Содержание:

Введение

Компьютеры уверенно вошли в жизнь людей и продолжают набирать популярность. Причины всеобщего увлечения ими просты: они дают возможность интересно проводить досуг, общаться с друзьями из любого города в чатах и на форумах, искать нужную информацию в Интернете.

Качественные изменения арсенала научно-технических средств, которые состоялись в последние годы под воздействием научно-технического прогресса, открыли новые возможности в работе с информацией, поставили перед человечеством задание усовершенствования старых и разрабатывание новых методик, технологий ее проработки.

Достаточно перспективными для использования, обработки и хранения информации есть гипертекстовые технологии. Они позволяют фиксировать и прорабатывать информацию в кратчайшие сроки, избегая сложных лабораторных процессов. Именно гипертекстовые технологии ставят работу с информацией на высокий уровень.

Целью данной курсовой работы является рассмотреть применение гипертекстовых языков.

Для достижения цели были поставлены следующие задачи:

1. Изучить литературу по данной теме;
2. Произвести сравнительный анализ литературы по данной тем;
3. Рассмотреть технологию гипертекста
4. Рассмотреть основные конструкции языков гипертекстовой разметки
5. Проанализировать практическое применение языков разметки гипертекста

Для написания работы были использованы 12 литературных источника, в том числе периодическая литература. Практическая значимость работы заключается в том, что материал может быть использован для дальнейших исследований в области информатики и смежных ей дисциплин.

Глава 1. Теоретические основы гипертекстовой технологии

История развития языков гипертекстовой разметки

появление первых версий языков гипертекстовой разметки относят к 1986 году, а в 1991 году HTML был существенно доработан и стал использоваться именно для передачи гипертекста по просторам всемирной паутины

Первая версия языка разметки гипертекста HTML была создана на основе стандарта обобщенного языка разметки SGML (Standard Generalized Markup Language), который в некотором роде можно считать прообразом расширяемого языка разметки данных XML (eXtensible Markup Language). Стандарт XML в наше время приобрел огромную популярность благодаря большому количеству своих расширений, используемых в компьютерных технологиях.

В результате существуют аббревиатуры SGML, HTML, XML и XHTML. Рассмотрим что каждая означает. SGML это не что иное, как набор правил, на основе которых можно строить любые языки разметки. HTML и есть один из этих языков - приложение SGML. Другими словами, SGML определяет то, как должны выглядеть элементы разметки, а HTML - какие именно должны быть элементы и как они должны интерпретироваться браузерами. XHTML, в свою очередь, является приложением XML, а сам XML ни что иное, как упрощенный вариант SGML. Языки HTML и XHTML, не смотря на то, что очень внешне похожи, имеют существенные скрытые отличия, которые, по большей части, заключаются в принципе их обработки[1].

Официальной спецификации HTML 1.0 не существует. До 1995 года существовало множество неофициальных стандартов HTML. Чтобы стандартная версия отличалась от них, ей сразу присвоили второй номер.

Версия 3 была предложена Консорциумом всемирной паутины (W3C) в марте 1995 года и обеспечивала много новых возможностей, таких как создание таблиц, «обтекание» изображений текстом и отображение сложных математических формул. Даже при том, что этот стандарт был совместим со второй версией, реализация его была сложна для браузеров того времени. Версия 3.1 официально

никогда не предлагалась, и следующей версией стандарта HTML стала 3.2, в которой были опущены многие нововведения версии 3.0, но добавлены нестандартные элементы, поддерживаемые браузерами Netscape Navigator и Mosaic[2].

В версии HTML 4.0 произошла некоторая «очистка» стандарта. Многие элементы были отмечены как устаревшие и нерекомендованные. В частности, элемент font, используемый для изменения свойств шрифта, был помечен как устаревший (вместо него рекомендуется использовать таблицы стилей CSS).

В 1998 году консорциум Всемирной паутины начал работу над новым языком разметки, основанном на HTML 4, но соответствующим синтаксису XML. Впоследствии новый язык получил название XHTML. Первая версия XHTML 1.0 одобрена в качестве Рекомендации консорциума Всемирной паутины 26 января 2000 года.

Планируемая версия XHTML 2.0 должна была разорвать совместимость со старыми версиями HTML и XHTML, но 2 июля 2009 года консорциум Всемирной паутины объявил, что полномочия рабочей группы XHTML2 истекают в конце 2009 года. Таким образом, была приостановлена вся дальнейшая разработка стандарта XHTML 2.0

В 2014 году вышла новая версия HTML 5.0. Цель разработки HTML5 — улучшение уровня поддержки мультимедиа-технологий с одновременным сохранением обратной совместимости, удобочитаемости кода для человека и простоты анализа для парсеров.

Определение содержания основных понятий

Гипертекст- это технология, которая базируется на средствах обработки больших, глубоко вложенных, структурированных, объединенных семантически, текстов и информации, организованных в виде фрагментов и принадлежащих к одной и той же системе объектов, которая расположена в вершине некоторой сети и выделяемая обычно цветом. Данная технология позволяет при машинной реализации моментально, нажатием одной или нескольких клавиш, вызывать и помещать в необходимое место просматриваемого или организуемого нового текста необходимые фрагменты гипертекста, которые "привязаны" к выделенному по цвету ключевому слову или словосочетанию, или нескольким слов.

Основными компонентами технологий, состоящих в применении гипертекстовой модели к информационным ресурсам, распределенным в Интернете, являются:

- URL - универсальный способ адресации ресурсов в сети;

- HTML - язык гипертекстовой разметки документов;

- HTTP (HyperText Transfer Protocol) - протокол обмена гипертекстовой информацией;

- дополнительные средства (CGI, Java, JavaScript).

Гипертекстовая база данных - это набор текстовых файлов, написанных на языке HTML, который определяет форму представления информации (разметка) и структуру связей этих файлов (гипертекстовые ссылки).

Такой подход предполагает наличие еще одной компоненты технологии - интерпретатора языка. В World Wide Web функции интерпретатора разделены между сервером гипертекстовой базы данных и интерфейсом пользователя.

Сервер, кроме обеспечения доступа к документам и реализации гипертекстовых ссылок, осуществляет также препроцессорную обработку документов, в то время как интерфейс пользователя проводит интерпретацию конструкций языка, связанных с представлением информации.

