

Оглавление

Как придумали Интернет?.....	2
Принцип организации сети с коммутацией пакетов	2
Что ждёт Интернет в будущем?.....	4
Понятие гипертекста как способа организации данных.	4
Введение в технологию создания веб-документов средствами HTML.....	8
С помощью чего создаются Web-страницы	9
Заголовки, текст, списки. Вставка изображений и ссылок.....	10
Понятие и структура веб-документов.....	12
Дизайн и свойства страницы.....	12
Работа с текстом.....	14
Списки.....	18
Работа с изображениями.....	20
Цвет фона и текста	22
HTML ссылки	24
Изменение цвета ссылок	26
Разберемся с таблицами в html	26
Веб-ресурсы образовательного назначения (обзор основных классов)....	32
Технология разработки гипертекстовых сред на основе формата HTML.	33
Создание web-сайта средствами web-редакторов.....	37

Как придумали Интернет?

Начальный импульс этому процессу, как ни странно, дали не США, а СССР. В октябре 1957 года СССР запустил первый искусственный спутник Земли, и эта новость в США произвела эффект разорвавшейся бомбы. Как ответный шаг в США было организовано агентство Advanced Research Projects Agency (ARPA), задачей которого стало применение различных научных и технологических разработок в оборонных целях. В том числе и использование компьютеров в структуре военного командования и управления. И вот в 1970-х Винтон Серф вместе с Бобом Каном по заказу ARPA стали заниматься небольшим проектом. В то время уже существовали компьютеры, объединённые в сети, но каждый производитель использовал свои собственные протоколы передачи информации. Военные не хотели стать заложниками какого-то одного производителя, и потому нашей задачей стала разработка общих стандартных сетевых протоколов, которые позволяли бы обмениваться информацией через сети, независимо от их размера, производителя оборудования и программного обеспечения. Это конкретная инженерная задача, Винтон Серф и Боб Кан не ставили перед собой цель изменить мир.

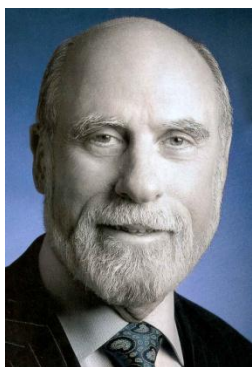


Рисунок 1. Винтон Серф

Идея состояла в том, чтобы использовать принципиально отличную от телефонной архитектуру. Телефонная сеть основана на коммутации каналов: когда вы набираете номер, между двумя телефонами создается прямой канал. Винтон Серф и Боб Кан же работали над сетью ARPANET с коммутацией пакетов - принципом, который обещал намного более широкие возможности.

Принцип организации сети с коммутацией пакетов

Представьте, что у вас есть толстая книга и вам нужно передать содержащуюся в ней информацию своему другу в другом городе, и при этом единственное, что у вас есть под рукой, - это почтовые открытки. Вы вырываете из книги все страницы, каждую разрезаете на части, чтобы она поместилась на открытке, пишете свой адрес и адрес получателя. Потом нумеруете открытки, делаете копию каждой (на случай, если потеряется) и бросаете их все в почтовый ящик.

Так вот, пакеты в протоколе, который сейчас называется Internet Protocol (IP), - это полный аналог открыток. Они также имеют адрес отправителя и получателя и несут некоторую информацию, также не обязательно приходят к адресату в том же порядке, в котором вы их отправили. Когда вы бросаете открытку в ящик, нет гарантии, что адресат ее получит, и в случае с пакетом тоже (это называется принципом «наилучшей попытки»). Но систему можно сделать надежной, если поверх этой технологии положить еще один слой (сейчас он называется Transmission Control Protocol, TCP). Поскольку вы не знаете, все ли открытки дошли, вы просите получателя прислать вам ответную открытку (пакет) со списком всех открыток, которые он получил, и пересылаете копии недостающих еще раз. Но и ответная открытка со списком может потеряться - в этом случае вы просто начинаете слать все открытки заново. И так до тех пор, пока не получите подтверждения. Ну вот, теперь вы знаете, как работает интернет.

