Содержание:

Введение

Совершенствование технических возможностей средств вычислительной техники, развитие коммуникационных средств и технологий управления информационными ресурсами в последние годы привели к появлению более крупных информационных систем. Речь идет о масштабах систем не только относительно объема поддерживаемых информационных ресурсов, но и числа их пользователей. Объем информационных ресурсов Web в настоящие время исчисляется многими миллионами страниц.

В связи с этим развитием информационных технологий, сетей, а также информационных систем получил широкое распространение язык гипертекстовой разметки HTML.

Цель курсовой работы – изучить теоретический материал по тематике курсовой работы и разработать страницу на HTML которая в достаточной мере отобразит суть и возможности языка.

Для достижения поставленной цели были выделены следующие задачи:

- проанализировать литературу по теме курсовой работы;
- рассмотреть и изучить понятие: «Язык гипертекстовой разметки»;
- охарактеризовать языки: HTML, XML, XHTML, SGML;
- обозначить новые тенденции в развитии HTML;
- рассмотреть и проанализировать спецификации HTML;
- разработать страницу на HTML которая в достаточной мере отобразит суть и возможности языка.

Структура курсовой работы: работа состоит из введения, трех глав, заключения, списка литературы, включающего в себя 4 источника и 1 приложение.

Первая глава посвящена изучению истории развития, основных понятий.

Вторая глава изучению спецификаций и разметки HTML.

Третья глава посвящена разработке страницы резюме разработчика на языке HTML.

Глава 1

В первой главе будут рассмотрены основные этапы развития языков гипертекстовой разметки, а также основные понятия.

Язык гипертекстовой разметки

Язык гипертекстовой разметки – это набор символов или последовательностей, которые добавляются в текст для определения внешнего вида и строения текста. В текстовом документе, при написании которого используется язык разметки, содержится не только сам текст как последовательность слов и знаков препинания, но и дополнительная информация о строении текста, например, списки, заголовки, выделенный текст, интерактивные элементы, ссылки и т.д. [1]

Языки гипертекстовой разметки используются, если необходим вывод форматированного текста, например в типографии (SGML, TeX, PostScript, PDF), пользовательских интерфейсах компьютеров (Microsoft Word, OpenOffice, troff), Всемирной Сети (HTML, XHTML, XML, WML, VML, PGML, SVG, XBRL).

Все языки разметки перемешивают текст документа с инструкциями разметки в файле или в потоке данных, однако текст может быть изолирован от разметки с помощью идентификаторов и других методов координации. Подобная «отдельная разметка» используется для внутреннего представления программ, которые работают с размеченными документами. Однако, внедренная или «междустрочная» более широко распространена. Приведем пример текста, размеченного с помощью HTML:

<h1> Anatidae </h1>

The family Anatidae includes ducks, geese, and swans, but not the closely-related screamers.

Программный код разметочных инструкций (теги) заключен в угловые скобки <тег>. Между тегами находится текст документа. Коды h1, p и em описывают

позицию текста, его назначение или значение (h1 означает «это заголовок первого уровня», p значит «это параграф», a em означает «это подчеркнутое слово или фраза»).

Интерпретирующая программа может применять эти правила или стили для показа различных частей текста с помощью различных гарнитур, размеров шрифта, отступов, цветов или других стилей. Например, тег H1 может быть представлен жирным шрифтом, или подчеркнутым, или более крупным, или даже может вообще не меняться.

Языки разметки

SGML

Первым языком с четким различием между структурой и видом документа был Scribe, созданный Брайаном Рейдом в 1980 году. Scribe был инновационным языком – в нем были введены стили, отделенные от текста документа и управляющие использованием описательных элементов. Scribe повлиял на разработку GML (позже SGML); кроме того от него произошли языки HTML и LaTeX.

В начале 1980х годов был создан SGML, воплотивший идею о том, что разметка должна быть сфокусирована на структуре документа, а внешнее представление документа должно быть прерогативой интерпретатора.

В SGML был точно определен синтаксис для включения разметки в текст. Кроме того отдельно были описаны, какие теги были разрешены и где. Программист получал возможность создавать и использовать любую желаемую разметку, выбирать теги для использования и самостоятельно давать им имена. SGML, таким образом, может считаться метаязыком. От него произошли многие другие языки разметки, например, TEI и DocBook.

SGML был широко распространен, несмотря на то, что он обладал рядом недостатков (он был слишком громоздким, трудным для изучения и слишком гибким).

SGML (англ. Standard Generalized Markup Language) — метаязык, который дает возможность определять язык разметки документов. SGML был разработан для совместного использования машинно-читаемых документов в больших

правительственных и аэрокосмических проектах.

HTML является приложением SGML

HTML

HTML (Hyper Text Markup Language) - язык разметки гипертекста. Он был разработан Тимом Бернерсом-Ли в рамках создания проекта распределенной гипертекстовой системы (World Wide Web). [2] HTML предназначен для написания гипертекстовых документов, публикуемых во всемирной сети. Документ на HTML может включать в себя стилизованный и форматированный текст, команды включения графических и звуковых файлов, гиперсвязи с различными ресурсами Internet, скрипты на языке JavaScript и VBScript, различные объекты, например Flash-анимацию. [3]

HTML имел много общего с другими языками разметки, основанными на синтаксисе SGML, однако освоить его было гораздо проще даже для тех специалистов, которые прежде никогда с ним не работали. Многие программисты считали HTML одним из основных факторов развития Web, поскольку этот язык был расширяемым и гибким. [4] В настоящее время HTML является одним из самых широко распространенных языков разметки.

HTML изначально был задуман как средство форматирования документов, которое не привязано к средствам воспроизведения; то есть HTML документ должен быть воспроизведен на любом оборудовании в задуманном виде без искажений. [5] Однако на практике такое не всегда возможно.

