

Содержание:

ВВЕДЕНИЕ

Актуальность темы исследования. Настоящее время характеризуется постоянно увеличивающимся ростом объемов информации в повседневной жизни. Большие информационные массивы, которые раньше нельзя было ни собрать, ни хранить, теперь создаются, обрабатываются и перемещаются с огромными скоростями. Если еще десятилетие назад основной составляющей передаваемых информационных массивов была текстовая информация — электронные книги, сведения, описания событий, то на сегодняшний день с такой же скоростью перемещаются и другие данные – графическая, звуковая, видеоинформация, исполняемые компьютерные программы.

Основой современных Интернет-технологий являются языки разметки гипертекста, написания сценариев и скриптов Web-документов. Впервые язык гипертекстовой разметки HTML был использован для создания пересылаемых по сети документов в 1990 г. С тех пор информационные технологии невообразимо изменились. Появилась Всемирная паутина и Консорциум W3C, разрабатывающий и внедряющий новые технологические стандарты. Учитывая скорость изменения современных технологий, постоянное обновление стандартов, и тот факт, что большинство Web-документов содержат описание разметки на языке HTML или XHTML, тема изучения существующих на сегодня языков гипертекстовой разметки является весьма актуальной.

Объектом исследования в работе являются современные Web-технологии, основанные на языках гипертекстовой разметки HTML и XHTML.

Предметом исследования основные принципы создания Web-страниц, Web-сайтов и разработки Internet-приложений с использованием языков гипертекстовой разметки HTML и XHTML.

Целью данной работы является изучение существующих языков разметки гипертекста и основных возможностей, реализуемых с помощью того или иного языка разметки. В связи с этим ставятся **задачи** изучить и описать:

- понятия гипертекста, гиперссылки, гипертекстовой системы, навигации;

- эволюцию и основные принципы создания языков SGML, HTML, XHTML, HTML5;
- общую структуру HTML-документа, использование основных тегов и атрибутов;
- различия в разметке HTML- и XHTML-документов;
- новые возможности языка HTML5.

В качестве источников в работе использованы:

- учебные пособия по информационным системам, информационным технологиям и Web-дизайну [1., 3., 4., 7., 8., 11., 13.];
- работы по теории гипертекста [2., 15., 16., 18.];
- монографии-руководства по созданию Web-сайтов с использованием различных языков гипертекстовой разметки [5., 6., 9., 10., 12., 14., 17.];
- документы Консорциума Всемирной паутины, содержащие точные описания (спецификации) изучаемых языков [20. – 28.];
- справочные материалы по языкам HTML, XHTML и HTML5, представленные и протестированные на сайте <http://htmlbook.ru/html> [19.].

Глава 1. ИСТОРИЯ СОЗДАНИЯ ЯЗЫКОВ РАЗМЕТКИ ГИПЕРТЕКСТА

1.1. Основные понятия концепции гипертекста

Гипертекст — термин, введённый Тедом Нельсоном в 1965 году для обозначения «текста ветвящегося или выполняющего действия по запросу»[\[1\]](#). Обычно гипертекст представляется набором текстов, содержащих узлы перехода между ними, которые позволяют избирать читаемые сведения или последовательность чтения.

По мнению Т. Нельсона, главной чертой гипертекста является отсутствие непрерывности и его нелинейная структура. Иными словами, «материал можно читать в любом порядке, образуя разные линейные тексты»[\[2\]](#).

Первая попытка описать «гипертекстовую машину» была предпринята ещё в 1945 г. Ваннавером Бушем в предложенном им проекте электронно-механического устройства Memex, предназначенного для «усиления мыслительных способностей» человека [18.]. В системе Memex человек «...сможет хранить свои книги, записи и

сообщения, и к которой сможет обращаться, как к консультанту, с растущей скоростью и гибкостью ответов. Для этого перед ним должны находиться два элемента хранения, спроектированные на экраны. Пользователь нажимает специальную кнопку, и эти два элемента соединяются навсегда. После этого в любой момент времени, когда один из них находится в поле зрения, другой может быть вызван простым нажатием кнопки». [3] Бушем было также предсказано всеобъемлющее индексирование текстов и мультимедийных ресурсов с возможностью быстрого поиска необходимой информации.

Приведем еще несколько определений гипертекста:

- *Гипертекст* позволяет связывать текст, аудио, фотографии, чертежи, карты, движущиеся картинки и другие формы информации в осмысленное целое, к которому может осуществляться доступ при помощи системы индексации, ориентированной на конкретные идеи, а не на конкретные слова в тексте».
- *Гипертекст* можно определить как нелинейную документацию, документацию, которая ветвится и взаимосвязывается, позволяя читателю исследовать содержащуюся в ней информацию, в последовательности, которую он сам выбирает. [4]
- *Гипертекст* — это обычный текст, содержащий ссылки на собственные фрагменты и другие тексты. Аналогом гипертекста можно считать книгу, оглавление которой по своей сути представляет ссылки на главы, разделы, страницы (рис. 1). [5]
- *Гипертекст* — это текст, содержащий ключевые слова-ссылки (гиперссылки) на другие источники информации (фрагменты текстов, словарь терминов, понятий и определений, персоналии, хрестоматийные статьи, литературу, электронные библиотеки, иллюстративный материал, образовательные сайты в Интернете и другие вспомогательные учебные материалы). [6]

Текст 2

Текст 4

Текст 1

Текст 3

Рис. 1.1. Структура гипертекста.

Гипертекст интерактивен, так как читатель сам выбирает способ и последовательность чтения гипертекста. Основным элементом гипертекста являются гиперссылки.

Гиперссылкой называется особым образом отформатированный текст или рисунок, содержащий скрытый адрес другого ресурса в Web. Щелчок по гиперссылке вызывает переход к заданному документу, который отобразится либо в активном окне обозревателя, либо во вновь открытом окне. [7]

В зависимости от места, на которое осуществляется переход, можно выделить *локальные* (ссылки на другие части того самого документа, откуда они осуществляются) и *глобальные* (ссылки на другие документы, в общем случае — на какие-либо ресурсы, расположенные вне данного документа) гиперссылки.

В зависимости от объекта, на который ссылаются гиперссылки, их можно разделить на:

- гиперссылки на web-страницу;
- гиперссылки на часть документа внутри текущей Web-страницы;
- гиперссылки на документ FTP-сервера
- гиперссылки на мультимедийный Internet-ресурс;
- гиперссылки на электронную почту;
- гиперссылки на сценарий на языке Java Script. [8]

Гипертекст в совокупности с различными видами мультимедийной информации и поддерживаемыми между ними связями называется **гипермедиа**.

Центральным понятием концепции гипертекста является **навигация** — управление процессом перемещения в гиперпространстве из произвольного узла отправления в узел прибытия. Специфической составляющей навигации является **броузинг** (от англ. «browse») — процесс беглого просматривания гипертекстовых документов или гипертекстовой базы данных с целью поиска определенных сведений. [9]

1.2. Обобщенный стандартный язык разметки SGML

Концепция гипертекста, предложенная Т. Нельсоном, получила свое дальнейшее развитие в конце 1960-х гг. в работах Чарльза Голдфарба. Группа исследователей из компании IBM под его руководством в 1969 г разработала «прототип языка для разметки технической документации, впоследствии названного GML (Generalized Markup Language)». [10] Основным принципом языка GML было четкое разграничение логической структуры документа и его оформления.

Для этой цели был задействован механизм определений типов документов (DTD — Document Type Definition). DTD, как и лист стилей, представляющий собой файл, внешний по отношению к документу. Разделение содержания, представления и проверки на допустимость обеспечивает чрезвычайную гибкость, поскольку как листы стилей, так и DTD могут быть легко модифицированы, в то время как структурная разметка документа может при этом оставаться неизменной. [11]

В 1986 г. Международная организация по стандартизации (ISO) приняла первый стандарт ISO-8879, основанный на GML и озаглавленный «Standard Generalized Markup Language (SGML)». Этот стандарт был посвящен описанию языка SGML — обобщенного метаязыка, позволяющего строить системы логической и структурной разметки любых разновидностей текстов. Слово *структурная* означает, что управляющие коды, вносимые в текст при такой разметке, не несут никакой информации о внешнем виде документа, а лишь указывают границы и соподчинение его составных частей, т. е. задают его логическую структуру.

Язык SGML был разработан для стандартизации обмена данными между разнородными программами и платформами. Он являлся не только мощным, но и очень сложным средством. Кроме того, SGML-документ было трудно интерпретировать без определений языка разметки, который хранится в определении типа документа DTD (document type definition). DTD необходимо было посылать вместе с SGML-документом или включать в документ, для того чтобы можно было распознать теги, созданные пользователем. Под тегами понимались специальные управляющие коды, записываемые в виде текста, ограниченного угловыми скобками. [12]

Принципы, на которых строится язык SGML, значительны и интересны; идеология языка оказала влияние на многие компьютерные разработки. Однако сам по себе SGML не получил заметного распространения до тех пор, пока он не был выбран в качестве основы для нового языка разметки гипертекстовых документов. [13]

1.3. Возникновение Всемирной паутины и языка HTML

Всемирная паутина (World Wide Web, WWW) зародилась в Европейской организации по ядерным исследованиям (CERN) в Женеве (Швейцария) в 1989 году. Программист по имени Тим Бернерс-Ли впервые предложил систему управления информацией, которая использовала «гипертекстовый» процесс для объединения связанных документов по сети. Он и его сотрудники создали прототип проекта и выпустили его для презентации. В первые несколько лет веб-страницы были только текстовыми. В 1992 году в мире было всего 50 веб-серверов.

Реальный рост популярности Всемирной паутины произошел в 1993 году, когда был представлен первый графический браузер (NCSA Mosaic). Это позволило Всемирной паутине превратиться из научного исследования в средство массовой информации. Развитие Всемирной паутины продолжается под наблюдением созданного в 1994 г. Консорциума Всемирной паутины (W3C).[\[14\]](#)

В основе Всемирной паутины лежит протокол передачи гипертекстовых сообщений HTTP (Hypertext Transfer Protocol), а сами страницы формируются с помощью гипертекстового языка описания документов HTML (HyperText Markup Language).[\[15\]](#)

Всемирную паутину образуют миллионы веб-серверов сети Интернет, расположенных по всему миру. Большинство ресурсов Всемирной паутины представляет собой гипертекст. Гипертекстовые документы, размещаемые во всемирной паутине, называются веб-страницами.

Веб-страница, или **веб-документ**, как правило, состоит из основного HTML-файла и некоторого количества ссылок на другие объекты разного типа: изображения, другие HTML-файлы, аудио- и видеофайлы.

HTML-файлом, **HTML-страницей** или **гипертекстовой страницей** называют файл, который содержит текст, написанный на языке HTML.[\[16\]](#)

Язык HTML, составивший основу WWW, был разработан Тимом Бернерсом-Ли на базе уже существующего стандартного языка разметки печатных документов — SGML (Standard Generalised Markup Language), который является стандартом ISO с 1986 г. Строго говоря, язык HTML есть подмножество метаязыка SGML.[\[17\]](#)

HTML, однако, был задуман как достаточно вольный стандарт. Иными словами, страница, созданная с его использованием, должна обрабатываться браузером несмотря ни на что, в любом случае, даже при наличии откровенных ошибок в коде.