Да, все просто. Но в этом принципе есть два очень важных критических решения, которые и предопределили популярность сети Интернет.

Первое - это то, что почтовая открытка (и пакет) не знает, какую информацию она несет: это могут быть текст, картинка, числа, уравнения, все что угодно. Все, что открытка (и пакет) знает, - это то, что она должна быть доставлена из точки А в точку Б. Следовательно, так можно передавать через сеть любую информацию, если договориться о том, что она означает. То есть для нового приложения не нужно менять архитектуру сети. Не нужно получать разрешение от каждого провайдера, который передает ваш трафик. Можно посылать и передавать данные из любого места сети.

Второй важный момент: почтовая открытка (и пакет) не знает, как ее доставляют к адресату - пешком, на лошади, поездом, судном, автомобилем или самолетом. Все, что имеет при этом значение, - что ее (возможно) доведут до адресата. За последние 40 лет технологии связи неоднократно менялись, но каждый раз удавалось организовать передачу пакетов поверх новой технологии, будь то телефонные провода, оптоволоконные кабели, мобильные радиоканалы, спутниковая или радиорелейная связь.

Вообще появление сети Интернет - это типичный «черный лебедь», то есть концепция, описывающая очень редкое, маловероятное событие, которое оказывает огромное влияние на дальнейший ход вещей. При этом ни само событие, ни его последствия не поддаются прогнозированию. Сам термин пришел из латыни, из цитаты древнеримского поэта Ювенала: *Rara avis in terris nigroque simillima*



сугно -«Хороший человек так же редок, как черный лебедь» (до XVII века черные лебеди не были известны в Европе).

Что ждёт Интернет в будущем?

Где-то в районе 2009 года произошло знаменательное событие: количество подключенных к интернету устройств превысило количество пользователей (людей). Эта дата считается рождением новой концепции - «Интернета вещей», то есть сети, где вещей больше, чем людей. В ближайшем будущем эта тенденция даже усилится - к 2020 году к сети будет подключено 50 млрд устройств. Причем не только компьютеров, планшетов и смартфонов, но и совершенно бытовых вещей - фоторамок, микроволновок, стиральных машин, а также самых разных автоматических датчиков, образующих сенсорные сети, которые будут превращать вашу комнату, здание и даже город в умный и единый организм. Общаясь через сеть между собой и управляющими компьютерами, такие устройства смогут, например, сгладить пик энергопотребления - просто включая технику не одновременно, а по очереди. И это уже не фантастика, а вполне реальные коммерческие устройства. Вообще фантазия разработчиков, оснащающих устройства сетевыми интерфейсами, похоже, не имеет пределов. Лет десять назад я шутил, что скоро даже лампочки будут иметь возможность подключения к интернету. Эта шутка перестала быть шуткой пару лет назад - сейчас такую лампочку с беспроводным подключением к сети можно купить в магазине примерно за \$20. Поэтому делать прогнозы даже на десять лет чрезвычайно сложно,

Понятие гипертекста как способа организации данных.

Язык гипертекстовой разметки HTML (HyperText Markup Language) был предложен Тимом Бернерсом-Ли в 1989 году в качестве одного из компонентов технологии разработки распределенной гипертекстовой системы World Wide Web. Когда Т. Бернерс-Ли предложил свою систему, в мире информационных технологий наблюдался повышенный интерес к новому и модному в то время направлению-гипертекстовым системам. Сама идея, но не термин, была введена В. Бушем в 1945 году в предложениях по созданию электромеханической информационной системы Memex. Несмотря на то, что Буш был советником по науке президента Рузвельта, идея не была реализована. В 1965 году Т. Нельсон ввел в обращение сам термин "гипертекст", развил и даже реализовал некоторые идеи, связанные с работой с "нелинейными" текстами. В 1968 году изобретатель манипулятора "мышь" Д.Енжильбард продемонстрировал работу с системой, имеющей типичный гипертекстовый интерфейс, и, что интересно, проведена эта демонстрация