Документы HTML являются текстовыми файлами, которые кроме самого текста страницы содержат специальные теги (или управляющие элементы) разметки. [6] Теги разметки указывают браузеру, каким образом следует отобразить страницу.

HTML не является языком программирования. Он предназначен для форматирования и структурирования текстового документа определенным образом. [7] Каждому элементу присваиваются определенные параметры, которые интерпретирует браузер, и таким образом текстовый документ принимает желаемый формат. [8] [9] Параметры могут быть заданы как для отдельного элемента, так и для группы или определенного типа элементов (например, таблицы, ячейки, ссылки, текст и т.д.). [10] [11] Также есть возможность

присваивать определенные свойства выбранным элементам персонально. [12]

Характеристики элементов указываются с помощью тегов. [13]

XML

XML (расширяемый язык разметки) – это мета-язык разметки, получивший широкое применение в настоящее время. XML более прост, чем SGML; он используется в основном при работе с документами в интернете. Поскольку XML является мета-языком, он позволяет пользователям создавать любые необходимые теги. [14] XML документы имели много общего с SGML документами, что помогало пользователям SGML легко освоить XML.

XML предназначен в основном для не полностью структурированной среды, например для публикаций и документов. Однако пользователи оценили его, поскольку XML обладает гибкостью при достаточной простоте. XML широко используется для передачи данных между программами. XML (англ. eXtensible Markup Language) — язык разметки, предназначенный для хранения структурированных данных и обмена информацией между программами и включающий общие синтаксические правила. [15]

XHTML

С начала 2000 годов XML получил более широкое распространение, чем SGML, поэтому был предложен термин XHTML (Extensible HyperText Markup Languge — Расширяемый Гипертекстовый Язык Разметки). [16] XHTML-документы оформляются как XML-документы, что позволяет создавать более четкие и точные документы, используя теги HTML.

XHTML (англ. Extensible Hypertext Markup Language) — язык разметки, созданный на базе XML. [17] Он соответствует спецификации SGML и по возможностям сравним с HTML. При работе с XHTML применимы многие технологии, разработанные для XML (например, XSLT и XPath). Анализ XHTML проще и быстрее, чем HTML. Обработка XHTML возможна даже на устройствах с низкой продуктивностью, поскольку синтаксис XML строже, чем SGML.

Одно из отличий XHTML от HTML заключается в том, что все теги должны быть закрытыми. [18] Другие атрибуты в тегах должны быть в кавычках. Кроме того, все

теги и имена атрибутов должны быть написаны в нижнем регистре, чтобы восприниматься правильно; HTML, напротив, невосприимчив к регистру.

Подобно HTML, он может быть охарактеризован как «контейнерный язык».

HTML 5.

С развитием технических возможностей более старые языки разметки, такие как XHTML, устарели, поскольку они не имеют штатных ресурсов для многих задач (например, проигрывание аудио и видеороликов, поддержка геолокации, возможность рисовать непосредственно в браузере, и т.д.). В XHTML подобные задачи возможно решить только с помощью сторонних плагинов. [19] [20]

В связи с необходимостью решения вышеуказанных задач, был создан язык HTML5.

HTML5 не является продолжением XHTML или HTML4, несмотря на похожее название, он является отдельным языком разметки. Разработка HTML5 не полностью завершена, однако с ним уже возможно работать в современных браузерах. [21]

HTML5 имеет следующие возможности:

- Поддержка геолокации — определение местоположения пользователя на карте и использование этой информации для определения маршрута его движения, вывода близлежащих магазинов, кинотеатров, кафе и других данных.
- Воспроизведение видеороликов.
- Воспроизведение аудиофайлов.
- Локальное хранилище данных — позволяет сайтам сохранять информацию на компьютере пользователя и обращаться к ней позднее.
- Фоновые вычисления — стандартный способ запуска JavaScript в браузере в фоновом режиме.
- Оффлайн приложения — страницы, способные работать при отключении Интернета.
- Рисование — внутри тега <canvas> с помощью JavaScript можно рисовать фигуры, линии, создавать градиенты или трансформировать объекты.
- Новые элементы форм: для даты, времени, поиска, чисел, выбора цвета и др.

Кроме того, в HTML5 добавлены дополнительные теги для разметки документов, исключены устаревшие теги, некоторые другие теги подверглись модификации. [22]

Тег в языках разметки.

Тег — это элемент языка разметки гипертекста. Текст между начальным и конечным тегом размещается и отображается в зависимости от свойств тега. [23]

Теги имеют следующий синтаксис. Чаще всего используются парные теги – открывающий и закрывающий (или начальный и конечный). Иногда возможно использование одиночного тега. Например, тег <p>, обозначающий отступ абзаца, может использоваться как в виде парного, так и в виде одиночного тега. Тег с пустым текстом: <HR></HR> (обозначает разрыв текста без сохранения отступов) может также иметь одиночную форму (<HR/>). [24]

Основные теги и их интерпретация определены организацией W3C.

SGML позволяет назначить другие символы для обрамления тега (например, фигурные скобки), однако такие подмножества SGML как HTML или XML подобной возможности не имеют.

Теги и элементы

Текст, заключенный между открывающим и закрывающим тегами, включая сами теги, называется элементом. Содержание элемента – это тест между тегами, не включая теги. Содержание элемента может включать в себя текст, а также другие элементы.

Атрибуты

Атрибуты – это свойства тега, которые дают дополнительные возможности для форматирования и структурирования текста. Атрибуты записываются следующим образом: имя атрибута – значения. При этом текстовые значения заключены в кавычки. Например, фрагмент текста может быть выделен определённым шрифтом, с помощью тега , при этом в теге следует указать название

шрифта и размер: оформляемый текст .

Таким образом были рассмотрена история создания языков гипертекстовой разметки, основные виды языков (SGML, HTML, XML, XHTML, HTML5) и основные понятия – тег, элемент, атрибут.

