МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

імені Володимира Даля

Кафедра компьютерных наук

Реферат

По дисциплине

Компьютерній дизайн и мультимедия

Студент Болдакова И.В.

Содержание

1. Вступление

2. Рассмотрим язык гипертекстовой разметки HTML

3. Инструментарий для создания Web-страниц

3.1 HTML-редакторы

4. Создание сайта при помощи CMS Joomla 1.5.7

Вывод

Литература

# 1. Вступление

World Wide Web - глобальная компьютерная сеть на сегодняшний день содержит миллионы сайтов, на которых размещена всевозможная информация. Люди получают доступ к этой информации посредством использования технологии Internet. Для навигации в WWW используются специальные программы - Web-браузеры, которые существенно облегчают путешествие по бескрайним просторам WWW.Вся информация в Web-браузере отображается в виде Web-страниц.

Web-страницы, поддерживая технологию мультимедиа, объединяют в себе различные виды информации: текст, графику, звук, анимацию и видео. От того, насколько качественно и красиво сделана та или иная Web-страница, зависит во многом ее успех в Сети.

Пользователю приятно посещать те Web-страницы, которые имеют стильное оформление, не отягощены чрезмерно графикой и анимацией, быстро загружаются и правильно отображаются в окне Web-браузера.

Создать Web-страницу непросто, но наверно каждый человек хотел бы попробовать себя в роли дизайнера. И я, в данном случае, не являюсь исключением, потому и выбрала такую тему для своей курсовой работы.

В своем реферате я сделала попытку разобраться в том, что необходимо знать и уметь для создания Web-страницы, какое программное обеспечение является инструментарием создания Web-страниц и как его эффективно использовать.

Также в данной работе мною рассмотрены основы языка программирования Web-страниц - HTML, который является общепринятым стандартом WWW. Это даст нам возможность ознакомиться со структура Web-страницы и приемами ее правильного оформления. А так же при помощи CMS Joomla мы рассмотрим создание сайта.

# 2. Рассмотрим язык гипертекстовой разметки HTML

Web-страницы могут существовать в любом формате, но в качестве стандарта принят *Hyper Text Markup Language* - язык разметки гипертекстов, предназначенный для создания форматированного текста, насыщенного изображениями, звуком, анимацией, видеоклипами и гипертекстовыми ссылками на другие документы.

Можно работать на Web без знания языка HTML, поскольку тексты HTML могут создаваться разными специальными редакторами и конвертерами. Но писать непосредственно на HTML нетрудно. Возможно, это даже легче, чем изучать HTML-редактор или конвертер, которые часто ограничены в своих возможностях, содержат ошибки или проводят плохой HTML код, который не работает на разных платформах.

Язык HTML существует в нескольких вариантах и продолжает развиваться, но конструкции HTML скорее всего будут использоваться и в дальнейшем. Изучая HTML и познавая его глубже, создавая документ в начале изучения HTML и расширяя его насколько это возможно, мы имеем возможность создавать Web-страницы, которые могут быть просмотрены многими браузерами Web, как сейчас, так и в будущем. Это не исключает возможности использования других методов, например, метод расширенных возможностей, который предоставляется Opera, *Google Chrome,* Internet Explorer или другими браузерами.

Работа по HTML - это способ усвоить особенности создания документов в стандартизированном языке, используя расширения, только если это действительно необходимо.

HTML был ратифицирован World Wide Web Consortium. Он поддерживается всеми браузерами.

Поскольку HTML-документы записываются в ASCI I-формате, то для ее создания может использован любой текстовый редактор.

Обычно HTML-документ - это файл с расширением.html или. htm, в котором текст размечен HTML-тегами (англ. tag - специальные встроенные указания). Средствами HTML задаются синтаксис и размещение тегов, в соответствии с которыми браузер отображает содержимое Веб-документа. Текст самих тегов Веб-браузером не отображается.

Все теги начинаются символом '<' и заканчиваются символом '>'. Обычно имеется пара тегов - стартовый (открывающий) и завершающий (закрывающий) тег (похоже на открывающиеся и закрывающиеся скобки в математике), между которыми помещается размечаемая информация:

<p>Информация</p>

Здесь стартовым тегом является тег <p>, а завершающим - </p>. Завершающий тег отличатся от стартового лишь тем, что у него перед текстом в скобках <> стоит символ '**/**' (слэш).

