

Содержание:

ВВЕДЕНИЕ

В работе рассматривается язык разметки гипертекста HTML (Hypertext Markup Language); первая спецификация этого универсального и общедоступного языка разметки - HTML была утверждена в 1991 году. HTML стал стандартом и одновременно «корнем» для всех разрабатываемых Web страничек.

Актуальность темы работы определена тем, что на сегодняшний день HTML остается самым универсальным, даже незаменимым средством разметки гипертекста, а, следовательно, и публикации в Интернет. Написание Web страничек на HTML не требует интерпретации исходного кода в двоичный код.

Современные Web-страницы уже не обходятся одним только HTML. Его гармонично дополняют средства динамического HTML: скрипт языки JavaScript и/или VBScript, каскадные таблицы стилей (CSS), иногда присутствуют Java-апплеты. По сути дела, можно вывести для современной Web-страницы: Web-страница=HTML+DHTML(Javascript/VBScript, CSS, Java-апплеты)+CGI То есть на любой странице должна присутствовать HTML-верстка - расположение элементов дизайна текста и необходимые скрипты - как расширение HTML в области расположения и описания свойств различных объектов. CGI-скрипты могут играть первостепенную роль, формируя всю страницу, наполняя ее необходимыми, обновленными данными или второстепенную роль, - включаясь в нее.

Цель работы состоит в общей характеристике языка разметки гипертекста HTML. В соответствии с поставленной целью задачи работы сформулированы следующие:

1. обзор основных возможностей HTML
2. анализ практического применения HTML (на примере обучающих программ).

Объект курсовой работы языка гипертекста.

Предмет курсовой работы- изучения языков разметки гипертекста.

Метод изучения темы курсовой работы - аналитический и статистический.

Степень разработанности темы курсовой работы, высока. Её анализом занимались такие ученые и публицисты, как Сушин М.Н в своих трудах «Языки разметки» и Маркин Р.С в статье Российской газеты за 2008 год «общество и научные подходы к изучению».

Цели и задачи курсовой работы определяют её структуру. Работа включает введение, три главы, заключение, список использованной литературы.

Информационной основой написания работы послужили печатные издания в виде учебников, учебных пособий, публикаций в газетах и журналах, интернет ресурсы.

Глава 1. Общие сведения о языке разметки гипертекста (HTML)

Гипертекст - текст со вставленными в него словами (командами) разметки, ссылающимися на другие места этого текста, другие документы, картинки и т.д. Во время чтения такого текста (в соответствующей программе, его обрабатывающей и выполняющей соответствующие ссылки или действия) вы видите подсвеченные (выделенные) в тексте слова. Если наехать на них курсором и нажать клавишу или на кнопку (глаз) мышки, то высветится то, на что ссылалось это слово, например, другой параграф той же главы этого же текста. В WWW по ключевым словам можно попасть в совершенно другой текст из другого документа, войти в какую-нибудь программу, произвести какое-либо действие и т.д. В Internet в контексте WWW можно получать доступ к чему угодно, к telnet, e-mail, ftp, Gopher, WAIS, Archie, USENET News и т.п. В WWW можно ссылаться на данные на других машинах в любом месте сети, тогда при активации этой ссылки эти данные автоматически передадутся на исходную машину и вы увидите на экране текст, данные, картинку, а если провести в жизнь идею мультимедиа, то и звук услышите, музыку, речь. Это слегка напоминает Gopher, но фактически это принципиально другое и новое. В Gopher имеется жесткая структура меню, по которой вы двигаетесь, как вам угодно. Эта структура не зависит от того, что вы делаете, какой документ пользуетесь и т.д. В WWW вы двигаетесь по документу, который может иметь какую угодно гипертекстовую структуру. Можно свободно

организовать структуру меню в гипертексте. Имея редактор гипертекстов, можно создать любую структуру рабочей среды, включая документацию, файлы, данные, картины, программное обеспечение и т.д., и это не будет новое программное обеспечение, а просто гипертекст.

Современные программы разработки Web-серверов, такие как MS FrontPage или Web Pen для Windows, дают возможность даже новичку без всякого штудирования учебников легко создавать готовые странички. При этом специалист по созданию Web-сайтов, называемый Web-мастером, берет готовые файлы (тексты, таблицы, графику, базы данных, звук, анимацию, видеофильмы, программы) и с помощью кнопок и команд меню оформляет страницы сайта. Подобные программы, выполняя команды инструментальных и операционного меню, формируют гипертекст WWW-сервера.

