~	Тильда

Общая пунктуация

Код Unicode	Десятичное значение	Сущность	Символ	Описание
U+02002	 	 		Обычный пробел
U+02003	 	 		Широкий пробел
U+02009	 	 		Узкий пробел
U+0200C	‌	‌	Непечатаемый	Запрет на создание лигатур
U+0200D	‍	‍	Непечатаемый	Разрешение на создание лигатур
U+0200E	‎	‎	Непечатаемый	Направление письма — слева направо
U+0200F	‏	‏	Непечатаемый	Направление письма — справа налево
U+02013	–	–	—	Короткое тире
U+02014	—	—	—	Длинное тире
U+02018	‘	‘	‘	Одинарная открывающая кавычка
U+02019	’	’	’	Одинарная закрывающая кавычка
U+0201A	‚	‚	,	Одинарная нижняя кавычка
U+0201C	“	“	“	Двойная открывающая кавычка
U+0201D	”	”	”	Двойная закрывающая кавычка
U+0201E	„	„	„	Двойная нижняя кавычка
U+02020	†	†	†	Типографский крестик
U+02021	‡	‡	‡	Двойной типографский крестик
U+02022	•	•	•	Буллит (маркер списка)
U+02026	…	&hellep;	...	Многоточие
U+02030	‰	‰	‰	Символ промилле (тысячная доля единицы)
U+02032	′	′	‘	Штрих, минуты, футы
U+02033	″	″	”	Двойной штрих, секунды, дюймы

Код Unicode	Десятичное значение	Сущность	Символ	Описание
U+02039	‹	‹	<	Одинарная угловая открывающая скобка
U+0203A	›	›	>	Одинарная угловая закрывающая скобка
U+0203E	‾	‾	-	Верхняя черта
U+02044	⁄	⁄	/	Символ дроби
U+020AC	€	€	€	Символ евро

ПРИМЕЧАНИЕ

Математические символы, используемые для обозначения эквивалентности (\equiv представлен символами ≡ либо ≡) и операции присваивания ($:=$ представлен символами ≔ либо ≔), обычно используются в качестве символов навигации при создании сайтов для мобильных устройств, поскольку они похожи на пиктограммы списка параметров, отображаемые на таких устройствах.

Обзор синтаксиса XHTML

XHTML (eXtensible HyperText Markup Language — расширяемый язык гипертекстовой разметки) представляет собой результат реструктурирования языка разметки HTML в соответствии с более строгими синтаксическими правилами XML (eXtensible Markup Language — расширяемый язык разметки). Код HTML5 можно записывать с применением синтаксиса XHTML (формально называемого “XML Serialization of HTML5” — XML-сериализация HTML5). В этом разделе описаны отличия синтаксиса XHTML от синтаксиса HTML.

Поскольку XHTML представляет собой разновидность XML, его синтаксис более строгий и отличается от HTML в следующем.

- Все названия элементов и атрибутов должны записываться с использованием нижнего регистра символов, например `...`.
- Все элементы должны завершаться, т.е. включать закрывающий тег, например `<p>...</p>`.
- Пустые элементы также должны закрываться. Для выполнения этой операции в конце тега ставится косая черта. Перед косой чертой обычно добавляется пробел, чтобы обеспечить обратную совместимость с устаревшими браузерами, например `<hr />`, `` и `<meta />`.
- Все значения атрибутов должны заключаться в кавычки (одинарные или двойные), например `<td colspan="2">`.
- Все значения атрибутов должны быть явными, нельзя их минимизировать до одного слова, как разрешено в HTML, например `checked="checked"`.
- Более строго соблюдаются ограничения вложения. Эти ограничения четко сформулированы:
 - элемент не может включать другой элемент;
 - элемент `pre` не может включать элементы `img`, `object`, `applet`, `big`, `small`, `sub`, `sup`, `font` и `basefont`.
 - элемент `form` не может содержать другие элементы `form`;

- элемент `button` не может включать элементы `a`, `form`, `input`, `select`, `textarea`, `label`, `button`, `iframe` и `isindex`.
- элемент `label` не может включать другие элементы `label`.
- Специальные символы `<`, `>`, `&`, `'` и `"` должны всегда представляться символьными сущностями, включая тот момент, когда они появляются в значениях атрибутов, например `<`, `>`, `&`, `'` и `"` соответственно.
- XHTML-документы должны использовать значение `id` вместо `name` для идентификации фрагментов документа и в элементах `a`, `applet`, `form`, `frame`, `iframe`, `img` и `map`.
- XHTML-документы должны выступать в качестве XML-приложений, а не текстовых документов HTML. В частности, сервер должен конфигурироваться таким образом, чтобы в XHTML-документах присваивать заголовку `Content-type` значение `application/xhtml+xml`. Если сервер сконфигурировать невозможно, то тип контента можно указать в элементе `meta` внутри элемента `head`, как показано в следующем примере кода.

```
<meta http-equiv="content-type"
      content="application/xhtml+xml; charset=UTF-8"
      />
```

К сожалению, некоторые популярные браузеры (например, Internet Explorer) не могут выполнять синтаксический анализ XHTML-документов в виде XML, что приводит к проблемам. Для получения дополнительных сведений по типам MIME в XHTML обратитесь на сайт www.w3.org/TR/xhtml-media-types/.

