**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯПРОФЕССИОНАЛЬНАЯ**

**ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ**

**«КУБАНСКИЙ ИНСТИТУТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ»**

**ИНДИВИДУАЛЬНЫЙ ПРОЕКТ**

**по дисциплине «Информатика»**

**на тему «Разработка электронного пособия HTML-язык гипертекстовой разметки»**

Выполнила студентка группы 20-Ф1-9

Специальность 33.02.01 Фармация

Хомич Яна Владимировна

Руководитель:

Преподаватель, Пясецкий С.А.

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Краснодар, 2021

СОДЕРЖАНИЕ

[Введение 2](#_Toc71816707)

[1 Редакторы HTML-документов 4](#_Toc71816708)

[1.1 Текстовые редакторы 4](#_Toc71816709)

[1.2 Специализированные HTML-редакторы 4](#_Toc71816710)

[1.3 Возможности других приложений по переводу в HTML 4](#_Toc71816711)

[2 Синтаксис HTML 6](#_Toc71816712)

[3 Структура HTML-документа 8](#_Toc71816713)

[4 Адресация ресурсов 9](#_Toc71816714)

[5 Графические веб-форматы 12](#_Toc71816715)

[6 Основные элементы HTML 13](#_Toc71816716)

[7 Комментарии 16](#_Toc71816717)

[8 Символьные объекты 17](#_Toc71816718)

[Практическая часть 19](#_Toc71816719)

[ЗАКЛЮЧЕНИЕ 21](#_Toc71816720)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc71816721)

# **ВВЕДЕНИЕ**

Язык HTML был разработан английским ученым Тимом Бернесом-ли в 1991 году как средство структурирования и форматирования документов с целью обмена научной и технической документацией. Текстовый формат представления веб-документов был выбран исходя из основных требований к веб-документу: простота, возможность непосредственной интерпретации в любой операционной системе, минимальный размер файла, удобство редактирования.

Язык разметки гипертекстовых документов HTML позволяет определить различные типы элементов, обеспечивающих структурную функциональность документа: заголовки, разделы, абзацы, списки, таблицы, изображения, гиперссылки, встроенные объекты различных мультимедийных форматов и т.д. Элементы HTML объявляются с помощью команд разметки, называемых тегами 1 (от английского tag – ярлык). Встречающиеся в тексте документа HTML-теги интерпретируются браузером при отображении документа. Совокупность допустимых элементов, правила их объявления, синтаксис определяются в стандарте HTML. Стандарты разрабатываются на основании многочисленных предложений, проходят ряд стадий. С 1994 года по настоящее время разработка проводится под эгидой надгосударственной организации World Wide Web Consortium (W3C). Вся документация на сайте W3C представлена на английском языке, однако есть и ссылки на переводы. Основными вехами на пути развития стандартов HTML можно считать принятие стандартов HTML 1.2, HTML 2.0, HTML 3.2, HTML 4.0. На момент написания этого пособия последней разработкой является стандарт HTML 4.01.

В Интернете представлены разные типы информационных ресурсов, передача которых с компьютера на компьютер подчиняется определенным сетевым протоколам. Для гипертекстовых документов системы WWW это протокол http (Hyper Text Transfer Protocol - протокол передачи гипертекстов, то есть тех самых HTML-документов, о которых идет речь).

Браузер - синоним программы-клиента. Программа-сервер обеспечивает хранение информационных ресурсов и выдачу их по запросам программ-клиентов. Программа-клиент (браузер) должна уметь формировать запросы серверу, принимать и интерпретировать для пользователя получаемую с сервера информацию. За годы развития Интернет был создан целый ряд программ-браузеров, первые из которых были текстовыми (то есть в принципе не были рассчитаны на показ графики).

• Браузеры разных фирм иногда по-разному интерпретируют встречающиеся в HTML-документе теги; есть возможности «за пределами стандарта», поддерживаемые только браузерами конкретной фирмы. При их использовании иной раз приходится для каждого браузера создавать свою версию сайта.

• в случае, когда в HTML-документе встречаются теги, неизвестные браузеру, он их просто игнорирует без всяких сообщений об ошибках.

• Сайт приходится тестировать в браузерах разных производителей и разных версий, при разных разрешениях экрана, добиваясь приемлемого результата.

