

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

Элемент thead	165
Элемент time	166
Элемент title	167
Элемент tr	168
Элемент track	169
Элемент u	170
Элемент ul	171
Элемент var	172
Элемент video	173
Элемент wbr	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

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 *структуры* 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/13xE0Br).
- **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 *полизаполнение* JavaScript (заплата), имитирующее поддержку HTML5. Полный набор существующих на данный момент полизаполнений можно найти в коллекции Modernizr, доступной по следующему адресу:

www.github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills

В качестве простого примера полизаполнения 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 Девериа.

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 видеодиаграмма, 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, а `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.

onabort	ondragleave	onload*	onratechange
onblur*	ondragover	onloadeddata	onreset
oncancel	ondragstart	onloadedmeta- data	onscroll
oncanplay	ondrop	onseeked	onseeked
oncanplay- through	onduration- change	onmousedown	onseeking
onchange	onemptied	onmousemove	onselect
onclick	onended	onmouseout	onshow
onclose**	onerror*	onmouseover	onsort**
oncontextmenu	onfocus*	onmouseup	onstalled
oncuechange	oninput	onmousewheel	onsubmit
ondblclick	oninvalid	onpause	onsuspend
ondrag	onkeydown	onplay	ontimeupdate
ondragend	onkeypress	onplaying	onvolume- change
ondragenter	onkeyup	onprogress	onwaiting

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

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