Браузер, читающий HTML-документ, отображает его в окне, используя структуру HTML-тегов. В каждом HTML-документе должны присутствовать три главных части:

A) Объявление HTML;

B) Заголовачная часть;

C) Тело документа**.**

*A) Объявление HTML*

<HTML> и </HTML>. Пара этих тегов сообщает программе просмотра (браузеру) что между ними заключен документ в формате HTML, причем первым тегом в документе должен быть тег <HTML> (в самом начале документа), а последним - </HTML> (в самом конце документа).

<HTML> </HTML>

*B) Заголовочная часть.*

<HEAD> и </HEAD>. Между этими тегами располагается информация о документе (название, ключевые слова для поиска, описание и т.д.). Однако наиболее важным является название документа, которое мы видим в верхней строке окна браузера и в списках "Избранное (BookMark)". Специальные программы-спайдеры поисковых систем используют название документа для построения своих баз данных. Для того чтобы дать название своему HTML-документу текст помещается между тегами <TITLE> и </TITLE>.

<HTML> <HEAD> <TITLE>Моя первая страница</TITLE> </HEAD> </HTML>

*C) Тело документа.*

Третьей главной частью документа является его тело. Оно следует сразу за заголовком и находится между тегами <BODY> и </BODY>. Первый из них должен стоять сразу после тега </HEAD>, а второй - перед тегом </HTML>. Тело HTML-документа - это место, куда автор помещает информацию, отформатированную средствами HTML.

<HTML> <HEAD> <TITLE> Моя первая страница</TITLE> </HEAD> <BODY> ..........

</BODY> </HTML>

Теперь мы можем написать HTML-код нашей странички:

<HTML> <HEAD> <TITLE>Моя первая страница</TITLE> </HEAD> <BODY> Здесь будут мои страницы! </BODY> </HTML>

В разделе BODY все символы табуляции и конца строк браузером игнорируются и никак не влияют на отображение страницы. Поэтому перевод строки в исходном тексте HTML-документа не приведет к началу новой строки в отображаемом обозревателем тексте при отсутствии специальных тегов. Это правило очень важно помнить и не забывать ставить разделяющие строки теги, иначе у текста не будет абзацев, и он станет нечитаемым.

Для начала новой строки используется тег <br> (сокр. от англ. break - прервать). Этот тег приводит к отображению браузером дальнейшего текста с начала следующей строки. Закрывающий для него тег не используется. Он удобен, если требуется с какого-то места писать с новой строки без начала нового абзаца, например, в стихотворении. Повторное его использование позволяет вставить одну или несколько пустых строк, отодвинув следующий фрагмент страницы вниз.

Сплошной текст без промежутков читается не очень легко, его неудобно просматривать и находить нужные места. Разбитый на абзацы, текст воспринимается гораздо быстрее. Для начала нового абзаца используется тег <p> (англ. paragraph - абзац). Этот тег, кроме начала новой строки, вставляет одну пустую строку. Но многократное повторение <p>, в отличие от <br>, не приведет к появлению нескольких пустых строк, останется все та же одна пустая строка.

Внутри скобок тега кроме его названия могут размещаться также атрибуты (англ. atributes - атрибуты). Они отделяются от названия и между собой пробелами (одним или несколькими), а пишутся в виде *имя\_атрибута*="*значение*". Если значение не содержит пробелов, то кавычки могут быть опущены, но так делать не рекомендуется. Тег <p> может содержать атрибут ALIGN, определяющий выравнивание абзаца. По умолчанию абзац выравнен влево ALIGN="left". Возможны также выравнивания вправо ALIGN="right" и по центру ALIGN="center". При использовании атрибутов, после форматируемого текста следует использовать закрывающий тег </p>. Если его нет, то новый тег <p> означает закрытие предыдущего, соответственно вложенные <p> невозможны. Выравнить текст по центру возможно также тегом <CENTER>.

Теперь мы можем поместить на нашу Web-страницу некоторый текст с различным выравниванием:

<HTML> <HEAD> <TITLE>Моя первая страница</TITLE> </HEAD> <BODY> <p align=center>Здесь будут мои личные страницы! <p align=left>На них Вы сможете найти: <bp> - рассказ обо мне и о моих увлечениях; <bp> - мои фотографии.

<p align=right>С одной из моих страниц можно будет<br>отправить мне электронное письмо.

</BODY> </HTML>

# 3. Инструментарий для создания Web-страниц

# 3.1 HTML-редакторы

Каждый выбирает свой инструмент для создания Web-страниц. Это может быть MS FrontPage или Macromedia DreamWeaver, Allaire HomeSite или 1st Page. А кто-то пользуется простым текстовым редактором, например Блокнотом (Notepad).