была с использованием системы телекоммуникаций. Однако внятно описать свою систему он не смог. В 1975 году идея гипертекста нашла воплощение в информационной системе внутреннего распорядка атомного авианосца "Карл Винстон". Идея гипертекстовой информационной системы состоит в том, что пользователь имеет возможность просматривать документы (страницы текста) в том порядке, в котором ему это больше нравится, а не последовательно, как это принято при чтении книг. Поэтому Т.Нельсон и определил гипертекст как нелинейный текст. Достигается это путем создания специального механизма связи различных страниц текста при помощи гипертекстовых ссылок, т.е. у обычного текста есть ссылки типа "следующий-предыдущий", а у гипертекста можно построить еще сколь угодно много других ссылок. Любимыми примерами специалистов по гипертексту являются энциклопедии, Библия, системы типа "Help". Простой, на первый взгляд, механизм построения ссылок оказывается довольно сложной задачей, т.к. можно построить статические ссылки, динамические ссылки, ассоциированные с документом в целом или только с отдельными его частями, т.е. контекстные ссылки. Дальнейшее развитие этого подхода приводит к расширению понятия гипертекста за счет других информационных ресурсов, включая графику, аудио- и видеоинформацию, до понятия гипермедиа. Тем, кто интересуется более подробно различными схемами и способами разработки гипертекстовых систем, стоит обратиться к специальной литературе. Разработчики HTML должны были решить две задачи:

- дать дизайнерам гипертекстовых баз данных простое средство создания документов;
- сделать это средство достаточно мощным, чтобы отразить имевшиеся на тот момент представления об интерфейсе пользователя гипертекстовых баз данных.

Первая задача была решена за счет выбора теговой¹ модели описания документа. Такая модель широко применяется в системах подготовки документов для печати. Примером такой системы является хорошо известный язык разметки научных документов TeX, предложенный Американским Математическим Обществом, и программы его интерпретации. К моменту создания HTML существовал стандарт языка разметки печатных документов - Standard Generalised Markup Language, который и был взят в качестве основы HTML. Предполагалось, что такое решение поможет использовать существующее программное обеспечение для интерпретации нового языка. Однако, будучи доступным широкому кругу пользователей Internet, HTML зажил своей собственной жизнью. Вероятно, многие администраторы баз данных WWW и разработчики программного обеспечения

¹ Тег – оформленная единица HTML-кода.

для этой системы имеют довольно смутное представление о стандартном языке разметки SGML. Язык HTML позволяет определять структуру электронного документа с полиграфическим уровнем оформления; результирующий документ может содержать самые разнообразные теги: иллюстрации, аудио- и видео- фрагменты и так далее. Язык включает в свой состав развитые средства для спецификации нескольких уровней заголовков, шрифтовых выделений, различных групп объектов, например, словари, каталоги или меню для размещения иллюстраций и других фрагментов, а также множество других возможностей. Вторым важным моментом, повлиявшим на судьбу HTML, стал выбор в качестве основы гипертекстовой базы данных обычного текстового файла, который хранится средствами файловой системы операционной среды компьютера. Такой выбор был сделан под влиянием следующих факторов:

- такой файл можно создать в любом текстовом редакторе на любой аппаратной платформе в среде любой операционной системы.
- к моменту разработки HTML существовал американский стандарт для разработки сетевых информационных систем - Z39.50, в котором в качестве единицы хранения указывался простой текстовый файл в кодировке LATIN1, что соответствует US ASCII.