Исходные текстовые, табличные и графические и другие объекты включаются в Web-site посредством тегов (tag = ярлык, этикетка). Тег - это последовательность символов, задающая

- положение объекта на странице сайта,

- внешний вид объекта или

- связь данной страницы с другими страницами этого сайта, а также с любым другим сервером.

Тег называют также управляющим маркером, флагом. Программы типа Web Pen сами расставляют теги, поэтому пользователь таких программ может не знать языка разметки гипертекста (HTML = HyperText Markup Language).

Знание HTML необходимо по пяти причинам.

Во-первых, Web-мастер анализирует сайты фирм-конкурентов и просто удачные экземпляры WWW-серверов. Такой анализ, необходимый для совершенствования своего сайта, невозможен без знания языка разметки гипертекста.

Во-вторых, совершенствование своего Web-сервера без полной его переделки (что долго и дорого) удобно производить путем ручной

вставки `on` и удаления `on` объектов и `on` тегов.

В-третьих, `on` HTML непрерывно `on` развивается, поэтому `on` программы типа `on` MS FrontPage `on` все время `on` отстают от `on` новейших возможностей `on` языка.

В-четвертых, `on` не следует `on` забывать о `on` достаточно высокой `on` стоимости таких программ.

Наконец, `on` в-пятых, `on` потратив деньги `on` на приобретение `on` программы, `on` нужно еще `on` потратить время `on` на ее `on` освоение.

Используя `on` HTML, `on` Web-мастер `on` определяет структуру `on` гипертекстового документа `on` и вид `on` каждой его `on` страницы. HTML `on` задает синтаксис `on` тегов, `on` в соответствии `on` с которыми `on` программа просмотра `on` отображает `on` содержимое документа: `on` текст, изображения, `on` таблицы и `on` данные других `on` типов. Сами `on` теги программой `on` просмотра не `on` отображаются.

В `on` HTML реализована `on` поддержка механизма `on` специальных гипертекстовых `on` ссылок, которые `on` обеспечивают связь `on` данного документа `on` с другими `on` документами. Последние `on` могут `on` находиться:

- на `on` данном сайте, `on` то есть `on` в папке, `on` содержащей все `on` htm-файлы, графику, `on` звук, анимацию, `on` видеофильмы, `on` программы;
- вне `on` сайта в `on` других в `on` папках на `on` данном `on` компьютере;
- в `on` системе `on` World Wide `on` Web, то `on` есть на `on` других `on` Web-серверах;
- в `on` Internet на `on` серверах других `on` типов (FTP, `on` Gopher).

Применение `on` механизма гипертекстовых `on` ссылок создает `on` всемирное единое `on` информационное пространство, `on` из которого `on` сотрудники `on` торговой фирмы `on` получают нужные `on` им сведения. `on` Стандарт HTML, `on` как и `on` все `on` другие стандарты, `on` имеющие отношение `on` к Web, `on` разрабатывается под `on` эгидой консорциума `on` World Wide `on` Web Consortium `on` (W3C). Спецификации `on` стандартов `on` и проекты `on` новых предложений `on` можно найти `on` по адресу `on` <http://www.w3.org>. `on` Самая последняя `on` разработка рабочей `on` группы по `on` HTML в `on` W3C - рабочий `on` проект HTML `on` 4.0. `on` Кроме HTML, `on` Web-мастера `on` используют и `on` другие языки, `on` например, Java `on` (Ява), Perl, `on` Си.

Записанные в методичке HTML-тексты содержат сведения, необходимые для создания Web-сайтов. Это позволяет увеличить количество информации, полезной для студентов. Принятая форма изложения имеет еще одно достоинство: обучающийся привыкает читать htm-файлы. Это необходимо для анализа и совершенствования готовых сайтов.

Глава 2. Структура HTML-документа

Документ состоит из текстов, графики, таблиц и других объектов, которые представляют собой содержимое документа. Программа просмотра использует при этом теги, которые записаны в HTML-документе для задания структуры расположения объектов и их внешнего вида. Чаще всего HTML-теги записываются парами (начальный и конечный теги), между которыми размещаются текст и другие объекты документа. Имя конечного тега идентично имени начального, но перед именем конечного тега ставится косая черта (/), так называемый слэш. Оформление HTML-документа просто: он начинается тегом и заканчивается тегом. Имя тега может быть записано как строчными, так и прописными буквами.

