

Содержание:

ВВЕДЕНИЕ

Объём и скорость обработки данных в современных компьютерных сетях обеспечило возможность создания и помещения в компьютерные сети Web-сайты и Web-страницы различного вида и направленности. Что повысило количество пользователей сети.

В связи с данным развитием компьютерных сетей для пользователей сети стали популярны языки гипертекстовой разметки, при помощи которых можно самому своими силами создать свой Web-сайт. К таким языкам относятся: PHP, HTML, HTML5, JavaScript и т.д.

Актуальность работы заключается в необходимости изучать современные информационные технологии, а именно процесс создания сайта при помощи языков гипертекстовой разметки.

Цель работы состоит в описании характеристики языков гипертекстовой разметки, изучить теоретический и практический материал данной темы курсовой работы.

Объект курсового исследования: популярные языки гипертекстовой разметки. Сайт интернет

Для достижения поставленной цели были выделены следующие задачи:

- изучить теоретическую практическую литературу по теме курсовой работы;
- описать языки гипертекстовой разметки;
- описать спецификации языков гипертекстовой разметки;
- привести пример практического применения HTML (на примере обучающего сайта).

Структура курсовой работы: работа состоит из введения, трех глав, заключения, списка литературы, включающего в себя 25 источников и двух приложений.

Первая глава посвящена изучению основных понятий касающихся технологии Web.

Вторая глава посвящена изучению технологии Web на основе языков гипертекстовой разметки: HTML, XML, XHTML.

В третьей главе рассматриваются практическое применение языка разметки гипертекста HTML.

В заключении описаны результаты проведённого исследования.

Курсовой проект может быть интересен преподавателям, инженерам-программистам, студентам и всем тем, кто интересуется Web-программированием.

Глава 1. Технология Web

Понятие «технологии Web»

Технология Web сформировалась в ходе процесса создания и совершенствования системы World Wide Web, данная технология оказывает влияние на ход развития разных видов информационных технологий. На её основе развиваются способы и методы для интегрирования неоднородных ресурсов в информационные системы. Разрабатываются новые способы создания распределённых систем. В ходе развития информационно-коммуникационной технологии Web-технологиям нашли применение в разных сферах деятельности, к примеру, в электронном бизнесе и электронных библиотеках. Также при помощи Web-технологий стало возможно создавать крупные научные и образовательные системы [1, 45].

Для дальнейшего развития технологии Web требуются радикальные и новые подходы. Нахождение способов для реализации новых подходов является главной задачей консорциума W3C (World Wide Web Consortium), начиная ещё с середины 90-х годов.

Термин «Web» в разных энциклопедиях и словарях имеет следующее определение: web – это глобальное информационное пространство, состоящее из связанных между собой гипертекстовыми ссылками документов, страниц и файлов, размещённых на физической инфраструктуре Internet, обмен данными в которой осуществляется при помощи протоколов передачи данных [3, 34].

Система «Web» имеет огромные интенсивно наращиваемые информационные ресурсы, большая часть, которых предоставлена в свободном доступе в любой отрезок времени. Созданная среда WWW способна взаимодействовать с ресурсами других информационных сервисов Internet, таких как Gopher, FTP, Archie, WAIS,

Telnet, электронная почта. Технология «Web» также обеспечивает доступ к базам данных через сеть коммуникаций. К наиболее часто используемым сервисам Internet относятся: WWW, электронная почта, сервисы передачи файлов - FTP, поддержка телеконференций, удаленный доступ к вычислительным ресурсам [2, 21].

Информационные сервисы Internet построены согласно архитектуре «клиент-сервер». Некоторые из этих сервисов, например WWW, могут поддерживать работу с распределенными информационными ресурсами [4, 32].

В большом количестве энциклопедий и словарей под Web - технологией понимается «технология, по которой была построена Всемирная паутина, позволяющая размещать и отображать разного вида документы, находящиеся в Интернете и связанные между собой системой ссылок»[2, 74].

Современные информационные Web технологии быстро преобразили современный мир и оказывают на него непосредственное влияние. Web-технологии стали применяться не только в бизнесе, но и в частной и в профессиональной жизни. Новейшие Web-технологии применяются в сферах жизни общества, изменяя способы общения и принцип ведения Web-проектов в современных компаниях. В быту и в бизнесе для переписки и торговли между людьми и организациями используются Web-технологии, создаются собственные Web-узлы, где предлагается информация, товары или услуги. Средства для создания Web-ресурсов развиваются довольно стремительно, позволяя создавать пользователям сложные Web-документы, без специальных знаний о структуре и внешнем виде Web-документа, позволяя пользователям тратить время на продуктивную творческую деятельность. Главное преимущество Web-технологий в современных условиях заключается в их простоте и как следствие в последующем повышении эффективности их применения.

1.2. Обзор языков программирования для Web

Самым распространенным среди языков гипертекстовой разметки является язык тегов HTML. Также стал пользоваться популярностью язык скриптов PHP. Менее популярными являются языки: ASP, GGI, Perl, SSI, ASP. XML, XHTML и DHTML к языкам веб-программирования не относятся, но служат хорошими дополнениями к ним. Данные языки, несмотря на свою непопулярность, представляют собой довольно хорошее средство для создания Web-ресурсов. Среди скриптовых языков можно

выделить только язык JavaScript и язык VBScript [6, 51].

HTML - это язык, предназначенный для логической разметки электронного документа. В этом языке довольно удобно и легко работать, именно поэтому пользователи и программисты предпочитают его. Во время использования HTML данные о тексте, картинке, эффекте, скрипте, расположенном элементе записываются при помощи языка тегов в один документ, в котором не содержится ничего кроме текста, браузером данная информация читается, в результате чего формируется веб-страница. Каждая деталь такого веб-документа помещается в отдельный тег и весь веб-документ сохраняется в формате.html. Для визуального оформления веб-документа используются коды CSS (стили) и скрипты JavaScript [7, 37].

Среди других языков гипертекстовой разметки наиболее популярным стал язык PHP. При помощи этого языка создаются довольно сложные веб-ресурсы, которые существенно отличаются от страниц, созданных при помощи языка HTML. Используя язык PHP можно создать веб-страницы, на которых будет отображаться информация о количестве посетителей сайта и их активности, создавать на сайте гостевые книги, чат, форум, счетчик, информационную систему и т.д. PHP может взаимодействовать со многими базами данных [7, 43].

Среди сложных языков гипертекстовой разметки можно выделить язык CGI. При помощи языка CGI можно создать интерактивное и динамическое интернет-приложение. CGI – представляет собой интерфейс, в котором позволяет веб-серверу согласно запросам браузера запускать и выполнять интернет-приложения, а результат их работы отправлять на сервер или отобразить в браузере.

Perl – является самым распространенным языком веб-программирования для приложений CGI. Он довольно удобен для ведения системного администрирования, сетевого программирования и других задач.

XML, DHTML и XHTML являются чем-то вроде дополнительных компонентов к HTML. Они нужны, чтобы сделать стандартную HTML страницу более динамической с интерактивными возможностями. Так как язык HTML довольно ограничен, и в нём трудно создавать интерактивные элементы.

Вывод по первой главе: Технология Web и её языки программирования созданы для того обеспечить возможность пользователям формировать Web-страницы и добавлять на них различные текстовые и графические данные с последующим их выкладыванием в сеть Интернет.

Глава 2. Язык гипертекстовой разметки

2.1 Язык гипертекстовой разметки HTML

Язык HTML (HyperText Markup Language, язык разметки гипертекста) относится к группе языков гипертекстовой разметки (markup languages). В данном определении «разметка» является общей служебной информацией, которая не выводится во время отображения документа, но именно она определяет, как будут выглядеть фрагменты документа. Например, в разметке можно задать параметр, чтобы нужные слова выделялись жирным или курсивом, или, к примеру, выделить отдельный абзац при помощи особого шрифта или оформить заголовок с увеличенным шрифтом [8, 36].

Любой язык разметки служит для решения двух важных задач:

- 1) определение синтаксиса разметки;
- 2) определение смысла разметки.

Язык разметки HTML был создан и был рекламирован как дополнение к языку SGML. В 1974 году Чарльзом Голдфарбом язык HTML после значительных доработок был принят как официальный стандарт ISO, а SGML (Standard Generalized Markup Language) стал представлять собой метаязык – который служит как система для описания других языков разметки [9, 38].

Сам стандарт SGML появился из-за необходимости создать среду, в которой можно совместно пользоваться данными разных приложений и операционных системам. В ходе анализа недостатков, которые имели многие нестандартные языки разметки, три учёных из компании IBM - Чарльз Гольдфарб (Charles Goldfarb), Эд Мошер (Ed Mosher) и Рэй Лори (Ray Lorie) - создали три общих принципа, следуя которым можно обеспечивать возможности для совместной работы с документами в среде разных операционных систем [9, 45]:

- 1) Использование единого принципа форматирования во всех программах, которые выполняют обработку документов. Должен быть в наличии единый набор синтаксической конструкции и общей семантики, что сможет заметно упростить взаимодействие между программами.

2) Специализация языков форматирования. Используя специализированные языки на базе наборов стандартных правил, программист перестал зависеть от внешней реализации и представления о потребности конечного пользователя.

3) Чёткое определение формата документа. Созданы правила, которые строго определяют формат документа, определяют количество и принцип маркировки языковой конструкции, используемой в документе. Применяя стандартный формат, пользователь может теперь точно знать структуру документа.

На этих трёх правилах был основан предшественник SGML - GML (Generalized Markup Language). Изучение и разработка GML продолжалось около 10 лет, до тех пор, пока согласно соглашению, которое было заключено международной группой разработчиков, был создан новый стандарт SGML [10, 57].

Стандарты языка HTML, который теперь является одним из ключевых стандартов Web, разрабатывались и поддерживались консорциумом W3C. Основал этот международный консорциум Тим Бернес-Ли (Tim Berners-Lee). Консорциум создаёт стандарты форматирования и занимается разработкой семантической сети (Semantic Web). При помощи средств языка HTML производится форматная разметка веб-документов, определяется гиперсвязь между документами или их фрагментами [11, 54].

В качестве среды для написания кода HTML используется обычный текстовый файл. В результате гипертекстовые базы данных в концепции WWW – представляют собой набор текстовых файлов, язык HTML отвечает за размещение представленной информации и структуры связи между файлами и другими ресурсами на гипертекстовом документе [11, 76].