Таким образом, гипертекстовая база данных в концепции WWW - это набор текстовых файлов, размеченных на языке HTML, который определяет форму представления информации (разметка) и структуру связей этих файлов (гипертекстовые ссылки). Такой подход предполагает наличие еще одной компоненты технологии - интерпретатора языка. В World Wide Web функции интерпретатора разделены между сервером гипертекстовой базы данных и интерфейсом пользователя. Сервер, кроме доступа к документам и обработки гипертекстовых ссылок, осуществляет также препроцессорную обработку документов, в то время как интерфейс пользователя осуществляет интерпретацию конструкций языка, связанных с представлением информации. Если первая версия языка (HTML 1.0) была направлена на представление языка как такового, где описание его возможностей носило скорее рекомендательный характер, вторая версия языка (HTML 2.0) фиксировала практику использования конструкций языка, версия ++ (HTML++) представляла новые возможности, расширяя набор тегов HTML в сторону отображения научной информации и таблиц, а также улучшения стиля компоновки изображений и текста, то версия 3.2 призвана упорядочить все нововведения и согласовать их с существующей практикой. Кроме этого, в версии 3.2 снова делается попытка формализации интерфейса пользователя гипертекстовой распределенной системы. По сравнению с версией 2.0, HTML 3.2 позволяет реализовать отображение таблиц (контейнер), выполнение мобильных кодов, обтекание графики текстом, а также отображение верхних и нижних индексов. Кроме этих возможно-

стей, которые фиксируют текущую практику использования HTML, современные программы просмотра HTML - документов позволяют реализовать и ряд других возможностей разметки текста, которые описаны в стандарте HTML 3.0 и расширениях HTML фирм-разработчиков программного обеспечения: * разметка математических формул (HTML 3.0); * дополнительные контейнеры заголовка (HTML 3.0; Netscape Extensions; Microsoft Extensions); * дополнительные атрибуты стандартных контейнеров тела документа (ALIGN; BGCOLOR; TARGET и т.п.); * разбиение страницы на фреймы; * открытие дополнительных окон и др. Сейчас World Wide Web Consortium (W3C) уже опубликовал рабочие материалы спецификации HTML 4.0.. Кроме возможностей разметки текста, включения мультимедиа и формирования гипертекстовых связей уже существовавших в предыдущих версиях HTML, в версию 4.0 включены дополнительные средства работы с мультимедиа, языки программирования, таблицы стилей, упрощенные средства печати изображений и документов, которые становятся более доступными для всех пользователей HTML 4.0. Эти дополнения служат интернационализации WWW и распространению ее по всему миру. Кроме этого, для управления сценариями просмотра страниц Website (гипертекстовой базы данных, выполненной в технологии World Wide Web) можно использовать языки программирования этих сценариев типа JavaScript, Java и VBScript.

Гипертекст - текст, устроенный таким образом, что он превращается в систему, иерархию текстов, одновременно составляя единство и множество текстов.

Простейший пример гипертекста - это любой словарь или энциклопедия, где каждая статья имеет отсылки к другим статьям этого же словаря. В результате читать такой текст можно по-разному: от одной статьи к другой, по мере надобности, игнорируя гипертекстовые отсылки; читать статьи подряд, справляясь с отсылками; наконец, пуститься в гипертекстовое плавание, то есть от одной отсылки переходить к другой.

Гипертекст - это нелинейный лабиринт, своеобразная картина мира.

Гипертекст - способ организации текста (и вид текста), появившийся с внедрением в жизнь компьютерных технологий обучения. Дает возможность читателю, работающему с одним текстом, мгновенно получить на экране другой (чаще всего, поясняющий, раскрывающий смысл некоторого понятия глубже, чем текст первоначальный). А затем вернуться обратно и продолжить чтение основного текста. Глубина "вложенности" текстов формально не ограничивается. Связь текстов между собой организуется с помощью гиперссылок (линков). Ссылки могут быть и перекрестными.

Гипертекст является системой представления информации, которая создает семантическую сеть сложной топологии, позволяющую организовывать информацию на разных уровнях обобщения.

В литературоведении гипертекст — это такая форма организации текстового материала, при которой его единицы представлены не в линейной последовательности, а как система явно указанных возможных переходов, связей между ними. Следуя этим связям, можно читать материал в любом порядке, образуя разные линейные тексты.

