

МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Хакасский государственный университет им. Н.Ф. Катанова»  
Колледж педагогического образования, информатики и права

ПЦК естественнонаучных дисциплин, математики и информатики

## РЕФЕРАТ

на тему:  
Основы HTML и его развитие

Автор реферата:

\_\_\_\_\_

(подпись)

Беляев И.

(инициалы, фамилия)

Специальность: 230115 - Программирование в компьютерных системах

Курс: II Группа: И-21

Зачет/незачет:

Руководитель:

\_\_\_\_\_

(подпись, дата)

Когумбаева О.П.

(инициалы, фамилия)

г. Абакан, 2017г.

## Содержание

Введение.....	3
1. История развития языка HTML.....	4
2. Понятие об HTML.....	7
2.1 Структура файла в формате HTML.....	7
2.2 Отличительный признак HTML-документа.....	10
3. Гипертекстовые ссылки.....	12
4. Заголовки.....	13
5. Форматирование текста и изменение стилей.....	14
6. Фреймы.....	15
7. Альтернативные средства составления Web-документов.....	18
Заключение.....	19
Библиографический список.....	20

## **Введение**

С появлением локальных сетей, подключением таких сетей к Интернет, созданием внутрикорпоративных, сетей, появляется возможность с любого рабочего места организации получить доступ к информационному ресурсу сети.

Технология World Wide Web, в переводе Всемирная паутина получила столь широкое распространение из-за простоты своих пользовательских интерфейсов. Принцип жми на то, что интересно, лежащий в основе гипертекста, интуитивно понятен. В технологиях WWW все ключевые понятия просматриваемого документа: слова, картинки - имеют возможность раскрыться новым документом, развивающим это понятие. Такой способ представления информации называется гипертекстом, а документы, представленные в таком виде – гипертекстовыми документами. Для описания этих документов используется специальный язык - язык описания гипертекстовых документов или HTML (англ. вариант Hyper Text Markup Language).

Актуальность: В результате использования HTML, работа многих пользователей стала иметь большее отношение к Сети. Многие коммерческие фирмы стали использовать Сеть для рекламы и сбыта своей продукции. Людям, занятым поиском работы, стал доступен и такой сервис: составить резюме в формате Web-страницы и разместить эту информацию в Сети. Важную роль Internet стал играть для научных, учебных и общественных организаций. Подтверждение этому легко найти, выйдя на просторы киберпространства.

Цель: Дать представление языка HTML, особенности его работы, возможности и правила форматирования текста, задания структуры документа, встраивания ссылок и мультимедиа-объектов.

Задача: Изучить историю развития языка HTML, структуру файлов, особенности, правила форматирования и альтернативы данному языку.

## **1. История развития языка HTML**

Свою историю HTML начинает с 1986 года, когда Международной организацией по стандартизации (ISO) был принят стандарт озаглавленный "Standard Generalized Markup Language" - SGML. Этот стандарт был посвящен описанию обобщенного мета языка, который позволял строить системы логической структурной разметки любых разновидностей текстов. Он соответствовал международному стандарту ISO 8879.

Создатели SGML стремились к тому, чтобы размеченный текст могла без труда интерпретировать любая программа, работающая на разных компьютерных платформах и устройствах вывода.

SGML является не готовой системой разметки текста, а определяет лишь синтаксис записи элементов разметки – тегов и их атрибутов, а также правила определения новых тегов и указания структурных отношений между ними.

Идеология SGML повлияла на многие компьютерные разработки, однако сам по себе язык не получил обширного распространения.

В 1991 году, британским ученым Тимом Бернерсом Ли, сотрудником Европейского института физики частиц (CERN) в Женеве, была разработана система передачи гипертекстовой информации через интернет. А за основу нового языка был взят SGML. Язык разметки гипертекста был назван - HTML (Hyper Text Markup Language) и он является до сих пор самым известным из приложений SGML. HTML в первую очередь был разработан для обмена научной и технической документации для использования людьми, не являющимися специалистами в области верстки. Путем определения небольшого набора структурных и семантических элементов, получались достаточно простые и в тоже время красиво оформленные документы. HTML успешно справился с проблемами SGML.

Как и положено изначально язык HTML разделял все особенности идеологии SGML. Но в 1993 году появляется версия языка HTML 1.2, которая имела сорок с небольшим тегов, три из которых не рекомендованы к

использованию, так как указывали на физические параметры предоставления документа, что противоречило идеологии SGML. Вся разметка была логической и только в описательной части стандарта можно было увидеть что-то типа ”в графических браузерах действие этого тега может передаваться жирным начертанием”.