Язык HTML позволяет производить разметку электронного документа, помещая в документ разнообразные метки, текст, изображения, аудио- и видеoinформацию. В составе языка имеются разные средства, чтобы создавать различные уровни заголовков, выделений шрифта, различные виды списков, таблиц и многих других элементов.

Гипертекстовые ссылки, служащие для установки связи между текстовыми документами, стали использоваться для объединения самых различных информационных ресурсов, в том числе звука и видео. Используя систему гиперссылок HTML можно построить систему с взаимосвязанными документами по различным критериям. Язык HTML содержит команды (теги), используемые для задания формы и размера шрифтов, задания размера и расположения

изображений, теги позволяют задавать переходы от фрагментов текста или изображений к другим html – документам, создавая гипертекстовую ссылку. Документ в html - формате является текстовым файлом, который содержит в себе все необходимые сведения о выводимых на экран данных. Чтобы управлять сценариями просмотра страниц на странице веб-сайта, можно воспользоваться языками для программирования сценариев, например, языками JavaScript, Java и VBScript.

Для создания форм, в которые пользователь может вводить данные, для обработки используются специальные серверные программы, написанные на языках PHP или Perl. При помощи языка HTML можно помещать на веб-страницы гипертекстовую ссылку и интерактивную кнопку, чтобы соединять Web-страницы с другими страницами на Web-сайте [10, 54].

Новая версия языка HTML уже не будет создаваться, но для дальнейшего развития HTML был создан расширенный язык разметки гипертекста XHTML (Extensible Hypertext Markup Language). XHTML по своим возможностям похож на HTML, но XHTML имеет более строгие требования к синтаксису кода. Как и HTML, XHTML тоже является подмножеством языка SGML, но XHTML, в отличие от предшественника, имеет больше соответствие со спецификациями XML. Вариант XHTML 1.0 был одобрен и внедрён Консорциумом Всемирной паутины (W3C) 26 января 2000 года [12, 61].

2.2 Расширяемый язык разметки XML

Во второй половине 90-х годов прошлого века началось технологическое переоснащение технологии Web. Среди языков гипертекстовой разметки стал играть важную роль разработанный в консорциуме W3C новый язык гипертекстовой разметки XML. Язык XML (Extensible Markup Language) – это расширяемый язык разметки, представляющий собой метаязык, который определяет процедуру создания языков разметки для специфических целей [12, 51].

Главной сферой применения стандарта платформы XML стали слабоструктурированные данные[1] Web-сайтов в формате XML-документов. Использование языка XML в данной области дало возможность представлять в Web-среде гипермедиа страницы в формате XML-документов и осуществлять поддержку связанных с ними метаданных[2]. Главной целью создания платформы XML

является добавление в Web-среду метаданных, которые нужны для описания свойств поддерживаемых информационных ресурсов[3].

Метаданные нужны для того, чтобы создавать принципиально новые высокоуровневые Web-приложения, основой которых является интеграция с информационными технологиями, занимающиеся интеграцией с неоднородными информационными ресурсами. На рисунке 1 иллюстрируется упрощенная архитектура системы, в которой метаданные обеспечивают интеграцию неоднородных информационных ресурсов.

Простейшей возможностью для задания семантики – является использование пространственных имён. В отличие от языка HTML, который обеспечивает форматную разметку текста, определяя его внешний вид на экране, XML используется для структурной разметки [13, 39].

Структурная разметка в XML позволяет выделить в тексте кода содержательную структурную единицу, данные единицы называются элементами XML-документа. Чтобы выделить каждый тип элемента используется тег, который указывает имя типа элемента. В результате с каждый тег может ассоциироваться семантикой соответствующего элемента XML-документа, к примеру с адресом организации, номер телефона и т. д. [13, 45].

Приложение WEB

Реляционное приложение

Механизм доступа XML

Сервер баз данных с расширением для XML

XML - документы

БАЗА данных SQL

Метаданные XML

SQL - схема базы данных

Преобразователь: метаданные XML⇒SQL

Преобразователь: метаданные SQL⇒XML

Объединенные метаданные XML

Объединенная схема SQL

Рис. 1. Упрощенная архитектура системы, обеспечивающей интеграцию информационных ресурсов XML и SQL

Если некоторым сообществом разработчиков и пользователей XML-документов будет осуществлена договорённость о единой интерпретации имен, которые будут принадлежать некоторому набору, то данный унифицированный набор, возможно, применяя описания его смысла (например, в вид обычного текста на естественном языке), может быть использован как пространство имён. Адрес веб-документа, представляющий этот набор имён, будет рассматриваться в виде уникального идентификатора пространства имён, в результате на него можно ссылаться в XML-документе, в котором используются принадлежащие этому пространству имена. В результате чего им будет придаваться некоторый смысл [15, 42].

Метаданные, представляющиеся средствами RDF, могут быть использованы, чтобы поисковые машины Web могли эффективно производить поиск ресурсов, в электронных библиотеках и коллекциях Web-страниц, составляя некоторый виртуальный документ, представляющий содержание информационных ресурсов по конкретным предметным областям, а также оказывающий поддержку различным Web-приложениям, нуждающимся в семантической информации о Web-ресурсах [15, 52].

В настоящее время создаются сервисы для регистрации и поддержки пространств имён согласно интересам разных сообществ разработчиков и пользователей. Зарегистрированное пространство имен представляет собой стандарт для сообщества клиентов сервиса с регистрацией [16, 35].

Большая работа по продвижению стандарта платформы XML на практике ведётся крупным международным, некоммерческим консорциум OASIS (расшифровывается как, Organization for Structured Information Standards - организация по стандартизации структурированной информации), в состав этого консорциума входит более 600 как корпоративных, так и индивидуальных членов из разных стран мира. OASIS занимается разработкой и координацией процесса разработки и распространения информации о методологии применения технологии и стандартов XML [16, 42].

На распространение стандарта XML-платформы существенным образом влияет политика W3C, которая направлена на обеспечение доступности данных спецификаций. Создавая ряд для свободного распространения синтаксических анализаторов языков.

Хоть язык XML и созданные на нём стандарты получают всё большее распространение, имеется и ряд факторов, сдерживающие массовое распространение XML в среде сетей Web [16, 51].

Технологии XML также распространяется и России. В результате чего началось формирование русскоязычной терминологии в этой области.

Платформа XML имеет довольно благоприятные перспективы для широкого применения. Благодаря богатым функциональным возможностям рассмотренных семейств Web-стандартов, и высокой активности в области разработки и развития сетевых стандартов, а также создания программного обеспечения, на основе XML.

Расширяемый язык разметки гипертекста XHTML

В ходе активного распространения технологии XML образовалась проблема обеспечения преемственности в среде Web, что требовало создание возможностей, которые позволили бы пользоваться огромными информационными ресурсами в рамках технологии HTML.

Один из подходов для решения данной проблемы реализован в стандарте XHTML 1.0 одобренный W3C в январе 2000 г. Так как HTML и XHTML являются подмножеством языка SGML, но язык XHTML, в отличие от предшественника, более соответствует спецификациям XML [17, 51].

Идея предложенного подхода заключается в том, что надо создать на основе XML, язык разметки, который мог быть эквивалентен по функциональности языку HTML. Аналогично в настоящее время разрабатываются многие другие языки гипертекстовой разметки с конкретизацией XML. Процесс создания таких языков сводится, к разрабатываю способов определения типов документов (DTD).

XHTML это преемник HTML. Потребность, чтобы создать более строгую версию HTML появилась в следствии того, что современный веб-контент более ориентируется на нетрадиционные виды устройств (например, сотовые телефоны, смартфоны, планшетные компьютеры), в которых бывают часто ограниченными в ресурсах,

поэтому web-страницы должны быть написаны на гибком и нетребовательном к ресурсам языке XHTML [17, 101].

Во всех современных браузерах имеется поддержка XHTML. Совместимость с ним поддерживается и в старых браузерах, т. к. в основу XHTML входит HTML. Данная совместимость приводит к замедлению процесса перехода от HTML к XHTML в сетях Web.

Настоящим преимуществом XHTML является его сочетание с каскадной таблицей стилей. Что позволило отделить процесс оформления документа от оформления содержимого документа.

Стандарт XHTML предлагает три варианта своего языка для отображения HTML-документов и три версии DTD:

- XHTML Strict (строгий XHTML);
- XHTML Transitional (переходный XHTML);
- XHTML Frameset (XHTML с фреймами).

Вариант XHTML Strict нужен, чтобы полностью отделить содержание документа от его оформления. Он используется для чистой структурной разметки веб-документа не применяя элементы форматирования. Для форматирования такого документа можно дополнительно воспользоваться языком каскадных таблиц стилей (CSS) [16, 43].

Вариант XHTML Transitional позволяет использовать таблицы стилей, с некоторыми небольшими корректировками в разметке, для того чтобы документ мог отображаться в старых браузерах, в которых может не быть поддержки таблиц стилей. Он предназначается для лёгкого перехода из HTML в XHTML.

Вариант XHTML Frameset добавляет поддержку фреймов в веб-документ. Что позволило разбить страницу сайта в окне браузера на несколько отдельных разделов (фреймов), в которых находится набор функций управления разделами сайта.

Также существуют специализированные дополнительные версии XHTML:

XHTML 1.1 Модульный (Module-based): программист может импортировать дополнительные свойства в разметку документа. В этой версии имеется поддержка руби-разметку, которая необходима для отображения контента на восточных языках.

XHTML Основной (Basic): является специальной облегчённой версией XHTML для устройств, которые поддерживают ограниченный набор элементов XHTML, такой язык используется в основном для миниатюрных устройств, к примеру, для мобильных телефонов. В будущем он должен заменить WML и C-HTML.

XHTML мобильного профиля (Mobile Profile): основан на архитектуре XHTML Basic, он позволяет добавлять специфические элементы для отображения на мобильных телефонах. Он являлся шагом вперед к созданию мобильных сервисов 3G. XHTML предоставляет пользователю доступ к полноцветному контенту, который имеет отличный вид и имеет удобный интерфейс [18, 65].

Обеспечение модульности языка воплощается в проекте новой версии стандарта - XHTML 1.1.

Со временем XHTML стал более доступен, так как используемые им пространственные имена практически идентичны с HTML 4.01, что позволяет взаимодействовать с DTD во всех современных браузерах. Работа с DTD (Document Type Definition) является критически важным аспектом для XML-документов. Пространство имён языка XML очень велико и для его определения используется именно DTD написанных специально для конкретных разновидностей. Для браузера во время отображения соответствующего XML-кода данные документы должны быть доступными. Преимуществом XHTML в том, что пространство имён можно сократить до списка которым используется пользователь на сайте [18, 78].