Глава 2.

Во второй главе рассмотрена структуреа HTML документа: синтаксис и назначение основных элементов.

Теги верхнего уровня

Каждая веб-страница имеет общую структуру, которая содержит типичные элементы, общие для сайтов различных типов и направленностей. [25] Рассмотрим типичный код веб-страницы:

<html>

<head>

<title>Заголовок страницы</title>

</head>

<body>

<!-- Комментарий -->

</body>

</html>

Разберем отдельные строки кода более подробно.

<html>…</html>

Тег <html> указывает на начало HTML-файла, внутри которого хранится заголовок (<head>) и тело документа (<body>). Закрывающий тег </html> всегда завершает

HTML код.

<head>…</head>

Заголовок документа <head> может содержать текст и теги, однако содержимое этого раздела не показано напрямую на странице, за исключением контейнера <title>. Закрывающий тег </head> также должен присутствовать в обязательном порядке, для обозначения окончания блока заголовка.

<title>…</title>

Тег <title> обозначает заголовок веб-страницы (в Windows текст заголовка показывается в левом верхнем углу браузера). [26] Тег <title> дожжен присутствовать в коде в обязательном порядке.

<body>…</body>

В теле документа <body> размещаются теги и содержательная часть веб-страницы. [27] Закрывающий тег </body> указывает на завершение тела документа.

<!-- Комментарий -->

Комментарии не отображаются в браузере, они служат для записи скрытых заметок, не влияющих на вид веб-страницы. Комментарий начинается тегом <!-- и заканчиваются тегом -->, все, что находится между этими тегами, не отображается на странице. Пользователь может прочитать комментарии, открыв исходный код документа.

<!DOCTYPE>

Элемент <!DOCTYPE> указывает тип текущего документа — DTD (document type definition, описание типа документа). [28] Определение типа документа необходимо для того, чтобы браузер понимал, как именно следует интерпретировать данный код, поскольку в зависимости от используемого языка разметки (различные версии HTML, XHTML, и т.д.) документ интерпретируется по-разному, поскольку различные языки имеют различный синтаксис. Таким образом, с помощью <!DOCTYPE> определяется стандарт, согласно которому отображается веб-страница. Ниже <!DOCTYPE> будет рассмотрен более подробно.

Существует несколько типов <!DOCTYPE> , которые отличаются в зависимости от версии языка разметки, на которую они ориентированы. В таблице 1. приведены основные типы документов с их описанием.

DOCTYPE	**Описание**
HTML 4.01	
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">	Строгий синтаксис HTML.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">	Переходный синтаксис HTML.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">	В HTML-документе применяются фреймы.
HTML 5	
<!DOCTYPE html>	В этой версии HTML только один доктайп.
XHTML 1.0	
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">	Строгий синтаксис XHTML.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">	Переходный синтаксис XHTML.
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">	Документ написан на XHTML и содержит фреймы.
XHTML 1.1	
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">	Разработчики XHTML 1.1 предполагают, что он постепенно вытеснит HTML. Как видите, никакого деления на виды это определение не имеет, поскольку синтаксис один и подчиняется четким правилам.

Таблица 1

Разница между строгим и переходным описанием документа заключается в различных подходах к написанию программного кода. Строгий HTML подразумевает жесткое соблюдение спецификаций HTML, ошибки в данном случае недопустимы. Напротив, в переходном HTML могут встречаться огрехи кода, не мешающие работе программы, поэтому переходный HTML в некоторых случаях более предпочтителен.

Например, в строгом HTML и XHTML наличие тега <title>, обязательно, а в переходном HTML его использование необязательно. Браузер будет показывать документ вне зависимости от того, соответствует ли он синтаксису или не соответствует. Проверка осуществляется валидатором и используется разработчиками для отслеживания ошибок в документе.

Иногда в коде HTML не указан <!DOCTYPE>, в этом случае документ все равно будет отображаться в браузере, однако есть вероятность, что один и тот же документ будет показан различным образом при использовании <!DOCTYPE> или при его отсутствии. Кроме того, подобные документы могут отображаться с ошибками, не так, как задумывал разработчик. Чтобы избежать подобных ситуаций, следует добавлять <!DOCTYPE> в начало документа в обязательном порядке.

Метатеги <meta>

Тег <meta> - универсальный тег. Он добавляет ряд дополнительных возможностей, например, при помощи метатегов можно изменить кодировку страницы, добавить описание документа, ключевые слова. Более подробно тег рассмотрен ниже.

Метатеги выполняют несколько важных функций:

- Влияют на то, как отображается и какую позицию занимает страница в индексе (выдаче). [29]
- Помогают продвигать ресурс по наиболее соответствующим Вашей целевой аудитории запросам. Позволяют потенциальному посетителю увидеть в поисковой выдаче нужную информацию, делая описание максимально информативным.
- Фиксируют данные о странице, помогают правильно отобразить информацию, настроить индексацию (анализ сайта и внесение его в базу для последующего включения в выдачу).

В настоящее время мета теги не оказывают сильного влияния на продвижение, однако они по-прежнему учитываются при оценке поисковыми системами, и предпочтение отдается ресурсу с грамотно сформированными тегами. [30]

Мета теги для SEO (поисковой) оптимизации

К данной категории тегов относятся три тега: title, description, keywords. Данные атрибуты сообщают поисковому роботу информацию о сайте, которая помогает грамотно определить тематику ресурса и ранжировать (сортировать) в выдаче. Потому составляя метатеги, разработчики обязательно используют ключевые слова

(слова и фразы, которые целевой пользователь вводит в строку поиска, чтобы найти нужный ресурс).

Title

Заголовок страницы. В индексе это крупная синяя ссылка на сайт, в браузерных вкладках — название.