Гипертекстовость (гиперлитература) — новое свойство/характеристика литературного прозаического произведения, для которого характерны черты гипертекста (внутренние корреляционные² ссылки, отсутствие линейного повествования). Используется для создания эффекта игры, свойственного постмодернистской литературе: количество значений изначального текста расширяется, благодаря читательскому формированию сюжетной линии.

Одно из ярчайших произведений, которое характеризуется гипертекстовостью — роман-лексикон М. Павича «Хазарский словарь». Следует также отметить наличие электронных гиперлитературных произведений (М. Джойс, «Полдень»)

Простейший пример гипертекста — любой словарь или энциклопедия, где каждая статья имеет отсылки и сноски к другим статьям.

Введение в технологию создания веб-документов средствами HTML.

HTML (HyperText Markup Language, язык гипертекстовой разметки) - специальные инструкции браузеру, с помощью которых создаются Веб-страницы.

Web-страницы - это документы в формате HTML, содержащие текст и специальные тэги (дескрипторы) HTML. По большому счету тэги HTML необходимы для форматирования текста (т.е. придания ему нужного вида), который "понимает" браузер. Документы HTML хранятся в виде файлов с расширением .htm или .html.

Тэги HTML сообщают браузеру информацию о структуре и особенностях форматирования Веб-страницы. Каждый тэг содержит определенную инструкцию и заключается в угловые скобки <>. Большинство тэгов состоят из открывающей и закрывающей частей и воздействуют на текст, заключенный внутри.

HTML – это язык разметки документов в среде WEB. То, что вы видите при просмотре страницы в Internet, это интерпретация вашим браузером HTML-текста. Чтобы браузер правильно отображал форматирование к примеру текста т.е. разделял его на абзацы, выделял цитаты, заголовки, списки и.т.д. ему надо как-то сообщить, что мол это заголовок, а это – параграф и.т.д. Этим как раз и занимается язык html. Чтобы увидеть HTML-коды страницы в Internet, кликните

² Взаимная связь, соотношение.

правой кнопкой мыши по странице, в выпавшем меню выберите пункт - view source (или "просмотр HTML кода").

С помощью чего создаются Web-страницы

Что такое Веб-страница? По сути дела - это простой текстовый документ, содержащий тэги (которые в свою очередь являются обычным текстом, заключенным в скобки). Т.е., исходя из вышеизложенного, Веб-страницы можно набрать в любом текстовом редакторе (Блокнот, WordPad, Word и т.д.). В этом случае абсолютно все надо набирать своими ручками. Основной плюс такого метода - код получается абсолютно минимизированным, т.е. "вес" такой html-страницы будет минимальным. К недостаткам следует отнести тот факт, что пользователь должен очень хорошо знать язык html-разметки, да и результаты своего труда нельзя будет сразу посмотреть. Поэтому таким методом создания Веб-страниц практически никто не пользуется. Посоветовать его можно разве что для создания простеньких Веб-страничек если нет возможности использования специализированных программ.

Практически все Веб-дизайнеры используют специальные программы, которые называются **HTML-редакторы**. Наиболее популярные – Front Page, Macromedia HomeSite, Dreamweaver MX,.

Чем же они хороши? Прежде всего, тем, что они как раз и предназначены для создания Веб-страниц. Синтаксис в таких программах обычно выделяется разными цветами, так что очень хорошо видно где находится текст, а где дескрипторы³, а где php-код. Многие сложные конструкции тэгов можно вводить одним щелчком мышки. Результат своего труда виден тут же, в соседнем окошке. Если вы серьезно занимаетесь веб-дизайном, то без такой программы вам просто не обойтись.

Кроме вышеуказанных программ существует еще один тип специализированных программ - это, так называемые, средства визуального проектирования (Microsoft FrontPage, HoTMetal Pro). Они позволяют создавать Веб-страницы таким образом, чтобы полностью исключить необходимость набора тэгов HTML вручную. Подобные программы выполняют все необходимые операции автоматически, по мере "рисования" Веб-страницы. К недостаткам таких программ следует отнести тот факт, что "вес" таких страниц оказывается большим из-за избыточного *HTML-кода*, да и изящную страничку без подправки кода "вручную" сделать весьма проблематично.