Спецификации технологий Web

2.4.1 О спецификации HTML

Спецификация HTML состоит из следующих разделов.

Введение. Во введении приводится описание занимаемого языком HTML места в схеме World Wide Web, приведено описание краткой истории развития языка HTML, имеется описание действий, которые можно сделать, используя HTML 4.0, и содержится описание процесса создания документов формата HTML [19, 41].

Информация о языке. В разделе «Информация о языке» содержатся следующие данные:

- Информация о символах, которые отображаются в документе HTML.
- Информация об основных типах, данных в документе HTML.
- Информация об элементах управления структурой документа HTML.
- Информация об элементах управления представления документов формата HTML.
- Информация об элементах управления интерактивностью документа HTML, включая описание форм для ввода данных пользователем и скрипты для управления активными документами.
- Информация о формальном SGML-определении HTML.

Список нормативных и информативных документов. В этом разделе содержится информация о нормативных и информационных документах, на основании которых был создан язык HTML.

Выше описанной спецификацией можно пользоваться несколькими способами:

- 1) Прочсть её от начала до конца. Узнать общее представление о языке HTML и его технических параметрах.
- 2) Обращаться к необходимой информации. Получить только нужную информацию, а именно информацию о максимальной скорости передачи информации и о синтаксисе языка.

В оперативной версии спецификаций имеются:

- 1) Ссылки на элементы или атрибуты, связанные с определением их спецификаций.
- 2) На каждой странице имеется ссылка на указатели, чтобы найти определение на какой-либо элемент или атрибут, использовать можно не более двух ссылок.
- 3) На каждой первой странице трёх разделов руководства к исходному оглавлению добавлена более подробная информация о разделах.

Согласно спецификации HTML названия элементов описываются символами в верхних регистрах (например, в теге BODY), а названия атрибутов описываются символами в нижних регистрах (например, в тегах lang, onsubmit). Главное помнить, что в HTML имена элемента и атрибута не учитываются регистром, это было придумано для более легкого чтения пользователем [19, 45].

В название элемента и атрибута в HTML документе задаётся разметкой, поэтому пользователь может генерировать названия особым образом.

На каждое определение атрибута устанавливается отдельный тип его значения. Если возможных значений может быть несколько, то список значений, разделяется вертикальной чертой (|).

Информация о типе каждого определения атрибута помещается в квадратные скобки («[]»).

О спецификации XML

Расширенный язык разметки XML представляет собой поднабор SGML и имеет описание в спецификациях. Он создавался, чтобы обеспечить обслуживание, передачу и обработку в сети исходного SGML таким же способом, которым пользуется HTML. XML разрабатывался, чтобы облегчить процесс создания конкретной реализации и взаимодействий с SGML и HTML [20, 46].

W3C способствовало привлечению внимания к данной спецификации, для её широкого распространения, чтобы расширить функциональные возможности сети.

XML (сокращённо Extensible Markup Language) занимается описанием классов объектов данных, которые называются XML-документами, и частично занимается описанием поведения компьютерных программ обрабатывающих XML-документы. XML можно считать профилем приложения или урезанным вариантом SGML - The Standard Generalized Markup Language. По своей структуре документы XML схожи с документами SGML [20, 51].

Документ XML состоит из единиц хранения, которые называются экземплярами, в них содержат данные разбираемого и неразбираемого вида.

Разбираемые данные состоят из символов, из некоторых образуются символьные данные, а из других разметка. Разметка занимается кодированием описаний схем и логических структур единиц хранения документа. XML может предоставлять механизмы, чтобы наложить ограничение на схемы и логические структуры единиц хранения [20, 60].

XML разработала XML Working Group, которая была сформирована руководством World Wide Web Consortium (W3C) в 1996 году [20, 65].

Цель создания XML:

1. Широкое распространение в сети Internet.

2. Поддержка большого диапазона приложений.
3. Совместимость с SGML.
4. Лёгкий для написания программ и обрабатывающих документов язык гипертекстовой разметки.
5. Обеспечение минимального количества свойств по выбору (optional) в идеале - к нулю.
6. Документы должны стать более разборчивыми и точными по смыслу.
7. Обеспечение быстрого дизайна.
8. Дизайн должен стать формальным и кратким.
9. Лёгкое создание документов.
10. Краткость должна иметь минимальное значение.

Данная спецификация, вместе с другими ассоциированными стандартами, может предоставить всю нужную информацию, чтобы начать понимать XML и создавать на нём компьютерные программы и документы [19, 75].

Символы – это разбираемые экземпляры содержащие текст, последовательности символов, которые могут представлять собой символьные данные или разметку. Текст является комбинацией символьных данных и разметки.

Комментарии могут помещаться в любое место документа, игнорируя прочую разметку. Кроме того, их можно помещать внутри объявления, о типе документа учитывая допустимую грамматику. Они не часть символьных данных документа, поэтому процессор XML может, не давая приложению возможности запрашивать тексты комментариев.

Инструкции процесса (ИП) используются, для ввода инструкций для приложения в текст документа. Раздел CDATA может появляться в тех же местах, что и символьные данные; этот раздел используется для создания escape-блоков текста, которые содержат символы, которые будут распознаваться в качестве разметки. Документ XML должен начинаться с объявления XML, в котором указана используемая версия XML [21, 47].

Поскольку будущие версии ещё не сформированы, эта конструкция даётся как средство предоставления возможности автоматического распознавания версии и должна, следовательно, быть включена обязательно. Процессоры могут сигнализировать об ошибке, если получают документ, помеченный неподдерживаемой версией [21, 67].

Функции разметки в документе XML состоят в описании структуры хранимых данных и их логической структуры, а также в ассоциировании пар атрибут-значение с их логической структурой. XML использует механизмы объявления типов документов, чтобы определить ограничения в их логической структуре, чтобы в последствии осуществить поддержку использования предопределённой единицы хранения. Документ XML считается правильным/valid, если в нём имеется ассоциированное объявление о типе документа, если документом выполняется ограничение [21, 69].

Объявление типа документа XML служит для указания на объявление разметки, предоставляя грамматику для классов документов. Объявление разметки представляет собой объявление типа какого-либо элемента, объявление списков атрибутов или объявление экземпляров, или объявление о нотациях. Данные объявления могут как полностью, так и частично иметься внутри каждого экземпляра параметров.

В каждом документе XML содержится один или более элементов, которые ограничены начальным и конечным тегом, либо тегами для пустых элементов. Каждый элемент имеет свой тип, с идентификацией по имени, которое называют «generic identifier» (GI) или родовой идентификатор, в результате чего может иметь в себе набор спецификаций по атрибутам. Каждая спецификация атрибута имеет своё имя и значение [21, 73].

Начало в каждом непустом элементе XML обозначается при помощи начального тега. Окончание в каждом элементе, который начатое начальным тегом, должно быть отмечено при помощи конечного тега, которое содержит имя, отражающее тип элемента, как в начальном теге. Текст, находящийся между начальным и конечным тегом называют «содержимым элемента».

Элемент без содержимого называют «пустым». Пустой элемент представляется либо при помощи начального тега, после которого добавляется конечный тег, либо при помощи тега пустого элемента. Тег пустого элемента имеет свою особую форму.

Структура элементов документа XML может быть ограничена для проверки, при помощи объявления типа элемента и списка его атрибутов. Объявление типов элемента может ограничивать содержимое элементов.

Объявления типов элементов часто используются для ограничения типов элементов, которые могут появиться как потомки элементов.

В типе элемента имеется содержимое элемента, если элемент данного типа обязан содержать в себе только дочерние элементы, а не символьные данные, дочерние элементы могут, по усмотрению, быть разделены пробелами [21, 82].

В этом случае ограничивается модель содержимого, а именно простая грамматика, которая управляет разрешённым типом дочерних элементов и их порядком, в котором они появляются.

В типе элемента также имеется смешанное содержимое, если элементы этого типа содержат в себе символьные данные, перенесёнными дочерними элементами.

Атрибут используется для ассоциации пар имя-значение с другими элементами. Спецификация атрибутов может указываться только в начальном теге и в теге пустого элемента; поэтому продукция, используемая чтобы их распознать, указывается в разделе.

Документ XML состоит из одной или более единиц хранения. Эти единицы называют экземплярами; в них имеется содержимое и все они идентифицируются согласно name\имени экземпляра. Содержимое в разбираемом экземпляре называют замещающим текстом; этот текст является неотъемлемой частью документа [22, 45].

Неразбираемый экземпляр представляет собой ресурс, содержимое которого может, или быть или не быть, текстом, и, в случае если это текст, может не быть XML. В каждом неразбираемом экземпляре имеется ассоциированная нотация, идентифицируемая по имени. Согласно требованию процессора XML делать идентификатор экземпляра и его нотации доступным приложению, а сам XML не налаживает ограничений на содержимое в неразбираемом экземпляре [17, 49].

Экземпляр параметров - это разбираемый экземпляр для пользования внутри ОТД. Этими двумя типами экземпляров используются разные формы ссылок, и они распознаются в разных контекстах. В результате чего ими заняты разные пространства имён; поэтому экземпляры параметра и общие экземпляры, имеющие

одно имя - это двое разных экземпляра.

Ссылка в символах ссылается на специфические символы в наборе символов ISO/IEC 10646, к примеру, ссылка на символ, который не доступен напрямую из устройства для ввода. Ссылка экземпляра может ссылаться на содержимое только именованного экземпляра [20, 61].

Если процессором XML обнаруживается ссылка на разбираемые экземпляры, то, для проверки документа, процессор нужно включить его экземпляр с замещающим текстом. Если экземпляр внешний, а процессор проверяет документ XML, это значит, что процессор может, не считая необходимость, включать замещающие тексты экземпляра. Если не проверяющим процессором не был включён замещающий текст, то он должен уведомить приложение, о том, что им был обнаружен, но не прочитан, экземпляр.

Данное правило основано на том, что автоматически распознавать, предоставляемые механизмы экземпляров SGML и XML, которые были созданы, чтобы поддерживать модульность в авторизацию, не обязательно в других приложениях, особенно во время просмотра документа. Браузеры, к примеру, когда обнаружены ссылки на внешние разбираемые экземпляры, могут отобразить визуальное предупреждение о том, что данный экземпляр существует, и запрашивает у пользователя команду для показа найденного экземпляра [22, 59].

Описанные ранее процессоры XML можно разделить на два класса: проверяющий и не проверяющий процессор. Проверяющий и не проверяющий процессоры нужны для вывода сообщений о нарушении ограничений в правильно сформированной спецификации содержимого экземпляра документа и любого другого разбираемого экземпляра, которые читаются [22, 63].

