**Глава 1. ВВЕДЕНИЕ В HTML5**

* 1. **Общие сведения**

Инициатором разработки стандарта HTML5 является рабочая группа WHATWG (Web Hypertext Application Technology Working Group) [2], учрежденная производителями браузеров: Apple, Mozilla Foundation и Opera Software. Основным направлением работы этой группы является развитие языка HTML и программных интерфейсов (API, Application Program Interface), применяемых при разработке web-приложений. Рабочая группа была образована по причине утраты интереса консорциума W3C к развитию стандарта HTML 4.01, который не обновлялся с 2000 года. С 2009 года консорциум W3C и рабочая группа WHATWG работают над этим проектом совместно. В 2011 году W3C ввел логотип для HTML5 (рис. 1.1).



Рис. 1.1. Логотип HTML5

В спецификации HTML5 следует выделить четыре принципиально новых момента.

1. Семантика элементов: каждый элемент обладает своей смысловой нагрузкой; поисковые системы могут получить более адекватную информацию о смысле данных сайта.
2. Независимость от плагинов: при воспроизведении звука, видео, анимации, игр можно использовать только браузер, не применяя сторонние программы.
3. Применение микроразметки: в элементах предусмотрены специальные атрибуты (itemscope, itemtype и itemprop), позволяющие «пояснить» поисковым системам смысл данных.
4. Можно использовать масштабируемую векторную графику (SVG, Scalable Vector Graphics) прямо в коде html-документа.

При разработке web-приложения разработчик должен решить, в каком объеме использовать HTML5. Если браузер, на который ориентировано приложение, не поддерживает какую-нибудь возможность, нет никакого смысла использовать ее, несмотря на все, что говорится в стандарте. У разработчика нет никаких шансов проверить потенциальную возможность на каждом браузере. К счастью, есть сайты, позволяющие отчасти справиться с этой задачей, например [3].

* 1. **Структура страницы HTML 5**

По сравнению предыдущей версией структура HTML-документа практически не изменилась. На рис. 1.2 приведен пример простейшей html-страницы, соответствующей стандарту HTML5. Сравнив с аналогичной страницей, реализованной на HTML 4.01 легко заметить, изменения в элементе **doctype (**не указывается версия стандарта) и **meta** (можно ограничиться лишь указанием кодировки). В остальном простейшие страницы практически не будут отличается ничем.

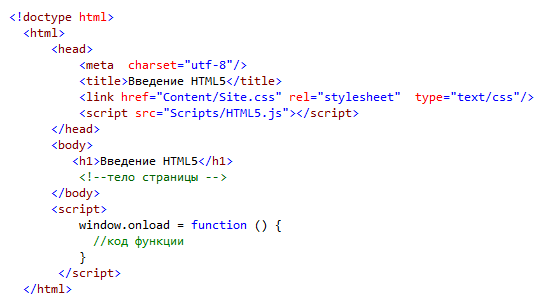


Рис. 1.2. Пример простейшей страницы HTML5

* 1. **Новые элементы HTML5**

В табл. 1.1 приведен список новых элементов языка HTML5 полностью или частично поддерживаемых современными браузерами.

Таблица 1.1. Новые элементы в языке HTML5

|  |  |
| --- | --- |
| Элемент | Назначение |
| <article> | Задает содержание сайта, блога, форму и пр. |
| <aside> | Определяет блок информации, обычно расположенный сбоку (сложился специальный термин «сайдбар»), для размещения ссылок на архив, меток |
| <datalist> | Задает список значений для элементов ввода |
| <details> | Определяет информацию, которая обычно свернута и может быть раскрыта пользователем по желанию. Вместе с этим элементом используется <summary> |
| <figure> | Используется для группировки любых элементов, например, изображений |
| <figcaption> | Содержит описание для <figure> . Должен быть первым или последним внутри <figure>. |
| <footer> | Задает «подвал» сайта. Обычно в нем располагается имя автора, дата документа, правовая информация |
| <header> | Определяет «шапку» сайта. |
| <hgroup> | Применяется для группировки заголовков web-страницы. Обычно содержит теги <h1>, <h2>…. |
| <keygen> | Генерирует открытый и закрытый ключ для безопасной связи с сервером |
| <mark> | Позволяет выделить важный текст |
| <meter> | Используется для отображения количественной меры в ограниченном диапазоне. |
| <nav> | Задает навигацию по сайту. Обычно содержит один или несколько элементов <a> |
| <output> | Задает область для вывода значения, сформированного в сценарии на JavaScript |
| <progress> | Отображает состояние выполнения какого-либо процесса |
| <section> | Задает раздел документа, который может включать в себя заголовки, <header> , текст, <footer> |
| <summary> | Определяет видимый заголовок элемента <details> |
| <time> | Помечает содержимое, как время, дата или одновременно время и дата. Значение может быть задано datatime |
| <audio> | Применяется для воспроизведения аудиофайлов |
| <canvas> | Применяется для 2D-отображения произвольных объектов |
| <source> | Используется совместно с элементами <audio> и <video> для указания нескольких источников |
| <video> | Применяется для воспроизведения видеофайлов |

Обратите внимание на появление элементов (<article>, <aside>, <header>, <footer>, <section>, <details> и пр.), предназначенных для структуризации документа. В предыдущей версии это обеспечивалось, по сути двумя элементами – блочным <div> и строковым <span>. Основное назначение этих элементов – более точно указать поисковым машинам тип информации, хранящихся в документе.