# 

# **1 Редакторы HTML-документов**

* 1. **Текстовые редакторы**

Поскольку HTML-документ представляет собой текстовый файл, его можно подготовить в простейшем текстовом редакторе, например, в блокноте (notepad). Очень удобен редактор textpad, позволяющий в одном из своих режимов вставлять в текст HTML-теги или символьные объекты HTML из предлагаемого списка. Прекрасно реализованы в этом редакторе и многие другие возможности, необходимые при работе с HTML-документами (а также с текстами программ на ряде языков программирования).

# **1.2 Специализированные HTML-редакторы**

Наиболее, часто используют специализированные редакторы, предназначенные именно для подготовки HTML-документов. Все они по интерфейсу принципиально не отличаются друг от друга, позволяя вставлять команды HTML-разметки с помощью кнопок панелей инструментов либо команд меню и задавать атрибуты HTML-элементов в диалоговых окнах. При этом разработчик видит документ таким, каким он будет при просмотре с помощью браузера. Возможен также просмотр и правка исходного текста HTML-документа.

# **1.3 Возможности других приложений по переводу в HTML**

HTML-документ может быть получен конвертацией в HTML из других форматов. Например, программы широко используемого пакета MS Office, начиная с версии 97, позволяет сохранить документ в формате HTML. При этом сохраняются, насколько это возможно, особенности форматирования символов и абзацев, изображения, таблицы, списки и так далее. При такой конвертации исходный текст HTML-документа получается крайне избыточным и нуждается в коррекции. Даже многие современные графические редакторы предоставляют возможность сохранить коллекцию картинок в виде HTML-документа.

Практика показывает, что наиболее рационально пользоваться специализированными HTML-редакторами в сочетании с хорошим текстовым редактором типа textpad для правки исходного HTML-кода.

# **2 Синтаксис HTML**

HTML-документ представляет собой текстовый файл, содержащий собственно текст, который должен быть отображен в окне браузера, и команды разметки - HTML-теги, задающие элементы документа, которые определяют его внешний вид и функциональность. HTML-тег записывается в угловых скобках (<>) и состоит из имени элемента, за которым может следовать список атрибутов (для многих элементов необязательный). Имена элементов и атрибутов представляют собой английские слова и аббревиатуры, и почти всегда их смысл прозрачен.

Элементы условно можно разделить на две группы. Большинство элементов (называемые также контейнерами) задаются с помощью трех компонентов: начальный тег, содержимое и конечный тег. В начальном теге в угловых скобках указывается имя элемента и список атрибутов, в конечном – только имя элемента, предваряемое символом слэш (/). Содержимое элемента располагается между начальным и конечным тегами и интерпретируется браузером согласно правилам, определенным в спецификации стандарта HTML. Например, элемент b (от английского bold) задает полужирное начертание для текста, расположенного между тегами и .

Некоторые элементы могут вкладываться друг в друга. Например, вложение элемента b (полужирное начертание) в элемент i (курсив) обеспечит полужирный курсив. Таким образом, фрагмент документа начало текста один два три продолжение текста будет отображен браузером так: начало текста один два три продолжение текста .

Элементы другой группы (называемые также автономными) не имеют содержимого и конечного тега. При их интерпретации в отображаемый документ вставляется тот или иной объект. Например, элемент, встречающийся в тексте HTML-документа, вызывает вставку графического изображения из файла pict.gif. Для большинства элементов можно задавать свойства, называемые атрибутами. Атрибуты в списке отделяются друг от друга одним или несколькими пробелами, либо символами табуляции, либо символами перевода строки; последовательность записи атрибутов несущественна. Подавляющее большинство атрибутов употребляются в виде пары название атрибута = значение атрибута.

В случае, когда значение атрибута представляет собой более чем одно слово, его следует заключить в одинарные или двойные кавычки (в спецификации HTML рекомендуется использовать кавычки всегда). Вот пример начального тега элемента img с атрибутами, смысл которых вполне понятен даже до разговора о структуре веб-документа и определениях основных HTML-элементов: Встретив такой тег в HTML-документе, браузер интерпретирует его, выводя изображение из файла cat.jpg, шириной 300 и высотой 200 пикселов. Элемент img относится к числу «автономных» и не имеет конечного тега.