В качестве элемента гипертекстовой базы данных, описываемой HTML, используется текстовый файл, который может легко передаваться по сети с использованием протокола HTTP. Эта особенность, а также то, что HTML является открытым стандартом, и огромное количество пользователей имеет возможность применять этот язык для оформления своих документов, безусловно, повлияли на рост популярности HTML и сделали его сегодня главным механизмом представления информации в Web. Преимуществами языка HTML являются простота, надплатформенность (возможность непосредственной интерпретации в любой операционной системе), минимальный размер файла, удобство редактирования и интерпретации.

Язык разметки гипертекстовых документов HTML позволяет определить различные типы элементов, обеспечивающих функциональность документа: текстовые фрагменты с заданными параметрами форматирования, списки, таблицы, изображения, гиперссылки и т. д. Элементы HTML объявляются с помощью команд разметки, называемых тегами (от англ. tag — «ярлык»). Встречающиеся в тексте документа HTML-теги интерпретируются браузером при отображении документа.

[\[18\]](#)

1.4. Эволюция версий языков HTML и XHTML

Представленная миру Тимом Бернерсом-Ли в 1991 г. версия HTML 1.0, насчитывала всего лишь 20 тегов. Описание возможностей носило скорее рекомендательный характер. Официальной спецификации для языка HTML 1.0 не существовало.

В апреле 1994 г. консорциумом Всемирной паутины (W3C) началась подготовка спецификации следующей версии языка — 2.0. Спецификация для HTML 2.0 была выпущена в сентябре 1995 г. [\[19\]](#) и содержала в основном конструкции, необходимые для создания жизнеспособной системы управления информацией. [\[20\]](#)

В январе 1997 г. W3C создает HTML 3.2 [\[21\]](#), затем в апреле 1998 г. — HTML 4.0 [\[22\]](#), а в декабре 1999 года — HTML 4.01 [\[23\]](#). Все перечисленные версии представляли собрание всех HTML-элементов, использовавшихся в то время. В них вошло

множество презентационных расширений HTML, подстроенных под работу доминирующих браузеров — Netscape Navigator, Mosaic и Internet Explorer, но все еще отсутствовали решения в виде отдельных таблиц стилей.

Стандарт HTML 4.01 определяется формально в виде трех SGML-определений типа документа: строгого (Strict DTD), переходного (Transitional DTD) и фреймового (Frameset DTD).[\[24\]](#) Строгое DTD (рекомендуемое к использованию) определяет только те элементы, которые не признаны нежелательными в стандарте 4.0. Переходное DTD включает в себя все эти нежелательные элементы и более точно отражает современное употребление HTML, в котором все еще широко используются устаревшие элементы. Фреймовое DTD позволяют разделить окно браузера на несколько фрагментов с тем, чтобы отображать в них разные документы.

Примерно в то же время, когда разрабатывалась версия HTML 4.01, сотрудники W3C осознали, что один язык разметки с ограниченными возможностями не получится использовать для описания всех видов информации (химических формул, математических уравнений, мультимедийных презентаций, финансовой информации и т. д.), которые можно распространять во Всемирной паутине. Их решение — XML (eXtensible Markup Language, Расширяемый язык разметки), метаязык для создания языков разметки, основанный на стандартном обобщенном языке разметки SGML. [\[25\]](#)

В итоге, HTML был заморожен в версии 4.01. В январе 2000 г. была выпущена спецификация под названием XHTML 1.0[\[26\]](#), которая представляла собой XML-версию HTML, использующую строгие правила синтаксиса XML — заключение атрибутов в кавычки, закрытие тегов и т. п. Были разработаны два стандарта: «переходной» XHTML Transitional и «золотой стандарт» XHTML Strict.[\[27\]](#)

В мае 2001 года W3C выпустил обновленный стандарт XHTML 1.1[\[28\]](#), представляющий собой еще более строгую версию XHTML по сравнению с XHTML 1.0, и имеющий больше расхождений с HTML 4.01. Возможно, стандарт и стал рафинированным с академической точки зрения, но пользоваться им практически невозможно.[\[29\]](#)

В 2002 году W3C начал разрабатывать XHTML 2.0, язык решили делать с нуля, что привело к проблемам совместимости с предыдущими версиями языков HTML и XHTML. Процесс написания и утверждения затянулся на годы. Без реализации в браузерах создание XHTML 2.0 застопорилось.[\[30\]](#)

1.5. Создание стандарта HTML5

История HTML5 начинается в 2004 году, когда крупные компаний, такие как Google, Mozilla, Opera, Apple и Microsoft, создают свою собственную рабочую группу под названием WHATWG (Web Hypertext Application Technology Working Group, Рабочая группа по разработке гипертекстовых приложений). Эти люди не считали, что будущее Интернет — это XML.

Перед тем, как начать работать над HTML5, рабочая группа WHATWG создала две спецификации: Web Forms 2.0 (веб-формы) и Web Apps 1.0 (веб-приложения), которые впоследствии стали частью спецификации HTML5.

В 2006 W3C всё еще продолжает работать над XHTML 2.0, но его разработчики решили, что они были чересчур оптимистичны, полагая, что весь мир перейдет на XML, и, естественно, к XHTML 2.0. «Важно развивать HTML шаг за шагом. Попытка заставить мир одновременно перейти на XML, включая кавычки вокруг атрибутов, закрытие пустых тегов и пространства имен, провалилась», — признал Тим Бернерс-Ли.[\[31\]](#)

Поэтому в этом же году W3C начал разработку своей версии HTML5, в её основе лежали наработки рабочей группы WHATWG. В конце 2009 г. W3C официально прекратил работу над XHTML 2.0[\[32\]](#), закрыв рабочую группу и направив свои ресурсы и усилия на HTML5.[\[33\]](#)

На данный момент, работа над языком HTML5 продолжается, создаются новые теги и технологии, всё это добавляется в спецификацию. Сама спецификация HTML5 была опубликована 28 октября 2014 г.[\[34\]](#)

Приведем основные характеристики HTML5.

- HTML5 основан на версии HTML 4.01, которая не содержит презентационных или иных устаревших элементов и атрибутов. Это означает, что подавляющее большинство элементов в языке HTML5 те же, которые использовались в течение многих лет, и корректно обрабатываются браузерами.
- В HTML5 не используется DTD (определение типа документа), представляющее собой отдельный документ, определяющий все элементы и атрибуты в языке разметки. Таким образом, язык полностью документируется в SGML, отсутствуют варианты DTD — Transitional, Strict и Frameset.

- HTML5 — первая спецификация HTML, которая содержит подробные инструкции о том, как браузеры должны обрабатывать неправильную и наследованную разметку.
- HTML5 также может быть написан в соответствии с более строгим синтаксисом XML (т.н. XML-сериализация HTML5). Некоторые разработчики стали предпочитать характеристики «опрятного» стандарта XHTML, так что этот способ верстки по-прежнему остается, хотя и не обязателен.
- В дополнение к разметке, HTML5 определяет ряд API-интерфейсов (интерфейсов прикладного программирования). Они облегчают связь с веб-приложениями, а также возлагают некоторые распространенные процессы (аудио-, видеопроигрыватели) на встроенный функционал браузеров.[\[35\]](#)

С ноября 2016 г. W3C рекомендовал к применению версию HTML5.1.[\[36\]](#)

В заключение отметим, что в настоящее время язык гипертекстовой разметки HTML5 вместе с каскадными таблицами стилей CSS3 и языком сценариев JavaScript являются основными инструментами для создания Web-страниц и Web-приложений.[\[37\]](#)

Выводы по главе 1.

Таким образом, в первой главе даны основные понятия и определения концепции гипертекста: гипертекст, гиперссылка, гипертекстовая система, навигация. Далее отмечено, что идеи гипертекста нашли свое воплощение в создании языка SGML (стандартного обобщенного языка гипертекстовой разметки), стандартизированного международной организацией ISO в 1986 г. Следующим шагом стало создание на базе SGML языка HTML (языка гипертекстовой разметки) и появление Всемирной паутины в 1989 году. В главе показана эволюция основных версий языка HTML (HTML 2.0, 3.2, 4.0 и 4.01) с 1991 по 1999 г., создание нового метаязыка XML (расширяемого языка разметки), разработка на основе XML языков XHTML 1.0, 1.1 и 2.0.

Отмечено, что с 2009 г. Консорциум Всемирной паутины перестал развивать XHTML, и все усилия бросил на развитие нового стандарта HTML5. Язык HTML5 имеет много новшеств по сравнению с предыдущими версиями, отличается простотой написания и корректной обработкой всеми браузерами. В 2014 г. он был

рекомендован к использованию Консорциумом Всемирной паутины и является на сегодняшний день общепринятым стандартом для разметки Web-документов.

Глава 2. ОСНОВНЫЕ ПРИНЦИПЫ РАЗМЕТКИ WEB-ДОКУМЕНТОВ С ПОМОЩЬЮ ЯЗЫКОВ HTML И XHTML

2.1. Основные возможности языка HTML 4.01

HTML-документ представляет собой обычный текстовый файл, содержащий текст документа и специальные языковые конструкции — *теги* или *дескрипторы*, используемые для разметки документа и управления его отображением. Для создания HTML-документа можно использовать любой простейший текстовый редактор. [\[38\]](#)

Определение типа документа. В первой главе были рассмотрены различные версии языков HTML, XML и XHTML. Для правильной интерпретации языка, на котором написан документ, браузеру следует сообщить, какой именно язык выбран. Это делается посредством специального дескриптора `<!DOCTYPE...>`, имеющего структуру:

```
<!DOCTYPE элемент_верхнего_уровня публичность  
"регистрация//организация//тип имя//язык" "url">.
```

Для HTML 4.01 наиболее употребительными являются следующие схема задания содержимого данного дескриптора: [\[39\]](#)

Строгая (Strict) схема:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 strict//EN"  
"http://www.w3.org/TR/HTML4.01/strict.dtd">
```

Переходная (Transitional) схема:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 transitional//EN"
```

"http://www.w3.org/TR/HTML4.01/loose.dtd">

Фреймовая (Frameset) схема:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 frameset//EN"

"http://www.w3.org/TR/HTML4.01/frameset.dtd">.

В настоящее время браузеры воспринимают HTML-код и без дескриптора <!DOCTYPE...>, но в этом случае не исключены непредвиденные ситуации, особенно в Internet Explorer, которых разработчик обязан избегать.

Структура HTML-документа. Любой документ в формате HTML начинается с открывающего тега <HTML> и заканчивается тегом </HTML>. Он состоит из двух частей:

- раздела заголовка (определяемого тегом HEAD);
- тела, которое включает содержимое документа и определяется тегом BODY. В общем виде документ HTML имеет следующую структуру:[\[40\]](#)

<HTML>

<HEAD>

Раздел заголовка

</HEAD>

<BODY>

Тело документа

</BODY>

</HTML>

Раздел заголовка содержит различного рода служебную информацию (например, ключевые слова, используемые поисковыми машинами), не считающуюся содержимым документа. Наиболее часто в заголовке применяются следующие теги:

<TITLE> — заголовок HTML-документа, который отображается в строке заголовка окна браузера;

<BASE> — базовый адрес, используемый при обработке относительных URL-адресов;

<LINK> — тег, используемый для связи с другими HTML-документами;

<META> — дополнительная информация об HTML-документе.