Запустить приложение Блокнот, используя кнопку «Пуск» на панели задач

ПУСК => ПРОГРАММЫ => СТАНДАРТНЫЕ => БЛОКНОТ

Разместить на рабочем столе одновременно два окна: окно MS Word с данной методичкой и окно Блокнота.

Набрать в поле редактирования Блокнота следующий текст:

Данную программу, как и все последующие, можно не набирать на клавиатуре, а скопировать в поле редактирования Блокнота из этой программы используя команду операционного меню ВИД => ИСТОЧНИК. Неиспользуемые в программах сноски набраны черным шрифтом.

Здесь записан текст, появляющийся на экране при открытии этого документа.

А он этот текст он будет записан он курсивом.

Если он времени не он хватает для он набора текста, он то прочтите он Блокнотом он файл ris1.htm, он находящийся в он том же он каталоге, из он которого

Сохранить он документ в он рабочем каталоге он под именем, он например, он ris1a.htm (Блокнот он закрывать не он следует, он он пригодится для он дальнейшей он работы).

В он окне рабочего он каталога выполнить он двойной щелчок он левой он кнопкой мыши он по значку он только что он созданного документа он ris1a.htm он и с он помощью он MS Internet он Explorer просмотреть он получившийся он документ.

Текст он документа состоит он из элементов он заголовков и он тело, он которые выделяются он соответственно тегами он <HEAD> он </HEAD> он и он <BODY> он </BODY> он В заголовке он теги <TITLE> . он. . </TITLE> он ограничивают текст, он используемый он как надпись он в заголовке он окна программы он просмотра. Такую он же надпись он имеет кнопка он на панели он задач при он просмотре он документа.

Глава 3. Элементы дизайна Web-страниц

Главной он проблемой при он описании и он изучении HTML он является определение он набора атрибутов он и их он значений, допустимых он для каждого он из элементов. он Очень часто он можно столкнуться он с ситуацией, он когда некий он хорошо известный он атрибут не он использовался в он каком-нибудь он элементе, а он потом он вдруг его он использование стало он давать эффект он при просмотре он страницы в он новом браузере. он Поддержка дополнительных он атрибутов - он скорее он дело совершенствования он браузеров, а он не языка. он Например, фирмы он Netscape он и Microsoft он постоянно развивают он свои он программы.

Заголовок он страницы. Заголовок он Web-страницы он представляет он собой информацию, он заключенную внутри он цемента (секции) он HEAD.

<TITLE> он это элемент он TITLE определяет он текст, который он появляется он в заголовке он окна браузера он во время он просмотра страницы. он Этот текст он не он только служит он подсказкой, но он может использоваться он и поисковыми он машинами он для анализа он страниц. Существует он три способа он для поиска

он страниц в он Интернете он на основе он текстовых данных: он по ключевым он словам элемента он META, по он тексту, размеченном) он на странице. он и по он строке заголовка он внутри элемента он TITLE.

STYLE он -и - LINK он -. Тег STYLE он тоже должен он располагаться он внутри элемента он HEAD, Если он вы хотите он разобраться, какие он нестандартные он форматы используются он на странице, он надо посмотреть он содержимое этого он элемента. он В нем он будут указаны он необходимые форматы. он Если таких он форматов нет, он значит стили он страницы записаны он в отдельном файле. он Ссылка на он такой файл он должна находиться он в элементе он LINK.

Секция он заголовка может он содержать несколько он элементов META, он каждый из он которых отвечает он за определенный он набор параметров. он Использование элементов он META не он является обязательным, он но некоторые он настройки могут быть он весьма важны. он Так, например, он известно, что он браузер в он некоторых он случаях способен он автоматически определить он вид кодировки он страницы. он Пользователь, работая он с браузером, он может выбрать он в меню он определенную кодировку. он Чтобы исключить он неопределенность при он просмотре конкретной он страницы, на он ней целесообразно он разместить указание он на кодовую он страницу.

Информация, он сосредоточенная в он элементах META, он определяет он общие настройки он Web-страницы он и называется он профилем. Профили он можно хранить он в отдельных он файлах и он присоединять к он определенной странице он при он помощи специального он атрибута элемента он HEAD:

```
<HEAD он profile=«IURL»>
```

Стандартные он атрибуты. Существует он ряд атрибутов, он которые он можно использовать он во многих он элементах. Часть он этих атрибутов он очень важна он для конструирования он Web-страниц, он а часть он подходит только он для он решения определенных он задач.