Универсальный идентификатор ресурсов (URL)[3]

Система универсальных идентификаторов ресурсов (URL) разработана для использования в системах Интернет и в ее основу заложены следующие принципы:

- расширяемость - новые адресные схемы должны были легко вписываться в существующий синтаксис;

- полнота - по возможности любая из существовавших схем должна была описываться посредством URL;

- читаемость - адрес должен легко пониматься человеком.

Формат URL включает:

- схему адреса;

- IP- или доменный адрес машины;

- номер TCP-порта;

- адрес ресурса на сервере (каталог или путь);

- имя HTML-файла и метку;

- критерий поиска данных.

Гипертекстовая разметка используется для указания, в какой части экрана и как должен отображаться текст, каким образом связаны между собой тексты, которые составляют гипертекстовую базу данных. Исходя из данных целей вводятся специальные управляющие символы. Текст с данными управляющими символами сохраняется в обычном текстовом файле в кодировке ASCII и может быть отредактирован почти любым текстовым редактором[4].

Язык гипертекстовой разметки HTML предложил Тим Бернерсон-Ли в 1989 г. До создания HTML был стандарт языка разметки для печатных документов - SGML (Standart Generalised Markup Language), который был взят в качестве основания HTML . Полагалось, что такое решение поддержит использование существующего программного обеспечения для истолкования нового языка гипертекстовой разметки[5].

В качестве элемента гипертекстовой базы данных для языка HTML был выбран обыкновенный текстовый файл, хранящийся средствами файловой системы операционной среды электронной вычислительной машины.

Глава 2.Основные конструкции языков гипертекстовой разметки

2.1 Язык гипертекстовой разметки HTML

За основу модели разметки документов в HTML принята теговая модель. *Теговая модель* описывает документ как совокупность контейнеров, каждый из которых начинается и заканчивается тегами. Т.е. документ HTML представляет собой не что иное, как обычный ASCII-файл, с добавленными в него управляющими HTML-кодами (тегами).

Теги HTML-документов в большинстве своем просты для понимания и использования, ибо они образованы с помощью общеупотребительных слов английского языка, понятных сокращений и обозначений. HTML-тег состоит из имени, за которым может следовать необязательный список атрибутов тега. Текст тега заключается в угловые скобки (< и >)[6]. Простейший вариант тега - имя, заключенное в угловые скобки, например <HEAD> или . Для более сложных тегов характерно различие атрибутов, которые могут иметь конкретные значения, определенные автором для видоизменения функции тега.

Атрибуты тега следуют за именем и отделяются друг от друга одним или несколькими знаками табуляции, пробелами или символами возврата к началу строки. Порядок записи атрибутов в теге значения не имеет. Значение атрибута, если таковое имеется, следует за знаком равенства, стоящим после имени атрибута. Если значение атрибута - одно слово или число, то его можно просто указать после знака равенства, не выделяя дополнительно. Все остальные значения необходимо заключать в одинарные или двойные кавычки, особенно если они содержат несколько разделенных пробелами слов. Длина значения атрибута ограничена 1024 символами. Регистр символов в именах тегов и атрибутов не учитывается, чего нельзя сказать о значениях атрибутов. Например, особенно важно использовать нужный регистр при вводе URL других документов в качестве значения атрибута HREF.

Чаще всего HTML-теги состоят из начального и конечного компонентов, между которыми размещаются текст и другие элементы документа. Имя конечного тега идентично имени начального, но перед именем конечного тега ставится косая черта (/) (например, для тега стиля шрифта - курсив закрывающая пара представляет собой , для тега заголовка <TITLE> закрывающей парой будет </TITLE>). Конечные теги никогда не содержат атрибутов. По своему значению теги близки к понятию скобок "begin/end" в универсальных языках программирования, которые задают области действия имен локальных переменных и т. п. Теги определяют область действия правил интерпретации текстовых тегов документа.

При использовании вложенных тегов в документе следует соблюдать особую аккуратность. Вложенные теги нужно закрывать, начиная с самого последнего и двигаясь к первому. Некоторые HTML-теги не имеют конечного компонента, поскольку они являются автономными элементами. Например, тег изображения <IMG>, который служит для вставки в документ графического изображения, конечного компонента не требует. К автономным тегам также относятся разрыв

строки (
), горизонтальная линейка (<HR>) и теги, содержащие такую информацию о документе, которая не влияет на его отображаемое содержимое, например теги <META> и <BASE>.

В некоторых случаях конечные теги в документе можно опускать. Большинство браузеров реализованы так, что при обработке текста документа начальный тег воспринимается как конечный тег предыдущего. Самый распространенный тег такого типа - тег абзаца <P>. Поскольку он используется в документе очень часто, то его обычно ставят только в начале каждого абзаца. Когда один абзац заканчивается, следующий тег <P> сигнализирует браузеру о том, что нужно завершить данный абзац и начать следующий. Большинство авторов тегом конца абзаца вообще не пользуются.

Есть и другие конечные теги, без которых браузеры отлично работают, например конечный тег </HTML>. Тем не менее, рекомендуется включать по возможности больше конечных тегов, чтобы избежать путаницы и ошибок при воспроизведении документа.

Общая схема построения контейнера в формате HTML может быть записана в следующем виде:

"контейнер" := <"имя тега" "список атрибутов">

содержание контейнера

</"имя тега">

2.2. Расширяемый язык разметки XML

Язык XML был создан для хранения, транспортировки и обмена данными, с его помощью можно реализовать обмен данными между различными системами. XML документ состоит из частей, называемых элементами. Элементы составляют основу XML-документов. Они образуют структуры, которые можно обрабатывать программно или с помощью таблиц стилей. Элементы размечают именованные разделы информации. Элементы строятся с помощью тегов разметки, обозначающих имя, начало и конец элемента. Элементы могут быть вложены друг в друга, на верхнем уровне находится элемент, называемый элементом документа или корневым элементом в котором содержатся остальные элементы. Например:

```
<?xml version="1.0"?>

<planets>

<planet ID="1">

<name>Mercury</name>

</planet>

<planet ID="2">

<name>Venus</name>

</planet>

<!-- There are more planets. -->

</planets>
```

Документ XML может располагаться в одном или нескольких файлах, причем некоторые из них могут находиться на разных машинах.