Проверяющий процессор обязан, по запросу пользователя, сообщать о нарушении ограничений, выраженные объявлением в ОТД, и невозможности выполнить ограничение правильности, данных в конкретной спецификации. Чтобы выполнить ограничение правильности, проверяющий процессор XML должен прочитать и обработать все ОТД и провести проверку всех внешних разбираемых экземпляров, на которые имеются в документе ссылки. Не проверяющие процессоры проверяют только экземпляры документов, вместе с внутренним поднабором ОТД, на правильность формирования. Формальная грамматика XML приведена в спецификации согласно используемым нотациям Extended Backus-Naur Form (EBNF). Каждое правило грамматики определено одним символом [22, 79].

2.4.3 О спецификации XHTML

Семейство типов документов XHTML базируется на XML и предназначено для работы с пользовательскими агентами на базе. Семейство XHTML является следующим шагом в эволюции Интернет. Переходя сегодня на XHTML, разработчики содержимого (контента) могут вступить в мир XML со всеми его преимуществами, сохраняя при этом совместимость содержимого с более старыми и более новыми версиями [15, 78].

Пространство имен XHTML может использоваться с другими пространствами XML в соответствии с [XMLNAMES], хотя такие документы не являются строго конформными XHTML 1.0 в соответствии с приведенным выше определением. В будущих работах W3C будут определены способы указания конформности документов, в которых используется несколько пространств имен [19, 61].

Конформный пользовательский агент должен соответствовать всем определенным в спецификации критериям.

Говорится о различиях которые присутствуют в языке XHTML. Поскольку XHTML является приложением XML, некоторые приемы, допустимые в языке HTML, основанном на SGML, должны быть изменены.

К документам XHTML 1.0 не предъявляется требование совместимости с существующими пользовательскими агентами, но на практике оно достаточно легко реализуемо.

Спецификация XHTML 1.0 закладывает основу семейства типов документов, которые будут расширениями и подмножествами XHTML, для поддержания широкого диапазона новых устройств и приложений путем определения модулей и механизма объединения этих модулей. Такой механизм позволит унифицировать способы расширения XHTML 1.0 и использования его подмножеств путем определения новых модулей [22, 71].

По мере перемещения XHTML с традиционных пользовательских агентов на рабочем столе на другие платформы становится ясно, что не все элементы XHTML будут необходимы на всех платформах.

Процесс модуляризации разбивает XHTML на ряд более мелких подмножеств элементов. Модуляризация дает определенные преимущества.

В профиле документа определяется синтаксис и семантика набора документов. Соответствие профилю документа обеспечивает основу гарантии совместимости. В профиле документа определяются возможности, необходимые для обработки документа этого типа.

HTML5 существует в двух вариациях - как разновидность XHTML и как модификация «исторического» HTML. Создатели веб-страниц могут применять любую из этих вариаций. В первой вариации новый язык полностью согласован с XML 1.0, так что разработчикам доступны все преимущества XML и смежных технологий. Во второй вариации HTML5 покончил с кошмарным наследием прошлого - формальной совместимостью с SGML. Все предыдущие стандарты HTML претендовали на эту совместимость, но ни один браузер не обрабатывал код HTML по замысловатым правилам SGML. Вместо этого использовались простые алгоритмы, распознающие набор конкретных элементов и атрибутов. HTML 5 узаконил такое поведение браузеров. Новый стандарт не оглядывается на SGML, а описывает частные правила для интерпретации конкретных тегов и других синтаксических конструкций. По сути, HTML 5 документирует принципы, по которым браузеры уже обрабатывают код последние пятнадцать лет [23, 79].

В отличие от предыдущих стандартов, предназначенных в основном для разработчиков кода, HTML5 строго и подробно описывает поведение браузеров. Он требует от браузеров обрабатывать не только грамотный код, но и по возможности корректировать ошибки, вплоть до употребления отсутствующих в стандарте элементов. Например, HTML 5 не позволяет авторам кода использовать элементы `tbody` или `marquee`, но описывает, как браузер должен поступать, если они все-таки встретятся в документе. Это соответствует очередному генеральному принципу HTML5 - максимальной совместимости с уже существующим наполнением сети. Миллионы действующих сайтов не должны пострадать от появления нового языка и поддерживающих его браузеров [23, 66].

Ещё одна особенность нового стандарта: он описывает документ HTML в терминах объектной модели (DOM). Перед тем, как отображать веб-страницу, браузер формирует древовидную структуру её элементов и других объектов. Предыдущие стандарты стыдливо замалчивали этот этап, а HTML5, наоборот, во многом опирается именно на такое представление документа. Это позволяет включить в спецификацию объекты, свойства и методы DOM, которые широко используются с

незапамятных времён, однако никогда не были стандартизованы: `window.open()`, `alert()`, `prompt()`, `document.links`, `document.images`, `innerHTML`, `contentEditable` и так далее. Наконец, в HTML 5 официально включён элемент `embed` [17, 31].

Однако HTML5 впитал не всё, что поддерживается в браузерах. Некоторые элементы, наоборот, удалены из стандарта: `acronym`, `applet`, `basefont`, `big`, `center`, `dir`, `font`, `isindex`, `s`, `strike`, `tt`, `u`. Убраны фреймы (`frame`, `frameset`, `noframes`), хотя остался элемент `iframe`. Нет больше атрибутов, которыми почти никто не пользовался (`accesskey`, `longdesc` и др.) или которыми не имеет смысла пользоваться (`align`, `border`, `valign`, `cellpadding`, `cellspacing` и др.). Некоторые старые элементы переосмыслены: например, `i` и `b` теперь обозначают разные формы смыслового выделения, а не курсив и полужирный шрифт; `hr` отвечает за разрыв в структуре документа. Элемент `menu` обрел вторую молодость и позволяет создавать настоящие меню, а `address` ущемлён в правах и предназначен строго для контактной информации в узком контексте [23, 81].

Наконец, в HTML5 добавилось много нового. Для статей, блогов, документации и подобных материалов будут полезны элементы `article` и `section`, а также новая система заголовков и рубрикации. «Шапка» и «подвал», панель навигации и боковая панель обозначаются элементами `header`, `footer`, `nav` и `aside`. Для мультимедийного содержания появились элементы `audio` и `video`. Они внедряют в страницу соответствующий ресурс и обеспечивают стандартные элементы управления - кнопки запуска, паузы и остановки, проматывание, регулятор громкости. Для тех браузеров, где эти элементы не поддерживаются, HTML5 позволяет дополнить их альтернативным содержанием - например, роликом Flash, с помощью которого сейчас внедряется большинство аудио- и видеозаписей. В продвинутых браузерах полноценно работает элемент `audio` или `video`, а во всех остальных - альтернативный ролик, так что все пользователи останутся довольны [16, 81].

Здесь проявляется ещё один генеральный принцип - новые возможности HTML5 должны создавать как можно меньше проблем в тех браузерах, где они не поддерживаются. Почти все новшества языка устроены так, что для старых браузеров можно обеспечить их реализацию с помощью альтернативного содержания, CSS, JavaScript и других технологий. Отдельного упоминания заслуживает элемент `canvas`, который предоставляет область для рисования изображений «на лету» средствами JavaScript. Собственно, массовый интерес к HTML 5 начался после того, как Firefox и Opera решили поддерживать этот элемент.

Формы в HTML5 изменились до неузнаваемости. Примитивные процедуры заполнения и отправки формы, описанные в прежних версиях HTML, уже сейчас никого не устраивают, поэтому разработчики сайтов выдумывают собственные расширения интерфейса, средства проверки, самодельные элементы управления. В HTML5 эта самодеятельность урегулирована стандартными элементами, атрибутами и интерфейсами DOM. В дополнение к обычным текстовым полям, появились поля для ввода определённых данных. Например, `input type=«number»` - это поле для ввода числа, а `input type=«range»` - ползунок для регулирования громкости или чего-то похожего. Атрибуты `min`, `max` и `step` определяют диапазон и точность допустимых значений в поле. Элемент `datalist` обеспечивает поле со списком, а элемент `output` - область для вывода динамически генерируемых результатов. Атрибут `required` для элемента управления обозначает, что этот элемент обязателен для заполнения, так что без него форма не будет отправлена. Атрибут `autofocus` позволяет установить фокус ввода в данный элемент. Метод `checkValidity()` проверяет правильность заполнения полей. До сих пор эти возможности имитировались на сайтах с помощью кустарных сценариев JavaScript [25, 71].

В стандарте предлагается элемент `datagrid` для организации динамических структур данных, элемент `command` для вызова пользовательских команд, элемент `progress` - индикатор хода выполнения операции, а также средства DOM для редактирования содержания прямо на веб-странице, перетаскивания, выделения, работы с буфером обмена и многого другого. Это отвечает общей тенденции развития Всемирной Паутины: если раньше сайты были чем-то вроде публикаций, то теперь они превратились в полноценные приложения, требующие функционального и удобного пользовательского интерфейса. Соответственно, новый HTML также должен стать не столько языком публикаций, сколько языком описания приложений [22, 89].

Преимущества HTML5:

1) Повышенная безопасность. В настоящее время, используемый множеством WEB - сайтами тег `iframe` содержит в себе явную угрозу безопасности для пользовательской системы. Дело в том, что функцией этого тега является отображение контента с другого сайта, например в плане рекламы или сведений о состоянии каких-либо данных. Но если сайт, передающий такую информацию подвергнется взлому, появляется большая вероятность того, что злоумышленники через него направят вредоносный код на сайт-приемник, который и подвергнет его пользователей опасности заражения вирусом. В HTML5, чтобы избежать подобной

ситуации для тегов `iframe` была спроектирована технология `Sandbox`. Благодаря ей, все скрипты, запускающиеся через `iframe` будут работать в специально ограниченной виртуальной среде, которая не выпустит результаты их работы за пределы браузера, а значит и не сможет поразить пользовательскую операционную систему. Даже если предположить, что запуск вируса посредством этого тега все же произойдет, какие-либо серьезные последствия он понести уже не сможет [25, 77].

2) Ввод данных. В HTML5 становится проще и безопаснее. Правильность введения данных контролируется в режиме реального времени непосредственно во время ввода. В HTML4.0 проверка ошибок и сообщение о них происходит уже после отправки документа, что может создавать проблемы пользователям.

3) Поддержка полноценной графики. До недавнего времени бесспорным лидером анимированной графики для браузеров была технология `Flash`. С появлением в новом HTML5 стандарте такого инструмента как `canvas` позиции `Flash`-графики будут, вероятно, сильно потеснены.