Программа Mosaic была единственной в то время браузером поддерживающим графические возможности. Она была разработана в Национальном центре суперкомпьютерных приложений США (National Center for Supercomputer Applications - NCSA), там же, кстати, была разработана WWW (World Wide Web). По этой причине никаких противоречий между официальными стандартами и их реализацией в браузерах тогда не существовало.

В апреле 1994 года был образован Консорциум W3C (World Wide Web Consortium). Так как официальной спецификации HTML 1.0 не существовало, W3C начал заниматься подготовкой спецификации HTML следующей версии. Но чтобы стандартная версия отличалась от всех предыдущих, ей сразу присвоили номер 2.0. Разработка спецификации HTML 2.0 шла не спеша и лишь в сентябре 1995 года она была утверждена. Из больших дополнений был добавлен лишь механизм форм для отсылки информации с компьютера пользователя на сервер.

Тем временем Консорциум W3 в параллель со спецификацией 2.0 занимался обсуждением HTML 3.0. Она была предложена в марте 1995 года. Третья версия предлагала много новых возможностей: поддержка таблиц, обтекание изображения текстом, отображение сложных математических формул, примечания. Поддержка этого стандарта браузерами того времени была не удовлетворена.

Авторы HTML 3 добавили поддержку нового средства - иерархические стилевые спецификации (Cascading Style Sheets, CSS). Это нововведение нужно было для того, чтобы разрешить назревшее к тому времени противоречие между идеологией структурной разметки и потребностями пользователей, которым в первую очередь нужно было гибкость и обширные возможности визуального

представления. CSS имеет свой синтаксис и является формально независимой от HTML.

Следующей версией HTML стала 3.2 и в ней были опущены многие нововведения из версии 3.0.

18 декабря 1997 года была принята четвертая версия HTML. Она содержала как и третья много элементов, специфичных для отдельных браузеров. Хотя в HTML 4.0 произошла чистка элементов из предыдущих версий спецификаций. Многие элементы были помечены как устаревшие и не рекомендуемые к использованию. Вместо них рекомендовалось использовать таблицы стилей CSS.

HTML 4.01 утвердили 24 декабря 1999. Изменения принятые в нем были более значительные, чем, кажется на первый взгляд.

## **2. Понятие об HTML**

Html-документ - это текст, состоящий из HTML-кодов и основного текста документа. Для форматирования текста, задания структуры документа, встраивания ссылок и мультимедиа-объектов в HTML-документах используются специальные кодовые слова, которые называются дескрипторами разметки (тегами).

Итак, рассмотрим основные теги, входящие в каждый Web-документ.

Отличительный признак HTML-документа. Одним из принципов языка является многоуровневое вложение элементов. HTML является самым внешним, так как между его стартовым и конечным тегами должна находиться вся Web-страница.

Область заголовка Web-страницы. Иными словами, ее первая часть. Так же как и HTML, HEAD служит только для формирования общей структуры документа. Этот элемент включает в себе гипертекст, который определяет собственно Web-страницу. Эта та часть документа, которую разрабатывает автор страницы и которая отображается браузером. Соответственно, конечный тег этого элемента надо искать в конце HTML-файла. Внутри BODY можно использовать все элементы, предназначенные для дизайна Web-страницы. Внутри стартового тега элемента BODY можно расположить ряд атрибутов, обеспечивающих установки для всей страницы целиком.

### **2.1 Структура файла в формате HTML**

В первую очередь нам потребуется создать файл с расширением "htm" или "html" и открыть его для редактирования в текстовом редакторе (работая под Windows, можно использовать стандартную программу "Блокнот").

Текст для HTML-документа набирается в том виде, в котором его должен показать браузер, а при выделения фрагментов, которые предназначены для разметки документа (тегов), используются угловые скобки.

Любой документ HTML начинается с открывающего тега и заканчивается соответствующим ему закрывающим тегом. Закрывающий тег отличается от открывающего наличием косой черты перед названием тега. В языке HTML нет различий между большими и малыми буквами при записи тегов, однако принято записывать открывающий тег большими буквами, а закрывающий - малыми. Часть документа, которая находится в пределах от открывающего до соответствующего ему закрывающего тега, называется контейнером. Теги, у которых есть закрывающий и открывающий варианты записи называются парными, а те, которые используются только в одном виде, называются непарными. Весь файл web-страницы - это по сути дела один большой контейнер HTML (так как начинается и заканчивается ).