В XML используется специальная разметка для интеграции содержимого разных файлов в один объект, который можно охарактеризовать как 9 логическую структуру. Благодаря тому, что документ не ограничен одним файлом, XML позволяет создавать документ из частей, которые могут располагаться где угодно.

Документ XML обычно содержит следующие разделы:

• XML-декларация;

• Пролог;

• Элементы;

• Атрибуты;

• Комментарии

XML-декларация обычно находится в первой строке XML-документа. XML-декларация не является обязательной. Однако, если она существует, она должна располагаться в первой строке документа, и до нее не должно быть больше ничего, в том числе пробелов. XML-декларация в схеме документа состоит из следующих

элементов:

1. Номер версии:

<?xml version="1.0"?>.

Это обязательный аргумент. Текущая версия — 1.0.

2. Декларация кодировки:

<?xml version="1.0" encoding="UTF-8"?>

Это необязательный параметр. Если он используется, то декларация кодировки должна располагаться сразу после информации о версии в XML-декларации. Декларация кодировки должна содержать значение, представляющее собой существующую кодировку

символов.

3. Декларация автономности, например:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>.

Декларация автономности, как и декларация кодировки, необязательны. Если декларация автономности используется, то она должна стоять на последнем месте в XML-декларации.

Прологом называются данные, расположенные после открывающего тега документа или после корневого элемента. Он включает сведения, относящиеся к документу в целом — кодировка символов, структура документа, таблицы стилей.

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="book.xsl"?>

<!DOCTYPE book SYSTEM "schema.dtd">

<!--Some comments-->

В языке XML есть возможность включения в документ инструкций, которые несут определенную информацию для приложений, которые будут обрабатывать тот или иной документ. Инструкции по обработке в XML создаются следующим образом.[7]

<? Приложение Содержимое ?>

В XML инструкции по обработке заключаются в угловые кавычки со знаком вопроса. В первой части процессинговой инструкции определяется приложение или система, которой предназначена вторая часть этой инструкции или ее содержимое. При этом инструкции по обработке действительны только для тех приложений, которым они адресованы. Примером процессинговой инструкции может быть следующая инструкция:

<?serv cache-document?>

Элементы в XML документе отвечают за организацию информации и являются основными структурными единицами языка XML. Элементы оформляются следующим образом:

<ElementName> Содержимое элемента </ElementName>

Теги устанавливают границы вокруг содержимого элемента, если таковое имеется. У каждого элемента должно быть имя. Имена XML-элементов должны подчиняться следующим правилам:

Названия могут содержать буквы, цифры и другие символы;

• Названия не могут начинаться с цифры или знака препинания;

• Названия не могут начинаться с букв xml;

• В названии не должно быть пробелов.

• Нельзя допускать пробелов у кавычек (<);

• Имена элементов являются регистрозависимыми;

• Все элементы должны иметь закрывающий тэг[8].

В XML элементы могут быть двух типов – пустые и непустые. Пустые элементы не содержат в себе никаких данных, таких как текст или другие конструкции, и могут сокращенно записываться следующим образом:

<ElementName />

В XML документе обязательно должен присутствовать единственный корневой элемент, все остальные элементы являются дочерними по отношению к

единственному корневому элементу. При этом должен строго соблюдаться порядок вложенности элементов:

<person>

<givenName>Peter</givenName>

<familyName>Kress</familyName>

</person>

В данном случае элемент <person> содержит два других элемента, <givenName> и <familyName>. Элемент <givenName> содержит текст Peter, а элемент <familyName> — текст Kress.

В XML элементы могут содержать атрибуты с присвоенными им значениями, которые помещаются в одинарные или двойные кавычки. Атрибуты позволяют добавлять сведения об элементе с помощью пар «имя-значение». Атрибуты часто используются для определения тех свойств элементов, которые не считаются содержимым элемента, хотя в некоторых случаях (например, HTML-элемент img) содержимое элемента определяется значениями атрибута. Атрибуты могут отображаться в открывающих или пустых тегах, но не в закрывающих тегах[9]. Атрибут для элемента задается следующим образом:

<myElement attribute="value" ></myElement>

Синтаксические правила создания атрибута:

• Декларируются в открывающем тэге;

• Количество атрибутов не ограничено;

• Несколько атрибутов разделяются пробелами;

• Атрибут состоит из имени и значения

• Каждое имя должно быть уникально в рамках одного элемента;

• Нельзя использовать пробелы в именах атрибутов;

• Значение атрибута должно быть в кавычках.

Значение атрибутов может заключаться как в одинарные, так и в двойные кавычки. Возможно также использование одних кавычек внутри других, например:

<myElement attribute="value" > </myElement>

<myElement attribute='value' ></myElement>

<myElement attribute='"value"' ></myElement>

2.3. Расширяемый язык разметки гипертекста XHTML

XHTML представляет собой семейство имеющихся на данный момент и могущих появиться в будущем типов документов и модулей, являющихся копиями, подмножествами или расширениями языка HTML 4. Семейство типов документов XHTML базируется на XML и предназначено для работы с пользовательскими агентами на базе.

Преимущества XHTML:

- Разработчики документов и создатели пользовательских агентов постоянно открывают новые способы выражения своих идей в новой разметке. В XML ввод новых элементов или атрибутов достаточно прост. Семейство XHTML разработано так, чтобы принимать расширения путем модулей и технологий XHTML для разработки новых соответствующих XHTML модулей (описанных в готовящейся спецификации Модуляризации XHTML). Модули позволят комбинировать существующие и новые наборы функций при разработке содержимого и создании новых пользовательских агентов.
- Постоянно вводятся альтернативные методы доступа в Интернет. По некоторым оценкам, в 2002 году 75% обращений к документам в Интернет будет выполняться с альтернативных платформ. Семейство XHTML создавалось с учетом общей совместимости пользовательских агентов. С помощью нового механизма профилирования пользовательских агентов и документов серверы, прокси и пользовательские агенты смогут преобразовывать содержимое наилучшим образом. В конечном счете станет возможной разработка соответствующего XHTML содержимого, пригодного для любого соответствующего XHTML пользовательского агента.