Суть такого мощного нововведения как `canvas` состоит в том, что для браузеров на экране выделяется специальное пространство, положение которого на веб-странице определяется разработчиком сайта. В этой области с помощью целого набора специальных графических HTML5 команд можно составлять графические изображения, в том числе и анимированные, практически любой сложности. Более того, для 3-D графики разрабатывается специальный набор команд аналогичных всемирно известному стандарту трехмерной графики `OpenGL`. Теперь браузер будет способен воспроизводить не только красивые двухмерные интерактивные приложения, но и полноценные трехмерные игры [24, 102].

Если для HTML5 и дальше продолжат развиваться направление трехмерных игр, в недалеком будущем мы имеем все шансы получить игровые проекты с современной шейдерной графикой и соответствующими спецэффектами для нее. Как только графику браузеров полностью переведут на вычислительную поддержку от видеокарт, в сети будут размещены ресурсы с он-лайн играми практически ничем не отличающимися от сегодняшних графически красивых, но отдельно устанавливаемых клиентов игровых приложений [25, 91].

4) Видео и аудио. Для того чтобы воспроизвести видео или аудио поток в браузере пользователю придется прибегать к помощи все той же `Flash` технологии, требуемые в HTML4.0.

Браузерам для этого приходится постоянно пользоваться Flash проигрывателями, а также скачивать постоянные обновления от Adobe для этой технологии, без которых пользователи сильно рискуют заразить свои компьютеры вирусами через бреши в старых версиях Flash.

HTML5 призван, наконец, избавить браузеры от этих дополнительных программ и дать им возможность полноценного воспроизведения аудио и видео контента своими силами. Для этого в HTML5 коде предусмотрены специальные теги, такие как `<video>` и `<audio>`. Впрочем, у этого нововведения имеются и свои недостатки, например отсутствие единого стандарта для видеокодеков, что не способствует пока широкому распространению этой удобной для пользователя и веб-разработчиков технологии [24, 81].

5) Многократное увеличение скорости работы. Сложно представить современные Web-сайты без большого числа интерактивных форм, видео и аудио вставок, анимированных графических элементов и разного рода оживляющих Web-сайты программ-скриптов. Конечно, мощности современных вычислительных машин вполне хватает, чтобы обрабатывать такое огромное количество элементов Web - страницы, но Интернет технологии не стоят на месте, и чем больше будет развиваться визуализация, интерактивность и качество медиа-информации, тем выше будут требования к «железу» пользователя.

Конечно же, проектировщики HTML5 не могли обойти эту проблему стороной. Новый стандарт обеспечивает поддержку многопоточности, которую именовали как Web Works. Таким образом, современные многоядерные системы получают возможность параллельно выполнять обработку скриптов и тем самым увеличить скорость работы сайта на несколько порядков в зависимости от его конструкции и сложности [25, 62].

6) Технология Web Storage. Пользователи комфортно работают с теми сайтами, которые способны сохранять изменения в настройках, например для внешнего вида рабочего пространства. Для того, чтобы сохранить какую-либо информацию о действиях пользователя сайт обычно пользуется технологией cookies, которая позволяет сохранять на компьютере посетителя текстовый файл со своими данными.

Технология Web Storage для HTML5 затмит cookies своими возможностями только потому, что представляет из себя не форматно ограниченный текстовый файл, а специально ориентированную базу данных для различных видов информации.

Теперь в такой базе сайт может хранить не только отдельные настройки сайта, а целые фрагменты его кода, что позволит работать с веб-приложениями даже при отсутствии подключения к сети Интернет.

7) Структурные возможности HTML5. Делают структуру Web-документа более простой, а код чище. Новые семантические элементы header, nav, section, article, aside и footer могут использоваться вместо контейнеров div, применяющихся сейчас в HTML4.0. новые элементы позволяют более конкретно описывать заголовок, нижний колонтитул, колонки текста и другие. Эти нововведения могут использоваться для автоматической генерации оглавления и организации более эффективной и простой навигации по Web -странице [25, 75].

Вывод по второй главе: объединив на одной Web-страницы возможности HTML и HTML5, можно создать интерактивный сайт, на котором можно выполнять любые действия пользователя.

Глава 3. Практическое применение языка разметки гипертекста HTML

В качестве примера мною был создан обучающий сайт, который представляет собой широкую иерархию с элементами нелинейной структуры. Первый уровень иерархии составляют основные рубрики «Главная», «Описание программ», «Практические упражнения», «Видео-уроки», «Контакты». Данные наименования рубрик выбраны в силу обучающей специализации сайта, на данном сайте будут размещены различные статьи и практические упражнения по работе в программах обработки компьютерной графики. Сам сайт будет называться «Обработка компьютерной графики».

Созданный мною обучающий сайт относится к сайтам дистанционного образования, так как на этом сайте будут размещены, различные статьи и практические упражнения выполняя, которые пользователь сайта сможет приобрести некоторые практические навыки работы в компьютерных программах. В качестве компьютерных программ, в которых пользователь сайта будет делать практические задания, мною выбраны следующие:

- растровый графический редактор AdobePhotoshop;
- векторный графический редактор Inkscape;

- программа для 3D моделирования Autodesk 3ds Max.

Эти программы выбраны мною в силу того, что сайт будет ориентирован на тех, кто учится быть графическим дизайнером или любым другим, чья работа будет связана с компьютерной графикой. Соответственно с этим название моего обучающего сайта будет «Обработка компьютерной графики».

Обучающий сайт «Обработка компьютерной графики», создавался при помощи языка гипертекстовой разметки HTML5. Создание исходного кода сайта происходило в программе Notepad++.

Notepad++ - это бесплатный текстовый редактор, предназначенный для работы с текстом и редактирования исходного кода компьютерной программы, сайта, веб-странички, компонентов системы и много другого [25, 54].

Созданные в редакторе Notepad++ Web-страницы сохранялись в формате .html и помещались в папку сайта «Обработка компьютерной графики». А пример исходного кода Web-страниц написанный в окне Notepad++ можно посмотреть в приложении 1.

В каждой Web-странице создавались перекрёстные ссылки, благодаря которым много созданных Web-страниц были объединены в один сайт. Применение возможностей языка HTML5 особенно заметно в оформлении дизайна сайта. Сайт имеет горизонтальное меню, наводя курсор на пункты меню можно заметить плавное изменение цвета надписей. При переходе на страницы сайта каждый раз изменяется цвет оформления сайта. Обучающий сайт «Обработка компьютерной графики» имеет пять Web-страниц и каждая из этих Web-страниц имеет своё цветовое оформление.

Во вкладке «Главная» размещена информация о назначении данного обучающего сайта, информация о назначении Web-страниц сайта, а также информация о практических работах, размещённых на сайте.

Во вкладке «Описание программ» размещено общее описание компьютерных программ, в которых надо будет выполнять практические работы, размещённые на сайте. В данной вкладке размещено описание растрового графического редактора Adobe Photoshop, редактора трёхмерной графики 3D Studio (3DS) Max, векторного графического редактора Inkscape.

Во вкладке «Практические упражнения», размещены практические задания, которые надо выполнять в вышеописанных программах. На сайте ссылки на эти практические упражнения представлены в виде раскрывающегося списка, который создан при помощи языка HTML5. При нажатии на заголовок списка под заголовком появляются ссылки на практические упражнения. Практические упражнения представляют собой статью с изображением, в каждом упражнении имеется кнопка возврата на предыдущую страницу, а в некоторых ссылки на другие вкладки сайта.

Во вкладке «Видео-уроки» размещены видео, в которых автор видео объясняет на практическом примере принципы работы в компьютерной программе. В данной вкладке размещены видео-уроки по работе в программе 3DS Max.

Во вкладке «Контакты» размещена контактная информация, используя которую пользователь может связаться с администрацией или специалистами сайта «Обработка компьютерной графики». Также в данной вкладке размещена форма обратной связи со специалистами сайта «Обработка компьютерной графики».

Скриншоты вышеописанных web-страниц сайта «Обработка компьютерной графики» можно просмотреть в приложении 2.

Вывод по третьей главе. Таким образом, при помощи языков гипертекстовой разметки HTML и HTML5 мною был создан Web-сайт обучающей направленности.

ЗАКЛЮЧЕНИЕ

Создание Web поправу можно считать одним из крупнейших научно - технических достижений последнего десятилетия XX века. Благодаря реализации этого проекта рождается целый ряд новых информационных технологий, имеющих весьма значимые социально-экономические последствия.

Современные информационные Web-технологии быстро изменяют наш мир и непосредственно влияют на развитие Web-технологий. Эта технологическая революция сильно повлияла на все сферы человеческой деятельности. Внутренняя сложность и предельная простота применения современных информационных Web-технологии делает их доступными каждому, кто ежедневно сталкивается с применением их в своей профессиональной деятельности.

Главное преимущество Web-технологий в современных условиях заключается в их простоте и как следствие в повышении эффективности их применения.

В этой курсовой работе мы попытались показать обширную проблематику технологий современных информационных систем, технологий Web, появления новых тенденций в развитии технологий Web.

В результате курсовой работы был разработан основной каркас сайта. Реализовано меню сайта, страницы со статьями и практическими упражнениями, а также форма обратной связи.

Поставленные задачи выполнены, был разработан обучающий сайт «Обработка компьютерной графики».