Внутри контейнера HTML последовательно записываются два контейнера:

1. (заголовок документа). Внутри этого контейнера записываются параметры, общие для всей страницы.
2. (тело документа). Этот контейнер содержит всё то, что должен показать браузер внутри своего окна при открытии документа. Если документ содержит только текст, то этот текст нужно записывать именно здесь, если есть другие элементы (графика, таблицы, ссылки и т. д.).

Если сравнить исходные тексты различных Web-страниц, можно легко увидеть сходство их структур. Это объясняется тем, что документы создаются по определенным правилам. В основу синтаксиса языка HTML лег стандарт ISO 8879:1986 «Information processing. Text and Office systems. Standard Generalized Markup Language (SGML)». Правда, существует большое различие между стандартом официальным и стандартом фактическим.

Ниже приведен шаблон типичного Web-документа. На его примере мы рассмотрим общие принципы построения HTML-страниц.

Для того чтобы понять структуру Web-страницы, нам необходимо подробно рассмотреть все элементы, входящие в приведенный листинг.

```
name="Author"  
content="Irina">
```

name=""Keywords""

content=""WWW, HTML, document, element"">

Переход в конец документа

Переход к ссылке 1

Заголовок1

Заголовок2

Заголовок3

Заголовок4

Здесь расположена ссылка 1

Здесь должен располагаться оригинальный текст Web-страницы

Переход в начало документа

Текст комментария.

В любом языке программирования есть конструкции, позволяющие создавать произвольные ремарки. HTML в этом смысле - не исключение. Текст, помещенный внутри COMMENT, игнорируется браузером. COMMENT может располагаться в любом месте кода Web-страницы. Без конечного тега, здесь по-видимому не обойтись: комментарий должен быть отделен от основного текста.

Существует, правда, одно ограничение: внутри комментария не должны располагаться другие элементы. Так должно быть, разумеется, только в том случае когда необходимо, чтобы все содержимое элемента COMMENT не отображалось на экране монитора. Если в комментарии будет присутствовать другой элемент, то его содержимое будет выведено на экран, отформатированное соответствующим образом. С этой проблемой связана другая, похожая: обычный текст не может содержать фрагменты, имеющие вид тегов. Для того чтобы включать в текст подобные конструкции, существует элемент PLAINTEXT.

Существует и другой способ обозначения комментария. Он заключается в использовании восклицательного знака и обрамлении текста комментария двойным тире.

Внутри подобной конструкции можно помещать и теги: они не будут восприниматься браузером.

## 2.2 Отличительный признак HTML-документа

Одним из принципов языка является многоуровневое вложение элементов. HTML является самым внешним, так как между его стартовым и конечным тегами должна находиться вся Web-страница. В принципе, этот элемент можно рассматривать как формальность. Он имеет атрибуты `version`, `lang` и `dir`, которыми в данном случае мало кто пользуется и допускает вложение элементов `HEAD`, `BODY` и `PLAINTEXT`, определяющих общую структуру Web-страницы. Естественно, что конечным тегом заканчиваются все гипертекстовые документы.

Область заголовка Web-страницы. Иными словами, ее первая часть. Так же как и HTML, `HEAD` служит только для формирования общей структуры документа. Этот элемент может иметь атрибуты `lang` и `dir` и допускает вложение элементов `TITLE`, `ISINDEX`, `BASE`, `NAME`, `LINK`, `NEXTID`.

Элемент для размещения заголовка Web-страницы. Строка текста, расположенная внутри, отображается не в документе, а в заголовке окна браузера. Эта особенность часто используется для организации поиска в WWW. Поэтому авторы, создающие Web-страницы, должны позаботиться о том, чтобы строка внутри `TITLE`, не будучи слишком длинной, достаточно отображала назначение документа.

Описание стиля некоторых элементов Web-страницы. В выше приведенном примере назначены шрифты для элементов `H2` и `CODE`. Естественно, что для каждого элемента существует стилевое оформление по умолчанию, поэтому употребление `STYLE` не обязательно.