Строго конформный документ XHTML - это документ, которому необходимы только возможности, описанные в настоящей спецификации как обязательные. Такой документ должен соответствовать всем следующим критериям:

1. Он должен проходить проверку корректности в соответствии с одним из трех DTD.
2. Корневым элементом документа должен быть элемент <html>.
3. Корневой элемент документа должен назначать пространство имен XHTML с использованием атрибута xmlns [XMLNAMES]. Пространство имен для XHTML определено в http://www.w3.org/1999/xhtml.
4. В документе до корневого элемента должно иметься объявление DOCTYPE. Открытый идентификатор, включаемый в объявление DOCTYPE, должен ссылаться на одно из трех DTD, приведенных в приложении A, с помощью соответствующего формального открытого идентификатора. Системный идентификатор может изменяться, отражая соглашения, принятые в локальной системе.

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"DTD/xhtml1-strict.dtd">

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"DTD/xhtml1-transitional.dtd">

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"

"DTD/xhtml1-frameset.dtd">

вот пример минимального документа XHTML.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ru" lang="ru">

<head>

<title>Виртуальная библиотека</title>

</head>

<body>

Переехала по адресу vlib.org.

</body>

</html>

Обратите внимание, что в данном примере включено объявление XML. Такое объявление XML не является обязательным для всех документов XML. Авторам документов XHTML настоятельно рекомендуется использовать объявления XML во всех своих документах. Такое объявление обязательно, если кодировка символов документа отличается от используемых по умолчанию UTF-8 или UTF-16.

Глава 3. Практическое применение языков разметки гипертекста

3.1. Создание документов в стандарте HTML

Создадим в HTML следующую таблицу:

Название	Автор	Тип переплета	Количество страниц	Цена

Для этого откроем блокнот и наберем следующий код:

<HTML>

<HEAD>

<TITLE>Каталог</TITLE>

</HEAD>

<BODY>

<H2>Книжный каталог</H2>

<TABLE BORDER="1" CELLPADDING="5">

<THEAD>

<TH>Название</TH>

<TH>Автор</TH>

<TH>Тип переплета</TH>

<TH>Количество страниц</TH>

<TH>Цена</TH>

</THEAD>

<TR ALIGN="center">

<TD>< STYLE="font-style:italic"></SPAN></TD>

<TD>< </TD>

<TD</TD>

<TD ></TD>

<TD ></TD>

</TR>

</TABLE>

</BODY>

</HTML>

Рассмотрим подробнее что выполняют данные теги. Любой HTML-документ состоит из структуры:

<html>

<head> <!-- Техническая информация о документе -->

<title>...</title> <!-- Задаем заголовок документа -->

</head>

<body> <!-- Основная часть документа -->

</body>

</html>

Тег <H2> -задает шрифт заголовка

Тег <TABLE> с атрибутами BORDER="1" CELLPADDING="5" задает таблицу с рамкой в 1 пиксель.

Теги <TD> и <TR> задают строки и столбцы.

Чтобы увидеть результат работы в браузере необходимо сохранить файл с расширением .html

Книжный каталог

Название	Автор	Тип переплета	Количество страниц	Цена

Рисунок 1. Выполнение кода в браузере.

Так же в HTML можно добавлять языки программирования PHP, JavaScript и другие.

HTML – код не может отправлять данные формы, поэтому мы использовали язык программирования php. PHP позволяет избежать этих проблем. Когда посетитель заполняет форму и нажимает кнопку отправить PHP-код сам отправляет данные формы администратору сайту. Кроме этого он генерирует для посетителей специальную HTML—страницу с подтверждением о том, что данные отправлены удачно.

Отправкой данных из формы занимается сервер, а не сам пользователь. Это дает администратору сайта гарантию, что он получит все данные на электронную почту.

На рисунках 2 и 3 показана схема передачи информации.