Тело документа содержится между тегами <BODY> и </BODY> и включает всю информацию, которая отображается в окне браузера.

В теле документа используются специальные теги, обеспечивающие форматирование текста документа и включающие в него изображения, таблицы и формы.

Теги форматирования текста. Приведем описание основных тегов, используемых для задания формата отображения текста в окне браузера.[\[41\]](#)

Заголовки. HTML-документ может содержать шесть уровней заголовков. Для задания заголовка используются пары тегов:

<H1> заголовок первого уровня </H1>

<H2> заголовок второго уровня </H2>

...

<H6> заголовок шестого уровня </H6>

В открывающем теге можно указать дополнительный атрибут ALIGN, определяющий способ выравнивания текста заголовка. Данному атрибуту можно задавать одно из трех значений:

LEFT — выравнивание по левому краю;

RIGHT — выравнивание по правому краю;

CENTER — выравнивание по центру.

Например, для создания заголовка первого уровня, выровненного по центру, используется следующая строка:

<H1 ALIGN=CENTER> Текст заголовка </H1>

Абзацы. Текст, относящийся к одному абзацу, заключается между тегами <P> и </P>. Каждый абзац отделяется от предыдущего увеличенным межстрочным интервалом. Так же как и для заголовков, для абзаца можно задавать способ выравнивания текста с помощью атрибута ALIGN. Если требуется начать текст с новой строки в пределах одного абзаца, используется тег
.

Списки. HTML-документ может содержать как маркированные, так и нумерованные списки. Для создания маркированных списков используются теги и , нумерованных — и . В обоих случаях каждый элемент списка помещается между тегами и . Допускается создание вложенных списков.

Выделение фрагментов текста. Язык HTML позволяет выделять отдельные слова и символы документа. Приведем основные теги, используемые при выделении:[\[42\]](#)

 ... — выделение полужирным шрифтом;

<I> ... </I> — выделение курсивом;

<U> ... </U> — выделение подчеркиванием.

Указанные теги задают способ выделения фрагмента текста явным образом. Наряду с ними можно использовать теги, которые лишь указывают, что текст должен быть выделен, не обозначая способа выделения. В этом случае выбор способа выделения определяется браузером:

 ... — выделенному тексту должно уделяться внимание;

 ... — выделенному тексту должно уделяться особое внимание;

<KBD> ... </KBD> — обозначение клавиши клавиатуры;

<VAR> ... </VAR> — обозначение переменной;

<CITE> ... </CITE> — цитата.

Рассмотрим пример HTML-документа, в котором используется большинство из рассмотренных тегов:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 transitional//EN"
```

```
"http://www.w3.org/TR/HTML4.01/loose.dtd">
```

<HTML>

<HEAD> <TITLE> Пример HTML-документа </TITLE> </HEAD>

<BODY>

<H2 ALIGN=CENTER> Пример использования списков </H2>

<P> Вложенный список:

<U>Элемент 1</U>

<I>Элемент</I> 1.1

<I>Элемент</I> 1.2

<U>Элемент 2</U>

<I>Элемент</I> 2.1

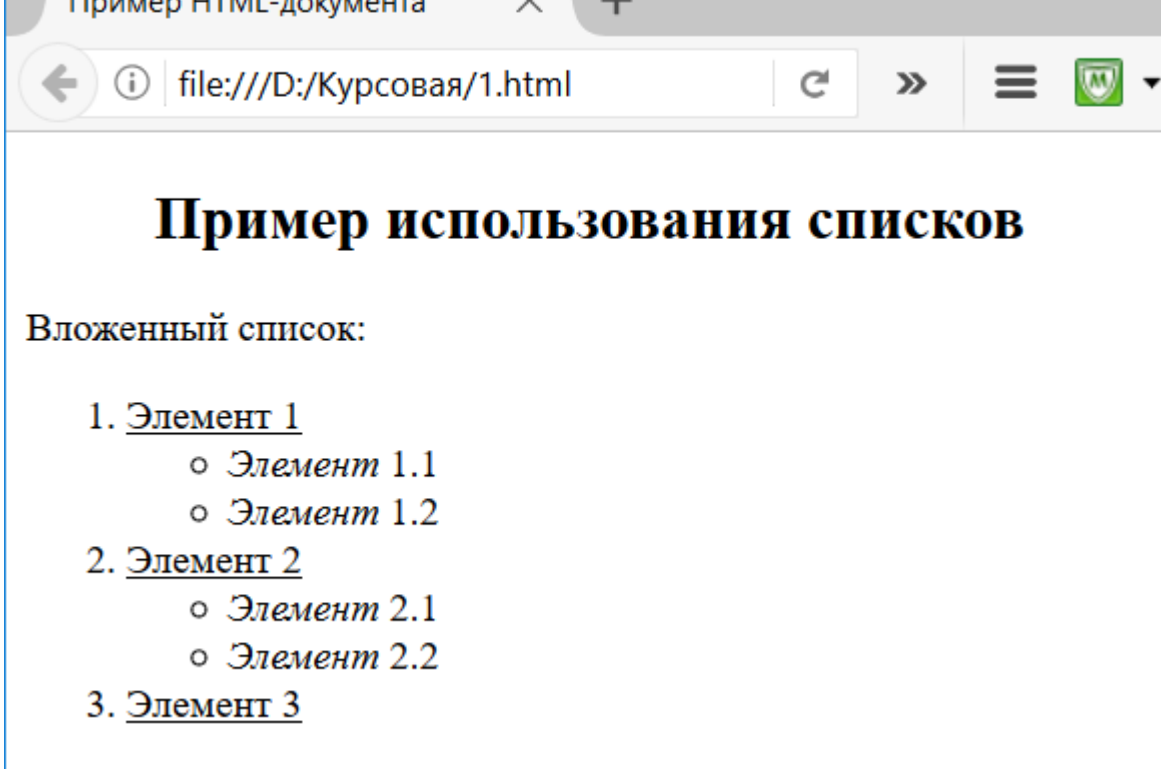
<I>Элемент</I> 2.2

<U>Элемент 3</U>

</P>

</BODY>

</HTML> [\[43\]](#)



Вид данного

документа в окне браузера Mozilla Firefox показан на рис. 2.1. В приведенном выше HTML-коде указан один параметр заголовка — заголовок окна браузера, задан один заголовок текста с выравниванием по центру и создан вложенный список.

Элементы первого уровня в списке представляют собой нумерованный список, а элементы второго уровня — маркированный список.

Рис. 2.1. Пример HTML-документа, содержащего списки

Гиперссылки. Гиперссылки обеспечивают связь между различными HTML-документами. Гиперссылка представляет собой фрагмент HTML-документа (текст или изображение), щелчок на котором приводит к загрузке другого документа.[\[44\]](#)

Для создания гиперссылки используется пара тегов `<A>` и ``. Заключение между ними фрагмент HTML-документа при просмотре будет отображаться как гиперссылка. Тег `<A>` обязательно должен использоваться совместно с атрибутом `href`. С помощью него задается ссылка на документ, к которому должен быть произведен переход при щелчке на гиперссылке.

Таким образом, фрагмент HTML-документа, задающий гиперссылку, в общем виде выглядит так:

` фрагмент документа `

Здесь URL-адрес, задаваемый атрибутом HREF, может быть двух видов: абсолютным и относительным:

- абсолютный URL-адрес содержит полную информацию о местоположении ресурса и протоколе обращения к ресурсу;
- относительный URL-адрес указывает расположение ресурса относительно местоположения текущего HTML-документа. Например, строка

```
<A HREF=doc1.html> Переход к документу 1 </A>
```

создает гиперссылку, указывающую на гипертекстовый документ doc1.html, который размещен в том же каталоге, что и текущий документ.

Если абсолютный адрес документа, содержащего приведенную гиперссылку, выглядит, например, как <http://www.domen.ru/information/main.html>, то абсолютным адресом, на который эта гиперссылка ссылается, будет <http://www.domen.ru/information/doc1.html>.

Тег <A> имеет еще одно назначение — кроме создания гиперссылок он позволяет устанавливать маркеры для организации переходов по гиперссылкам в пределах одного документа HTML. Для задания маркера тег <A> используется совместно с атрибутом NAME:

```
<A NAME="имя_маркера"> текст </A>
```

В этом случае текст, заключенный между тегам <A> и , при отображении никак не выделяется, но к помеченному таким образом фрагменту HTML-документа можно перейти с помощью гиперссылки следующего вида:

```
<A HREF="#имя_маркера"> текст </A>
```

Гиперссылки такого вида удобно использовать в документах большого объема.

Формы. Формы предназначены для организации интерактивного режима работы пользователя, обеспечивая взаимодействие между пользователем, работающим на клиентской машине, и веб-приложениями, выполняющимися на стороне сервера. Для создания формы используется пара тегов <FORM> и </FORM>. Между ними располагаются строки, описывающие различные элементы управления: кнопки, поля ввода, флажки и т. п.:[\[45\]](#)

```
<FORM> описание элементов управления </FORM>
```

Совместно с тегом <FORM> практически всегда используются атрибуты ACTION и METHOD:

- атрибут ACTION предназначен для указания URL-адреса программы (сценария), которая будет выполнять обработку данных, введенных пользователем;
- атрибут METHOD определяет метод, с помощью которого данные, введенные пользователем, будут передаваться на сервер. Он может принимать одно из двух значений: GET или POST. Основные элементы управления создаются с помощью тега <INPUT>, который используется без закрывающего тега. Тип управляющего элемента задается с помощью атрибута TYPE данного тега. Кроме атрибута TYPE, тег <INPUT> содержит ряд других атрибутов, определяющих параметры элемента управления.

Поля ввода. Для создания полей ввода атрибуту TYPE следует присвоить значение "TEXT". Параметры поля ввода задаются следующими атрибутами тега <INPUT>: [\[46\]](#)

NAME — идентификатор элемента управления;

VALUE — начальное значение, отображаемое в поле ввода сразу после загрузки документа;

SIZE — максимальное количество отображаемых символов;

MAXLENGTH — максимальное количество символов, которые могут быть введены с помощью данного поля ввода. Например, приведенный ниже фрагмент кода создает форму, содержащую текстовое поле txt1 длиной 19 символов:

```
<FORM METHOD="POST"
```

```
ACTION="http://domen.ru/scripts/test.cgi">
```

```
Имя: <INPUT TYPE="TEXT" SIZE=19 NAME="txt1">
```

```
</FORM>
```

Имеется еще одна разновидность полей ввода — поля, предназначенные для ввода пароля. Для создания такого поля ввода атрибуту TYPE следует задать значение "PASSWORD". Все символы, вводимые в этом поле ввода, будут отображаться на экране в виде звездочек (*). В остальном этот элемент управления ничем не отличается от обычного поля ввода.

Флажки. Для создания флажка атрибуту TYPE тега <INPUT> задается значение "CHECKBOX". Параметры флажка определяются следующими атрибутами:

NAME — идентификатор элемента управления;

VALUE — атрибут, определяющий значение, которое передается на сервер в случае, если флажок установлен;

CHECKED — атрибут, указывающий, что после загрузки документа флажок должен быть установлен. Данному атрибуту не задается никакого значения. Например, следующий код создает флажок, передающий на сервер значение "YES" и по умолчанию являющийся установленным:

```
<FORM METHOD="POST"
```

```
ACTION="http://domen.ru/scripts/test.cgi">
```

```
<INPUT TYPE="CHECKBOX" NAME="chk1" VALUE="YES" CHECKED>
```

Запомнить

```
</FORM>
```

Переключатели. Переключатель представляет собой группу элементов управления, подобных флажкам. Однако в отличие от последних в установленном состоянии может находиться только один из элементов управления, входящих в группу. Для создания переключателей атрибуту TYPE задается значение "RADIO". Параметры переключателей определяются следующими атрибутами:[\[47\]](#)

NAME — идентификатор переключателя. Он должен быть одинаковым для всех элементов управления, входящих в одну группу;

VALUE — значение, передаваемое серверу при установленном значении элемента управления;

CHECKED — атрибут, указывающий, какой из элементов управления должен быть установлен в группе при загрузке документа.

Следующий фрагмент HTML-кода содержит описание переключателя с тремя положениями:

```
<FORM METHOD="POST"
```

```
ACTION="http://domen.ru/scripts/test.cgi">  
<INPUT TYPE="RADIO" NAME="rb1" VALUE="1" CHECKED>  
Сегодня <BR>  
<INPUT TYPE="RADIO" NAME="rb1" VALUE="2">  
за последнюю неделю <BR>  
<INPUT TYPE="RADIO" NAME="rb1" VALUE="3">  
за последний месяц <BR>  
</FORM>
```

Кнопки. Различают два вида кнопок:[\[48\]](#)

- кнопка SUBMIT производит передачу введенных пользователем данных на сервер;
- кнопка RESET сбрасывает все элементы управления в исходные состояния.

Для создания кнопок атрибуту TYPE задается значение либо "SUBMIT", либо "RESET" — в зависимости от того, какую кнопку требуется создать. Надпись на кнопке задается с помощью атрибута VALUE. Приведенный ниже фрагмент кода создает пару кнопок, одна из которых имеет тип SUBMIT, а вторая — RESET:

```
<FORM METHOD="POST"  
ACTION="http://domen.ru/scripts/test.cgi">  
<INPUT TYPE="SUBMIT" VALUE="OK" CHECKED>  
<INPUT TYPE="RESET" VALUE="CANCEL">  
</FORM>
```

Приведем пример формы, содержащей основные элементы управления:

```
<HTML>  
<HEAD> <TITLE> Пример HTML-формы </TITLE>  
</HEAD>
```

<BODY>

<H2 ALIGN=CENTER> Пример создания форм </H2>

<FORM METHOD="POST"

ACTION="http://domen.ru/scripts/test.cgi">

Имя: <INPUT TYPE="TEXT" SIZE=19 NAME="txt1">

<INPUT TYPE="CHECKBOX" NAME="chk1" VALUE="YES" CHECKED>

Запомнить

<INPUT TYPE="RADIO" NAME="rb1" VALUE="1" CHECKED>

сегодня

<INPUT TYPE="RADIO" NAME="rb1" VALUE="2">

за последнюю неделю

<INPUT TYPE="RADIO" NAME="rb1" VALUE="3">

за последний месяц

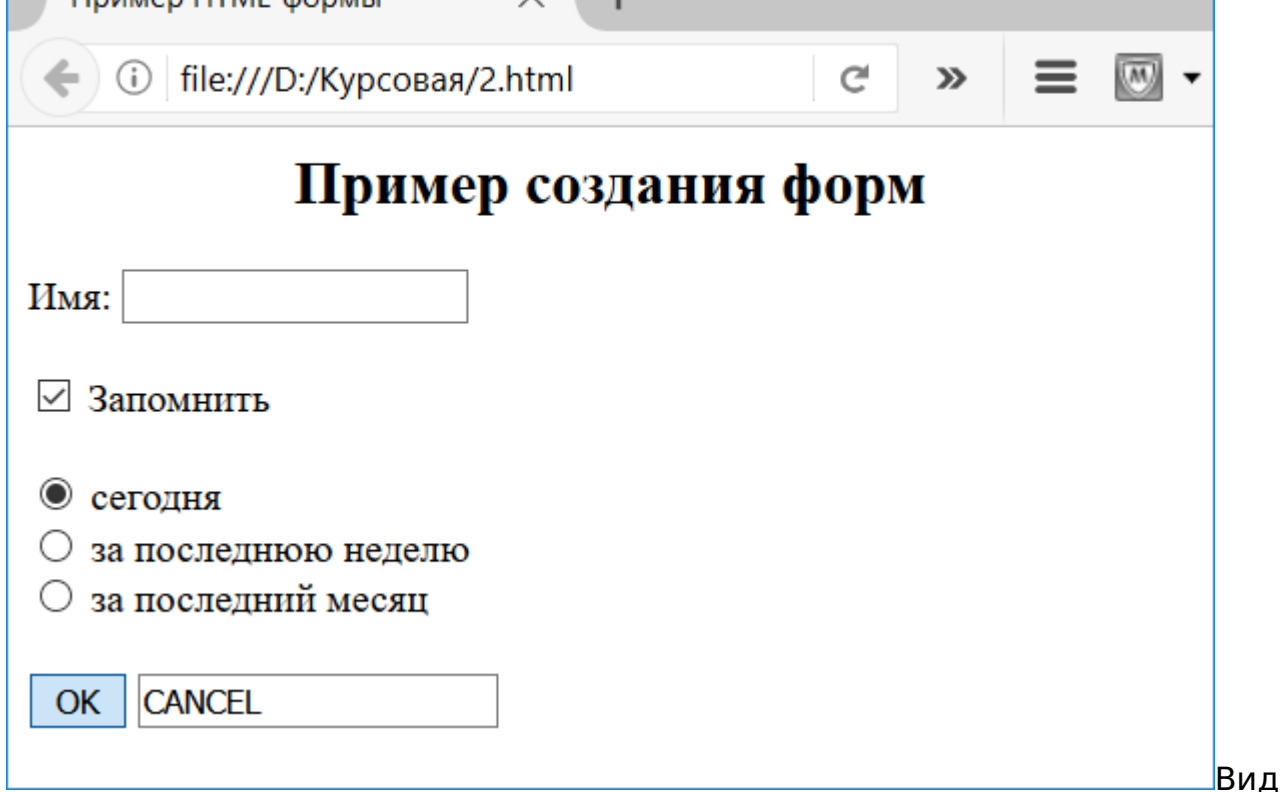
<INPUT TYPE="SUBMIT" VALUE="OK" CHECKED>

<INPUT TYPE="RESET" VALUE="CANCEL">

</FORM>

</BODY>

</HTML> [\[49\]](#)



приведенного HTML-документа в окне браузера Mozilla Firefox показан на рис. 2.2.

Рис. 2.2. Пример HTML-формы.

Таблицы. Для создания таблиц в HTML 4 применяется тег <TABLE>, форма записи которого в общем виде следующая:[\[50\]](#)

```
<TABLE ALIGN="ЗНАЧЕНИЕ" WIDTH="ЗНАЧЕНИЕ" BORDER="ЦЕЛОЕ ЧИСЛО">
```

```
CELLSPACING="ЦЕЛОЕ ЧИСЛО" CELLPADDING="ЦЕЛОЕ ЧИСЛО"
```

```
<CAPTION> ЗАГОЛОВОК ТАБЛИЦЫ </CAPTION>
```

```
<TR ALIGN="ЗНАЧЕНИЕ" VALIGN="ЗНАЧЕНИЕ">
```

```
<TD ALIGN="ЗНАЧЕНИЕ" VALIGN="ЗНАЧЕНИЕ"
```

```
COLSPAN = "ЦЕЛОЕ ЧИСЛО" ROWSPAN = "ЦЕЛОЕ ЧИСЛО"
```

```
HEIGHT="ЗНАЧЕНИЕ" WIDTH="ЗНАЧЕНИЕ">
```

```
Содержимое ячеек</TD>
```

```
</TR>
```

</TABLE>

Основополагающим тегом в блоке описания таблицы является директива <TABLE>, в нее вкладываются все остальные команды блока. Данный тег может использовать следующие атрибуты:[\[51\]](#)

ALIGN — задает горизонтальное выравнивание всей таблицы в целом, может принимать значения LEFT, RIGHT или CENTER;

WIDTH — определяет ширину всей таблицы и может принимать значение целого числа, если ширина таблицы указывается в пикселах, либо числа от 1 до 100 с символом «%» (например «60%»), если ширина таблицы задается в процентах от ширины экрана пользователя;

BORDER — указывает на толщину обрамления столбцов и ячеек таблицы в пикселах. Если значение данного атрибута равно нулю, сетка таблицы становится невидимой;

CELLSPACING — задает промежуток между ячейками в пикселах;

CELLPADDING — задает промежуток между содержимым ячейки и рамкой вокруг ячейки в пикселах.

Тег <CAPTION> задает заголовок всей таблицы.

В теги <TABLE> и </TABLE> вкладываются теги <TR> и </TR>, определяющие строки таблицы. Количество данных тегов зависит от реального количества строк. Команда <TR> может использовать атрибут ALIGN, определяющий горизонтальное выравнивание содержимого ячейки внутри самой ячейки, а также атрибут VALIGN, задающий вертикальное выравнивание содержимого ячейки. Первый из них может принимать значения LEFT, RIGHT и CENTER, второй — значения TOP, MIDDLE, BOTTOM и BASELINE.

В контейнер, образуемый тегами <TR> и </TR>, вкладываются команды <TD> и </TD>, описывающие сами ячейки. Тег <TD> оперирует атрибутами ALIGN и VALIGN, действие и параметры которых аналогичны таковым для тега <TR>. Кроме того, тег <TD> оперирует следующими необязательными атрибутами: COLSPAN — число столбцов, перекрываемых ячейкой; ROWSPAN — число строк, перекрываемых ячейкой. HEIGHT и WIDTH — соответственно высота и ширина ячейки в пикселах, либо в процентах от высоты таблицы.[\[52\]](#)

Приведем пример верстки простой таблицы, состоящей из четырех строк и двух столбцов, включающей объединенные ячейки:[\[53\]](#)

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ТАБЛИЦА С ОБЪЕДИНЕНИЕМ ЯЧЕЕК</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<TABLE WIDTH="200" BORDER="1" ALIGN="CENTER"
```

```
CELLPADDING="4" CELLSPACING="0">
```

```
<CAPTION> <H2> ПРИМЕР ТАБЛИЦЫ </H2> </CAPTION>
```

```
<TR>
```

```
<TD COLSPAN="2" ALIGN="CENTER">ЯЧЕЙКА 1</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>ЯЧЕЙКА 2</TD>
```

```
<TD>ЯЧЕЙКА 3</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD ROWSPAN="2">ЯЧЕЙКА 4</TD>
```

```
<TD>ЯЧЕЙКА 5</TD>
```