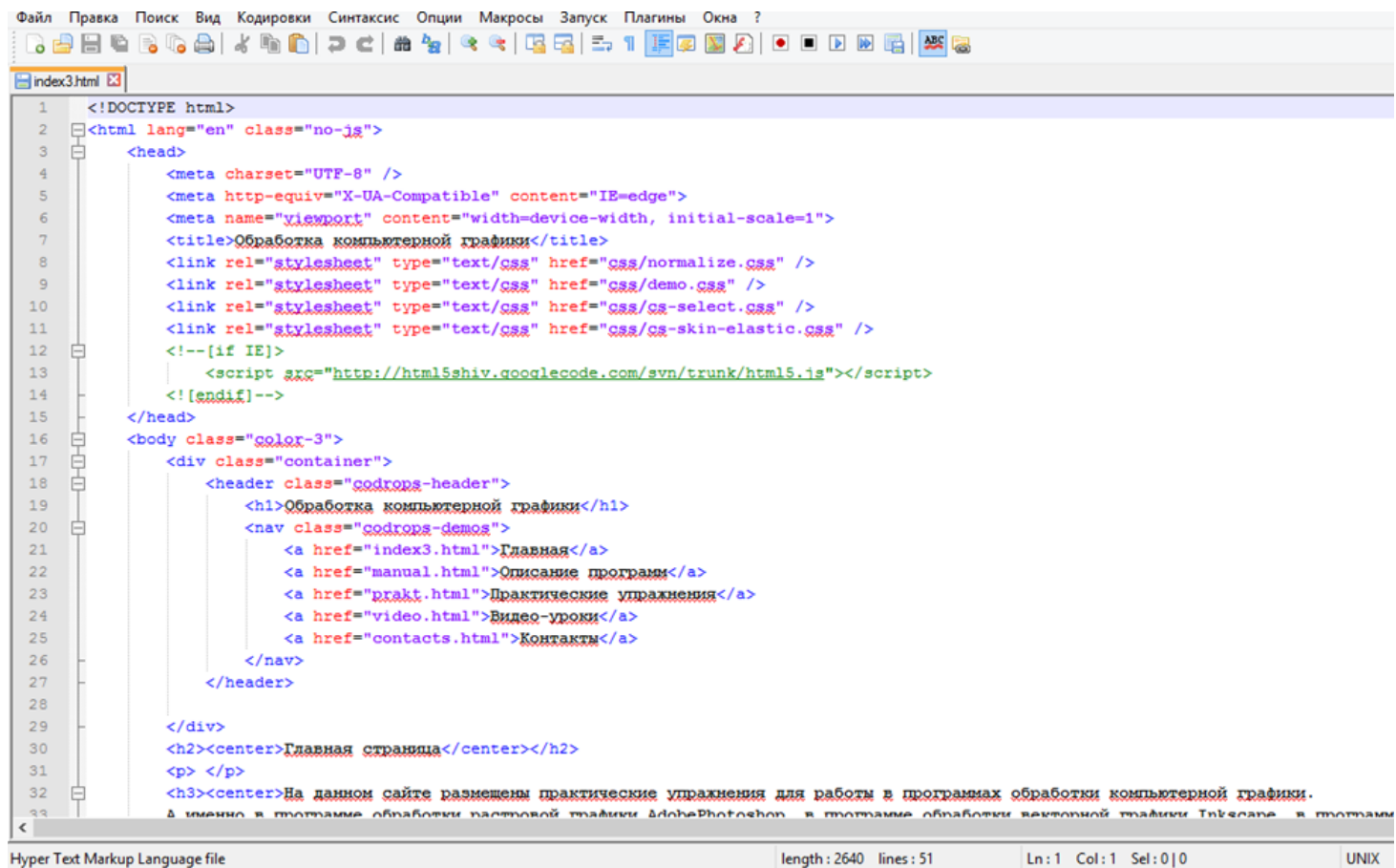
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бенкен, Е.С. PHP, MySQL, XML: программирование для Интернета (+CD-ROM) / Е.С. Бенкен, - 3-е изд. - СПб: БХВ-Петербург, 2017. - 304 с.
2. Бенкен, Е.С. PHP, MySQL, XML: программирование для Интернета / Е.С. Бенкен - СПб: БХВ-Петербург, 2015. - 336 с.
3. Васильев, В.В., Сороколетова Н.В., Хливненко Л.В. Практикум по Web-технологиям / В.В. Васильев, Сороколетова Н.В., Хливненко Л.В. - М.:Форум, НИЦ ИНФРА-М, 2015. - 416 с.
4. Вейл, Э., Вильчинский, Н., Сивченко, О. HTML 5.Разработка приложений для мобильных устройств / Э. Вейл; Пер. с англ. Н. Вильчинского, О. Сивченко. - СПб.: Питер, 2015. - 480с.
5. Веселкова, Т.В., Кабанов, А.С. Эффективная эксплуатация сайта: Практическое пособие / Т.В. Веселкова, А.С. Кабанов - М.: Дашков и К, 2015. - 176 с.
6. Гендина, Н.И., Колкова, Н.И. Создание официальных сайтов учреждений культуры и образования: теория и практика: Сборник научных трудов / Науч. ред. Н.И. Гендина, Н.И. Колкова - СПб: Профессия, 2015. - 384 с.
7. Грингард, С. Интернет вещей: Будущее уже здесь / С. Грингард - М.: Альпина Пабл., 2016. - 188 с.
8. Дронов, В.А. Django. Практика создания Web-сайтов на Python / В.А. Дронов - СПб: БХВ-Петербург, 2016. - 528 с.
9. Дронов, В.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / В.А. Дронов - СПб: БХВ-Петербург, 2016. - 768с.

10. Дронов, В.А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов / В.А. Дронов - СПб:БХВ-Петербург, 2016. - 688 с.
11. Дронов, В.А. PHP, MySQL, HTML5 и CSS3. Разработка современных динамических Web-сайтов / В.А. Дронов - СПб: БХВ-Петербург, 2016 - 688с.
12. Евсеев, Д.А., Трофимов В. В. Web-дизайн в примерах и задачах: Учебное пособие / Д.А. Евсеев; Под ред. Трофимова В. В. - М.: КноРус, 2016. - 264 с.
13. Зараменских, Е.П., Артемьев И.Е. Интернет вещей. Исследования и область применения : монография. / Е.П. Зараменских, И.Е. Артемьев. - М. : ИНФРА-М, 2017. - 188 с.
14. Колисниченко, Д.Н. PHP и MySQL. Разработка Web-приложений / Колисниченко Д.Н., 5-е изд., перераб. и доп. - СПб: БХВ-Петербург, 2015. - 592 с.
15. Колокольцева, Т.Н., Лутовинова, О.В. Интернет-коммуникация как новая речевая формация: Коллективная монография / Науч. ред. Т.Н. Колокольцева, - 3-е изд., стер. - М.:Флинта, Наука, 2016. - 328 с.
16. Мак-Дональд, М., Коротяева, Т. Создание Web-сайта. Недостающее руководство / М. Мак-Дональд; Пер. с англ. Т. Коротяевой - 3-е изд. - СПб.: БХВ-Петербург, 2015. - 624 с.
17. Немцова, Т.И., Назарова, Ю.В., Гагарина, Л.Г. Компьютерная графика и Web-дизайн. Практикум: Учебное пособие / Т.И. Немцова, Ю.В. Назарова; Под ред. Л.Г. Гагариной - М.:ИД ФОРУМ, НИЦ ИНФРА-М, 2016. - 288 с.
18. Никхил, А. Веб-программирование для чайников / А. Никхил - М.: Вильямс, 2016. - 304 с.
19. Пухаренко, Ю.В., Норин, В.А. Метрология, стандартизация и сертификация. Интернет-тестирование базовых знаний: Учебное пособие / Ю.В. Пухаренко, В.А. Норин - СПб: Лань, 2016. - 308 с.
20. Пьюривал, С. Основы разработки веб-приложений / С. Пьюривал - СПб: Питер, 2015. - 272 с.
21. Робсон, Э., Фримен, Э., Черник, В.В. Изучаем HTML, XHTML и CSS / Э. Робсон, Э. Фримен; Пер. Черника В.В., - 2-е изд. - СПб: Питер, 2017. - 720с.
22. Сергеев, А.Н. Создание сайтов на основе WordPress: Учебное пособие / А.Н., Сергеев - СПб: Лань, 2015. - 128 с.
23. Тузовский, А.Ф. Проектирование и разработка web-приложений: Учебное пособие для академического бакалавриата / А.Ф. Тузовский - М.: Юрайт, 2016. - 218 с.
24. Фленов, М.Е. PHP глазами хакера / М.Е. Фленов, - 3-е изд., перераб. и доп. - СПб: БХВ-Петербург, 2016. - 256 с.

25. Фрейн, Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрейн, - 2-е изд. - СПб: Питер, 2017. - 272 с.

ПРИЛОЖЕНИЕ 1



```
1 <!DOCTYPE html>
2 <html lang="en" class="no-js">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <title>Обработка компьютерной графики</title>
8     <link rel="stylesheet" type="text/css" href="css/normalize.css" />
9     <link rel="stylesheet" type="text/css" href="css/demo.css" />
10    <link rel="stylesheet" type="text/css" href="css/cs-select.css" />
11    <link rel="stylesheet" type="text/css" href="css/cs-skin-elastic.css" />
12    <!--[if IE]>
13      <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
14    <![endif]-->
15  </head>
16  <body class="color-3">
17    <div class="container">
18      <header class="codrops-header">
19        <h1>Обработка компьютерной графики</h1>
20        <nav class="codrops-demos">
21          <a href="index3.html">Главная</a>
22          <a href="manual.html">Описание программы</a>
23          <a href="prakt.html">Практические упражнения</a>
24          <a href="video.html">Видео-уроки</a>
25          <a href="contacts.html">Контакты</a>
26        </nav>
27      </header>
28    </div>
29    <h2><center>Главная страница</center></h2>
30    <p></p>
31    <h3><center>На данном сайте размещены практические упражнения для работы в программах обработки компьютерной графики.
32    А именно в программах обработки растровой графики Adobe Photoshop, в программах обработки векторной графики Inkscape, в программах
33
```

Hyper Text Markup Language file length: 2640 lines: 51 Ln: 1 Col: 1 Sel: 0 | 0 UNIX

Рис. 1. Страница «Главная» написанная в программе Notepad++

```
Файл Правка Поиск Вид Кодировки Синтаксис Опции Макросы Запуск Плагины Окна ?
manual.html
25  li {
26  list-style-type: none; /* Убираем маркеры */
27  }
28  ul {
29  margin-left: 0; /* Отступ слева в браузере IE и Opera */
30  padding-left: 0; /* Отступ слева в браузере Firefox, Safari, Chrome */
31  }
32  </style>
33  </head>
34  <body class="color-4">
35  <div class="container">
36  <header class="codrops-header">
37  <h1>Обработка компьютерной графики</h1>
38  <nav class="codrops-demos">
39  <a href="index3.html">Главная</a>
40  <a href="manual.html">Описание программы</a>
41  <a href="prakt.html">Практические упражнения</a>
42  <a href="video.html">Видео-уроки</a>
43  <a href="contacts.html">Контакты</a>
44  </nav>
45  </header>
46
47  </div>
48  <h2><center>Описание программы</center></h2>
49  <h3><p>
50  Adobe Photoshop - многофункциональный графический редактор, разработанный и распространяемый фирмой Adobe Systems.
51  В основном работает с растровыми изображениями, однако имеет некоторые векторные инструменты. Продукт является лидером рынка в
52  области коммерческих средств редактирования растровых изображений и наиболее известным продуктом фирмы Adobe. В настоящее время
53  Photoshop доступен на платформах macOS, Windows, в мобильных системах iOS, Windows Phone и Android. Также существует версия Photoshop
54  Express для Windows Phone 8 и 8.1.
55  </h3>
56  </p>
57  <h3><p> Основные возможности Adobe Photoshop:</p></h3>
58  </h3>
```