Атрибут должен содержать основные ключевые слова и конкурентные преимущества. Важно, чтобы пользователь сразу понял, что сайт ему предлагает, и почему данный ресурс ему подойдет. Основная сложность — вместить нужную информацию в четко заданный поисковыми системами лимит символов. У Google максимум — 70 (12 слов), у Яндекс — 80 (15 слов). Пример приведен в таблице 2.

Дано	**Неудачный title**	**Удачный title**
Ключевые слова «купить кресло», «купить кресло недорого», «купить удобное кресло»	Купить кресло. Купить удобное кресло, купить кресло недорого	Как купить удобное кресло недорого с доставкой и получить скидку 15%?

Таблица 2

В первом примере переизбыток ключевых фраз делает предложение трудным для восприятия пользователем. Второй вариант метатега объединяет все нужные ключевые фразы в логичной последовательности и указывает на конкурентное преимущество компании (скидку).

Description

Description – это краткий «анонс» страницы в результатах поиска, несколько коротких предложений, в которых используются наиболее значимые ключевые слова конкретной страницы. Рекомендуемая длина meta тега — 160-200 символов.

Мета теги Title и Description составляются отдельно для каждой страницы сайта, поскольку все разделы должны иметь собственный список ключевых фраз, соответствующий тематике и содержанию.

С помощью meta name description поисковым роботом описание страницы (сниппет) отображается в результатах поиска под title.

Если атрибут не заполнен либо сформирован не по правилам (значительно превышен лимит символов, избыток ключевых слов), то поисковый робот составляет описание автоматически, составляя его из отрывков фраз на странице. Результат работы подобной программы делает продажи сайте малоэффективными. Пример в таблице 3.

Дано	**Неудачный Description**	**Удачный Description**
Магазин продажи книг; ключевые слова «интересная книга», «купить книгу»	Книги издавна являются одними из самых близких «друзей» человека. Интересная книга помогает скоротать время, дает полезные знания, раскрывает занимательные факты, пробуждает эмоции. Потому стоит задуматься о том, чтобы купить книгу в нашем магазине.	Нужна интересная книга? В нашем магазине — широкий выбор тематик: от узкоспециальной литературы до мировых бестселлеров. У нас Вы можете купить книгу с доставкой в течение 1-5 дней в любой регион РФ. Первым покупателям — скидка 10%.

Таблица 3

Первый вариант — не специально составленный meta тег, а взятый со страницы кусок статьи. Возможно, пользователю интересно почитать литературный текст, но, обращаясь к поисковой системе, он предпочтет получить ответ на конкретный запрос о покупке. Потому метатеги для сайта второго типа более привлекательны: они сразу дают понимание, какой ассортимент и преимущества предлагает магазин.

Keywords

Сообщает поисковому роботу ключевые слова, под которые «заточен» контент. Поисковый робот сверяет фразы в содержимом страницы с указанными в данном теге, а полученные результаты учитывает в дальнейшем при определении позиций ресурса.

Сейчас поисковики не придают данному атрибуту такой вес, как раньше. Но грамотное использование его не будет лишним.

Используйте 3-6 ключевых слов/фраз через запятую. Чрезмерное употребление может расцениваться поисковиком как переспам. В список должны включаться только используемые на странице запросы, под которые оптимизирован текст, названия товаров и т.д.!

Мета теги технического характера

Атрибуты внутренней оптимизации ресурса. Такие мета теги настраивают информацию, которая отображается пользователю в браузере, и включают отдельные опции (к примеру, дают возможность закрыть для индексации отдельные разделы).

Наиболее распространенные технические метатеги приведены в таблице 4.

Пример	**Описание**
<meta http-equiv="Content-type" content="text/html;charset=windows-1251">	Content-type. Включает данные о типе документа и кодировке. Влияет на то, насколько корректно страница будет отражаться в браузере. Особенно актуален при неверной настройке браузера или сервера. В такой ситуации документ будет показан на другой кодировке, и пользователь не сумеет ничего изменить даже попыткой переключить вручную. Для кириллицы используется атрибут windows-1251 или KOI-8R

<meta http-equiv="Content-Language" content="ru">	Содержит информацию о языке контента (содержимого). Как и с предыдущим тегом, сейчас браузеры определяют язык автоматически. Но в случае неверных настроек данный тег будет полезен.
<meta name="robots" content="noindex">	Показывает поисковому роботу, какие страницы следует включать в индекс, а какие — нет. Данный тег предполагает ряд дополнительных атрибутов: index/noindex (можно/нельзя индексировать); follow/nofollow (учитывать/не учитывать гиперссылки); all/none (открыть/закрыть для индексации страницу и гиперссылки).
<meta http-equiv="refresh" content="N; url=URL">	Позволяет задать время обновления страницы при загрузке либо переадресации пользователя в другой раздел. Тег полезен в случае обновления сайта: вместо страницы ошибки 404 или длительной загрузки перенаправляет пользователя в новый раздел.
	Author и Copyright. Этими тегами можно прописать авторство графического и текстового содержимого страницы.
<meta name="Author" content="Иван Иванович Иванов">	автор страницы
<meta name="Copyright" content="Петр Петрович Петров">	принадлежность авторских прав

<meta name="Address" content="г. Москва, Ярославское шоссе 25">	адрес автора/собственника

Таблица 4

Все вышеперечисленные метатеги для сайта призваны, прежде всего, улучшить восприятие ресурса поисковой системой. Потому следует обратить внимание на грамотное составление данных атрибутов, (следует отражать только правдивую информацию и соблюдать разумный подход при использовании ключевых слов).

Следует помнить, что сейчас поисковые системы оценивают ресурсы, в первую очередь, по качеству контента, кода, структуры. Потому при составлении метатегов важно ставить себя на позицию пользователя: указывать данные, способные повлиять на восприятие и отображение страницы, преподносить полезную информацию последовательно, логично, с ключевыми выгодами для клиента.