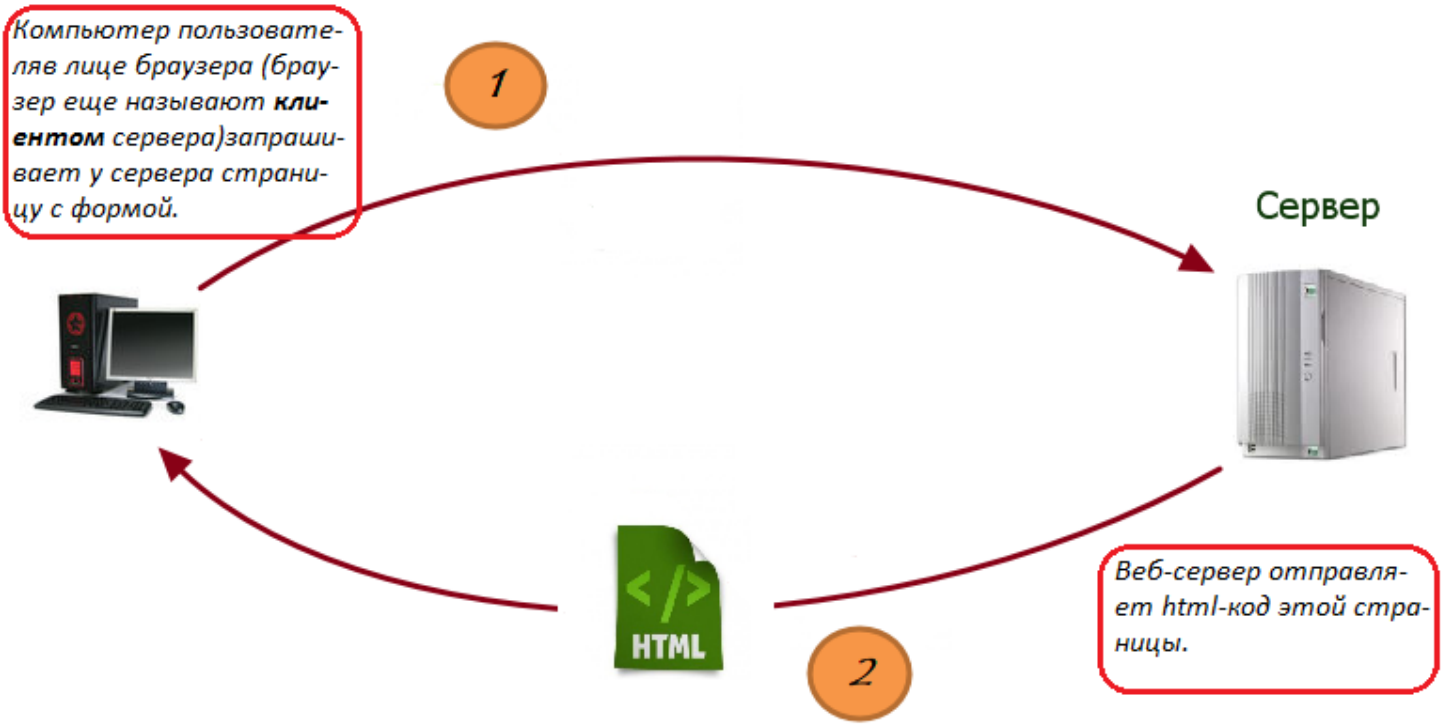

Рисунок 2. Пользователь запрашивает страницу с формой у сервера

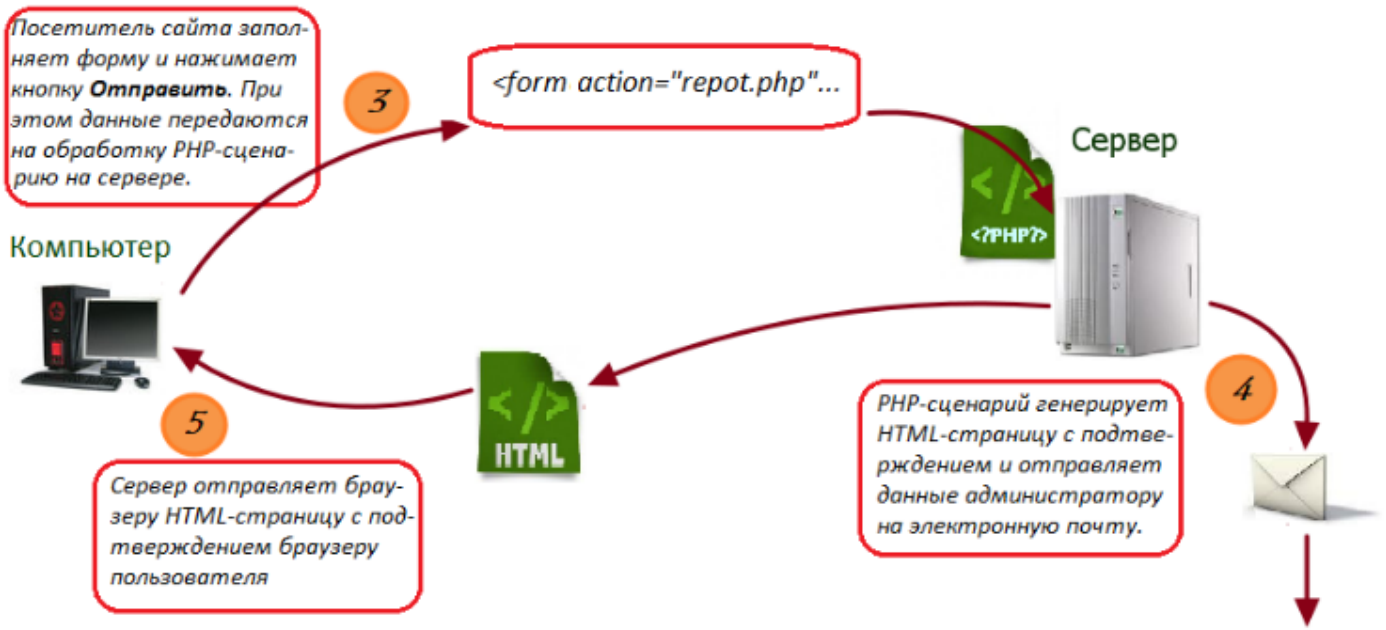

Рисунок 3. Передача данных на электронную почту

Таким образом, учитывая схему передачи данных был реализован php-сценарий, передающий информацию на электронную почту администратора..

Ниже описаны некоторые аспекты программы.

В тег $kuda помещается адрес, на который будет передаваться информация.

В тег $zagolovok помещается название формы.

$pol=$_POST['pol']; разделяет форму на секции. Для каждого вопроса своя секция.

```
if ($pol == men)

{

$sex = 'Мужской';

}

else if ($pol == women)

{
$sex = 'Женский';

}

else

{$sex = 'Не указан';}
```

Цикл условия, звучит он так: “Если ответ на вопрос “Ваш пол” не “мужской”, то значение становится “женский”, но если он не женский, то присваивается значение “Не указан”. Следующие 15 вопросов состоят из такого же цикла.

$messages

Сообщение, которое придет на адрес, указанному в теге $kuda.

Все результаты приходят на почту администратора. На рисунке 5 показан пример заполненной формы.