Рис. 2. Страница «Описание программ» написанная в программе Notepad++

```
Файл Плавка Поиск Вид Кодировки Синтаксис Опции Макросы Запуск Плагины Окна ?
prakt.html
76         <a href="contacts.html">Контакты</a>
77     </nav>
78 </header>
79
80 </div>
81 <h2><center><b>Практические упражнения</b></center></h2>
82
83 <ul class="cd-accordion-menu animated">
84 <li class="has-children">
85     <input type="checkbox" name ="group-1" id="group-1">
86     <label for="group-1">AdobePhotoshop</label>
87
88     <ul>
89     <li class="has-children">
90         <input type="checkbox" name ="sub-group-1" id="sub-group-1">
91         <label for="sub-group-1">Обработка фотографий</label>
92
93         <ul>
94             <li><a href="urok1.1.html">Эффект объёмного изображения на фото</a></li>
95             <li><a href="urok1.2.html">Создаем коллаж с абстрактными фигурами в фотошоп</a></li>
96         </ul>
97     </li>
98     </ul>
99 </li>
100 <li class="has-children">
101     <input type="checkbox" name ="sub-group-2" id="sub-group-2">
102     <label for="sub-group-2">Создание рисунков</label>
103
104     <ul>
105         <li><a href="urok2.1.html">Создаем макет визитной карточки в фотошоп</a></li>
106         <li><a href="urok2.2.html">Добавляем огненные эффекты в фотошоп</a></li>
107     </ul>
108 </li>
109 </ul>
110 </ul>
Hyper Text Markup Language file      length : 4544  lines : 149      Ln : 1  Col : 1  Sel : 0 | 0
```

Рис. 3. Страница «Практические упражнения» написанная в программе Notepad++


```
Файл Правка Поиск Вид Кодировки Синтаксис Опции Макросы Запуск Плагины Окна ?
video.html
16 <body class="color-9">
17 <div class="container">
18 <header class="codrops-header">
19 <h1>Обработка компьютерной графики</h1>
20 <nav class="codrops-demo">
21 <a href="index3.html">Главная</a>
22 <a href="manual.html">Описание программы</a>
23 <a href="prakt.html">Практические упражнения</a>
24 <a href="video.html">Видео-уроки</a>
25 <a href="contacts.html">Контакты</a>
26 </nav>
27 </header>
28
29 </div>
30 <h2><center>Видео-уроки</center></h2>
31 <p></p>
32 <h3><p>3D MAX для начинающих Урок 1 Интерфейс</p></h3>
33 <video src="video/3D MAX для начинающих Урок 1 Интерфейс.mp4" controls></video>
34 <h3><p>3D MAX для начинающих Урок 2 Моделирование из примитивов</p></h3>
35 <video src="video/3D MAX для начинающих Урок 2 Моделирование из примитивов.mp4" controls></video>
36 <script src="js/classie.js"></script>
37 <script src="js/selectFx.js"></script>
38
39 <script>
40 (function() {
41     [].slice.call( document.querySelectorAll( 'select.cs-select' ) ).forEach( function(el) {
42         new SelectFx(el);
43     } );
44 })();
45 </script>
46
47 </body>
48 </html>
Hyper Text Markup Language file length: 2050 lines: 48 Ln: 1 Col: 1 Sel: 0|0
```

Рис. 4. Страница «Видео-уроки» написанная в программе Notepad++

```
Файл  Плавка  Поиск  Вид  Кодировки  Синтаксис  Опции  Макросы  Запуск  Плагины  Окна  ?
contacts.html x
22      <a href="manual.html">Описание программ</a>
23      <a href="prakt.html">Практические упражнения</a>
24      <a href="video.html">Видео-уроки</a>
25      <a href="contacts.html">Контакты</a>
26      </nav>
27      </header>
28
29      </div>
30      <h2><center>Контакты</center></h2>
31      <p> </p>
32      <h3><p>E-mail администратора сайта: Admin1@site.ru</p></h3>
33      <h3><p>E-mail технического отдела: techno-otdel@site.ru</p></h3>
34      <h3><p>Справка по телефону: 4-23-19, 5-89-14</p></h3>
35      <h3 style="color:#FFFFFF">Форма для отправки сообщения специалисту</h3>
36      <form method="post" action="sendmail.html">
37      <table width="600">
38      <tr>
39      <td>
40      <u>Фамилия Имя Отчество:</u>
41      </td>
42      <td>
43      <input name="fio" type="text" style="width:100%;"/>
44      </td>
45      </tr>
46      <tr>
47      <td>
48      <u>Номер телефона:</u>
49      </td>
50      <td>
51      <input name="tel" type="text" style="width:100%;"/>
52      </td>
53      </tr>
54      <tr>
55      <td>
```

Рис. 5. Страница «Контакты» написанная в программе Notepad++

```
Файл Правка Поиск Вид Кодировки Синтаксис Опции Макросы Запуск Плагины Окна ?
urok1.1.html
22 <a href="manual.html">Описание программы</a>
23 <a href="prakt.html">Практические упражнения</a>
24 <a href="video.html">Видео-уроки</a>
25 <a href="contacts.html">Контакты</a>
26 </nav>
27 </header>
28
29 </div>
30 <p><a href="prakt.html"></a></p>
31 <h2><center>Эффект объемного изображения на фото</center></h2>
32 <p></p>
33 <h3><p>В этом уроке вы научитесь создавать в Photoshop эффект объемной картинки, который оживит Вашу фотографию.</p>
34 </h3>
35 <p></p>
36 <h3><p> Шаг 1. Откройте картинку, с которой вы хотите работать.</p></h3>
37 <p></p>
38 <h3><p>Шаг 2. Создадим рамку, для этого создайте новый слой и с помощью Marquee tool (Прямоугольник), создадим квадрат,
39 перекрывающий картинку. После этого инвертируем выделение Select - Inverse (Выделение - Инверсия выделения) или Shift+Ctrl+I.
40 После этого залдем полученную область цветом, который выбрали для рамки, в данном случае белым. Затем, переместим слой с картинкой,
41 чтобы он находился перед слоем с полученной рамкой.</p></h3>
42 <p></p>
43 <h3><p> Шаг 3. Выберите слой с картинкой. Снова выделим нашу рамку, для этого идем в Select - Reselect (Выделение - Выделить снова),
44 и создадим маску слоя Layer - Layer mask - Hide Selection (Слой - Слой-маска - Скрыть выделенные области). Таким образом, мы спрячем
45 часть рисунка, которая перекрывается рамкой.</p></h3>
46 <p></p>
47 <h3><p> Шаг 4. Выбираем маску слоя и мягкой небольшой кистью белого цвета начнем рисовать сверху тех областей рисунка,
48 которые мы хотим сделать объемными (на примере это края листика). Если вы хотите что-то исправить, то достаточно поменять
49 цвет кисти на черный и стереть ненужную область. Это не очень легко, поэтому не торопитесь, и вы получите желаемый результат.</p></h3>
50 <p></p>
51 <h3><p> Шаг 5. Создание тени. Создадим новый слой и расположим его между слоем с картинкой и слоем с рамкой. Проверьте, что черный
52 или серый является вашим основным цветом.</p></h3>
53 <h3><p> Возьмите кисть и начинайте рисовать по той области, которую хотите сделать объемной. Когда вы закончите, используйте
54 Filter - Blur - Gaussian blur (фильтр - Размытие - Размытие по Гауссу). Выберите подходящий радиус размытия, в соответствии с
55 размером той части, которую вы хотите сделать объемной. Я использовала радиус в 5 пикселей </p></h3>
Hyper Text Markup Language file length: 5926 lines: 74 Ln: 1 Col: 1 Sel: 0|0 UNIX ANSI as UTF-8
```

Рис. 6. Страница с одним из практических заданий написанная в программе Notepad++

ПРИЛОЖЕНИЕ 2

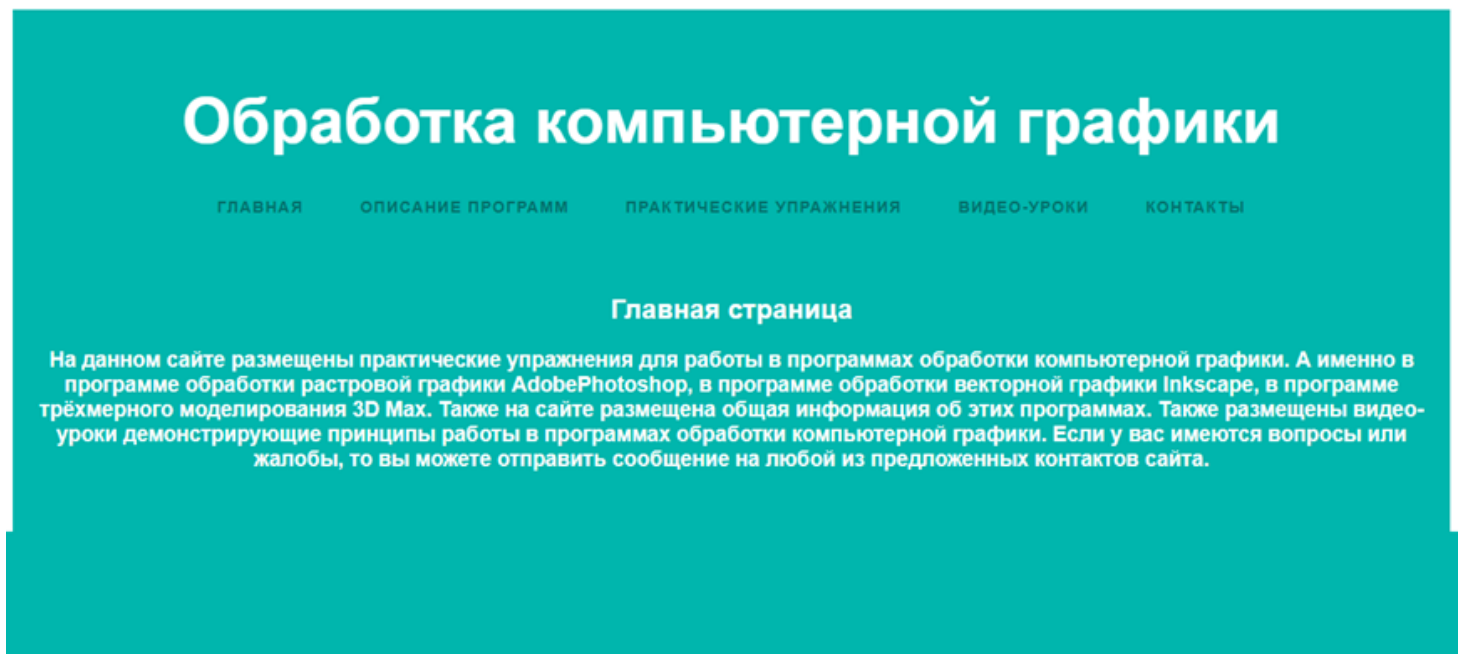


Рис. 7. Скриншот страницы «Главная» на сайте «Обработка компьютерной графики»