Стили (CSS)

Связанные стили

При использовании связанных стилей описание селекторов и их значений располагается в отдельном файле, как правило, с расширением css. Для связывания документа с этим файлом применяется тег <link>. [31] Данный тег помещается в контейнер <head>, как показано в примере ниже.

```
<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

<title>Стили</title>

<link rel="stylesheet" href="style.css">
```

```
</head>

<body>

<h1>Заголовок</h1>

<p>Текст</p>

</body>

</html>
```

Значение атрибута тега <link> — rel остаётся неизменным независимо от кода, как приведено в данном примере. Значение href задаёт путь к CSS-файлу, он может быть задан как относительно, так и абсолютно. [32] Важно, что таким образом можно подключать таблицу стилей, которая находится на другом сайте.

Содержимое файла mysite.css подключаемого посредством тега <link> приведено в примере.

```
H1 {

color: #000080;

font-size: 200%;

font-family: Arial, Verdana, sans-serif;

text-align: center;

}

P {

padding-left: 20px;

}
```

Как видно из данного примера, файл со стилем не хранит никаких данных, кроме синтаксиса CSS. В свою очередь и HTML-документ содержит только ссылку на файл со стилем, т. е. таким способом в полной мере реализуется принцип разделения кода и оформления сайта. Поэтому использование связанных стилей является наиболее универсальным и удобным методом добавления стиля на сайт. Ведь

стили хранятся в одном файле, а в HTML-документах указывается только ссылка на него.

Глобальные стили

При использовании глобальных стилей свойства CSS описываются в самом документе и располагаются в заголовке веб-страницы. [33] По своей гибкости и возможностям этот способ добавления стиля уступает предыдущему, но также позволяет хранить стили в одном месте, в данном случае непосредственно на той же странице с помощью контейнера <style>, как показано в примере.

```
<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

<title>Глобальные стили</title>

<style>

H1 {

font-size: 120%;

font-family: Verdana, Arial, Helvetica, sans-serif;

color: #333366;

}

</style>

</head>

<body>

<h1>Hello, world!</h1>

</body>
```

</html>

В данном примере задан стиль тега <h1>, который затем можно использовать во всех необходимых местах на данной веб-странице.

Внутренние стили

Внутренний или встроенный стиль является по существу расширением для одиночного тега, используемого на текущей веб-странице. [34] Для определения стиля используется атрибут style, а его значением выступает набор стилевых правил.

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

<title>Внутренние стили</title>

</head>

<body>

Пример текста

</body>

</html>

В данном примере стиль тега задаётся с помощью атрибута style, в котором через точку с запятой перечисляются стилевые свойства

Скрипты <script>…</script>

В данном разделе мы рассмотрим все основные элементы javascript.

Javascript подключается напрямую в HTML-файл. [35] Самый простой способ - это написать javascript-команды внутри тега <script> в теле страницы.

Подключение в любом месте

Когда браузер читает HTML-страничку, и видит <script> - он первым делом читает и выполняет код, а только потом продолжает читать страницу дальше.

Так, в следующем примере будет показано начало страницы, затем три раза выполнится функция alert, которая выводит окошко с информацией, а только потом появится остальная часть страницы.

```
<body>

<h1>Считаем кроликов</h1>

<script type="text/javascript">

for(var i=1; i<=3; i++) {

alert("Из шляпы достали "+i+" кролика!")

}

</script>

<h1>...Посчитали</h1>

</body>
```

В этом примере использовались следующие элементы.

<script type="text/javascript"> ... </script>

Тег <script> сообщает браузеру о том, что внутри находится исполняемый скрипт. Атрибут type говорит о том, что это javascript. Атрибут может и отсутствовать, но с точки зрения стандарта его следует указать. [36]

Конструкция for.

Это стандартный цикл, по синтаксису аналогичный другим языкам программирования.

Объявление var i.

Объявление переменной внутри цикла: i - локальная переменная.

Функция alert.

Выводит сообщение на экран и ждет, пока посетитель не нажмет OK.

Вынос скриптов в заголовок HEAD.

Обычно javascript стараются отделить от собственно документа.

Для этого его помещают внутрь тега HEAD, а в теле страницы по возможности оставляется чистая верстка.

В следующем примере javascript-код только описывает функцию count_rabbits, а ее вызов осуществляется по нажатию на кнопку input.

```
<html>

<head>

<script type="text/javascript">

function count_rabbits() {

for(var i=1; i<=3; i++) {

alert("Из шляпы достали "+i+" кролика!")

}

}

</script>

</head>

<body>

<input type="button" onclick="count_rabbits()" value="Считать кролей!"/>

</body>

</html>
```

Для указания запускаемой по клику функции в input был использован атрибут onclick. Это лишь вершина мощного айсберга javascript-событий.

Внешние скрипты.

Обычно javascript-код не пишут в HTML, а подключают отдельный файл со скриптом. Делается это следующим образом:

<script src="/script.js"></script>

При этом файл /my/script.js содержит javascript-код, который иначе мог бы находиться внутри тега <script>.

Это очень удобно, потому что один и тот же файл со скриптами можно подключать на разных страницах. При правильных настройках сервера браузер закеширует его и не будет скачивать каждый раз заново.

Чтобы подключить несколько скриптов, следует использовать несколько подобных тегов:

<script src="/js/script1.js"></script>

<script src="/js/script2.js"></script>

При указании атрибута src содержимое тега игнорируется.

То есть одновременно подключить внешний файл и написать что-то внутри тега нельзя. Придется делать два разных тега <script>: первый с src, второй - с командами, которые будут выполнены после выполнения внешнего файла.

Блочные элементы

Блочные элементы характеризуются тем, что занимают всю доступную ширину, высота элемента определяется его содержимым, и он всегда начинается с новой строки. [37]

*** … ***

Тег относится к универсальным блочным контейнерам и применяется в тех случаях, где нужны блочные элементы без дополнительных свойств. [38] Также с помощью тега можно выравнивать текст внутри этого контейнера с

помощью атрибута align.