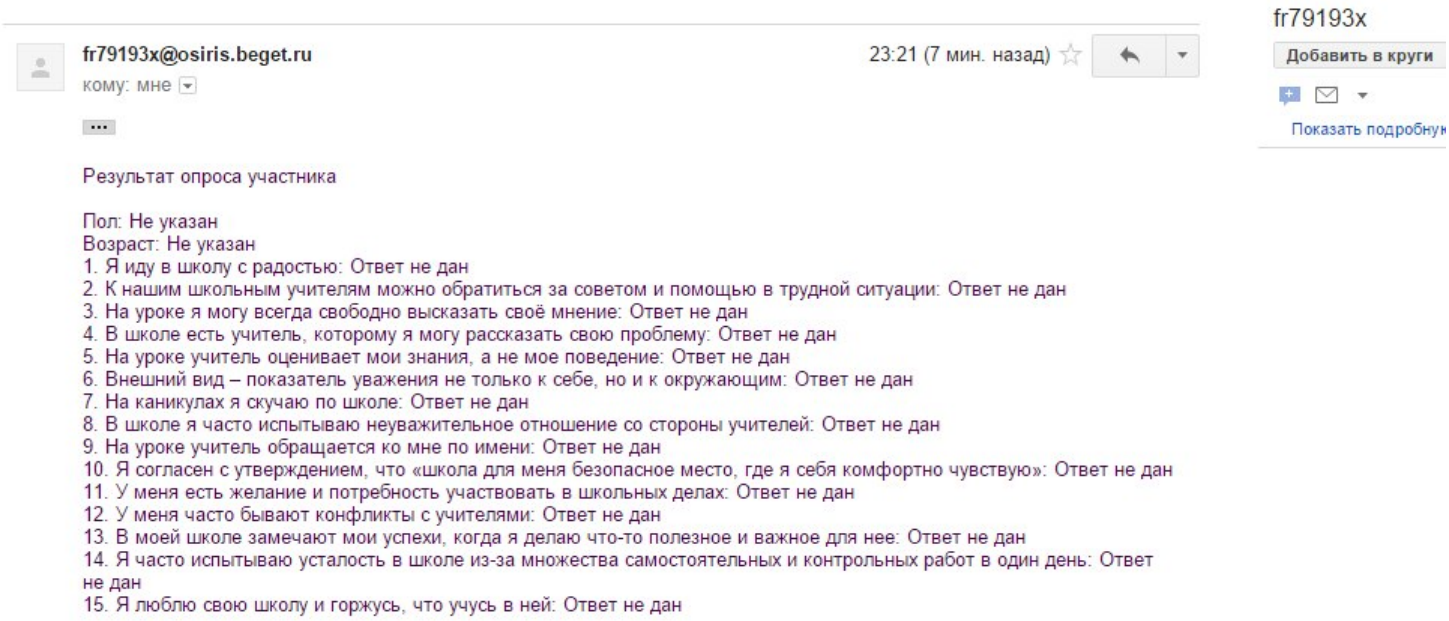

Рисунок 4. Результат заполненной почты

Ниже предоставлен фрагмент кода:

```
<?

$kuda='meison.2011@mail.ru';

$zagolovok='Результат опроса';

$headers='Content-type: text; charset="utf-8"';

if (isset($_POST['send'])){

$pol=$_POST['pol'];

$age=$_POST['age'];

$vp1=$_POST['vp1'];

$vp2=$_POST['vp2'];

$vp3=$_POST['vp3'];

$vp4=$_POST['vp4'];

$vp5=$_POST['vp5'];
```

```
$vp6=$_POST['vp6'];
$vp7=$_POST['vp7'];
$vp8=$_POST['vp8'];
$vp9=$_POST['vp9'];
$vp10=$_POST['vp10'];
$vp11=$_POST['vp11'];
$vp12=$_POST['vp12'];
$vp13=$_POST['vp13'];
$vp14=$_POST['vp14'];
$vp15=$_POST['vp15'];
if ($pol == men)
{
$sex = 'Мужской';
}
else if ($pol == women)
{
$sex = 'Женский';
}
Else
{
$sex = 'Не указан';
}
$age = trim($age);
```

```
if (empty($age))
{
$old = 'Не указан';
}else
{
$old = $age;
}
if ($vp1 == da)
{
$q1 = 'Да';
}
else if ($vp1 == net)
{
$q1 = 'Нет';
}
Else
{
$q1 = 'Ответ не дан';
}
if ($vp2 == da)
{
$q2 = 'Да';
}
```

```
else if ($vp2 == net)
{
$q2 = 'Нет';
}
Else
{
$q2 = 'Ответ не дан';
}
if ($vp3 == da)
{$q3 = 'Да';
}else if ($vp3 == net)
{
$q3 = 'Нет';
}else
{
$q3 = 'Ответ не дан';
}
if ($vp4 == da)
{
$q4 = 'Да'; }
else if ($vp4 == net)
{
$q4 = 'Нет';
```

```
}
Else
{
$q4 = 'Ответ не дан';
}
if ($vp5 == da)
{
$q5 = 'Да'; }
else if ($vp5 == net)
{
$q5 = 'Нет';
}
Else
{
$q5 = 'Ответ не дан';
}
if ($vp6 == da)
{
$q6 = 'Да';
}
else if ($vp6 == net)
{
$q6 = 'Нет';
```

```
}
Else
{
$q6 = 'Ответ не дан';
}
if ($vp7 == da)
{
$q7 = 'Да';
}
else if ($vp7 == net)
{
$q7 = 'Нет';
}
Else
{
$q7 = 'Ответ не дан';
}
if ($vp8 == da)
{$q8 = 'Да';}
else if ($vp8 == net)
{
$q8 = 'Нет';
}
```

```
Else
{
$q8 = 'Ответ не дан';
}
if ($vp9 == da)
{
$q9 = 'Да';
}else if ($vp9 == net)
{$q9 = 'Нет';
}
Else
{
$q9 = 'Ответ не дан';
}
if ($vp10 == da)
{
$q10 = 'Да';
}
else if ($vp10 == net)
{
$q10 = 'Нет';
}
Else
```

```
{
$q10 = 'Ответ не дан';
}
if ($vp11 == da)
{
$q11 = 'Да';
}
else if ($vp11 == net)
{$q11 = 'Нет';
}else
{
$q11 = 'Ответ не дан';
}
if ($vp12 == da)
{
$q12 = 'Да';
}
else if ($vp12 == net) {
$q12 = 'Нет';
}
Else
{
$q12 = 'Ответ не дан';
```

```
}
if ($vp13 == da)
{$q13 = 'Да';
}
else if ($vp13 == net)
{
$q13 = 'Нет';
}
Else
{
$q13 = 'Ответ не дан';
}
if ($vp14 == da)
{
$q14 = 'Да';
}
else if ($vp14 == net)
{
$q14 = 'Нет';
}
Else
{
$q14 = 'Ответ не дан';
```

```
}

if ($vp15 == da)

{

$q15 = 'Да';

}

else if ($vp15 == net)

{

$q15 = 'Нет';

}

else

{

$q15 = 'Ответ не дан';

}

$messages = "Результат опроса участника\n\nПол: ".$sex."\nВозраст: ".$old."\n1. Я иду в школу с радостью: ".$q1."\n2. К нашим школьным учителям можно обратиться за советом и помощью в трудной ситуации: ".$q2."\n3. На уроке я могу всегда свободно высказать своё мнение: ".$q3."\n4. В школе есть учитель, которому я могу рассказать свою проблему: ".$q4."\n5. На уроке учитель оценивает мои знания, а не мое поведение: ".$q5."\n6. Внешний вид – показатель уважения не только к себе, но и к окружающим: ".$q6."\n7. На каникулах я скучаю по школе: ".$q7."\n8. В школе я часто испытываю неуважительное отношение со стороны учителей: ".$q8."\n9. На уроке учитель обращается ко мне по имени: ".$q9."\n10. Я согласен с утверждением, что «школа для меня безопасное место, где я себя комфортно чувствую»: ".$q10."\n11. У меня есть желание и потребность участвовать в школьных делах: ".$q11."\n12. У меня часто бывают конфликты с учителями: ".$q12."\n13. В моей школе замечают мои успехи, когда я делаю что-то полезное и важное для нее: ".$q13."\n14. Я часто испытываю усталость в школе из-за множества самостоятельных и контрольных работ в один день: ".$q14."\n15. Я люблю свою школу и горжусь, что учусь в ней: ".$q15;
```

```
if (mail($kuda,$zagolovok,$messages,$headers)){

echo "<meta http-equiv='refresh' content='0; URL='/glavnaya.stranica.html'>";

}
}?>
```