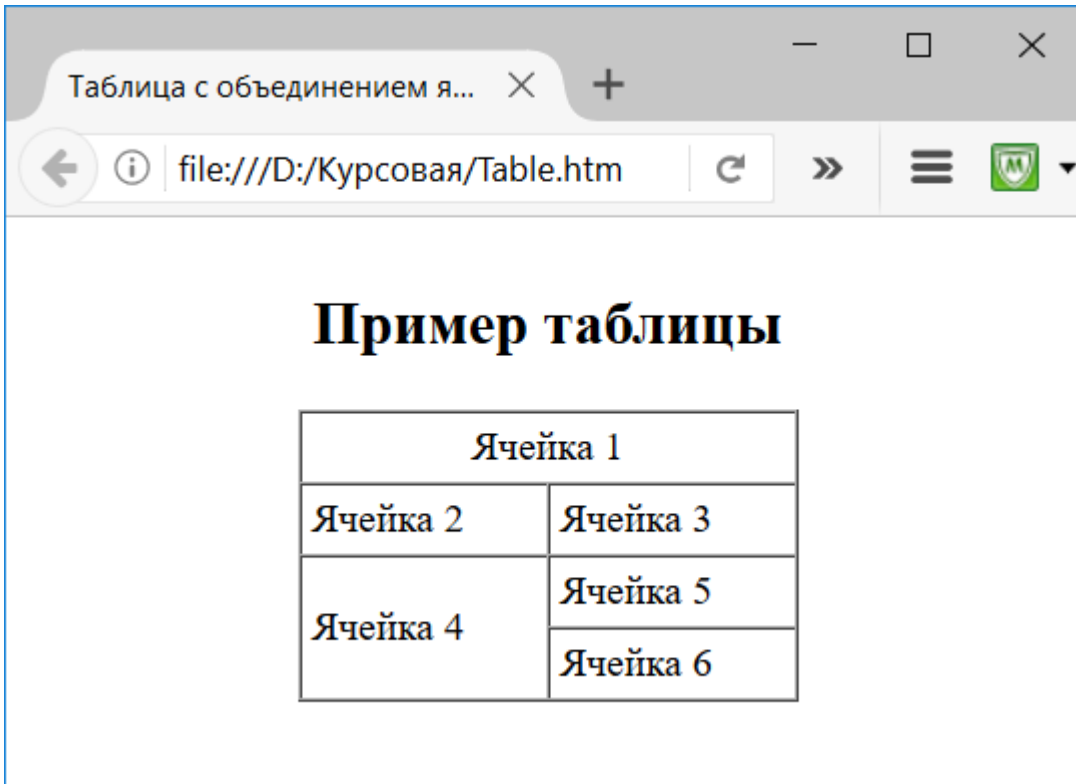
```
</TR>
```

```
<TR>
```

```
<TD>ЯЧЕЙКА 6</TD>
```


</TR>

</TABLE>



Вид приведенного

HTML-документа в окне браузера Mozilla Firefox показан на рис. 2.3.

Рис. 2.3. Пример таблицы в HTML.

Фреймы. Фреймы разбивают видимое на экране изображение на несколько частей (окон, рамок, ячеек). В каждом окне может быть размещен документ с собственным доменным адресом. Фреймы дают возможность в одном окне размещать текущую информацию, а в другом окне отображать меню. Это позволяет легко перемещаться по сайту, благодаря тому, что при прокрутке просматриваемой страницы элементы навигации остаются постоянно перед глазами.[\[54\]](#)

Заметим, что рядом расположенные фреймы могут содержать страницы, расположенные в разных папках, на разных сайтах и даже на разных серверах.

Рассмотрим синтаксис описания страницы, которая содержит фреймы. Такая страница не содержит контейнер `<BODY>... </BODY>`. Вместо него используется другой контейнер:[\[55\]](#)

```
<FRAMESET>
```

```
...
```

```
</FRAMESET>
```

При описании размеров фреймов используются абсолютные единицы измерения (пиксели), проценты и относительные единицы. Если сделать запись:

```
<FRAMESET COLS="150,10%,*">
```

то будет создано три вертикальных фрейма (три столбца, cols). При этом ширина крайнего левого столбца составит 150 пикселей, ширина среднего столбца составит 10% от оставшегося пространства, а третий столбец займет свободную справа площадь. Аналогично формируются горизонтальные фреймы.

Тег `<FRAME>` определяет свойства отдельного фрейма, на которые делится окно браузера. Этот элемент должен располагаться в контейнере `<FRAMESET>`, который задает способ разметки страницы на отдельные области. В каждую из таких областей загружается самостоятельная веб-страница, определяемая с помощью параметра SRC. Хотя обязательных атрибутов у тега `<FRAME>` и нет, рекомендуется задавать каждому фрейму его имя через атрибут NAME.[\[56\]](#)

Ниже приведен пример создания HTML-страницы с тремя фреймами, где в качестве загружаемых в фреймы файлов использованы созданные ранее документы 1.html, 2.html, Table.html. Перед тегом `<HTML>` необходимо добавить описание типа документа (DTD) для работы с фреймами:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 FRAMESET//EN"
```

```
"HTTP://WWW.W3.ORG/TR/HTML4/FRAMESET.DTD">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>FRAME</TITLE>
```

</HEAD>

<FRAMESET ROWS="48%,*" COLS="*">

<FRAME SRC="1.HTML" NAME="TOPFRAME" SCROLLING="NO">

<FRAMESET COLS="50%,*">

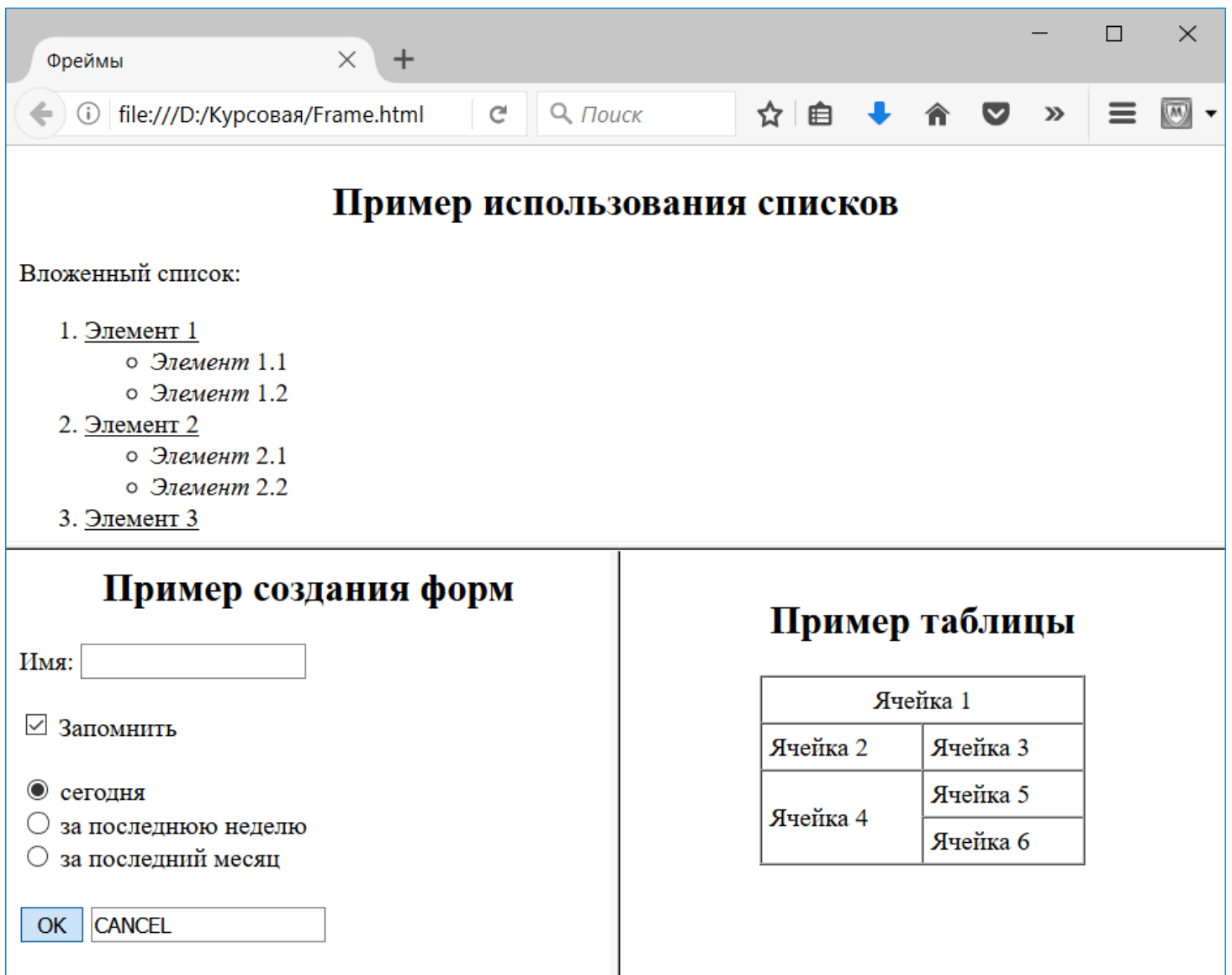
<FRAME SRC="2.HTML" NAME="LEFTFRAME">

<FRAME SRC="TABLE.HTML" NAME="MAINFRAME">

</FRAMESET>

</FRAMESET>

</HTML> [\[57\]](#)



Вид приведенного HTML-документа с тремя фреймами в окне браузера Mozilla Firefox показан на рис. 2.4.

Рис. 2.4. Пример создания фреймов в HTML.

Фреймы позволяют создавать удобные системы навигации по сайту. Однако многие специалисты Web-дизайна рекомендуют воздержаться от широкого использования фремов.[\[58\]](#)

2.2. Особенности разметки XHTML

Определение типа документа. Для документа, создаваемого с помощью языка XHTML 1.0, можно задать один из приведенных ниже вариантов дескриптора DTD:

Строгая (Strict) схема:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Переходная (Transitional) схема:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Фреймовая (Frameset) схема:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset// EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Требования к разметке XHTML Как отмечалось в первой главе, язык XHTML предполагает соблюдение строго синтаксиса XML и имеет более жесткие требования относительно того, как должна выполняться разметка:[\[59\]](#)

- Имена элементов и атрибутов следует указывать в нижнем регистре. В стандарте HTML регистр не учитывается.
- Все элементы должны быть закрыты (замкнуты). Пустые элементы закрываются добавлением слеша перед закрывающей скобкой (например, `
`, `<input />`).
- Значения атрибутов необходимо заключать в кавычки (например, `width="100px"`). Одинарные кавычки приемлемы наравне с двойными, так как они используются повсеместно. Кроме того, не должно быть никаких лишних пробельных символов (пробелов или переводов строк) до или после значения атрибута внутри кавычек.
- Все атрибуты обязаны иметь явные значения. Формат XML (и, следовательно, XHTML) не поддерживает минимизацию атрибутов, которая практиковалась языком SGML и позволяла сводить написание некоторых атрибутов к простому указанию их значений. Таким образом, если в HTML-документе можно просто написать `CHECKED`, что обозначает, что элемент формы будет активен в момент загрузки страницы, то в рамках стандарта XHTML необходимо явно указать `checked="checked"`.
- Правильное вложение документов строго обязательно. У некоторых элементов появились новые ограничения по вложению.

- Для вставки специальных символов следует указывать их символьную ссылку, (например, &ашр для символа &).
- В качестве идентификатора используется атрибут id, а не name.
- Сценарии должны располагаться в разделе CDATA, таким образом они будут рассматриваться как обычные текстовые символы и не станут анализироваться как разметка XML.

Приведем небольшой пример корректного XHTML-кода:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>XHTML</title>
</head>
<body>
<p></p>
</body>
</html> \[60\]
```

Выводы по главе 2.