<h1>Заголовок</h1> (h1…h6)

В HTML используются шесть текстовых заголовков разного уровня, с помощью которых можно разделять текст на секции разного уровня. Таким образом, тег <h1> используется для наиболее важных заголовков (заголовки первого уровня), а тег тег <h6> показывает заголовки шестого уровня, являющиеся наименее важными. [39] Заголовок первого уровня по умолчанию отображается самым крупным жирным шрифтом, заголовки следующих уровней имеют менее крупный шрифт.

Поскольку теги <h1>…<h6> являются блочными элементами, они должны начинаться с новой строки, а следующие за ними элементы отображаются на следующей строке. Кроме того, перед заголовком и после него добавляется пустое пространство. [40]

*** Абзац ***

Тег указывает на начало абзаца (параграфа) текста. Если нет закрывающего тега, то конец данного абзаца совпадает с началом следующего абзаца.

Поскольку, тег является блочным элементом, текст всегда начинается с новой строки, а абзацы, следующие один за другим, разделены пустым пространством (или отбивкой).

Строчные элементы

Строчными называются такие элементы веб-страницы, которые являются непосредственной частью другого элемента, например, текстового абзаца. В основном они используются для изменения вида текста или его логического выделения. [41]

Разница между блочными и строчными элементами следующая.

- Строчные элементы могут содержать только данные или другие строчные элементы, а в блочные допустимо вкладывать другие блочные элементы, строчные элементы, а также данные. Иными словами, строчные элементы никак не могут хранить блочные элементы. [42]

- Блочные элементы всегда начинаются с новой строки, а строчные таким способом не акцентируются.
- Блочные элементы занимают всю доступную ширину, например, окна браузера, а ширина строчных элементов равна их содержимому плюс значения отступов, полей и границ.

<a>…

Тег <a> является одним из важных элементов HTML и предназначен для создания ссылок. [43] В зависимости от присутствия атрибутов name или href тег <a> устанавливает ссылку или якорь. Ссылки являются основой гипертекстовых документов и позволяют переходить с одной веб-страницы на другую. Особенность их состоит в том, что сама ссылка может вести не только на HTML-файлы, но и на файл любого типа, причем этот файл может размещаться совсем на другом сайте. [44] Главное, чтобы к документу, на который делается ссылка, был доступ. Иными словами, если путь к файлу можно указать в адресной строке браузера, и файл при этом будет открыт, то на него можно сделать ссылку.

…

Универсальный тег, предназначенный для определения строчного элемента внутри документа. [45]

…

Определяет жирное начертание шрифта.

…

Устанавливает курсивное начертание шрифта.

***
***

Тег
 устанавливает перевод строки в том месте, где этот тег находится. В отличие от тега параграфа <p>, использование тега
 не добавляет пустой отступ перед строкой.

<img />

Тег <img> предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Если необходимо, то рисунок можно сделать ссылкой на другой файл, поместив тег <img> в контейнер <a>. При этом

вокруг изображения отображается рамка, которую можно убрать, добавив атрибут border="0" в тег <img>.

Списки

Списком называется взаимосвязанный набор отдельных фраз или предложений, которые начинаются с маркера или цифры. [46] Списки предоставляют возможность упорядочить и систематизировать разные данные и представить их в наглядном и удобном для пользователя виде.

***
…
***

Тег <ol> устанавливает нумерованный список, т.е. каждый элемент списка начинается с числа или буквы и увеличивается по нарастающей. [47]

***
…
***

Устанавливает маркированный список, каждый элемент которого начинается с небольшого символа — маркера.

***
…
***

Тег <li> определяет отдельный элемент списка. [48]Внешний тег <ul> или <ol> устанавливает тип списка — маркированный или нумерованный.

<dd>, <dt>, <dl>

Тройка элементов предназначена для создания списка определений. Каждый такой список начинается с контейнера <dl>, куда входит тег <dt> создающий термин и тег <dd> задающий определение этого термина. [49] Закрывающий тег </dd> не обязателен, поскольку следующий тег сообщает о завершении предыдущего элемента. Тем не менее, предпочтительнее закрывать все теги.

Таблицы

Таблица состоит из строк и столбцов ячеек, которые могут содержать текст и рисунки. [50] Обычно таблицы используются для упорядочения и представления табличных данных.

<table>…</table>

Служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов <tr> и <td>.

<td>…</td>

Предназначен для создания одной ячейки таблицы. Тег <td> должен размещаться внутри контейнера <tr>, который в свою очередь располагается внутри тега <table>.

<th>…</th>

Тег <th> предназначен для создания одной ячейки таблицы, которая обозначается как заголовочная. [51] Текст в такой ячейке отображается браузером обычно жирным шрифтом и выравнивается по центру.

<tr>…</tr>

Тег <tr> служит контейнером для создания строки таблицы.

Таким образом, во второй главе были рассмотрены основные элементы HTML кода: теги верхнего уровня (<html>, <head>, <body>), комментарии, <DOCTYPE>, метатеги (<meta>, <title>, description, keywords, content-type, content-language, robots, refresh, Author, Copyright, Address), CSS (связанные стили, глобальные стили, встроенные стили), скрипты (<script>), блочные элементы (, <h1> - <h6>, <p>), строчные элементы (<a>, , <b>, <i>,
, <img>), списки (нумерованный, маркированный), таблицы.

Глава 3

В данной главе рассмотрена страница резюме программиста, написанная на HTML5. Все элементы кода снабжены комментариями. Полная версия кода страницы представлена в приложении 1. Все элементы, использованные при написании страницы, были рассмотрены в предыдущих главах.

HTML документ начинается с объявления типа документа посредством тега DOCTYPE. Для написания данной работы выбран HTML5, поэтому DOCTYPE записан следующим образом:

<!DOCTYPE html>

Разметка документа начинается с объявления тега <html>.