3.2. Создание документов в стандарте XML

Заполним выше созданную таблицу данными с помощью стандарта XML.

Создадим файл следующего содержания с расширением .xml:

```
<?xml version="1.0" encoding="windows-1251" standalone="yes"?>

<!-- File Name: Каталог.xml -->

<КАТАЛОГ>

<КНИГА>

<НАЗВАНИЕ>Приключения Гекльберри Финна</НАЗВАНИЕ>

<АВТОР>Марк Твен</АВТОР>

<ТИП_ПЕРЕПЛЕТА>мягкий</ТИП_ПЕРЕПЛЕТА>

<СТРАНИЦЫ>298</СТРАНИЦЫ>

<ЦЕНА>549</ЦЕНА>

</КНИГА>

<КНИГА>

<НАЗВАНИЕ>Моби Дик</НАЗВАНИЕ>

<АВТОР>Герман Мелвилл</АВТОР>

<ТИП_ПЕРЕПЛЕТА>твердый</ТИП_ПЕРЕПЛЕТА>

<СТРАНИЦЫ>605</СТРАНИЦЫ>
```

<ЦЕНА>495</ЦЕНА>

</КНИГА>

<КНИГА>

<НАЗВАНИЕ>Унесенные ветром</НАЗВАНИЕ>

<АВТОР>Маргарет Митчелл</АВТОР>

<ТИП_ПЕРЕПЛЕТА>твердый</ТИП_ПЕРЕПЛЕТА>

<СТРАНИЦЫ>853</СТРАНИЦЫ>

<ЦЕНА>725</ЦЕНА>

</КНИГА>

<КНИГА>

<НАЗВАНИЕ>Легенда Сонной Лощины</НАЗВАНИЕ>

<АВТОР>Вашингтон Ирвинг</АВТОР>

<ТИП_ПЕРЕПЛЕТА>твердый</ТИП_ПЕРЕПЛЕТА>

<СТРАНИЦЫ>98</СТРАНИЦЫ>

<ЦЕНА>295</ЦЕНА>

</КНИГА>

<КНИГА>

<НАЗВАНИЕ>Приключения Тома Сойера</НАЗВАНИЕ>

<АВТОР>Марк Твен</АВТОР>

<ТИП_ПЕРЕПЛЕТА>мягкий</ТИП_ПЕРЕПЛЕТА>

<СТРАНИЦЫ>243</СТРАНИЦЫ>

<ЦЕНА>249</ЦЕНА>

</КНИГА>

</КАТАЛОГ>

Рассмотрим что выполняет данный код.

<?xml version="1.0" encoding="windows-1251" standalone="yes"?>- задает версию стандарта и его кодировку.

В XML документе, как правило, определяется хотя бы один элемент, называемый корневым и с него программы-анализаторы начинают просмотр документа. В приведенном примере этим элементом является <КАТАЛОГ>

Набором всех элементов, содержащихся в документе, задается его структура и определяются все иерархическое соотношения. В <КАТАЛОГ> входят элемент <КНИГА> со своими под элементами <ТИП_ПЕРЕПЛЕТА>, <СТРАНИЦЫ>,<ЦЕНА>.

Чтобы данные отобразились в браузере, испровим код файла .html.

<HTML>

<HEAD>

<TITLE>Каталог</TITLE>

<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7">

</HEAD>

<BODY>

<XML ID="dsoInventory" SRC="Каталог.xml"></XML>

<H2>Книжный каталог</H2>

<TABLE DATASRC="#dsoInventory" BORDER="1" CELLPADDING="5">

<THEAD>

<TH>Название</TH>

<TH>Автор</TH>

<TH>Тип переплета</TH>

<TH>Количество страниц</TH>

<TH>Цена</TH>

</THEAD>

<TR ALIGN="center">

<TD><SPAN DATAFLD="НАЗВАНИЕ"

STYLE="font-style:italic"></SPAN></TD>

<TD><SPAN DATAFLD="АВТОР"></SPAN></TD>

<TD><SPAN DATAFLD="ТИП_ПЕРЕПЛЕТА"></SPAN></TD>

<TD><SPAN DATAFLD="СТРАНИЦЫ"></SPAN></TD>

<TD><SPAN DATAFLD="ЦЕНА"></SPAN></TD>

</TR>

</TABLE>

</BODY>

</HTML>

Рассмотрим подробнее добавленные строки:

<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7"> - определяет метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем[10].

<XML ID="dsoInventory" SRC="Каталог.xml"> - предполагается, что XML-документ связан со страницей через фрагмент данных с именем dsoInventory.

В тег <TABLE> добавили атрибут DATASRC="#dsoInventory". Требуется для задания источника данных в атрибуте **datasrc**. Поддерживается IE начиная с 4.0.

В теги <TD> добавили атрибут SPAN DATAFLD="НАЗВАНИЕ", который будет считывать данные соответствующего поля

Запустим файл через браузер и увидим полученный результат.

Книжный каталог

Название	Автор	Тип переплета	Количество страниц	Цена
Приключения Гекльберри Финна	Марк Твен	мягкий	298	549
Моби Дик	Герман Мелвилл	твердый	605	495
Унесенные ветром	Маргарет Митчелл	твердый	853	725
Легенда Сонной Лощины	Вашингтон Ирвинг	твердый	98	295
Приключения Тома Сойера	Марк Твен	мягкий	243	249

Рисунок 5- Выполнение кода.