³ **ДЕСКРИПТОР** (лат . descriptor - описывающий), лексическая единица (слово, словосочетание) информационно-поискового языка, выражающая основное смысловое содержание какого-либо текста. Используется при информационном поиске документов в информационно-поисковых системах.

Браузер и Блокнот (или аналогичный простой редактор) - вот всё, что вам необходимо для создания веб-странички средствами HTML. Почему простой текстовый редактор идеально подходит для изучения HTML и CSS? Потому, что он не изменяет вводимый вами код.

ТЕГ

Единицей HTML-кода является тег. Например, <html >, <body>, <h1>, <h2> и так далее. Все тэги имеют одинаковый формат: они начинаются знаком "<" и заканчиваются знаком ">". Обычно имеются два тэга – открывающий и закрывающий. Различие в том, что в закрывающем имеется слэш "/". Желательно вводить все теги в нижнем регистре, хотя не обязательно.

Заголовки, текст, списки. Вставка изображений и ссылок.

Откроем блокнот и запишем следующие теги:

```
<h1> Самый крупный заголовок Н1 </h1>  
<h2> Заголовок чуть поменьше Н2 </h2>  
<h6> Самый мелкий заголовок Н6 </h6>  
<h0> Заголовок Н0 </h0>
```

Создадим папку **My Site**. Сохраним текстовый файл **index** с расширением **HTML**.

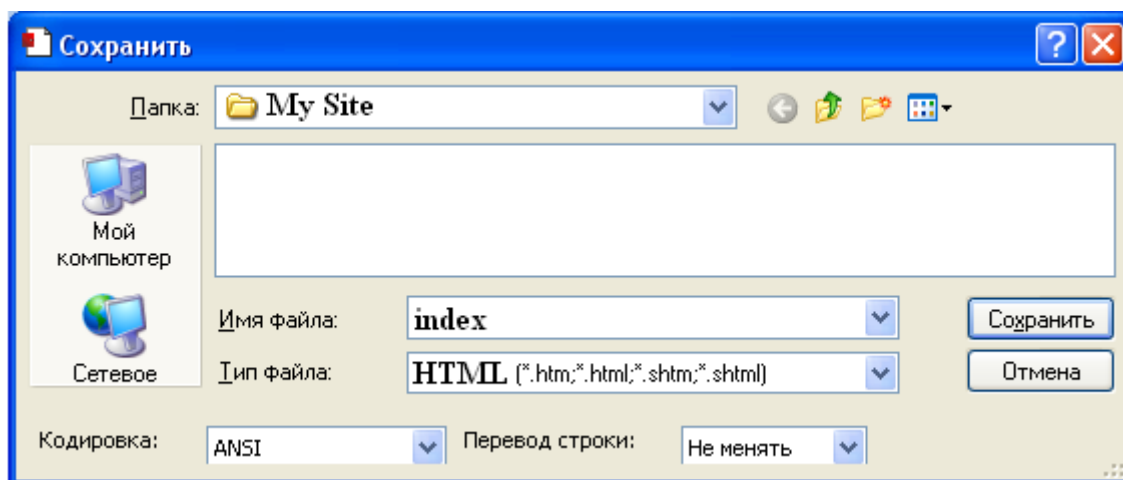


Рис. 1. Сохранение файла в текстовом редакторе Bred

Откроем файл в проводнике интернет.