Таким образом, во второй главе изложены основные принципы разметки Web-документа с помощью языков HTML и XHTML. Рассмотрены различные формы задания типа документа (DTD). Приведены теги языка HTML, определяющие заголовок, тело документа, теги форматирования текста, теги для создания заголовков различных уровней, теги для создания гиперссылок, нумерованных и маркированных списков, форм (включая поля ввода, флажки, переключатели,

кнопки), таблиц. Рассмотрена фреймовая форма задания структуры HTML-документа. Для большинства из перечисленных элементов представлены фрагменты кода с их использованием, а также показан вид документа в окне браузера Mozilla FireFox.

Описаны различные формы определения типа (DTD) для XHTML-документов. Отмечено, что язык XHTML следует строгим правилам языка XML, в связи с этим представлены требования к разметке XHTML.

Глава 3. НОВЫЕ ВОЗМОЖНОСТИ В HTML5

3.1. Структура Web-документа в HTML5

Определение типа документа. Декларация HTML5 имеет вид:[\[61\]](#)

```
<!DOCTYPE html>.
```

В языке HTML5 введен ряд новых элементов, отсутствовавших в версиях HTML и XHTML. Они приведены в таблице 1.

Таблица 1.

Новые элементы в HTML5

article	datalist	header	output	source
aside	details	hgroup	progress	summary
audio	embed	keygen	rp	time
bdi	figcaption	mark	rt	track
canvas	figure	meter	ruby	video

command footer nav section wbr

Источник: [14. , С. 225].

Помимо этого, в HTML5 добавлены новые типы ввода данных: color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url, week.

Также введен ряд новых атрибутов, в частности: contenteditable, contextmenu, draggable, dropzone, hidden, role.

В HTML5 некоторые элементы HTML 4.01 были признаны «устаревшими», так как носят презентационный характер, вышли из употребления или плохо поддерживаются браузерами. К ним относятся: acronym, applet, basefont, big, dir, font, frame, frameset, noframes, strike, tt.[\[62\]](#)

Рассмотрим пример создания структуры документа с использованием новых в HTML5 элементов article, nav, header, footer, aside.

Тег <article> используется для оформления независимых работ: журнальных и газетных статей, сообщений в блогах, комментариев или других элементов, которые можно извлечь для внешнего использования. Тег <nav> задает основную навигацию на странице. Теги <header> и <footer> задают верхний и нижний колонтитулы страницы, раздела или статьи. Тег <aside> используется для оформления информации во врезках, определяет контент, косвенно связанный с окружающим содержимым.[\[63\]](#)

Приведем документ, иллюстрирующий использование перечисленных выше элементов для задания структуры страницы:[\[64\]](#)

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Структура страницы</title>

</head>
```



```
<body>

<header>

<h2>Навигация по сайту</h2>

</header>

<nav>

<h3>Меню</h3>

<a href="1.html">Списки</a> | <a href="2.html">
Формы</a> | <a href="Table.html">Таблицы</a>

</nav>

<article>

<h2>HTML 4.01</h2>

<p>Рекомендован W3C к применению в декабре 1999 года</p>

</article>

<article>

<h2>HTML5</h2>

<p>Рекомендован W3C к применению в октябре 2014 года</p>

</article>

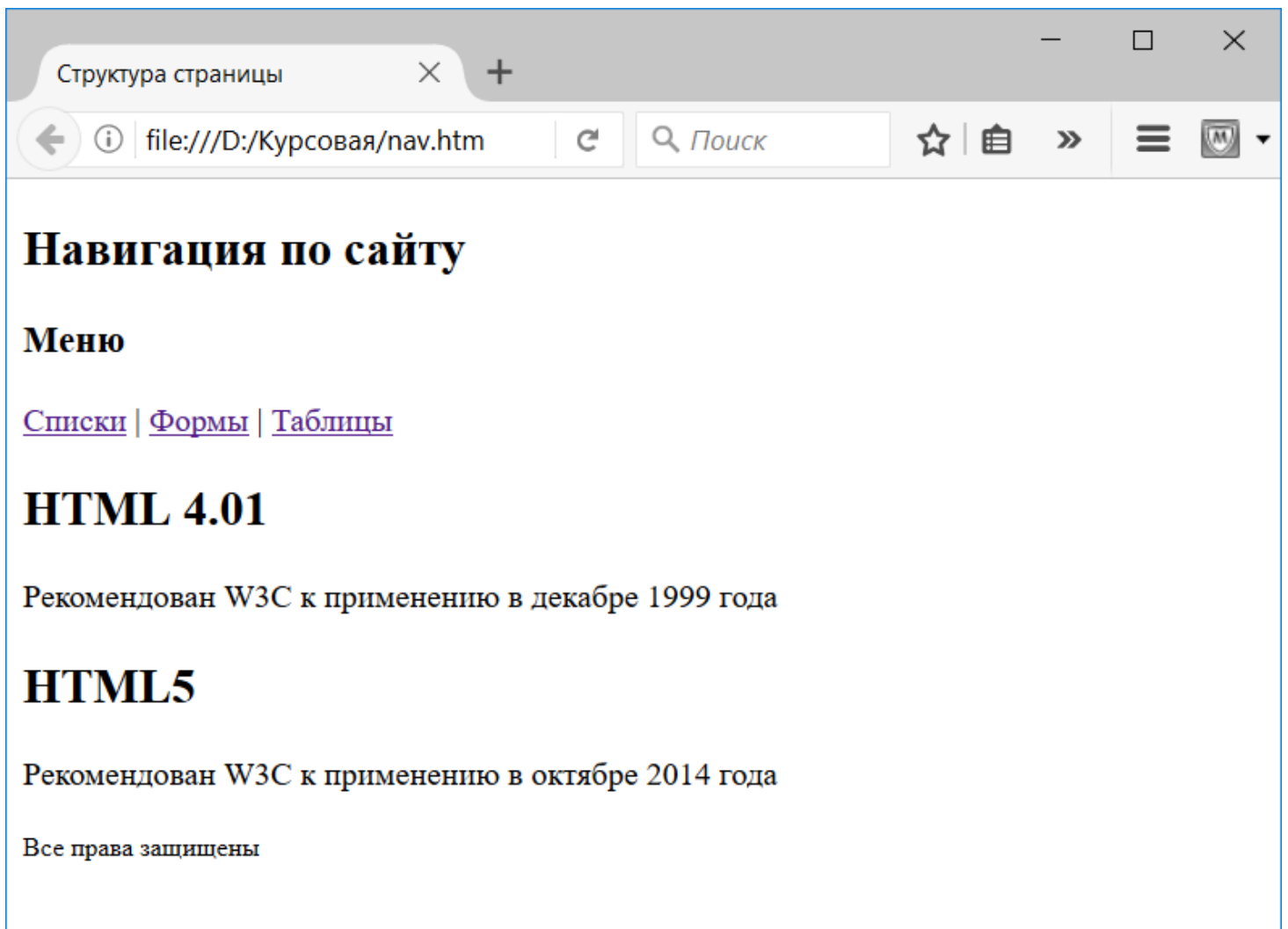
<footer>

Все права защищены

</footer>

</body>

</html>
```



Вид данного документа в окне браузера в окне браузера Mozilla Firefox показан на рис. 3.1.

Рис. 3.1. Пример задания структуры страницы в HTML5

3.2. Новые типы ввода данных

Во второй главе нами был рассмотрен пример создания в HTML 4.01 формы с помощью тега `<form>` и основных элементов управления с помощью тега `<input>`. В HTML5 добавлены типы ввода данных в формы путем задания новых значений атрибута `type`. Приведем их:[\[65\]](#)

`<input type=email>` — создает поле для ввода адреса электронной почты; сообщает браузеру, что форма не должна быть отправлена, пока пользователь не ввел нечто, похожее на правильный адрес, то есть браузер определяет правильность его формата;[\[66\]](#)

`<input type=url>` — создает поле для ввода URL-адреса и заставляет браузер проверять, что пользователь ввел правильный по структуре URL-адрес;

`<input type=date>` — создает поле для ввода даты, а также открывает доступ к встроенному в браузер виджету выбора даты;

`<input type=number>` — создает поле для ввода числа и заставляет браузер проверять, что пользователь ввел именно число;

`<input type=range>` — создает ползунок с тем, чтобы пользователь мог выбрать значение при помощи ползунка;

`<input type=color>` — создает поле для задания цвета, позволяет пользователю ввести значение цвета, выбрав нужный вариант в палитре.

Рассмотрим также совместное использование тега `<input>` и нового элемента `<datalist>`. Сочетание элемента `<input>` с атрибутом `list` и элементом `<datalist>` дает комбинированное поле — комбинацию раскрывающегося списка и текстового поля из одной строки. Оно позволяет пользователям вводить свой собственный текст, если их не устраивают предложенные в списке варианты.[\[67\]](#)

Список создается внутри нового элемента `<datalist>`, `id` которого указывается в качестве значения атрибута `list`. Сам `<datalist>` не отображается, но появляется в виде значений поля, похожего на поле выбора.

Ниже приведены фрагменты кода HTML5 для создания формы, включающей перечисленные выше значения атрибута `type` тега `<input>`, а также совместное использование тега `<input>` и элемента `<datalist>`.