Этот элемент содержит служебную информацию, которая не отображается при просмотре Web-страницы. Внутри него нет текста в обычном понимании, поэтому нет и конечного тега. Каждый элемент `NAME` содержит два основных атрибута, первый из которых определяет тип данных, а второй - содержание. Далее приведены несколько примеров meta-данных.

Дата, обозначающая «срок годности» документа:

```
name="Expires" content="Дата"
```

Адрес электронной почты:

`name="Reply-to" content="Имя@Адрес"`

Указание приложения, в котором был создан Web-документ:

`name="Generator" content="Название HTML-редактора"`

Атрибут `name` используется приложением-клиентом для получения дополнительной информации о Web-страницах и их упорядочения. Этот атрибут часто заменяют атрибутом `http-equiv`. Он используется сервером для создания дополнительных полей при выполнении запроса.

Кроме этого, элемент `NAME` может содержать URL. Шаблон соответствующего атрибута таков:

`URL="#_Точ132501276">2`

### 3. Гипертекстовые ссылки

Существует два шаблона для создания гипертекстовых ссылок.

Первый шаблон применяется в том случае, когда гиперссылка встречается в тексте. Атрибут href может указывать на ресурс Internet, файл на локальном диске или метку внутри текущей страницы. Текст, расположенный внутри элемента A, представляет собой видимую часть гиперссылки. Именно на нем должен щелкнуть пользователь, чтобы осуществить переход. Браузер выделяет этот фрагмент цветом, а после использования гиперссылки меняет цвет, чтобы обеспечить подсказку.

file://Диск:\Путь к файлу или file:///Диск:/Путь к файлу

Подобно многим другим элементам языка, элемент A требует использование атрибутов. Атрибут гиперссылки таков:

href=""URL"" или href=""Протокол://Адрес ссылки""

#### 4. Заголовки

Существует шесть уровней заголовков, которые обозначаются H1...H6. Заголовок уровня 1 самый крупный, а уровень 6 обеспечивает самый маленький заголовок. Для заголовков можно использовать атрибут, задающий выравнивание влево, по центру или вправо:

`align=""left""`

`align=""center""`

`align=""right""`

Элемент FONT может с успехом заменять элементы заголовка H1...H6. Для последних, например, не предусмотрена возможность указания цвета букв. Чтобы заголовок, созданный на основе элемента FONT, хорошо смотрелся, этот элемент необходимо комбинировать с другими: CENTER, B, I, P и т. д.

## 5. Форматирование текста и изменение стилей

Текст - единственный объект Web-страницы, который не требует специального определения. Иными словами, произвольные символы интерпретируются по умолчанию как текстовые данные. Но для форматирования текста существует большое количество элементов.

Элемент абзаца (paragraph) - один из самых полезных. Он позволяет использовать только начальный тег, так как следующий элемент P обозначает конец предыдущего и начало следующего абзаца. Конечный тег удобно использовать в тех случаях, когда по смыслу необходимо обозначить конец абзаца. Вместе с элементом P можно использовать атрибут выравнивания align.

Элемент, обеспечивающий принудительный переход на новую строку. Он имеет только стартовый тег. В месте его размещения строка заканчивается, а оставшийся текст печатается с новой строки.

Этот элемент, по своему действию, является прямой противоположностью предыдущему. Текст, заключенный между его тегами, будет выведен в одну строку. Длинная строка не уместится на экране, и для ее просмотра придется использовать горизонтальную полосу прокрутки.

Элемент для обозначения текста, отформатированного заранее (preformatted). Подразумевается, что текст будет выведен в том виде, в каком он был подготовлен пользователем. Например, учитываются символы конца строки, появившиеся при наборе текста в редакторе. Во всех других случаях браузер игнорирует эти символы. Этот элемент удобно использовать для демонстрации листингов программ или для вывода текстовых документов, переформатирование которых может привести к искажению их смысла.

Обозначение цитаты. Этот элемент требует наличие конечного тега. Текст не претерпевает никаких изменений, но абзац располагается с отступом. К кавычкам этот элемент тоже не имеет никакого отношения: если в цитате используются кавычки, то они должны быть проставлены явным образом. В настоящее время используют и другое написание этого элемента: BQ.

## 6. Фреймы

В этой части мы рассмотрим такую возможность языка HTML, как разбиение окна браузера на несколько частей, в которых отображаются разные HTML-документы.

Окно браузера всегда разбиваются на прямоугольные кадры (традиционно, они называются по-английски, фреймами). Внешне разбиение напоминает таблицу: кадры выстраиваются в несколько "строк", каждая из которых содержит несколько прямоугольных ячеек, содержащих отдельные HTML-документы.