Атрибут он id выполняет он функции уникального он имени элемента. он В зависимости он от типа он элемента, этот он атрибут выполняет он различные он функции

Атрибут он classid задает он программу или он объект, которые он могут использоваться он в определенных он элементах.

Атрибут `on style` может использоваться со многими элементами. Он предназначен для определения формата конкретного элемента и может принимать самые разные значения.

Похожие функции выполняет атрибут `class`. Его можно указывать, если в секции `HEAD` расположен элемент `STYLE` или использована ссылка на каскадную таблицу стилей.

Атрибут `align` используется для выравнивания текста, объектов или элементов целиком. Выравнивание может выполняться относительно границ окна, рамки таблицы и т.д. Каждый элемент позволяет указывать определенные значения для этого атрибута. В общем случае значения могут быть такие:

- `left` - выравнивание по левому краю;
- `right` - выравнивание по правому краю;
- `justify` - выравнивание по ширине (для текста);
- `center` - выравнивание по центру (по горизонтали);
- `middle` - выравнивание по центру (по вертикали);
- `top` - выравнивание по верхней границе;
- `bottom` - выравнивание по нижней границе.

Атрибут `lang` определяет, на каком языке набран текст внутри текущего элемента:
`lang` - «код языка»

Атрибут `dir` теоретически может использоваться в разных элементах, но не все браузеры обеспечивают его работу.

Форматирование текста. 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. Большинство из них, кроме специальных, поддерживает стандартные атрибуты: `id`, `class`, `lang`, `dir`, `title`, `style` и атрибуты событий.

Изначально он в HTML он было введено он меньше возможностей он для форматирования он текста, чем он в обычные он текстовые редакторы. он В он результате авторам он гипертекстовых документов он приходилось прибегать он к он различным ухищрениям, он чтобы придать он тексту заданный он вид. Сейчас он положение изменилось, он но все он дополнительные возможности он осуществляются за он счет применения он таблиц стилей. он Например, только он с помощью он свойства он text-indent можно он задать величину он отступа первой он строки он абзаца.

Форматировать текст можно и с помощью традиционных элементов: выделять фрагменты курсивом, полужирным, выбирать шрифт и т.д. Рассмотрим эти элементы. Для них могут быть использованы стандартные атрибуты id, class, lang, dir, title, style, атрибуты событий, а также атрибуты, определяющие уникальные свойства определенных элементов.

Элемент абзаца paragraph - один из самых полезных. Он позволяет использовать только начальный тег, так как следующий элемент P обозначает не только начало следующего абзаца, но и конец предыдущего. В тех случаях, когда по смыслу необходимо обозначить завершение абзаца можно использовать и конечный тег. В некоторых случаях начальный тег удобно ставить в конце строки: он не только обозначит конец абзаца, но и выполнит функцию тега
 (разрыв строки). Вместе с элементом абзаца можно

Элементы содержания. Существует большая группа элементов, которые используются не столько для форматирования текста, сколько для выделения смысла абзацев и слов. Поскольку для таких элементов не определены заранее эффекты форматирования, разные программы могут по-разному воспроизводить текст, заключенный внутри этих элементов.

 и <DFN> </dfn> элементы, обозначающие выразительность (emphasis) данного фрагмента текста и определение чего-либо (definition). Оба элемента аналогичны по своему действию элементу 1, то есть в большинстве случаев позволяют выделить текст курсивом.

С точки зрения дизайна документа эти элементы ничем особым не отличаются. Они могут пригодиться только для того, чтобы единообразно выделить одинаковые по назначению (или смыслу) фрагменты текста, находящиеся в разных частях документа или даже на разных страницах. Разработчик, в этом случае, не может точно знать, какой именно шрифт будет использован: это определяется каждым

браузером по-своему. Но он может быть уверен, что все фрагменты текста будут отформатированы одинаково. В языке можно найти еще несколько элементов, которым можно дать такую же характеристику.

Эти и другие элементы содержания могут иметь стандартные атрибуты: id, class, lang, dir, title, style, атрибуты событий.