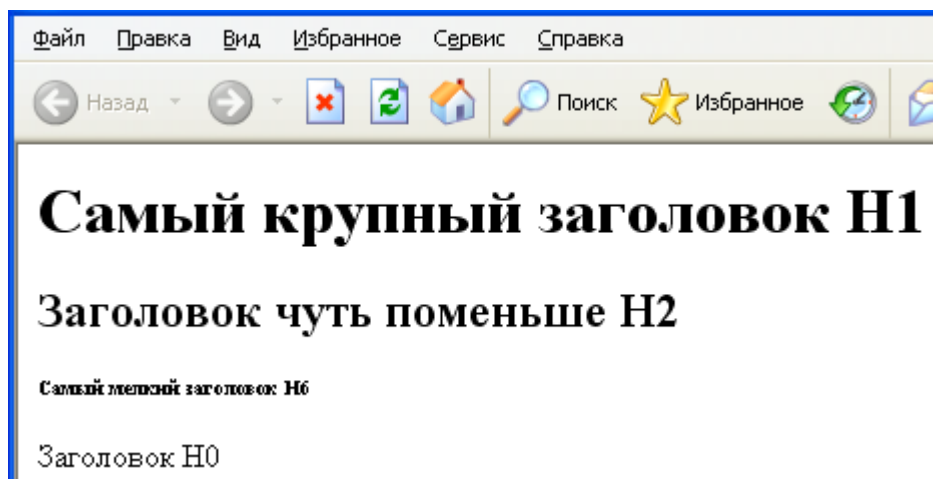


Рис. 2. Вид сформированной страницы в браузере

Всего существует семь уровней заголовков (0 – 6): тег <h1> означает заголовок первого уровня, а тег <h2> - второго уровня. Вначале тег открывается, затем идет его содержимое, потом он закрывается.

Атрибут

Атрибут тега – это дополнительная информация. Атрибуты всегда записываются внутри тега, затем следует знак равенства и детали атрибута, заключённые в двойные кавычки. Введем атрибут выравнивания по центру для заголовка второго уровня.

`<h2 align="center"> Заголовок чуть меньше </h2>`
 Внесенные изменения изменят вид текста на странице (рис.3).

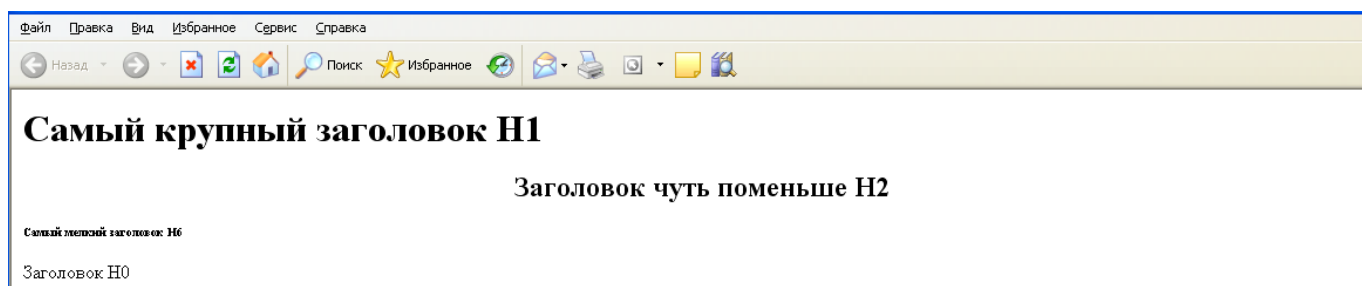


Рис. 3. Атрибут выравнивания по центру

Элемент – понятие, введенное чисто для удобства объяснения. Например, элемент HEAD состоит из двух тегов – открывающего <HEAD> и закрывающего </HEAD>. Таким образом, элемент – более общее понятие, обозначающее пару тегов и участок документа между тегами, на который распространяется их влияние.

Понятие и структура веб-документов.

Дизайн и свойства страницы.

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, должен начинаться со строки объявления версии HTML `<!DOCTYPE...>`, которая обычно выглядит примерно так:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

В данном случае мы говорим браузеру, что HTML соответствует международной спецификации версии 4.01. Не пытайтесь запомнить эту строчку наизусть, главное - вы должны знать, что она необходима. Ведь именно она помогает сделать сайт, который будет одинаково смотреться во всех браузерах (обычно это три основных Internet Explorer, Opera, Firefox).