Элементы <audio> и <video> позволяют применять аудио и видео файлы для проигрывания их на html-странице.

С более полным описанием новых элементов в языке HTML5 и примерами их применения можно ознакомиться в [4].

* 1. **Элементы претерпевшие изменения в HTML5**

Практически все элементы в той или иной степени претерпели изменения. В основном изменения сводятся к появлению или устареванию атрибутов или их значений. В HTML5 расширился набор глобальных атрибутов – атрибутов, являющимися общими для всех элементов языка. В таблице 1.2 перечислены новые глобальные атрибуты.

Таблица 1.2. Новые глобальные атрибуты в языке HTML5

|  |  |
| --- | --- |
| Атрибут | Назначение |
| contenteditable | Позволяет редактировать содержимое элемента |
| contenxtmenu | Позволяет задавать контекстное меню для элемента |
| dragable | Позволяет пользователю перетаскивать элемент |
| hiden | Позволяет скрывать элемент |
| spellcheck | Позволяет проверять правописание в содержимом элемента |

Значительно расширись возможности элементов форм. Появление их, существенно облегчает разработку сценариев на JavaScript.

В табл. 1.3 перечислены новые атрибуты элемента <form>.

Таблица 1.3. Новые атрибуты элемента <form> в языке HTML5

|  |  |
| --- | --- |
| Атрибут | Назначение |
| autocomplete | Включает или отключает автозаполнение формы |
| novalidate | Отключает встроенную проверку данных для полей <input type=”email”> и <input type=”url”> |

Элемент <input> в HTML5 существенно расширил свои возможности. В табл. 1.4 перечислены новые атрибуты элемента <input>, в табл. 1.5 – новые значения атрибута type.

Таблица 1.4. Новые атрибуты элемента <input> в языке HTML5

|  |  |
| --- | --- |
| Атрибут | Назначение |
| autocomplete | Включает или отключает автозаполнение элемента |
| autofocus | Установка фокуса на элемент |
| form | Связывает элемент с элементом <form> по атрибуту id |
| formaction | Указывает URL приложения, обрабатывающая введенные элементом (аналогично атрибуту action в <form>) |
| formenctype | Указывает способ кодирования при их отправке их на сервер |
| formmethod | Указание метод отправки данных на сервер (GET или POST) |
| formtarget | Указывает окно в которое будет загружаться результат обработки данных (аналогично атрибуту target в <form>) |
| list | Связывает элемент с элементом <datalist> по атрибуту id |
| max | Указывает максимальное значение для элемента <input type=”number”> |
| min | Указывает минимальное значение для элемента <input type=”number”> |
| multiple | Позволяет ввести несколько значений для элемента <input type=”email”> |
| pattern | Указывает на регулярное выражение, позволяющее осуществить проверку введенных данных |
| placeholder | Указывает на строку, которая находится в текстовом поле и исчезает при вводе |
| readonly | Указывается для поля, которое предназначено только для отображения данных |
| required | Указывается для поля, которое обязательно должно быть заполнено |
| step | Указывает шаг значений для элементов <input type=”number”> |

Таблица 1.5. Новые значения атрибута type элемента <input> в языке HTML5

|  |  |
| --- | --- |
| Значение | Назначение |
| color | Создает поле для выбора цвета с помощью виджета |
| date | Создает поле для выбора даты с помощью виджета |
| datetime | Создает поле для выбора даты и времени |
| datetime- local | Создает поле для выбора местной даты и времени |
| email | Создает поле для ввода адреса e-mail |
| month | Создает поле для ввода месяца |
| number | Создает поле для ввода числа |
| range | Создает виджет (ползунок) ввода числа |
| tel | Создает поле для ввода телефона |
| time | Создает поле для ввода времени |
| url | Создает поле для ввода URL |
| week | Создает поле для ввода номера недели |

* 1. **Устаревшие элементы**

Устаревшие теги, перечисленные в табл. 1.6, с точки зрения спецификации языка HTML5 использовать не рекомендуется. Однако, для обеспечения совместимости, они поддерживаются всеми браузерами.