Текстовые редакторы возможно использовать только для создания небольших страниц, так как у них есть много минусов: не поддерживаются проекты, отсутствует "подсветка" текста., в общем, работать крайне неудобно.

Основным недостатком MS FrontPage является то, что он генерирует очень большой HTML-код (слишком много лишнего), поэтому страницы получаются большими, что сказывается на скорости загрузки. Более того, при создании Web-страниц в этом редакторе видишь одно, а в окне браузера - совсем другое. Странички получаются какими-то кривыми, поэтому для создания качественных Web-страниц рекомендуется использовать пакеты, которые будут рассмотрены ниже.

Начнем мы с популярного Macromedia DreamWeaver. Компания Macromedia считается лидером по производству программ для создания веб-сайтов, а также законодателем моды в этой области.

***DreamWeaver 3.0***

Но DreamWeaver на несколько шагов опережает другие редакторы, использующие технологию WYSIWYG, в первую очередь тем, что генерирует очень чистый HTML-код. DreamWeaver позволяет вам избавиться от однотипной работы при создании страниц (например, верстка текста) при помощи использования *опции "запись последовательности команд" (*вы записываете последовательность производимых вами команд, потом нажимаете, например, CTRL+P, и DreamWeaver воспроизводит все в той же последовательности.

***HomeSite 4.0***

Следующий редактор - HomeSite 4 - для создания страниц вручную, т.е. для знатоков HTML. Вы получаете полный контроль над HTML-кодом, причем существует возможность оптимизировать свою страничку под один из трех популярных браузеров (MSIE, Google Phrome, Opera).

HomeSite содержит два основных режима: Edit и Design. Режим Design - это подобие WYSIWYG-редактора, выдающее HTML-код, причем, если вы загрузите чужой HTML-код, то HomeSite все перепишет по-своему.

Еще одна отличительная особенность HomeSite - это его "склейка" с Dreamweaver. HomeSite обладает кнопкой "Dreamweaver", а также входит в его стандартный пакет поставки. Впрочем, и DreamWeaver имеет возможность подключения HomeSite, как редактора для коректировки HTML-кода.

Одним из последних HTML-редакторов является EVR Soft 1st Page v2.

Его лозунг - "Create 1st class websites!" ("Создавайте первоклассные веб-сайты!"). Редактор содержит несколько режимов - Normal, Easy, Advanced/Expert и Hardcore, то есть вы можете выбрать свой уровень, а со временем перейти на более высокий. Еще одна особенность - довольно большая коллекция скриптов на JavaScript и DHTML. Все это довольно удобно разбито по категориям.

# 3.2 Графические редакторы

Создание и оптимизация графики - сложная и капризная задача. Безусловно, возможно создание Web-страницы и без использования графики - при помощи шрифтов, скриптов и таблиц стилей (CSS) - и это будет красиво и стильно. Но ведь окончательный вид документа зависит от большого числа различных факторов, таких как: ширина окна браузера, предварительные настройки браузера, принятые по умолчанию размер шрифта, его имя и цвет. К тому же не все скрипты и стили поддерживаются всеми браузерами. Если же будет использована графика, то посетитель вашей страницы увидит ее точно такой, какой сделали и видите ее вы.

Основная сложность работы с Web-графикой состоит в том, что пропускная способность каналов Интернета, в большинстве случаев, очень низкая и перед вами сразу встанут проблемы - как сделать графический файл небольшой по объему, но хорошего качества, какие программы и приемы использовать при его оптимизации.

Именно этому посвящен раздел о векторные и растровые графических редакторах, которые являются мощным инструментом обработки изображения в умелых руках.

Прежде чем рассмотреть векторные и растровые графические редакторы, следует уяснить себе, в чем состоит различие между векторным и растровым представлением изображения.

**Растровая графика** представляет собой сетку (растр), ячейки которой называются пикселами. Каждый пиксел в растровом изображении имеет строго определенное местоположение и цвет, следовательно любой объект представляется программой как набор окрашенных пикселов. Это значит, что пользователь, работая с растровыми изображениями, работает не над конкретными объектами, а над составляющими их группами пикселов.

Растровые изображения обеспечивают высокую точность передачи градаций цветов и полутонов, а также высокую детализацию изображения, поэтому они являются оптимальным средством представления тоновых изображений, таких как сканированные фотографии.