3.3. Создание документов в стандарте XHTML

Если разработать простой документ на XHTML,без использования других языков разметки, то разницы между HTML не увидеть. Однако если в документе есть инструменты, основанные на XML (таких как XSLT для преобразования документов) и можно заметить преимущества использования XHTML. Например, технология XForms позволит вам редактировать документы XHTML

Напишем обычный код для создания таблицы:

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

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ru">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

</head>

<body>

<table border="1" style="width: 100%; height: 100%;">

```
<tr>
<td colspan="2" style="height: 25px;">ячейка 1
</td>
</tr>
<tr>
<td>ячейка 2</td>
<td>ячейка 2</td>
</tr>
<tr>
<td colspan="2">ячейка 4</td>
</tr>
</table>
</body>
</html>
```

Внешний вид документа показан на рисунке

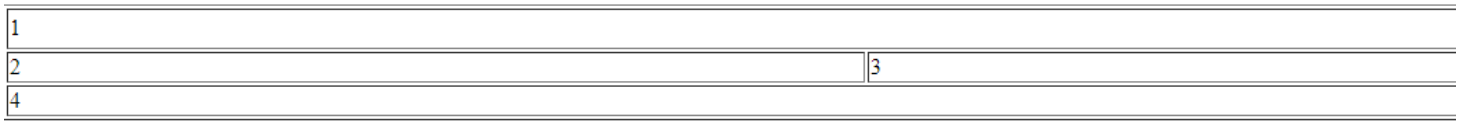

Рисунок 6. Внешний вид страницы

Существуют возможности преобразования XML файла в XHTML/

Заключение

Гипертекстовая технология позволяет быстро и точно осуществить поиск необходимой информации не только в рамках отдельных документов на

компьютера и web-страниц, созданных с их использованием, но и в локальных вычислительных сетях и всемирной сети Internet.

В данной работе рассматривается языки разметки гипертекста HTML, XML, XHTML.

Первая спецификация универсального и общедоступного языка разметки - HTML была утверждена в 1991 году. HTML стал стандартом и одновременно «корнем» для всех разрабатываемых Web страничек.

Язык XML – язык разметки, разработанный специально для размещения информации в World Wide Web, аналогично языку гипертекстовой разметки HTML (Hypertext Markup Language), который изначально стал стандартным языком создания Web-страниц.

Основное назначение языка XML – облегчить работу с документами в Web.

XHTML - это основанный на XML язык разметки гипертекста, максимально приближенный к текущим стандартам HTML. XHTML отличается от HTML строгостью написания кода.

Список использованной литературы

1. Антипов СВ. Современные технологии разработки Web-сайтов //Информатика и образование. - 2004. - №3.
2. Балафанов, Е.К. Новые информационные технологии. 30 уроков информатики / Е.К. Балафанов, Б.Б. Бурибаев, А.Б. Даулеткулов. - Алма-Ата. : Патриот, 2009. - 220 с
3. Велихов, А. В. Основы информатики и компьютерной техники: Учебное пособие А.В. Велихов: Букпресс, 2006. - 544 с.
4. Демин И.С. Концепция многослойного гипертекста // Модели экономических систем и информационные технологии. Сборник научных трудов, вып. VII - М.: Финансовая академия при Правительстве РФ, 2002 -725 с.
5. Максимов, Н.В. Современные информационные технологии: Учебное пособие / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. - М.: Форум, 2013. - 512 с.
6. Молли Э. Использование HTML и XHTML. Специальное издание. /Э. Молли, М.: Издательский дом “Вильямс” – 2004. - 736с
7. Логистика складирования: учебник: по специальности 080506 "Логистика и управление цепями поставок" / В. В. Дыбская. – М.: Инфра-М, 2012. – 557 с.

8. Новоселова Е.Н., Кадыров И.Р. Создание Web-страниц с помощью HTML //Информатика и образование. - 2005. - №1-3.
9. Пауэлл Т.А. Полное руководство по HTML. / Т. А. Пауэлл -Мн.: ООО "Попурри", 2001. - 912 с.
10. Титтел Э. . HTML 4 для "чайников". 5-е издание. / Э. Титтел, М. Бурмейстер, М.: Издательский дом “Диалектика -Вильямс” – 2007. - 368с.
11. Федеральный закон «Об информации, информационных технологиях и о защите информации» от 27.07.2006 № 149-ФЗ// СЗ РФ. 2006. № 31. Ст. 3448. (ред. от 06.04.2011 г.)
12. Холмогоров В. Основы Web-мастерства. Учебный курс. / В. Холмогоров - СПб.: Питер, 2001. - 352 с.

1. Максимов, Н.В. Современные информационные технологии: Учебное пособие / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. - М.: Форум, 2013. - 512 с. ↑

2. Титтел Э. . HTML 4 для "чайников". 5-е издание. / Э. Титтел, М. Бурмейстер, М.: Издательский дом “Диалектика -Вильямс” – 2007. - 368с. ↑

3. Титтел Э. . HTML 4 для "чайников". 5-е издание. / Э. Титтел, М. Бурмейстер, М.: Издательский дом “Диалектика -Вильямс” – 2007. - 368с. ↑

4. Максимов, Н.В. Современные информационные технологии: Учебное пособие / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. - М.: Форум, 2013. - 512 с. ↑

5. Правовая информатика и кибернетика. Учебник / Атанесян Г.А., Гаврилов О.А., Дёри П., Каблуков А.Г., и др.; Под ред.: Полевой Н.С. - М.: Юрид. лит., 2008. - 528 с. ↑

6. Антипов СВ. Современные технологии разработки Web-сайтов //Информатика и образование. - 2004. - №3. ↑

7. Антипов СВ. Современные технологии разработки Web-сайтов //Информатика и образование. - 2004. - №3. ↑

8. Холмогоров В. Основы Web-мастерства. Учебный курс. / В. Холмогоров - СПб.: Питер, 2001. - 352 с. ↑

9. Титтел Э. . HTML 4 для "чайников". 5-е издание. / Э. Титтел, М. Бурмейстер, М.: Издательский дом “Диалектика -Вильямс” – 2007. - 368с. ↑

10. Антипов СВ. Современные технологии разработки Web-сайтов //Информатика и образование. - 2004. - №3. ↑