Таблица 1.6. Элементы HTML 4.01, устаревшие в языке HTML5

|  |  |
| --- | --- |
| Элемент | Назначение |
| <acronym> | Указывает на то, что заданный текст является акронимом (устоявшееся сокращение, которое применяется как самостоятельное слово) |
| <applet> | Используется для внедрения Java-апплета |
| <basefont> | Применяется для форматирование всей html-страницы |
| <dir> | Создает список, содержащий названия директорий |
| <font> | Форматирует, заданный текст |
| <listindex> | Задает поисковую строку |
| <listing> | Содержит код программы |
| <object> | Применяется для внедрения плагина |
| <strike> | Содержит зачеркнутый текст |
| <tt> | Содержит текст, отображаемый моноширным шрифтом |
| <u> | Содержит подчеркнутый текст |
| <xmp> | Содержит текст «как есть» |

* 1. **Новые программные интерфейсы HTML5**

В табл. 1.7 приведен перечень новых программных интерфейсов HTML5 доступных разработчику в программах на языке JavaScript.

Таблица 1.6. Новые программные интерфейсы HTML5

|  |  |
| --- | --- |
| Программный  интерфейс | Назначение |
| Audio API | Позволяет управлять работой элемента <audio>. Например: запустить/остановить воспроизведение, увеличивать/уменьшать звук, осуществлять задержки звука и т. д. |
| Canvas 2D API | Позволяет отображать на холсте элемента <canvas> различные фигуры, менять цвет, толщину линии, выполнять различные преобразования, отображать текст и изображения и т.д. |
| Drag &Drop API | Позволяет захватить мышью и перенести элемент или блоки элементов |
| Geolocation API | Позволяет определить местоположение пользователя |
| IndexedDB API | Позволяет работать с локальной базой данных, хранящей объекты JavaScript |
| History API | Позволяет управлять историей браузера: просматривать, добавлять и изменять элементы истории, реагировать на события истории |
| Selectors API | Позволяет использовать с помощью CSS-селекторов выбирать элементы html-документа (аналогично jQuery) |
| Video API | Позволяет управлять работой элемента <video>. Например: запустить/остановить воспроизведение, увеличивать/уменьшать звук, изменение скорости воспроизведения, отобразить поток web-камеры и микрофона и т.д. |
| WebSockets API | Позволяет поддерживать открытое подключение к web-службе , отправлять сообщения web-службе и обрабатывать сообщения от web-службы |
| WebStorage API | Позволяет сохранять данные (до 10 МБ для одного домена) в виде пар ключ/значение |
| WebSQL API | Позволяет работать с локальной SQL-базой данных, поддерживаемой браузером |
| WebWorkers API | Позволяет в рамках браузера организовать фоновую обработку в виде отдельных потоков |

* 1. **Автономные web-приложения**

Автономные web-приложения – web-приложения HTML5, способные работать без подключения к серверу после того, как они загрузятся на клиентский браузер.

Основой автономного web-приложения является файл манифеста кэша. Первоначально файл манифеста кэша хранится на стороне сервера, указывается в элементе <html> (рис. 1.3) и пересылается вместе этой страницей браузеру.

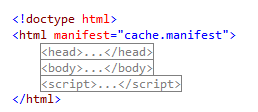
****

Рис. 1.3. Файл манифеста кэша указывается с помощью

атрибута manifest элемента <html>

В общем случае файл манифеста кэша может содержать три секции: CACHE, FALLBACK и NETWORK. В секции CACHE указывается перечень файлов, которые следует загрузить в кэш браузера. Секция FALLBACK указывает файл, который отобразится, если автономное приложение выполняет обратиться к файлу, который не был кэширован. Файлы, которые никогда не должны быть кэшированы, указываются в секции NETWORK.

Для более подробного знакомства с принципами разработки автономных web-приложений рекомендуется [5]

* 1. **Библиотека Modernizr**

При создании web-приложения разработчик, как правило, не может предположить в среде какого браузера будет выполняться это приложение. Поэтому прежде чем использовать некоторую возможность HTML5 следует убедиться, поддерживается ли она данным браузером. Для решения этой проблемы разработана небольшая библиотека с именем Modernizr [6]. Библиотека создает объект с именем Modernizr, включающем в себя свойства для каждой возможности HTML5 (рис. 1.4). Каждое такое свойство принимает одно из двух значений: true (возможность поддерживается), false (возможность не поддерживается).

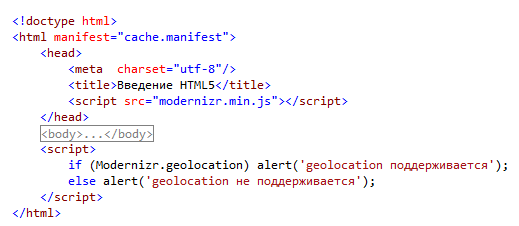


Рис. 1.4. Пример применения библиотеки Modrnizr: проверка возможности использования Geolocation API