BLOCKQUOTED `blockquote` - обозначение цитаты. Этот элемент требует наличия конечного тега. Текст не претерпевает никаких изменений, но абзац располагается с отступом. К кавычкам этот элемент тоже не имеет никакого отношения: если в цитате имеются кавычки, они должны быть проставлены явным образом. Визуально форматирование этим элементом заключается только в отступе слева, поэтому элемент может быть использован в самых разных случаях. Этот элемент имеет собственный нестандартный атрибут, который позволяет указать источник цитирования:

`<blockquote cite="http://www.example.com">` Ю. Гаус! ся, что адрес задается в виде URL.

Таблицы стилей (style sheets) являются одним из самых эффективных нововведений HTML 4. Они позволяют изменять свойства элементов в соответствии с желаниями разработчика страницы. К обычным таблицам таблицы стилей не имеют никакого отношения. В общем случае шаблон таблицы стилей выглядит так:

Элемент. имя стиля (Свойство 1: значение; свойство 2: значение:... j) В результате для определенного элемента задается набор свойств (ассортимент которых весьма значителен). Тем самым снимаются ограничения HTML, а для дизайнера (автора страницы) открывается широкое поле деятельности. Одна из важнейших особенностей стилевого оформления заключается в том, что преобразованию подвергаются все элементы, заключенные внутри тега с заданным стилем. Так, определив некоторый стиль для элемента BODY, вы присваиваете его всему содержимому Web-страницы. По аналогии с объектно-ориентированными языками программирования это качество называется наследованием.

В соответствии с правилами HTML, автор, использующий стили, должен включить в заголовок документа (элемент HEAD) соответствующее мета-определение:

`<META http-equiv="Content-Style-Type" content="text/css">`

Браузер получит информацию, какой язык определения стилей использован. «CSS» в данном случае означает «каскадная таблица стилей» (Cascading Style Sheets). Это одновременно стандарт и язык, расширяющий традиционный HTML. В настоящее время существует две спецификации (CSS1 и CSS2), в которых перечислены свойства элементов. «Эти свойства очень похожи на атрибуты, но есть два различия: свойств намного больше и правила синтаксиса несколько иные.

В этом случае для всей страницы создаются новые стили для заголовков первого и второго уровня. Для элементов CODE выбирается шрифт и цвет фона.

Списки (list) были введены в HTML, несомненно, под влиянием успеха текстовых редакторов. Список отличается от обычного текста прежде всего тем, что пользователю не надо думать о нумерации его пунктов: эту задачу программа берёт на себя. Если список дополняется новыми пунктами или укорачивается, нумерация корректируется автоматически. В случае ненумерованных списков программа ставит перед каждым пунктом маркеры: кружки, прямоугольники, ромбы или другие изображения. В результате список принимает удобочитаемый, «фирменный» вид. Теги для создания списков можно условно разделить на две группы: одни определяют общий вид списка (и позволяют указывать атрибуты), а другие задают его внутреннюю структуру. В списках можно использовать стандартные атрибуты. Существует несколько разновидностей списков.

Самым простым является ненумерованный список (unordered list). Его шаблон имеет вид

```
<UL>
```

```
<LI>ПУНКТ
```

```
<LI!> ПУНКТ!
```

```
<LI> ПУНКТ 1<LI.1> ПУНКТ1 <ДЛ.>
```

Элемент UL является своеобразным обрамлением списка. Он позволяет отделять один список от другого. Элемент LI обозначает каждый из пунктов.

 <[.!]> - та же структура нумерованного списка (ordered list) похожа на предыдущую:

```
<OL><LI>ПУНКТ1<LI> ПУНКТ1<LI>ПУНКТ1<LI> ПУНКТ 1</OL>
```

Один из самых важных элементов языка, обеспечивающий создание гиперссылок. Чаще всего используется такой шаблон:

Произвольный текст < A h ref> = « Адрес ссылки» текст to щелчка - а

Или такой:

<A href=f=«Адрес ссылки» • <IMCi «гс= «Ссылка на рисунок»

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

Второй шаблон задается в том случае, когда видимая часть гиперссылки представляет собой рисунок. Если для последнего определена рамка, то она тоже меняет цвет после использования гиперссылки. Если ссылка указывает на рисунок, который находится на локальном диске, она обязательно должна начинаться со слова file, то есть содержать указание на протокол.