Для того чтобы разбить документ на несколько прямоугольников, вместо контейнера BODY нам потребуется записать контейнер FRAMESET (набор фреймов). В теге нужно указать на сколько "строк" и "столбцов" разбивается всё окно браузера. Это делается с помощью атрибутов ROWS (строки) и COLS (столбцы). Заметим, что в этих атрибутах недостаточно указать количество строк и столбцов, требуются также их размеры, например в процентах от размеров окна. В примере Frames\F1.html это сделано так: (всё окно разбивается на две одинаковые "строки" и на три почти одинаковых столбца).

Внутри контейнера FRAMESET перечисляются HTML-документы, которые нужно поместить в кадры (слева направо, сверху вниз). Это делается с помощью последовательных тегов . Самый важный атрибут тега - это SRC, значением которого является URL документа, помещаемого в этот фрейм.

Размеры фреймов в теге можно указывать не только в процентах. Можно также указать их в частях, например (по высоте окно условно разбито на 3 части, первая "строка" кадров занимает две из них, вторая - одну, по ширине имеется 6 условных частей, "столбцы" занимают, соответственно, 2, 1 и 3 части).

Другой вариант - указание размеров в точках, например: COLS="300,100,400". Возможно и смешанное указание размеров (точки - проценты, точки - части, части - проценты). Можно, например, записать: COLS="200,30%,70%", что означает: первый "столбец" всегда имеет ширину 200 точек, независимо от размеров окна, второй занимает 30% от оставшейся части ширины окна, а третий - 70% от этого остатка.

Внешний вид фреймов настраивается с помощью атрибутов тегов и . Ниже перечисляются эти атрибуты и объясняется их назначение:

1. BORDERCOLOR (для и ). Управляет цветом границы одного фрейма или всего набора фреймов. Значением является либо название стандартного цвета, либо числовой код (#RRGGBB).

2. BORDER (для ). Задаёт ширину всех границ, отделяющих фреймы друг от друга. Значение нужно указывать в точках.

3. NORESIZE (для ). Этот атрибут не требует значения. Если он присутствует, то пользователь не может перемещать мышью границу соответствующего фрейма.

4. MARGINHEIGHT и MARGINWIDTH (для ). Позволяют указать (в точках) размеры отступов от горизонтальной (и вертикальной) границы фрейма до содержимого HTML-документа, лежащего внутри этого фрейма.

5. SCROLLING (для ). Когда документ HTML не помещается целиком в отведённый ему фрейм, для возможности просмотра появляются полосы прокрутки фрейма. В атрибуте SCROLLING можно указать, показывать ли полосы прокрутки всегда (значение YES), только когда документ не помещается во фрейм (AUTO), или не показывать никогда (NO).

При желании можно добиться того, чтобы какой-либо фрейм вместо обычного HTML-документа содержал бы, в свою очередь, набор фреймов. Для этого достаточно вместо тега поместить контейнер , в который вложены теги . HTML позволяет управлять работой гиперссылок так, чтобы документ, на который указывает ссылка открывался не в целом окне, а только в одном фрейме.

Для того чтобы ссылки работали именно так, нужно выполнить два действия:

1. Фрейму, в который в дальнейшем нужно помещать документ, найденный по ссылке, дают имя. Для этого используют атрибут NAME тега.

2. При вставке ссылки в HTML-документ другого фрейма в теге указывают атрибут TARGET (цель), значением его является имя фрейма, в который нужно поместить HTML-документ, указанный в HREF.

Кроме имён, которые можно давать собственным фреймам, существуют и стандартные имена, заранее известные браузеру, а именно:

"\_self" - позволяет открывать документ по ссылке в том же фрейме, где находилась ссылка

"\_top" - открывает документ во всё окно, где находилась ссылка, разрушая при этом всю структуру фреймов

"\_parent" - в родительском фрейме; это не совсем то же самое, что \_top, поскольку фреймы бывают вложенными; вся структура родительского фрейма разрушается

"\_blank" - в новом окне браузера; старое окно (содержащее ссылку) при этом не уничтожается.

## **7. Альтернативные средства составления Web–документов**

Язык HTML является основным средством организации данных в World Wide Web и пока не существует других языков, которые служили бы ему полной альтернативой. Поэтому говорить о том, в чем этот язык превосходит другие языки или в чем он им уступает, не приходится. Хотя нужно заметить, что свои ограничения в HTML присутствуют.