```
<form>
```

```
<p>Заполните форму (все поля обязательны)</p>
```

```
<p>Имя: <input name="name" required></p>
```

```
<p>e-mail: <input type="email" name="email" required></p>
```

```
<p>Сайт: <input type="url" name="site" required></p>
```

```
<p><input type="submit" value="Отправить"></p>
```

```
</form>
```

<p>Выберите язык разметки:</p>

<p><input list="character">

<datalist id="character">

<option value="HTML 4.01"></option>

<option value="XHTML 1.0"></option>

<option value="HTML5"></option>

</datalist></p>

<form>

<p>Введите данные:</p>

<p>Дата <input type="date"></p>

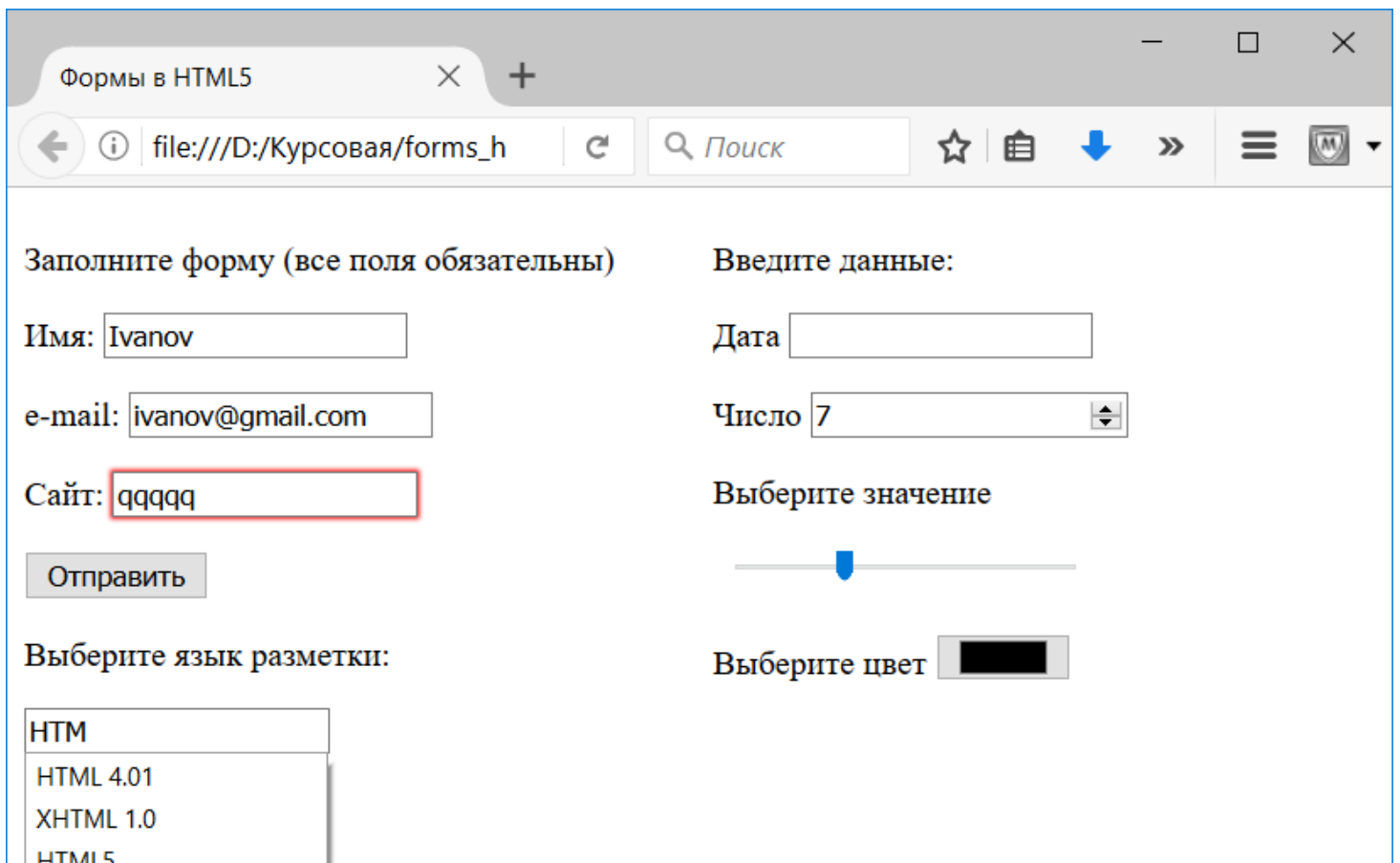
<p>Число <input type="number"></p>

<p>Выберите значение</p>

<p><input type="range"></p>

<p>Выберите цвет <input type="color"></p>

</form> [\[68\]](#)



Вид документа в окне браузера показан на рис. 3.2.

Рис. 3.2. Использование новых типов ввода данных в HTML5.

3.3. Размещение аудио- и видеоконтента на странице

В самом простом случае для размещения видео на странице в HTML5 потребуется такой код:[\[69\]](#)

```
<video src=video.webm></video>
```

Расширение файла webm используется здесь, чтобы указать на формат видео, закодированного с помощью WebM.

Для предоставления пользователю возможности управлять проигрыванием видео необходимо добавить в тег <video> атрибут controls. Для задания размера видео в окне браузера, необходимо добавить в тег <video> атрибуты width и height, которые задают ширину и высоту отображаемого видео в пикселях.

Аналогично, для добавления на страницу аудио используется код:

```
<audio src=audio.ogg controls></audio>,
```

в котором атрибут controls указывает на то, что в браузере будет отображаться панель проигрывания аудио.

Следует заметить, что современные браузеры поддерживают не все форматы аудио- и видеофайлов. Наиболее употребительными для аудиофайлов являются следующие форматы: ogg/vorbis, wav, mp3 и AAC; для видеофайлов — ogg/theora, H.264, WebM. Для универсального воспроизведения в разных браузерах аудио и видео кодируют с помощью разных кодеков и добавляют несколько файлов одновременно через тег <source>.

Приведем фрагмент кода для вставки аудио и видео в документ HTML:[\[70\]](#)

```
<h2>Вставка аудио- и видео</h2>
```

```
<h4>Слушаем Моцарта</h4>
```

```
<audio controls>
```

```
<source src="mozart.ogg" type="audio/ogg; codecs=vorbis">
```

```
</audio>
```

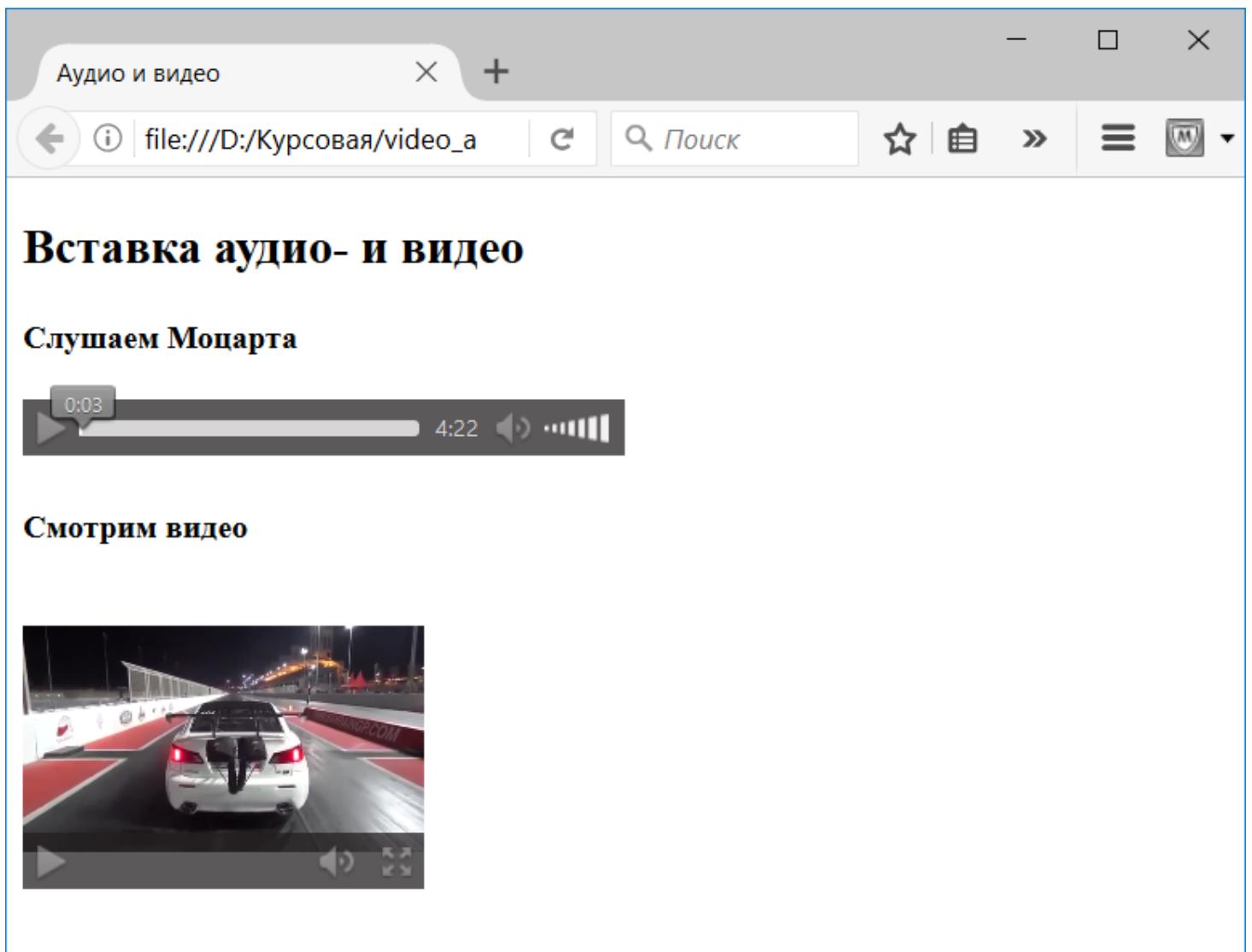
```
<h4>Смотрим видео</h4>
```