В HTML для абзаца предусмотрен соответствующий атрибут, и он называется align. Начальный тег объявляет новый абзац с указанным выравниванием, потом идет содержимое абзаца (текст, встроенные изображения и т.д.), конечный тег объявляет конец абзаца. После конечного тега можно объявлять новый абзац, возможно, с другим атрибутом выравнивания. Регистр символов в записи имен элементов и имен атрибутов значения не имеет, а вот значения атрибутов могут быть регистрозависимыми. Например, регистр важен при указании путевых имен файлов.

# **3 Структура HTML-документа**

HTML-документ заключается в теги. Между этими тегами располагаются два раздела: раздел заголовка (элемент head) и раздел тела документа (элемент body для простого документа либо элемент frameset, задающий набор кадров). Все указанные элементы имеют начальный и конечный тег. Раздел заголовка содержит описание параметров, используемых при отображении документа, но не отражающихся непосредственно в окне браузера.

Раздел тела документа содержит текст, предназначенный для отображения браузером и теги, указывающие на способ форматирования текста, определяющие графическое оформление документа, задающие параметры гиперссылок и так далее. Формально, согласно спецификации, HTML 4, первым в документе должен указываться элемент doctype, сообщающий браузеру об использованной версии HTML . В элементе doctype указывается также адрес, с которого браузер может загрузить определение типа документа - DTD (Document Type Definition). На практике же этот элемент зачастую опускают без ущерба для отображения документа. Вот пример самого простого HTML-документа:

Браузер отобразит этот документ, выведя в строке заголовка своего окна текст «Самый простой HTML-документ», а в самом окне текст «Проще не бывает», причем параметры шрифта и цвет фона будут зависеть от умолчаний конкретного браузера. На практике в разделе заголовка следует задавать определенную информацию. Для элемента body могут быть указаны атрибуты, определяющие цвета текста и фон документа. Согласно спецификации, HTML 4 лишь два элемента документа являются обязательными: doctype и title.

# **4 Адресация ресурсов**

Местоположение информационных ресурсов в WWW может быть определено в абсолютной и относительной форме.

Адресация в абсолютной форме используется при ссылках на ресурсы, находящиеся на других серверах. Универсальный адрес, определяющий местонахождение информационного ресурса, носит название URL (Uniform Resource Locator - унифицированный адрес ресурса). URL состоит из двух частей, разделенных двоеточием. Первая часть указывает на тип сетевого протокола, который зависит от типа ресурса. Например, если ресурс располагается на WWW-сервере, это протокол http. Вторая часть включает в себя имя компьютера (сервера) в доменной системе имен и путевое имя файла. При записи путевого имени названия каталогов разделяются прямой косой чертой (символ /). В именах файлов и каталогов различаются прописные и строчные буквы; пробелы не допускаются. Примеры URL: http://www.design.ru ftp://bookstore.com/soft/prog.zip mailto:

Ivanov@hotmail.com

http://vvz.nw.ru/Lessons/HTML/index.htm

Последний URL, указывающий на протокол mailto и адрес электронной почты, стоит особняком в ряду других, поскольку не ссылается ни на какой информационный ресурс. Это один из двух типов URL, при записи которого не ставятся после двоеточия две косые черты (второй такой тип – это URL, ссылающийся на сценарий JavaScript).

Адресация в относительной форме используется при ссылках на ресурсы, находящиеся на том же сервере. При записи пути к файлу названия каталогов разделяются косой чертой (/). Переход по каталогам компьютера на один уровень вверх обозначается двумя точками (..). Представим фрагмент файловой структуры некоего сайта, включающий распределенные по соответствующим каталогам HTML-файлы и изображения в формате jpg и gif

В качестве «легенды» примем следующую. Это сайт организации, предоставляющей своим сотрудникам в каталоге Staff подкаталоги для организации «персональных страничек». Иванову выделен подкаталог Ivanov. По названиям подкаталогов и файлов, Иванов увлекается фотографией и публикует фотогалерею (файл photo\_gallery.htm). Фотографии находятся в подкаталоге Photo, в котором есть вложенный подкаталог Nature с фотографиями объектов природы. Стартовая страница сайта лежит в корневом каталоге и называется index.htm. Стартовая страница Иванова также называется index.htm и лежит во вложенном каталоге Staff/Ivanov. Приведем несколько относительных адресов для возможных актуальных ссылок.