<html>

Далее идет блок head, в котором содержится служебная информация.

<head>

Далее идет набор мета тегов, в которых мы указываем кодировку, описание страницы и ключевые слова для поисковиков и автора документа. Также в блоке head расположен контейнер <title> с указанием заголовка окна.

<meta charset="utf-8" />

<meta name="description" content="Страница резюме приложение к курсовой работе">

<meta name="keywords" content="Дмитрий Акмулин резюме визитка курсовая работа">

<meta name="author" content="Дмитрий Акмулин">

<title>Резюме</title>

</head>

Как большинство элементов HTML, мы закрываем тег <head> и приступаем к описанию тела страницы посредством тега <body>. В теле страницы содержится информация, которая будет отображаться в браузере.

<body>

Описание начинается с вывода ФИО посредством заголовка первого уровня (тег <h1>). Далее идут два абзаца, содержащие дату рождения и семейное положение (тег). Наименование информации дополнительно выделено жирным шрифтом посредством тега .

<h1>Акмулин Дмитрий Валерьевич</h1>

Дата рождения: 10 марта 1988 года

Семейное положение: женат, трое детей.

Далее выводится информация о контактах, отделенная от основного текста посредством заголовка второго уровня <h2>. Внутри этого блока информации представлены адрес, представленный обычным текстом, и три ссылки (номер телефона, адрес электронной почты и адрес веб-сайта) посредством тега <a> (атрибут href тега <a> содержит дополнительную информацию о типе ссылки – tel: для номера телефона и mailto: для электронной почты).

<h2>Контакты:</h2>

Адрес: 141700, город Москва, улица Генерала Белова, дом 49, корпус 2, квартира 228.

Телефон: +7(916)402-24-06

E-mail: akmulin.dmitrij@gmail.com

Веб сайт: <a href="www.lab-irt.com">>

Следующий блок информации отделен заголовком второго уровня <h2> и содержит несколько абзацев, в которых описывается цель подачи резюме, образование соискателя и опыт работы с выделением наименований информации жирным шрифтом.

<h2>Информация:</h2>

Цель: практическое применение знаний и опыта, а также приобретение новых.

Образование: среднее специальное.

Опыт работы:
С 2009 года ФГБУ «Центральная Аэрологическая обсерватория» техник, с 2011 инженер. Разработка приборов измерения параметров окружающей среды и состава атмосферы. Проведение НИР и ОКР. Работа над БПЛА и миниатюрными измерительными приборами. Имеются публикации.

Далее представлен блоки сферы профессиональных интересов и навыков, отделенных от остального текста посредством обычных абзацев, выделенных жирным шрифтом, и представленных в виде маркированных списков (теги <ul> и <li>).

Сфера профессиональных интересов:

разработка программного обеспечения;

разработка электронных блоков;

программирование микроконтроллеров.

Навыки:

Опытный пользователь ПК (Linux (Mandriva/OpenSUSE/Ubuntu/RedHat), Windows (98/XP/Vista/7), Microsoft Office, OpenOffice/LibreOffice, QtCreator, Eagle PCB, Keil, STM32Cube, SolidWorks);

Языки программирования (C++ (Framework Qt), Python, PHP, JavaScript);

Знание ЕСКД;

Водительские права категории «B».

Далее представлена таблица, содержащая информацию об уровне владения навыками посредством использования тегов <table>, <tr>, <th>, <td>. В атрибутах тега <table> указана ширина границы в два пикселя (border), внутренние отступа ячеек, равные 10 пикселям (cellpadding), ширина таблицы, равная 500 пикселям (width) и выравнивание таблицы по центру по горизонтали относительно всей страницы (align="center").

<table border="2" cellpadding="10" width="500" align="center">

<tr>

<th>Навык</th>

<th>Уровень</th>

</tr>

<tr>

<td>Qt</td>

<td>senior</td>

</tr>

<tr>

<td>C++</td>

<td>midle</td>

</tr>

<tr>

<td>PHP</td>

<td>midle</td>

</tr>

<tr>

<td>JavaScript</td>

<td>midle</td>

</tr>

<tr>

<td>Python</td>

<td>junior</td>

</tr>

<tr>

<td>HTML</td>

<td>midle</td>

```
</tr>

<tr>

<td>CSS</td>

<td>midle</td>

</tr>

</table>
```

Далее идет блок с дополнительной информацией отделенный заголовком второго уровня и содержащий информацию о личных качествах и увлечениях.

```
<h2>Дополнительная информация</h2>

<p>

<b>Личные качества:</b><br />Пунктуальность, ответственность, добросовестность, самостоятельность, высокая работоспособность, высокая обучаемость, способность самостоятельно осваивать материал и технологии, мобильность, аналитический склад ума, способность принятия нестандартных решений.

</p>

<p>

<b>Увлечения:</b><br />Походы, спелеология, промышленный альпинизм, страйкбол, компьютерные игры, исторические ролевые игры.

</p>
```

В конце мы завершаем разметку страницы закрывающим тегом,

```
</body>
```

а так же завершаем саму страницу

```
</html>
```

Таким образом, в рамках данной работы была создана страница резюме программиста с помощью HTML5, которая была подробно описана в третьей главе.

Полный код страницы можно найти в приложении 1.

Заключение

В рамках курсовой работы было проведено исследование языков гипертекстовой разметки, а также была создана страница резюме программиста.

В первой главе была рассмотрена история создания языков гипертекстовой разметки, а также дана характеристика следующим языкам: SGML, HTML, XML, XHTML, HTML5. Кроме того были рассмотрены основные понятия языков разметки – тег, элемент, атрибут.