Обработка компьютерной графики

[ГЛАВНАЯ](#)

[ОПИСАНИЕ ПРОГРАММ](#)

[ПРАКТИЧЕСКИЕ УПРАЖНЕНИЯ](#)

[ВИДЕО-УРОКИ](#)

[КОНТАКТЫ](#)

Описание программ



Adobe Photoshop - многофункциональный графический редактор, разработанный и распространяемый фирмой Adobe Systems. В основном работает с растровыми изображениями, однако имеет некоторые векторные инструменты. Продукт является лидером рынка в области коммерческих средств редактирования растровых изображений и наиболее известным продуктом фирмы Adobe. В настоящее время Photoshop доступен на платформах macOS, Windows, в мобильных системах iOS, Windows Phone и Android. Также существует версия Photoshop Express для Windows Phone 8 и 8.1.

Основные возможности Adobe Photoshop:

- Поддержка большинства растровых и некоторых векторных графических форматов.
- Наличие всех необходимых инструментов для рисования и редактирования изображений.
- Поддержка цветовых моделей RGB, LAB, CMYK, Градации серого, Черно-белое, Duotone.

- Глубина цвета изображений: 8 бит, 16 бит, 32 бита.

Рис. 8. Скриншот страницы «Описание программ» на сайте «Обработка компьютерной графики»

Обработка компьютерной графики

[ГЛАВНАЯ](#)


[ОПИСАНИЕ ПРОГРАММ](#)


[ПРАКТИЧЕСКИЕ УПРАЖНЕНИЯ](#)

[ВИДЕО-УРОКИ](#)

[КОНТАКТЫ](#)

Практические упражнения

>  AdobePhotoshop

>  Inkscape


>  3D Max

Рис. 9. Скриншот страницы «Практические упражнения» на сайте «Обработка компьютерной графики»

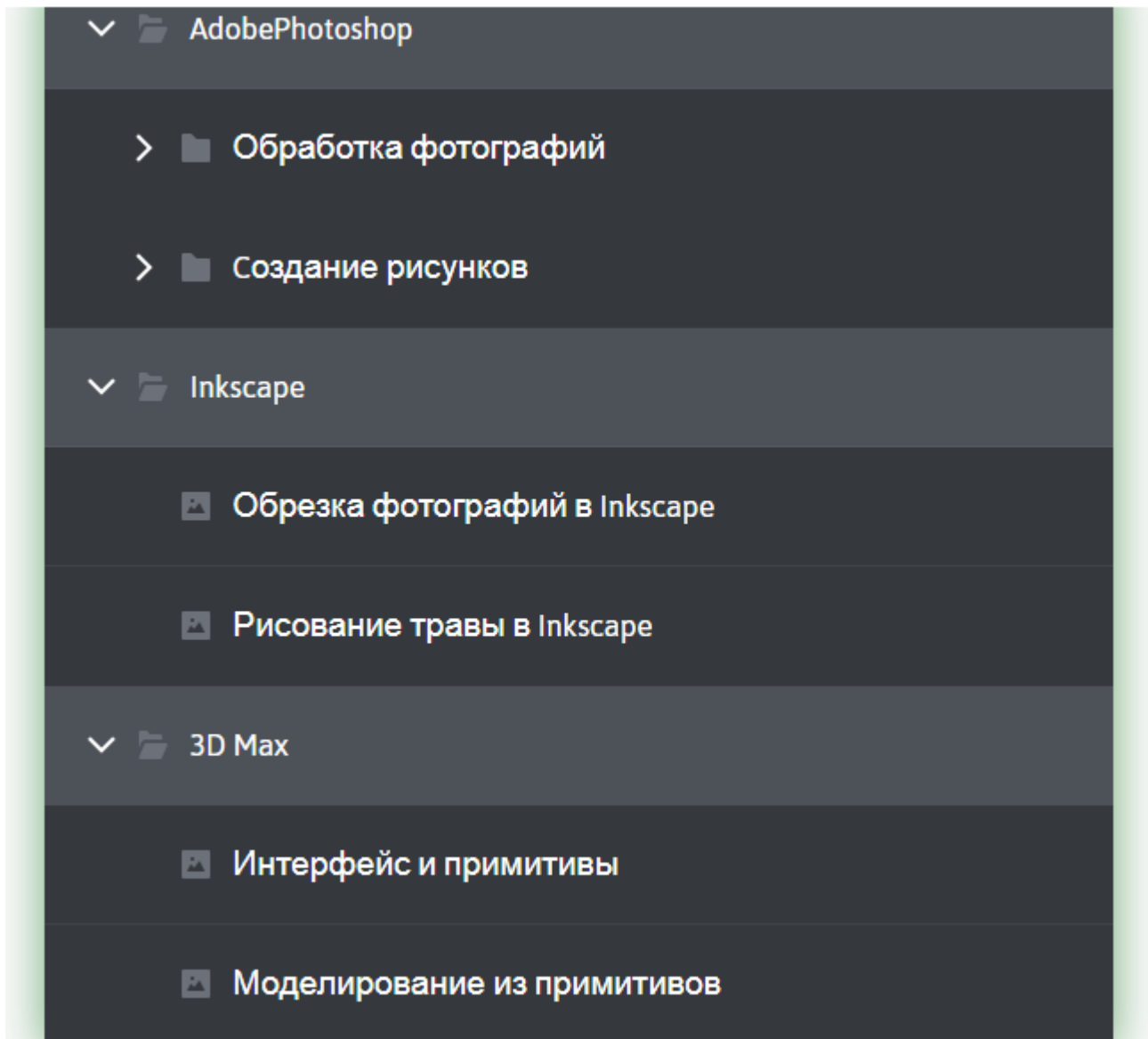


Рис. 10. Скриншот раскрывающегося списка упражнений на странице «Практические упражнения» на сайте «Обработка компьютерной графики»

Обработка компьютерной графики

[ГЛАВНАЯ](#)[ОПИСАНИЕ ПРОГРАММ](#)[ПРАКТИЧЕСКИЕ УПРАЖНЕНИЯ](#)[ВИДЕО-УРОКИ](#)[КОНТАКТЫ](#)

Обрезка фотографий в Inkscape

Inkscape — это векторный графический редактор, то есть для редактирования фотографий его обычно не используют. Между тем, в Inkscape есть некоторые интересные инструменты обработки растровых изображений, которые вполне заслуживают внимания.

Простая обрезка фотографий в Inkscape

1. Для начала, нам нужно проимпортировать фотографию в Inkscape. Как всем наверняка известно, Inkscape вполне умеет импортировать в текущий документ растровые изображения и работать с ними наряду с векторными объектами. Вы можете просто перетащить файл изображения с исходной папки в окно Inkscape, либо же выбрать в меню File → Import. После этого появится окошко с дополнительными опциями, в котором вы можете, например, выбрать опцию встраивания (Embed)

Рис. 11. Скриншот одного из упражнений на странице «Практические упражнения» на сайте «Обработка компьютерной графики»

Обработка компьютерной графики

[ГЛАВНАЯ](#)[ОПИСАНИЕ ПРОГРАММ](#)[ПРАКТИЧЕСКИЕ УПРАЖНЕНИЯ](#)[ВИДЕО-УРОКИ](#)[КОНТАКТЫ](#)

Видео-уроки

3D MAX для начинающих Урок 1 Интерфейс

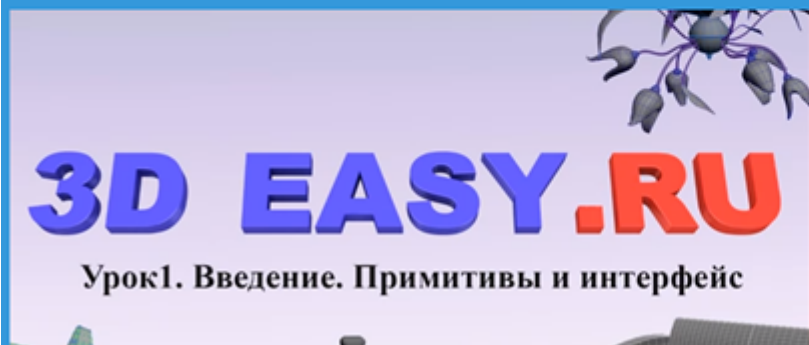


Рис. 12. Скриншот страницы «Видео-уроки» на сайте «Обработка компьютерной графики»

Обработка компьютерной графики

ГЛАВНАЯ

ОПИСАНИЕ ПРОГРАММ

ПРАКТИЧЕСКИЕ УПРАЖНЕНИЯ

ВИДЕО-УРОКИ

КОНТАКТЫ

Контакты

E-mail администратора сайта: Admin1@site.ru

E-mail технического отдела: techno-otdel@site.ru

Справка по телефону: 4-23-19, 5-89-14

Форма для отправки сообщения специалисту

Фамилия Имя Отчество:

Номер телефона:

Email:

Рис. 12. Скриншот страницы «Контакты» на сайте «Обработка компьютерной графики»

1. Термин «слабоструктурированные данные» означает такие данные, которые в отличие от данных в БД не имеют регулярной структуры, определяемой с помощью предписывающей схемы. [↑](#)
2. **Метаданные** — свойства данных, определяющие их структуру, допустимые значения и способы их представления, взаимосвязи с другими данными, размещение и другие характеристики данных, которые помогают правильно их интерпретировать и использовать. Иначе говоря, это данные о данных. [↑](#)
3. **3 Информационный ресурс** — используемые в приложениях данные, которые представлены в базах данных, базах знаний, на Web-сайтах, в отдельных файлах различной природы или в процедурной форме с помощью продуцирующих их программных средств. [↑](#)