• Ссылка со стартовой страницы сайта на фотогалерею Иванова: Staff/Ivanov/photo\_gallery.htm

•Ссылканаизображениеforest1.jpg: Photo/Nature/forest1.jpg

• Ссылка со стартовой страницы Иванова «на главную»: ../index.htm

• Ссылка на логотип организации со стартовой страницы Иванова: ../Images/logo.gif При всей искусственности примера, он соответствует принципам, которым нужно следовать:

• Файлы каждого отдельного раздела сайта и его подразделов следует размещать в отдельных вложенных каталогах. Эта иерархическая файловая структура должна соответствовать логической структуре сайта и может иметь любую степень вложенности.

• Имена файлов и каталогов могут содержать латинские символы, цифры, символ подчеркивания (\_) и дефис (-). Никаких пробелов и символов кириллицы. В именах файлов и каталогов важен регистр символов. Для операционной системы Unix (а подавляющее большинство веб-серверов работает именно под ОС этого семейства) прописные и строчные буквы различаются. То есть, например, logo.gif, logo.GIF и Logo.gif – это разные файлы.

• Всегда типичным именем стартовой страницы сайта было имя index. И сейчас имена index.htm, index.html, index.php являются именами стартовых страниц по умолчанию для большинства популярных программ-серверов (например, для веб-сервера Apache). Разумно также имя index давать и стартовым страницам разделов сайта, файлы которых лежат во вложенных каталогах. Именно так поступил Иванов из примера. При адресации WWW-ресурсов (то есть HTML-документов) есть возможность ссылаться не только на файлы, но и на фрагменты HTML-документов .

# **5 Графические веб-форматы**

Браузеры поддерживают три графических формата:

* GIF–GraphicsInterchangeFormat.

При сохранении изображения в этом формате количество используемых цветов не должно превышать 256. Обычно этот формат используют для небольших изображений, содержащих области однородной окраски с четкими границами. Формат GIF позволяет создать прозрачные фрагменты изображения. Этот формат поддерживает анимацию. Анимированное GIF-изображение представляет собой набор сменяющих друг друга кадров, упакованных в один файл.

* JPEG (JPG) - Joint Photography Experts Group Format.

Этот формат допускает сохранение изображений, содержащих миллионы цветов, с тонкими цветовыми переходами. Например, фотографии обычно сохраняются в формате JPEG. Изображения в этом форматене могут иметь прозрачных областей. Анимация не поддерживается.

* PNG - PortableNetworkGraphicsFormat. Этот формат является развитием формата GIF, обладая рядом дополнительных преимуществ, например,:

• более эффективное сжатие; алгоритм сжатия открытый и бесплатный, без лицензионных ограничений;

• сохранение полноцветных изображений с глубиной цвета до 48 бит на пиксел и черно-белых изображений с глубиной цвета до 16 бит на пиксел;

• возможность задания степени прозрачности от нулевой до 100% для каждой точки изображения; Формат PNG не позволяет хранить в одном файле несколько кадров-изображений не поддерживает анимацию.

# **6 Основные элементы HTML**

Концепции HTML 4 Стандарт HTML 4.0 был рекомендован в 1998 году. В дальнейшем были опубликованы версии 4 с уточнениями. Придя на смену крайне неудачному HTML 3.2, стандарт HTML 4 окончательно определил место этого языка разметки в описании веб-ресурсов. Язык HTML предназначен для структурной (логической) разметки документа, и его не следует применять для задания параметров внешнего отображения.

• в стандарте HTML 4.0 и последующих его редакциях сделано несколько акцентов, главные из которых:

* Разделение структуры и представления. Язык разметки HTML следует использовать для описания структуры документа, в то время как параметры внешнего отображения следует задавать с помощью каскадных таблиц стилей CSS.
* Универсальность доступа к Web. Предлагается ряд возможностей для пользователей с физическими недостатками, многоязыковая поддержка. При этом, эффективность вновь создаваемых веб-ресурсов будет зависеть как от разработчиков самих ресурсов, так и от поддержки декларируемых возможностей HTML производителями браузеров.

• в рамках концепции разделения структуры и внешнего представления целый ряд элементов и атрибутов, связанных с внешним представлением документа, объявлены нежелательными (в оригинале deprecated). В настоящее время многие из них все еще используются.Для обеспечения совместимости с предыдущими версиями браузеры еще неопределенно долго будут поддерживать эти элементы и атрибуты. Однако веб-разработчики должны по мере возможности отказываться от «нежелательного» использования конструкций HTML. Следование принципу разделения структуры и внешнего представления позволяет наиболее рациональным образом организовать процесс дальнейшего сопровождения веб-ресурса, связанного с появлением материалов новых форматов, изменением дизайнерской концепции и т.д.