```
<video width="200" height="150" controls="controls">
```

```
<source src="Lexus.webm" type='video/webm;
```

```
codecs="vp8, vorbis">
```

```
</video>
```



Вид документа в окне браузера Mozilla Firefox показан на рис. 3.3.

Рис. 3.3. Пример размещения аудио и видео на странице.

Выводы по главе 3.

Таким образом, в третьей главе рассмотрены некоторые новые возможности, введенные в HTML5. Изучены элементы, предназначенные для формирования структуры документа: `article`, `nav`, `header`, `footer`, `aside`. Рассмотрены новые типы, предназначенные для ввода на странице и проверки корректности в браузере таких данных, как имя, электронный адрес, URL-адрес, дата, число, число из заданного диапазона, цвет и др. Рассмотрено добавление в форму текстового поля с раскрывающимся списком-подсказкой.

Изучены теги для размещения на странице аудио- и видеоконтента. Представлены фрагменты кода, использующие перечисленные новые элементы, а также показано их отображение в браузере Mozilla FireFox.

ЗАКЛЮЧЕНИЕ

В данной курсовой работе изучены языки разметки гипертекста, составляющие основу создания современных Web-документов и Internet-приложений. В процессе выполнения курсовой работы рассмотрены понятия гипертекста, гиперссылки, гипертекстовой системы, навигации, положившие основу для создания первых языков разметки гипертекста и появления Всемирной паутины. Изучена эволюция версий языков HTML и XHTML в 1991 – 2009 гг под руководством Консорциума W3C и появление нового стандарта HTML5 — результата совместных усилий W3C и ведущих компаний-разработчиков современных браузеров.

Далее были изучены основные принципы разметки Web-документов с помощью языка HTML 4.01. Рассмотрены теги, задающие определение типа документа, заголовков и тело документа; теги форматирования текста, создания гиперссылок; теги, предназначенные для создания списков, таблиц, форм, полей ввода, флажков, переключателей, кнопок. Изучена фреймовая форма HTML-документа. Рассмотрены основные требования к разметке XHTML-документов.

В последней главе работы изучены новые возможности, предоставляемые в языке разметки HTML5. Рассмотрены новые теги для создания структуры документа, новые типы ввода данных, активно использующиеся в современных виджетах. Изучены теги для размещения на странице аудио- и видеоконтента.

В ходе выполнения работы были созданы HTML-документы, демонстрирующие использование тех или иных тегов, и представлен их вид в окне браузера Mozilla Firefox.

Следует отметить, что язык гипертекстовой разметки HTML5, благодаря своей простоте, корректной обработке браузерами и широким возможностям использования в современных Internet-технологиях, заслуженно является на сегодняшний день рекомендованным стандартом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алексеев А.П. Введение в Web-дизайн: учебное пособие – М.: СОЛОН-ПРЕСС, 2008. – 185 с.
2. Беляев М.И. Принцип реализации структуры гипертекста – один из основных принципов создания электронных учебников // Вестн. РУДН. Сер. Информатизация образования. – 2012. – № 3. – С. 71-82.
3. Гаврилов М.В., Климов В.А. Информатика и информационные технологии: учебник для СПО / 4-е изд. – М.: Издательство Юрайт, 2015. – 383 с.
4. Гасов В.М., Цыганенко А.М. Методы и средства подготовки электронных изданий Учебное пособие. – М.: МГУП, 2001. URL: <http://www.hi-edu.ru/e-books/xbook081/01/> (Дата обращения: 15.03.2017)
5. Дронов В.А. HTML5, CSS3 и Web 2.0. Разработка современных Web-сайтов. – СПб.: БХВ-Петербург, 2011. – 416 с.
6. Дунаев В.В. HTML, скрипты и стили / 3-е изд. – СПб.: БХВ-Петербург, 2011. – 816 с.
7. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – 544 с.
8. Информационные технологии: Учебник для вузов / Б.Я. Советов, В.В. Цехановский. – М.: Высш. шк., 2006. – 263 с.
9. Кирсанов Д. Веб-дизайн. – СПб.: Символ-Плюс, 2015. – 368 с.
10. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста / 2-е изд. – СПб.: Питер, 2012. – 304 с.
11. Могилев А.В., Листрова Л.В. Технологии обработки текстовой информации. Технологии обработки графической и мультимедийной информации. – СПб.: БХВ-Петербург, 2010. – 304 с.
12. Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство / 6-е изд. – СПб.: Символ-Плюс, 2008. – 752 с.
13. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов / 4-е изд. – СПб.: Питер, 2010. – 944 с.
14. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / 4-е изд. – М.: Эксмо, 2014. – 528 с.
15. Соболева О.В. Понятие и основные характеристики гипертекста // Альманах современной науки и образования.– 2011. – №12(55). – С. 163-164.
16. Субботин М.М. Гипертекст. Новая форма письменной коммуникации // Итоги науки и техники. Сер. Информатика. – М.: ВИНТИ, 1994. – Т. 18. – С. 22-36.
17. Холмогоров В. Основы веб-мастерства: учебный курс. – СПб.: Питер, 2003. – 316 с.

18. Эпштейн В.Л. Введение в гипертекст и гипертекстовые системы. URL: <http://www.lingvolab.chat.ru/library/hypertext.htm> (Дата обращения: 15.03.2017)
19. HTML (Справочник по HTML). URL: <http://htmlbook.ru/html>
20. HTML-2.0. URL: https://www.w3.org/MarkUp/html-spec/html-spec_2.html
21. HTML 3.2. Reference Specification URL: <https://www.w3.org/TR/REC-html32>
22. HTML 4.0 Specification. URL: <https://www.w3.org/TR/1998/REC-html40-19980424/>
23. HTML 4.01 Specification. URL: <https://www.w3.org/TR/html401/>
24. HTML5. URL: <https://www.w3.org/TR/html5/>
25. HTML5.1. URL: <https://www.w3.org/TR/html51/>
26. XHTML 1.0. The Extensible HyperText Markup Language. URL: <https://www.w3.org/TR/xhtml1/>
27. XHTML 1.1 - Module-based XHTML. URL: <https://www.w3.org/TR/xhtml11/>
28. XHTML 2.0. URL: <https://www.w3.org/TR/xhtml2/>

1. Цит. по Соболева О.В. Понятие и основные характеристики гипертекста // Альманах современной науки и образования.- 2011. - №12(55). - С. 163. [↑](#)
2. Цит. по Субботин М.М. Гипертекст. Новая форма письменной коммуникации // Итоги науки и техники. Сер. Информатика. - М.: ВИНТИ, 1994. - Т. 18. - С. 22-36. [↑](#)
3. Цит. по Эпштейн В.Л. Введение в гипертекст и гипертекстовые системы. URL: <http://www.lingvolab.chat.ru/library/hypertext.htm> (Дата обращения: 15.03.2017) [↑](#)
4. Цит. по Эпштейн В.Л. Введение в гипертекст и гипертекстовые системы. URL: <http://www.lingvolab.chat.ru/library/hypertext.htm> (дата обращения: 15.03.2017) [↑](#)
5. Информационные технологии: Учеб. для вузов / Б.Я. Советов, В.В. Цехановский. - М.: Высш. шк., 2006. - С. 88. [↑](#)
6. Беляев М.И. Принцип реализации структуры гипертекста - один из основных принципов создания электронных учебников // Вестн. РУДН. Сер. Информатизация образования. - 2012. - № 3. - С. 74. [↑](#)

7. Соболева О.В. Понятие и основные характеристики гипертекста // Альманах современной науки и образования.– 2011. – №12(55). – С. 164. [↑](#)
8. Соболева О.В. Понятие и основные характеристики гипертекста // Альманах современной науки и образования.– 2011. – №12(55). – С. 164. [↑](#)
9. Эпштейн В.Л. Введение в гипертекст и гипертекстовые системы. URL: <http://www.lingvolab.chat.ru/library/hypertext.htm> (дата обращения: 15.03.2017) [↑](#)
10. Кирсанов Д. Веб-дизайн. – СПб.: Символ-Плюс, 2015. – С. 5. [↑](#)
11. Могилев А.В., Листрова Л.В. Технологии обработки текстовой информации. Технологии обработки графической и мультимедийной информации. – СПб.: БХВ-Петербург, 2010. – С. 146-147. [↑](#)
12. Гасов В.М., Цыганенко А.М. Методы и средства подготовки электронных изданий Учебное пособие. – М.: МГУП, 2001. URL. <http://www.hi-edu.ru/e-books/xbook081/01/> (Дата обращения: 15.03.2017) [↑](#)
13. Кирсанов Д. Веб-дизайн. – СПб.: Символ-Плюс, 2015. – С. 7. [↑](#)
14. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / 4-е изд. – М.: Эксмо, 2014. – С. 39. [↑](#)
15. Алексеев А.П. Введение в Web-дизайн: учебное пособие – М.: СОЛОН-ПРЕСС, 2008. – С. 17. [↑](#)
16. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов / 4-е изд. – СПб.: Питер, 2010. – С. 802. [↑](#)
17. Могилев А.В., Листрова Л.В. Технологии обработки текстовой информации. Технологии обработки графической и мультимедийной информации. – СПб.: БХВ-Петербург, 2010. – С. 150. [↑](#)

18. Могилев А.В., Листрова Л.В. Технологии обработки текстовой информации. Технологии обработки графической и мультимедийной информации. – СПб.: БХВ-Петербург, 2010. – С. 150. [↑](#)
19. Hypertext Markup Language - 2.0. URL: https://www.w3.org/MarkUp/html-spec/html-spec_2.html [↑](#)
20. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / 4-е изд. – М.: Эксмо, 2014. С. 220. [↑](#)
21. HTML 3.2 Reference Specification URL: <https://www.w3.org/TR/REC-html32/>, [↑](#)
22. HTML 4.0 Specification. URL: <https://www.w3.org/TR/1998/REC-html40-19980424/> [↑](#)
23. HTML 4.01 Specification. URL: <https://www.w3.org/TR/html401/> [↑](#)
24. Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство / 6-е изд. – СПб.: Символ-Плюс, 2008. – С. 660. [↑](#)
25. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / 4-е изд. – М.: Эксмо, 2014. С. 220. [↑](#)
26. XHTML 1.0. The Extensible HyperText Markup Language. URL: <https://www.w3.org/TR/xhtml1/> [↑](#)
27. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста / 2-е изд. – СПб.: Питер, 2012. – С. 17. [↑](#)
28. XHTML 1.1 - Module-based XHTML. URL: <https://www.w3.org/TR/xhtml11/> [↑](#)
29. Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство / 6-е изд. – СПб.: Символ-Плюс, 2008. – С. 582. [↑](#)

30. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / 4-е изд. – М.: Эксмо, 2014. С. 221. [↑](#)
31. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста / 2-е изд. – СПб.: Питер, 2012. – С. 17. [↑](#)
32. XHTML 2.0. URL: <https://www.w3.org/TR/xhtml2/> [↑](#)
33. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста / 2-е изд. – СПб.: Питер, 2012. – С. 18. [↑](#)
34. HTML5. URL: <https://www.w3.org/TR/html5/> [↑](#)
35. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / 4-е изд. – М.: Эксмо, 2014. С. 223. [↑](#)
36. HTML 5.1. URL: <https://www.w3.org/TR/html51/> [↑](#)
37. Дронов В.А. HTML5, CSS3 и Web 2.0. Разработка современных Web-сайтов. – СПб.: БХВ-Петербург, 2011. –С. 12 [↑](#)
38. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 483. [↑](#)
39. Дунаев В.В. HTML, скрипты и стили / 3-е изд. – СПб.: БХВ-Петербург, 2011. – С. 24. [↑](#)
40. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 483. [↑](#)
41. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 484. [↑](#)

42. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 485. [↑](#)
43. Пример частично взят из Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 486. [↑](#)
44. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 487. [↑](#)
45. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 488. [↑](#)
46. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 488. [↑](#)
47. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 489. [↑](#)
48. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 489. [↑](#)
49. Пример частично взят из Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров, А.А. Васильев и др.: Под ред. В.Н. Петрова. – СПб.: Питер, 2011. – С. 490. [↑](#)
50. Холмогоров В. Основы веб-мастерства: учебный курс. – СПб.: Питер, 2003. – С. 147. [↑](#)
51. Холмогоров В. Основы веб-мастерства: учебный курс. – СПб.: Питер, 2003. – С. 147. [↑](#)
52. Холмогоров В. Основы веб-мастерства: учебный курс. – СПб.: Питер, 2003. – С. 148. [↑](#)

53. HTML (Справочник по HTML). URL: <http://htmlbook.ru/html/td/colspan> (Дата обращения 20.03.2017) [↑](#)
54. Алексеев А.П. Введение в Web-дизайн: учебное пособие – М.: СОЛОН-ПРЕСС, 2008. – С. 81. [↑](#)
55. Алексеев А.П. Введение в Web-дизайн: учебное пособие – М.: СОЛОН-ПРЕСС, 2008. – С. 82. [↑](#)
56. HTML (Справочник по HTML). URL: <http://htmlbook.ru/html/frame> (Дата обращения 20.03.2017) [↑](#)
57. Пример частично взят из <http://htmlbook.ru/html/frameset> (Дата обращения 20.03.2017) [↑](#)
58. Алексеев А.П. Введение в Web-дизайн: учебное пособие – М.: СОЛОН-ПРЕСС, 2008. – С. 84. [↑](#)
59. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / 4-е изд. – М.: Эксмо, 2014. С. 221 [↑](#)
60. Пример частично взят из HTML (Справочник по HTML). URL: <http://htmlbook.ru/html> [↑](#)
61. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / 4-е изд. – М.: Эксмо, 2014. С. 223. [↑](#)
62. Там же, С. 225. [↑](#)
63. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / 4-е изд. – М.: Эксмо, 2014. С. 103. [↑](#)

64. Пример частично взят из Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста / 2-е изд. – СПб.: Питер, 2012. – С. 33. [↑](#)
65. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста / 2-е изд. – СПб.: Питер, 2012. – С. 101. [↑](#)
66. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста / 2-е изд. – СПб.: Питер, 2012. – С. 103. [↑](#)
67. Там же, С. 108. [↑](#)
68. Пример частично взят из Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста / 2-е изд. – СПб.: Питер, 2012. – С. 102-108. [↑](#)
69. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста / 2-е изд. – СПб.: Питер, 2012. – С. 131. [↑](#)
70. Пример частично взят из HTML (Справочник по HTML). URL: <http://htmlbook.ru/html/audio>, <http://htmlbook.ru/html/video> [↑](#)