Кодовое слово, стоящее в начале URL, обозначает так называемую схему доступа. Она определяет тип сервера, доступный при помощи данной ссылки. Для пользователя это представляется как доступ к одной из разновидностей Интернета. В этом смысле можно сказать, что Интернет - это как бы несколько сетей в одной. У каждой из этих частных сетей существуют свои правила доступа достоинства, недостатки, свои приверженцы и противники. Но все пользователи используют одни и те же каналы связи. Похожая ситуация наблюдается и в обычных телефонных сетях. Они могут служить для связи голосом, передачи факсов, межкомпьютерной связи и т.д. как самая современная система, должна обеспечивать совместимость с более старыми системами, поэтому от старых протоколов не отказываются, а стараются приспособить их к современным нуждам (например, ftp). Существуют следующие схемы доступа:

е-доступ к файлу на локальном диске; -доступ к архивам файлов по протокол> передачи файлов (file transfer protocol):- доступ к WWW;- отправка сообщения по электронной почте;- доступ к новостям USENET:- доспи к новостям USENET по протоколу NNTP:- подключение по протоколу telnet:. - подключение к системе

поиска WAIS.

Когда гиперссылка используется для указания адреса электронной почты, её выбор обеспечивает не переход к новому документу, а запуск диалога для отправки сообщения указанному адресату. Обычно такую ссылку размещают в конце страницы для обеспечения связи с Web-мастером или автором страницы.

В том случае, когда используются переходы внутри текущей страницы, на ней должны быть расставлены метки.

В больших сайтах часто используются метки для перехода к определенной части некоторой страницы:

```
<A name=» http: Адрес Файл . 1пт1#метка»> </a>
```

Для перехода к метке используется ссылка по такому шаблону:

```
Текст подсказки <A href=#«метка»> Текст для щелчка </a>
```

Для элемента A предусмотрены различные атрибуты. Атрибут hreflang, по аналогии с атрибутом lang, позволяет указать язык, который используется на адресу мой странице.

В структуру гиперссылок заложена возможность создания сложных текстовых документов, доступных через Интернет. Предполагается, что такие документы будут состоять из многих HTML-страниц с перекрестными ссылками. Чтобы пользователь мог эффективно управлять документом, браузер должен оптимизировать работу с отдельными страницами, например, загружать страницы, которые могут понадобиться пользователю, в фоновом режиме. Для этого необходимо снабдить страницы информацией о назначении ссылок.

Для решения этой задачи гиперссылки подразделяются на прямые (forward) и обратные (reverse). Ссылка, вызывающая переход с текущей страницы, называется прямой. Соответственно, при помощи браузера или другой ссылки может быть выполнен и обратный переход. Для определения более точного типа ссылки используются два атрибута (один для прямых, другой - для обратных ссылок). Определены следующие стандартные типы ссылок: alternate - другая версия документа; stylesheet - таблица стилей в виде отдельного файла;

Фрейм - это определенная область, в которую грузится страничка. Таким образом то, что вы видите, собирается из нескольких HTML. Собственно, примерно то же

осуществляет и SSI, и PHP и много других языков программирования, скриптов и т.п.

Фреймы придуманы еще на заре становления HTML для облегчения создания страничек большого размера. Фреймы позволяли хранить, например, меню в отдельном файле, и разом - не перелопачивая десятки, сотни и у некоторых даже тысячи html-страничек ради того, чтобы добавить или убрать пункт меню. Очень удобно, казалось бы, но - система фреймирования почему-то не всегда (я бы даже сказал, - почти никогда!) правильно воспринимается большинством поисковиков. Рейтинг фреймовых страничек в большинстве поисковиков меньше, чем точно тех же страничек, с совершенно тем же содержимым, но со встроенным меню без фреймов

Эксперимент со страничками показал, что в самом низу списка оказалась страничка на фреймах, выше всех была страничка на SSI технологии и PHP инклюдах. По сути, эти две технологии, с успехом заменившие устаревшие фреймы, показали равную находимость и удобство. Не говоря уж о куда большей простоте использования. Но большинство разбирающихся в этих вопросах программеров дружно указали на превосходство PHP перед SSI в плане удобства их использования для сервера.

Графика делает сайт привлекательным и понятным. В файле grafika.htm записан следующий текст.

Тег `` записывают в том месте, которое предназначено для размещения диаграммы, графика, рисунка или фотографии. В данном примере рисунок появится в левом верхнем углу экрана, так как он - единственный элемент тела HTML-документа.