• Стандарт HTML 4.0 и последующие редакции предлагают ряд общих атрибутов (core attributes), которые могут использоваться с большинством HTML-элементов и служат в основном для назначения стиля элемента, либо для ссылки на сценарий обработки события. Использование этих атрибутов позволяет органично сочетать HTML, каскадные таблицы стилей CSS и сценарии JavaScript. Ниже приведен список основных атрибутов. Наиболее универсальные общие атрибуты:

* id = имя задает элементу уникальный в пределах документа идентификатор, который может быть использован как якорь для гиперссылок для ссылки на сценарий
* style = стиль CSS задает стиль элемента
* class = имя класса стиля CSS задает класс стиля элемента title = строка задает заголовок элемента. В современных браузерах текстовая строка, заданная как значение атрибута
* title, «всплывает» при наведении курсора мыши на элемент

Атрибуты, определяющие сценарий (например, на наиболее универсальном языке JavaScript), который выполняется при наступлении соответствующего события в области элемента:

* onClick Нажата и отпущена кнопка мыши
* onDblClick дважды нажата и отпущена кнопка мыши
* onKeyDown Нажата клавиша
* onKeyPress Нажата и отпущена клавиша
* onKeyUp Отпущена клавиша
* onMouseDown Нажата кнопка мыши
* onMouseMove Перемещение курсора мыши
* onMouseOut Увод курсора мыши
* onMouseOver Наведение курсора мыши
* onMouseUp Отпущена кнопка мыши
* Редко используемые общие атрибуты:
* dir Направление вывода текста (значения: ltr - слева направо; rtl - справа налево)
* lang Язык содержимого элемента

# **7 Комментарии**

Текст комментария размещается между ограничителями и не отображается в окне браузера. Комментарии могут быть многострочными. Комментарии используются в двух случаях. Первое - «по прямому назначению» - в комментарии разработчик напоминает себе: «Здесь начинается таблица, параметры которой потом предложит дизайнер. Не забыть описать во внешнем файле!». Второе- удобно заключить в 24 комментарии недоработанные ,либо временно ненужные фрагменты документа.

# **8 Символьные объекты**

Символы,<,> и & являются специальными символами HTML, поэтому при необходимости использовать их в тексте, отображаемом браузером, следует применять особые средства. Есть и другие символы (символы национальных алфавитов), которые требуют кодировки для их отображения при разработке документов на нескольких языках. Иногда необходимо вывести в одном документе не только символы стандартной латиницы и своего национального алфавита. Для того, чтобы сослаться на название немецкого, французского или чешского источника, либо привести цитату на греческом языке.В HTML для этой цели предусмотрены два типа ссылок на символы: мнемонические (character entity references) и числовые (numeric character references). Все мнемонические ссылки начинаются символом & и заканчиваются точкой с запятой; между этими символами задается соответствующий мнемонический код. Часто применяемый символ неразрывного пробела (nonbraking space) имеет код. Символы <,> , " и &, являющиеся служебными для HTML, также имеют хорошо запоминающиеся мнемонические коды: < (less than) , > (greater than), " (qoutation) и & (ampersand). В справочниках можно найти подобные коды для различных спецсимволов и символов нестандартной латиницы. Вот еще несколько примеров:

Таблица 1 – Примеры символов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код | &sect | &copy | &plusmn | &frac34 | &auml | &acirc |
| символ | § | © | ± |  | Ä | â |

Есть много десятков мнемонических кодов, полный список которых приведен в описаниях стандартов HTML и справочниках.

А числовых кодов миллионы (универсальная двухбайтовая кодировка Unicode, объявленная стандартной в HTML 4.0, потенциально позволяет закодировать 65536 символов). Числовые ссылки на символы имеют формат. Число может быть представлено в десятичном или шестнадцатеричном формате.Например, стандартная латиница начинается с A, греческий алфавит - с Α, а кириллица - лишь с А . Упомянем о нескольких актуальных символьных объектах:

Таблица 2-актуальные символьные объекты

|  |  |  |
| --- | --- | --- |
| Символ | мнемонический код | числовой код |
| Неразрывный пробел | &nbsp | &#160 |
| < | &it | &#60 |
| > | &gt | &#62 |
| & | &amp | &#38 |
| “ | &quot | &#34 |
| Знак авторского права © | &copy | &#169 |
| Зарегистрированный товарный знак ® | &reg | &#174 |

Удобнее использовать мнемонические коды. Например, для того, чтобы в окне браузера вывести формулу 2 <3, в тексте HTML-документа придется записать: 2 <3.