Вторая глава была посвящена структуре кода HTML документа. Были рассмотрены следующие элементы: теги верхнего уровня (<html>, <head>, <body>), комментарии, <DOCTYPE>, метатеги (<meta>, <title>, description, keywords, content-type, content-language, robots, refresh, Author, Copyright, Address), CSS (связанные стили, глобальные стили, встроенные стили), скрипты (<script>), блочные элементы (, <h1> - <h6>, <p>), строчные элементы (<a>, , <b>, <i>,
, <img>), списки (нумерованный, маркированный), таблицы.

В третьей главе подробно описана страница резюме программиста, созданная с помощью HTML5. Каждый элемент снабжен подробным комментарием. Полный код страницы находится в приложении 1.

Библиография

[1]Робсон. Э., Фримен Э. Изучаем HTML, XHTML, и CSS. 2-е изд. - СПб.: Питер, 2014. - стр. 9.

[2]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр. 453

[3]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр. 29

[4]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр. 38

[5]Робсон. Э., Фримен Э. Изучаем HTML, XHTML, и CSS. 2-е изд. - СПб.: Питер, 2014. - стр. 13

[6]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр 18

[7]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр 26

[8]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр27

[9]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр 28

[10]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр 29

[11]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр 30

[12]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр 32

[13]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр 35

[14]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр. 30

[15]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр. 546

[16]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр. 571

[17]Робсон. Э., Фримен Э. Изучаем HTML, XHTML, и CSS. 2-е изд. - СПб.: Питер, 2014. - стр. 716

[18]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр. 27

[19]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр 219

[20]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр 223

[21]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. - СПБ.: Питер, 2012. - стр. 16

[22]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. - СПБ.: Питер, 2012. - стр. 23

[23]Робсон. Э., Фримен Э. Изучаем HTML, XHTML, и CSS. 2-е изд. - СПб.: Питер, 2014. - стр.30

[24]Робсон. Э., Фримен Э. Изучаем HTML, XHTML, и CSS. 2-е изд. - СПб.: Питер, 2014. - стр.31

[25]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. - СПБ.: Питер, 2012. - стр. 26

[26]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. - СПБ.: Питер, 2012. - стр. 26

[27]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. - СПБ.: Питер, 2012. - стр. 26

[28]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. -

[29]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. - СПБ.: Питер, 2012. - стр. 26

[30]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. - СПБ.: Питер, 2012. - стр. 26

[31]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр.16

[32]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр.62

[33]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр.27

[34]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр.50

[35]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр.62

[36]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр.50

[37]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр.63

[38]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр.99

[39]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр.74

[40]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр.114

[41]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр.63

[42]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр.114

[43]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. - СПБ.: Питер, 2012. - стр.39

[44]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр.288

[45]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр.72

[46]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. - СПБ.: Питер, 2012. - стр.40

[47]Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. - СПБ.: Питер, 2012. - стр.88

[48]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр.262

[49]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр.57

[50]Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6-е издание. - Пер. с англ. - СПб: Симпол-Плюс, 2008. - стр.59

[51]Роббинс Дж. HTML5, CSS2 и JavaScript. Исчерпывающее руководство. / Дженнифер Роббинс; Пер. с англ. 4-е изд. - М.: Эксмо, 2014. - стр.166

Приложение 1

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="description" content="Страница резюме приложение к курсовой работе">

<meta name="keywords" content="Дмитрий Акмулин резюме визитка курсовая работа">

<meta name="author" content="Дмитрий Акмулин">

<title>Резюме</title>

</head>

<body>

<h1>Акмулин Дмитрий Валерьевич</h1>

<p><b>Дата рождения:</b> 10 марта 1988 года</p>
```

Семейное положение: женат, трое детей.

<h2>Контакты:</h2>

Адрес: 141700, город Москва, улица Генерала Белова, дом 49, корпус 2, квартира 228.

Телефон: +7(916)402-24-06

E-mail: akmulin.dmitrij@gmail.com

Веб сайт: www.lab-irt.com

<h2>Информация:</h2>

Цель: практическое применение знаний и опыта, а также приобретение новых.

Образование: среднее специальное.

Опыт работы:

С 2009 года ФГБУ «Центральная Аэрологическая обсерватория» техник, с 2011 инженер. Разработка приборов измерения параметров окружающей среды и состава атмосферы. Проведение НИР и ОКР. Работа над БПЛА и миниатюрными измерительными приборами. Имеются публикации (См. приложение 1).

Сфера профессиональных интересов:

разработка программного обеспечения;

разработка электронных блоков;

программирование микроконтроллеров.

Навыки:

Опытный пользователь ПК (Linux (Mandriva/OpenSUSE/Ubuntu/RedHat), Windows (98/XP/Vista/7), Microsoft Office, OpenOffice/LibreOffice, QtCreator, Eagle PCB, Keil, STM32Cube, SolidWorks);

Языки программирования (C++ (Framework Qt), Python, PHP, JavaScript);

Знание ЕСКД;

Водительские права категории «B».

<table border="2" cellpadding="10" width="500" align="center">

<tr>

<th>Навык</th>

<th>Уровень</th>

</tr>

<tr>

<td>Qt</td>

<td>senior</td>

</tr>

<tr>

<td>C++</td>

<td>midle</td>

</tr>

<tr>

<td>PHP</td>

<td>midle</td>

</tr>

<tr>

<td>JavaScript</td>

<td>midle</td>

</tr>

<tr>

<td>Python</td>

<td>junior</td>

</tr>

<tr>

<td>HTML</td>

<td>midle</td>

</tr>

<tr>

<td>CSS</td>

<td>midle</td>

</tr>

</table>

Личные качества:

Пунктуальность, ответственность, добросовестность, самостоятельность, высокая работоспособность, высокая обучаемость, способность самостоятельно осваивать материал и технологии, мобильность, аналитический склад ума, способность принятия нестандартных решений.

Увлечения:

Походы, спелеология, промышленный альпинизм, страйкбол, компьютерные игры, исторические ролевые игры.

</body>

</html>