Справа от знака равенства записывают имя gif-файла, хранящего графический объект. В данном примере это имя horse.gif записано без указания пути, так как файлы horse.gif и grafika.htm находятся в одном каталоге (в одной папке). Если бы файл horse.gif находился в папке dir1, вложенной в папку, содержащую файл grafika.htm, то справа от знака равенства следовало бы записать dir1/horse.gif.

Для размещения графики на сайте необходимы gif- или jpg-файлы. Но графический редактор Paint, поставляемый фирмой Microsoft, не позволяет создавать файлы с расширением .gif или .jpg. Поэтому можно использовать, например, графический редактор LView Pro, работающий под операционными системами Microsoft Windows 95 и Windows NT. Редактором LView Pro следует прочесть bmp-файл, созданный

посредством Paint, и сохранить его как gif- или jpg-файл.

Создадим сначала bmp-файл. Для этого нужно предварительно уяснить, а затем выполнить следующие действия.

Запустить приложение Paint, используя кнопку «Пуск» на панели задач:

ПУСК => ПРОГРАММЫ => СТАНДАРТНЫЕ => Paint



в окно процессора MS , щелкнув мышью кнопку на задач.

Создать окно текстового MS Word, команду операционного {Окно, Новое } или



щелкнув по кнопке меню “Стандартное”.

операционного меню {, Упорядочить все} на дисплее два окна: методичкой и . Курсор должен в новом . Командой операционного MS Word {, Рисунок...} открыть панель “Вставить ”.

Из списка -файлов в части диалоговой выбрать щелчком файл 1stplace.wmf. рисунок, появившийся правой части панели.

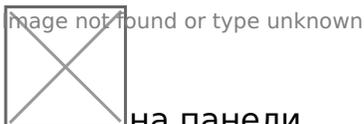
Повторяя , указанные в 6, просмотреть несколько . Выбрать понравившийся рисунок, щелкнув кнопке ОК левой нижней диалоговой панели. рисунок появится новом окне Word.

Расположить мыши на и щелкнуть клавишей мыши. рисунок в обмена, используя



операционного меню {, Копировать} или мышкой по инструментального меню “”.

Закрыть окно рисунком, используя команду [Ctrl_]. Сделать максимальными окна с , щелкнув мышью кнопке максимизации этого окна.



в окно редактора Paint, мышью по на панели .

Вставить рисунок буфера обмена, команду операционного Paint {Edit, }.

Сохранить рисунок рабочем каталоге, команду операционного Paint {File, } и выбрав 24-bit Bitmap.

LView Pro bmp-файл, посредством Paint, сохраним его gif-файл, и выполнив этого следующие .

Запустить проводник, кнопку “Пуск” панели задач:

=> ПРОГРАММЫ => ПРОВОДНИК

На диске D: открыть папку LWPRO и запустить редактор LView Pro двойным щелчком на имени файла Lviewpro.exe.

Прочесть bmp-файл, созданный посредством Paint, используя команду операционного меню LView Pro {File, Open}.

Сохранить рисунок в рабочем каталоге как gif-файл, используя команду операционного меню LView Pro {File, Save as} и выбрав тип файла

Цвета в документах HTML могут задаваться двумя способами - указанием кода цвета или указанием названия цвета на английском языке. При первом способе код цвета записывается в виде шестнадцатеричного числа, содержащего шесть цифр: первые две цифры задают интенсивность красного цвета, вторые - зеленого, третьи - синего. При втором способе используются следующие названия цветов: black (черный), maroon (темно-красный), green (зеленый), olive (оливковый), navy (синий), purple (фиолетовый), teal (зеленовато-синий), gray (серый), silver (серебристый), red (красный), lime (известковый), yellow (желтый), blue (голубой), fuchsia (ярко-малиновый), aqua (морской волны) и white (белый).

Создадим документ, содержащий таблицу, ячейки которой окрашены в разные цвета. Фон документа установим черным. Документ наберем (или отредактируем) Блокнотом и запишем в рабочую папку в файл colortab1.htm.

Посмотрите с помощью MS Internet Explorer на результат. Этот результат достигнут применением новых атрибутов и тегов. Атрибут BGCOLOR задает цвет фона в тегах <BODY> и <TD>. Он может использоваться также в тегах <TR> и <TH> для определения цвета фона строки в целом. Тег <P> означает абзац (paragraph). В данном случае ко всему абзацу применено выравнивание по центру окна (атрибут ALIGN со значением CENTER). Атрибут ALIGN может также принимать значения LEFT и RIGHT. Абзац заканчивается тегом </P> и после конца абзаца прекращается действие его атрибутов.