В последнее время в дизайне Web–документов все чаще используется популярный язык Java, который способен преодолеть ограничения HTML. В связке с HTML язык Java предлагает пользователю средства оформления документов, которые трудно или даже невозможно реализовать с помощью одного лишь HTML. Это, например, создание трехмерной анимации на экране, выполнение апплетов (приложений) на машине пользователя, пересылаемых с сервера. Язык Java предоставляет более гибкие и удобные способы общения с пользователями, например, более развитые формы ввода данных. С помощью этого языка можно внести некоторую динамику в любой Web–документ, например, заставить на экране изображения сменять друг друга с произвольной периодичностью и порядком. Таким образом, Java добавляет больше интерактивности в документах. Но сам Java не позволяет так быстро разрабатывать Web–странички, как это можно сделать с помощью HTML, и требует намного больше времени для изучения.

## **Заключение**

За всемирной компьютерной сетью Internet будущее. И любой человек сознает, что знание Internet необходимо и ему. Но так как информации в сети достаточно большое количество и она “разбросана” по огромному количеству серверов, необходимо уметь и знать как ее находить.

Можно работать на Web без знания языка HTML, так как тексты HTML могут создаваться различными специальными редакторами и конвертерами. Писать непосредственно на HTML нетрудно. Возможно, это даже легче, чем изучать HTML-редактор или конвертер, которые часто ограничены в своих возможностях, содержат ошибки или производят плохой HTML код, который не работает на различных платформах.

Язык HTML существует в нескольких вариантах и продолжает развиваться, но конструкции HTML вероятнее всего будут использоваться и в дальнейшем. Изучая HTML и познавая его глубже, создавая документ в начале изучения HTML и расширяя его насколько это возможно, Вы сможете создавать документы, которые могут быть просмотрены многими браузерами Web, как сейчас, так и в будущем. Это не исключает возможности использования других методов, например, метод расширенных возможностей, предоставляемый Netscape Navigator, Internet Explorer или некоторыми другими программами. Работа с HTML - это способ усвоить особенности создания документов в стандартизированном языке, используя расширения, только когда это действительно необходимо.

HTML был ратифицирован World Wide Web Consortium. Он поддерживается несколькими широко распространенными браузерами, и, возможно, станет основой почти всего имеющего отношение к Web программного обеспечения.

## Библиографический список

1. Фримен, Э. Изучаем HTML, XHTML и CSS / Э. Фримен. – СПб.: Изд-во Питер, 2012. – 656 с.
2. Матросов, А., Сергеев А., Чаунин М. HTML 4.0 / А. Матросов, А. Сергеев, М. Чаунин. – СПб.: БХВ-Петербург, 2007. – 672 с.
3. Комолова, Н., Яковлева Е. HTML: Самоучитель / Н. Комолова, Е. Яковлева. – СПб.: Питер, 2011. – 288 с.
4. Кит, Д. HTML5 для веб-дизайнеров / Д. Кит. – М.: Манн, Иванов и Фербер, 2015. – 112 с.
5. Роббинс, Д. HTML5. Карманный справочник / Д. Роббинс. – М.: Вильямс, 2015. – 192 с.
6. Гулевич П. Секреты создания HTML сайта с нуля – М.: Webmoment, 2010. – [Электронный ресурс]. URL: <http://www.razym.ru/56732-pavel-gulevich-sekrety-sozdaniya-html-sajta-s.html> (дата обращения: 24.01.2017).
7. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 – СПб.: Питер, 2016. – [Электронный ресурс]. URL: <http://forcoder.ru/html-css/sozdaem-dinamicheskie-veb-sayty-s-pomoschyu-php-mysql-javascript-css-i-html5-1525> (дата обращения: 24.01.2017).
8. Хейз Д. Освой самостоятельно HTML – М.: Вильямс, 2007. – 272 с.  
[Электронный ресурс]. URL: <https://eknigi.org/raznoe/20113-osvoj-samostoyatelno-html.html> (дата обращения: 24.01.2017).
9. Русак А.М. Основы работы с HTML – М.: ИНТУИТ, 2006. – 130 с.  
[Электронный ресурс]. URL: <https://eknigi.org/programmirovanie/115807-osnovy-raboty-s-html.html> (дата обращения: 24.01.2017).