Для изображения растровой графики всегда используется фиксированное количество пикселов, т.е. качество растрового изображения напрямую зависит от разрешающей способности оборудования. Это значит, что любое изменение изображения (поворот, увеличение и т.д.) приводит к неизменному искажению картинки и границы объектов получаются неровными.

**Векторные изображения** формируются на основе математически описанных фигур, называемых векторами, а вид изображения определяется параметрами векторов. Другими словами, векторная графика состоит из кривых, имеющих координаты, цвет и прочие параметры, а также замкнутых областей, заполненных определенным цветом. Границы этих областей также описываются кривыми. Файл с векторной картинкой содержит координаты и параметры кривых.

Результаты обработки векторных изображений не зависят от разрешающей способности оборудования, поэтому вы можете произвольно изменять их параметры (размер, цвет, форму и т.д.) - качество не ухудшится. Векторная графика применяется при создании цифровых объектов с использованием мелких кеглей (размеров шрифта) или таких объектов, как логотипы, для которых важно сохранять четкие контуры, при неограниченном масштабировании.

Графические пакеты (редакторы) тоже делятся на два типа: растровые и векторные. Давайте теперь рассмотрим наиболее популярные из них.

### **А) Редакторы растровой графики**



### Microsoft Paint- простой (или лучше сказать - простейший) редактор, входящий в стандартную поставку операционных систем Microsoft. Он обладает набором простейших функций (кисточка, карандаш, резинка и т.д.), которые позволяют создавать незамысловатые картинки. К сожалению, для обработки графики он практически не пригоден. Картинку, которую вы видете справа - это большее, на что способен этот редактор.

**Adobe Photoshop** - на сегодняшний день это самый мощный пакет для профессиональной обработки растровой графики. Это целый комплекс, обладающий многочисленными возможностями модификации растрового рисунка, имеющий огромный набор различных фильтров и эффектов, причем есть возможность подключать инструменты независимых производителей.



Пакет предлагает, например, средства для восстановления поврежденных изображений, ретуширования фотографий или создания самых фантастических коллажей, которые только может позволить себе наше воображение. В общем, потенциал этого пакета поистине огромен. Начиная с версии 5.5 в пакет включена программа **Adobe ImageReady**, предоставляющие огромные возможности по обработке графики под WEB (оптимизация изображений, создание анимированных gif, "разрезание" картинок на более мелкие и т.д.). Девиз разработчиков Adobe Photoshop - "Camera of your mind" - предполагает не только техническое совершенство, но и полную свободу творчества, на которую человек, работающий с этой программой, просто обречен.

**PhotoPaint** - еще один не менее известный графический редактор (из пакета Corel Draw) для обработки растровой графики, конкурирующий с Adobe Photoshop. Здесь также имеются все необходимые инструменты для обработки графики, разнообразные фильтры, текстуры. Разница лишь в удобстве работы, интерфейсе и скорости наложения фильтров - наложение происходит немного медленнее.

**Painter** - редактор предоставляет великолепные возможности для эмуляции реальных инструментов рисования: графит, мел, масло и т.д. Также позволяет имитировать фактуру поверхности материалов, живопись, создавать анимацию. Очень удобен для разработки фоновых рисунков или Web-страниц в стиле живописи. Пользуясь это программой чувствуешь себя настоящим художником.

Существует еще ряд редакторов (**Microsoft Photo Editor, Microsoft Photo DRAW**), также позволяющих реализовать простейшие задачи, но не удовлетворяющих запросам профессионалов.

### **B) Редакторы векторной графики**



**Adobe Illustrator** - пакет позволяет создавать, обрабатывать и редактировать векторную графику. По своей мощности он эквивалентен

растровому редактору Adobe Photoshop: имеет аналогичный интерфейс, позволяет подключать различные фильтры и эффекты, понимает многие графические форматы, даже такие как. cdr (Corel Draw) и. swf (Flash).

**CorelDraw** - безусловно, такой известный графический пакет не мог обойтись без средств для обработки векторной графики. Пакет по своей мощности практически не уступает графическим редакторам Adobe Photoshop и Adobe Illustrator. Помимо обработки векторной графики, в этом пакете существует обработчик растровой графики (Photo Paint), трассировщик изображений, редактор шрифтов, подготовки текстур и создания штрихкодов, а также огромные коллекции с изображениями (CorelGallery).

**Adobe Streamline** - еще один продукт фирмы Adobe, предназначенный для трассировки (перевода) растровой графики в векторную. Это небольшой, но очень полезный и мощный продукт. Особенно полезен, если вы создаете Web-страницы с использованием векторной графики, например, технологии Flash.