Информация о документе в разделе заголовка. В разделе заголовка указывается информация о документе. Кроме заголовка документа - элемента titlle -, эта информация не является синтаксически обязательной, однако правильное заполнение секции head может положительно повлиять на классификацию документа поисковыми машинами. Ограничимся упоминанием лишь двух элементов раздела заголовка: title и meta. Начиная со стандарта HTML 4.0, обязательным является объявление в разделе head с помощью элемента title информации, идентифицирующей документ. Текст, заключенный между тегами, отображается в заголовке окна браузера.Это первая информация о содержании документа, которую видит пользователь еще до загрузки документа. Заголовок должен быть достаточно кратким и в то же время емким, отражать суть документа и заинтересовывать потенциальную аудиторию.

В разделе заголовка обычно помещается и ряд элементов meta с различными атрибутами, предоставляющими дополнительную информацию (метаинформацию) о документе. Элемент meta – автономный, конечного тега не имеет. Обязательным для элемента meta является атрибут content, в котором указывается значение метаинформации. В качестве второго атрибута может использоваться атрибут httpequiv (эквивалентное HTTP-имя метаинформации, которое будет включено в HTTP-заголовок HTML-документа при передаче его клиенту) либо атрибут name, задающий имя метаинформации.

# Практическим заданиемв моём проект является создание электронного пособия в виде презентации. При создании я совершала действия для создания презентации, выбор дизайна, анимации ,поиска нужной информации,её фильтрации. Изучала саму программу MicrosoftPowerPoint. В ней я сделала создание слайдов, переходы, дизайн:

На рисунке 1 изображен скриншот как создать слайд.

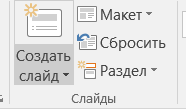


Рисунок 1 - Скриншот создания слайда

На рисунке 2 я показала как выбрать дизайн и где его найти

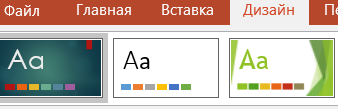


Рисунок 2 - Выбор дизайна

Так же изучила как правильно оформляется титульная страница-рисунок 3



Рисунок 3 - Титульный лист

Так же я закрепила навык использования переходов – рисунок 4

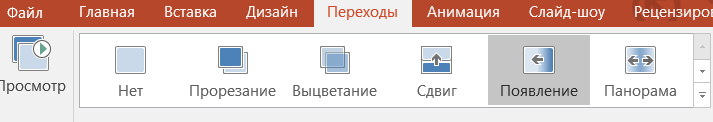


Рисунок 4 - Использование переходов

# **ЗАКЛЮЧЕНИЕ**

В ходе данной работы по подготовке курсовой мною были углублены знания по теме «Язык гипертекстовой разметки HTML». Из проделанной работы можно узнать :

-как преобразовать символы в эквивалентные конструкции;

- как предотвратить ввод пользователями разметки HTML в интерактивных web-приложениях (форум);

-как разместить элементы на странице при её отображении на экране монитора;

-как пользоваться языком разметки гипертекста и что это такое;

-как форматировать текст, таблицы и списки;

-организация ссылок на другие ресурсы;

-вставка изображений и расширений HTML.

Данная работа поможет пользователям понять, как работать с программой HTML. Она помогает разместить элементы на мониторе ,преобразовать символы, форматировать текст ,таблицы и списки ,вставлять изображения и организовать ссылки на другие ресурсы.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Босова,Л.Л.Информатика.10 класс. Базовый уровень/Л.Л.Босова, А.Ю.Босова.-М.:БИНОМ. Лаборатория знаний,2017.-288с
2. Семакин,И.Г.Информатика11класс.Базовый:учебник/И.Г.Семакин ,Е.К.Хеннер, Т.Ю.Шеина.-М,:БИНОМ.Лаборатория знаний ,2016 .-224с
3. Электронное учебное пособие «Язык разметки гипертекста HTML» https://ru.calameo.com/books/001347096f2198d4d740a