Предметный указатель

A

API
 Canvas 15
 Drag and Drop 15
 Editing 15
 File 16
 Geolocation 16
 Media 15
 Offline Web Applications 15
 Session History 15
 TextTrack 15
 Web Sockets 16
 Web Storage 15
 Web Workers 16

D

DNS 21
DOCTYPE 16
DOM 12
DTD 12

H

HTML5.1 13
HTTP 73; 121
 метод формы 88

N

NCR 179

R

Ruby 137

U

Unicode 179
URI 37
URL 36; 105
UTC 58

W

W3C 11; 12
WHATWG 11; 12

X

XHTML 187
XML 17; 187
 сущность 180

Абзац 133
Автозаполнение 55
Адрес электронной почты 95
Акроним 27
Аннотация Ruby 139
Аудиофайл 33

Б

Базовое имя 36

В

Видео 173
Виджет замыкания 58
Внедренный объект 65
Время 94; 104; 166
 всемирное координирован-
 ное 58
Всплывающий календарь 97
Встроенный фрейм 81

Г

Гиперссылка 112
Глобальный атрибут 19
Группа столбцов 50

Д

Дата 93
Двунаправленная изоляция 37
Двунаправленное
 переопределение 39
Диалоговое окно 60
Документ 16

З

Заголовок 74; 76
 документа 167

Звуковой файл 33
Знак пунктуации 184

И

Изображение 83
Индекс
 верхний 154
 нижний 152
Индикатор выполнения 136

K

Карта изображений
 клиентская 29; 85; 115
 серверная 84
Ключ 108
Кнопка 42
 замена изображением 97
 настраиваемая 92
 передачи данных 102
 сброса данных 101
Команда 52
Комментарий 54
Контактные данные 28
Контроль ввода данных 14
Кросс-доменный доступ к
 файлам 84
Курсив 65; 80

M

Меню 117
 всплывающее 118
 с несколькими вариантами вы-
 бора 145
Мера 122
Метаданные 120

N

Набор символов 180

O

Область щелчков 29
Обработчик событий 23
Обратный DNS 21
Объектная модель документа 12
Определение типа документа 12

P

Пароль 99
Переключатель 99
Подчеркивание 170
Поле
 поиска 101
 со списком 55
Ползунок 100
Полизаполнение JavaScript 17
Полужирный текст 36; 150
Предварительное
 форматирование 135
Проверка достоверности 19

R

Раздел
 документа 144
 навигационный 123
Разрыв строки 42
Регулярное выражение 90

C

Секционирующий корень 40
Символьная сущность 179
Список
 неупорядоченный 171
 описаний 56; 62
 связей 62
 упорядоченный 128
Статья 31
Стилевое правило 150
Столбец 49
 группа 50
Сценарий 142
Счетчик 98

T

Таблица 155
 заголовок 163
 колонтитул 162
 строка 168
 тело 157
 ячейка 158
Текст 103
 вставка 106

выделенный 116
поле ввода 159
Телефонный номер 102
Тело документа 40

Ф

Файл 96
Флажок 92
Форма 72
 владелец 43
 кодирование 44
 метод HTTP 88
Фрейм 81
Футер 71

Х

Холст 45

Ц

Цвет 93
Цитата
 длинная 39
 короткая 136

Ч

Часовой пояс 58
Числовая ссылка на символ 179

Э

Экранирование символа 179

Элемент
 a 25
 abbr 27
 address 28
 area 29
 article 31
 aside 32
 audio 33
 b 36
 base 36
 bdi 37
 bdo 39
 blockquote 39; 137
 body 41
 button 42

canvas 45
caption 47
cite 48
code 48
colgroup 50
col 49
command 52
datalist 55
data 54
dd 56
del 57
details 58
dfn 60
dialog 60
div 61
dt 64
embed 65
em 65
fieldset 67
figure 70
form 72
h1-h6 74
header 76
head 75
hgroup 77
hr 78
i 80
iframe 81
img 83
input 85
ins 106
kbd 107
keygen 108
label 109
legend 111
link 112
li 111
main 114
map 115
mark 116
menuitem 118
menu 117
meter 122
nav 123
noscript 124

object 126
ol 128
optgroup 129
option 130
output 132
p 133
param 134
pre 135
progress 136
q 136
rp 137
rt 138
ruby 139
s 140
samp 141
script 141
section 144
select 145
small 147
source 148
span 149
strong 150
style 150
sub 152
summary 153

sup 154
table 155
tbody 157
td 158
textarea 159
tfoot 162
thead 165
th 163
time 166
title 167
track 169
tr 168
u 170
ul 171
var 172
video 173
wbr 176

корневой 79
устаревший 14
Элемент управления 85
скрытый 96

Я

Якорь 25