# 4. Создание сайта при помощи CMS Joomla 1.5.7

Плавно, по принципу от простого к сложному, переходим от статических страниц WWW к динамическим. Что бы это осуществить, нам понадобиться знание Web - языков PHP, Jscript, Perl, *jQuery* и хотя бы язык структурных запросов MySQL. Ничего страшного, всего лишь какие то пару лет и вы профи, способные делать такие многофункциональные сайты, у которых страницы генерируются на лету, в которых присутствует базы данных клиетов форумов, логины, пароли, идентификация, аутентификация, подключение анимации и много другое.

Но не обязательно обладать знаниями всех языков, что бы сделать динамический многофункциональный сайт. Мне хватило знания HTML, SCC, PHP, 3Ds Max, Photoshop и SMC Joomla.

Joomla - система управления содержимым, написанная на языках PHP и JavaScript, использующая в качестве хранилища базу данных MySQL. Является свободным программным обеспечением, распространяемым под лицензией GNU GPL.

Основные возможности:

1. Функциональность можно расширять с помощью дополнительных модулей (расширений, плагинов).

2. Модуль безопасности для многоуровневой аутентификации пользователей и администраторов.

3. Система шаблонов позволяет легко изменять внешний вид сайта.

4. Настраиваемые схемы расположения модулей, включая левый, правый и центральный блоки меню.

5. К преимуществам системы можно отнести то, что все модули, компоненты, плагины, шаблоны можно написать самому, разместить их в структурированном каталоге расширений или отредактировать существующее расширение по своему усмотрению.

Можно работать с Joomla на сервере хостера в интернете (дополнительные расходы, а так же зависит многое от скорости интернета). Я предпочитаю работать на локальном сервере. Для этого выбрала Apache/2.2.4 (Win32) mod\_ssl/2.2.4 OpenSSL/0.9.8d PHP/5.2.4 Server at 127.0.0.1 Port 80.

Итак, установили локальный сервер. В директории localhost/home/www/ создаем папку, в которую помещаем саму Joomla. Хочу заметить, что под PHP5.3.1 Joomla работает не корректно. Постоянно компоненты форумов FireBoard и Kunena выдает ошибки на сайте, приходится править файл php. ini и отключать вывод об ошибках, что само действие сулит непредвиденных обстоятельств для сайта. По этому, рекомендую выбирать версию PHP 5.2.4, работа стабильная для всех плагинов, модулей и компонентов. После установки Joomla через браузер создаем в MySQL пользователя-админа и под его логином и паролем заходим на в админ панель. Далее надо выбрать шаблон для сайта. С первого взгляда покажется, что это очень легко, но капнув глубже убеждаешься, что 80% времени уходит для модернизации этого шаблона. Поэтому рекомендую выбирать максимально подходящий шаблон под нужды, не забывая о выводимых модулях. Далее подключаем весь функционал сайта: регистрацию, форум, слайд-шоу, баннеры, файлы для скачек, видео on line, аудио, редактируем, все выводимые страницы. Работаем в основном с файлом Index. php и template. css и конечно конфигурируем под себя все подключенные расширения, которых около 6000 для Joomla. Далее остается самое приятное - это дизайн сайта. При помощи растрового программы Photoshop и 3Ds Max (полнофункциональная профессиональная программная система для создания и редактирования трёхмерной графики и анимации, разработанная компанией Autodesk, содержит самые современные средства для художников и специалистов в области мультимедиа) создаем дизайн шаблона. Редактируем header, делаем нужный вид.

# Вывод

В связи с быстрым развитием Web-программирования и создания множественных CMS, была облегчена работа Web-дизайнера. Теперь для создания многофункционального мощного динамического web портала достаточно освоить несколько языков программирования (если будем дописывать расширения и скрипты), умело владеть графическими и растровыми программами, немного вдохновения, инноваций и мастерства.

# Литература

1. Которев Д. В, Костарев А.Ф. "PHP 5 или наиболее полное руководство в подлиннике." Петербугр 2005

2. А. Гончаров "Самоучитель HTML".

3. А. Матросов, А. Сергеев, М. Чаунин "HTML 4.0 в подлиннике"

4. Айзекс С. "Dynamic HTML"

5. Байенс Дж. "Примочки программирования в WEB"

6. Бурсов М.В. и др. "Основы работы с HTML-редактором Dreamweaver. Учебно-методическое пособие"

7. Велихов В. "Справочник по HTML 4.0"