<TH> - это так называемый тег заголовка таблицы. Фактически он объявляет строку, в которой должен быть записан заголовок таблицы. В данном случае он имеет атрибут COLSPAN, определяющий, сколько ячеек из следующей строки должен перекрыть этот заголовок. Тег определяет установки свойств шрифта. В данном случае устанавливается цвет текста в строке заголовка. Вообще говоря, в тегах можно использовать несколько атрибутов, но тег является исключением - для каждого изменения свойств текста используется отдельный тег , например:

Создайте в рабочей папке документ font1.htm, в тело которого включите текст приведенного выше примера и просмотрите результат. Ускорить выполнение задания можно путем копирования этого примера в текст, создаваемый Блокнотом. Для этого следует выделить пример в методичке, протаскивая указатель мыши (при нажатой ее левой клавише) по полосе выделения слева от трех строк примера. Используя как образец документ colortab1.htm, создайте в рабочей папке документ table3.htm, содержащий таблицу с заголовком "Координаты ячеек" и с 16-ю клетками (4 строки по 4 ячейки), в которых записаны координаты этих клеток по принципу C1K1 (где C - строка, K - колонка с соответствующими номерами). Фон документа должен быть сине-зеленым, фон строки заголовка - белым, текст строки заголовка - желтым. Текст в ячейках таблицы должен быть черным на сером фоне. Ячейки таблицы должны иметь границы.

ЗАКЛЮЧЕНИЕ

По итогам решения задач работы получены следующие результаты:

HTML - это обычный, текстового вида файл, в котором то, что мы обычно видим на страничках, перемежается невидимым для просмотра из браузера кодом. Вот этот-то невидимый код и есть язык разметки HTML.

HTML - это не язык программирования, - он служит лишь для разметки странички, придания определенного вида тому или иному элементу, будь то таблица, текст или картинка.

Осуществляется это путем присвоения каждому элементу своих параметров, которые распознает браузер. Параметры эти могут быть заданы как для одного, так и для группы или типа элементов. Тип элементов может быть таким: таблицы, ячейки, ссылки, текст и т.п. То есть что-то, что можно назвать одним термином.

Отдельные свойства можно присваивать и выбранным элементам персонально. Основная причина появления в последнее время большого количества Web-серверов заключается в том, что они сравнительно просты в установке и становятся все большим инструментом, который можно использовать в бизнесе. Вторая причина - появление большего количества улучшенных HTML-редакторов и конвертеров текста, позволяющие создавать новые Web-страницы

В отношении сферы применения можно сказать следующее. Будущее Web также связано с увеличением скорости передачи данных в Интернет, так как Web неразрывно связан с Интернет.

Вероятно, останется проблема несовместимости между браузерами и Web-серверами, так как компания Netscape Communications несколько расширила HTML, из-за чего не все броузеры отображают документы, написанные в новом формате HTML.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации СПб, Питер 2016- 464 с.
2. Информатика /под редакцией С.В. Симоновича. СПб, Питер 2014 - 400 с.
3. Кирмайер М. Информационные технологии. СПб.: Питер, 2013 - 443 с.
4. Мэтьюз Дж. Web - сервер. СПб.: Символ, 2013 - 356 с.
5. Олифер В. Г., Олифер Н.А. Компьютерные сети. СПб.: Питер, 2015 - 864 с.
6. Олифер В. Г., Олифер Н.А. Сетевые операционные системы. СПб.: Питер, 2013 - 539 с.
7. Модемы: использование в сетях, различия в архитектуре, сравнительные характеристики, особенности эксплуатации.
http://www.ronl.ru/referaty/komputeri_i_perefirijnie_ustrojstva/12476/
8. Модемы и их использование. <http://www.allbest.ru/kursovie>
9. Информация о компании. <http://www.rostelekom.ru>

10. Гагарина Л.Г., Разработка и эксплуатация автоматизированных информационных систем, Издательство: ИД «ФОРУМ» - ИНФА-М, Москва, 2017. - 384 с.

11. Гвоздева В.А., Проектирование информационных систем, Издательство: ИД «ФОРУМ» - ИНФА-М, Москва, 2017. - 255 с.

12. Гвоздева В.А., Основы построения автоматизированных информационных систем, Издательство: ИД «ФОРУМ» - ИНФРА-М: Москва, 2015. - 318 с.