Если страничка изготовлена с помощью Dreamweaver CS4, это строка кода будет выглядеть следующим образом:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

Очень полезным при создании сайтов является вставка комментария, потому, что через месяц после создания, вам будет трудно разобратся что к чему, ведь кода будет очень много. А если вы будете комментировать свои действия - тогда разобратся будет значительно легче! Пример тега комментариев:

```
<!--определим кодировку страницы документа -->
```

Комментарии нужны только создающему или редактирующему сайт, при просмотре документа через браузер его видно не будет. Комментарий открывают тегом <!-- затем пишут текст комментария и закрывают тегом -->⁴.

Вероятно вам приходилось открывать странички, в которых текст изображался непонятными знаками несуществующего языка. Для того, чтобы текст, написанный вами на кириллице выглядел точно так, нужно указать браузеру, какая кодировка используется в тексте страницы. В нашем случае это **windows-1251**, а тег выглядит таким образом:

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
```

Далее обозначается начало и конец документа тегам <html> и </html> соответственно. Внутри этих тегов должны находиться теги головы (<head></head>) и тела документа(<body></body>).

⁴ **Примечание:** тег <!-- --> внутри элемента TITLE не действует.

Обычно основой головы документа является элемент TITLE - заголовок документа. Также там содержится вся техническая информация, различные таблицы стилей и т.п. А основное содержимое: текст, таблицы, картинки, - находится в теле документа.



Рис. 4. Структура HTML документа

Как вы видите голова находится над телом, поэтому никогда не размещайте голову документа в теле документа (или наоборот). Сначала закрываем голову документа </HEAD>, и лишь затем открываем тело <BODY>. И еще, у документа одна голова и одно тело, и не стоит пытаться создавать их большее количество.

С учетом сказанного текст нашей странички выглядит так:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transi-
tional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!--определим кодировку страницы документа -->
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251" />
  <html>
    <head>
      <title> Главная </title>
      <!--Это название моей ПЕРВОЙ страницы-->
    </head>
    <body>
      <h1> Самый крупный заголовок H1 </h1>
      <h2 align="center"> Заголовок чуть поменьше H2 </h2>

```

```

<h6> Самый мелкий заголовок H6 </h6>
<h0> Заголовок H0 </h0>
</body>
</html>

```

Сохраним файл и посмотрим в браузере. По сравнению с предыдущим просмотром изменился только заголовок окна документа браузера.

Чтобы внести изменения в страничку, нужно открыть данный файл через блокнот. В браузере Opera можно менять текст странички непосредственно в браузере. Для этого можете просто запустить свою страничку, в контекстном меню выбрать "Исходный Код". После внесения изменений нажать на кнопку "Применить изменения".

Работа с текстом

Основными элементами форматирования текста в html являются:

- P Используется для разбивания текста на параграфы
- H1,H2,...H6 Применяются для создания заголовков 1,2...6 уровней
- BR Используется для переноса строки
- DIV , SPAN Используются для выделения части документа определенным способом.

Атрибуты:

ALIGN - определяет способ горизонтального выравнивания параграфа . Возможные значения: **left, center, right, justify**⁵. Это соответственно по левому краю, по центру, по правому краю, и по ширине. По умолчанию имеет значение **left**.

Примечание: следите чтобы содержимое в кавычках было написано без пробелов, т.е. `<palign="right">` а не `<p align=" right ">` иначе не будет работать!

BR - данный элемент осуществляет перенос строки. Не имеет закрывающего тега.

DIV - используется как контейнер для блоков html кода страницы, которым легко динамически манипулировать – перемещать, регулировать отступы, скрывать и т.п. Обязателен закрывающий тег!

Атрибуты:

⁵ Выравнивание по ширине не применяется для заголовков

ALIGN - определяет способ горизонтального выравнивания контейнера. Возможные значения: **left**, **center**, **right**, **justify**. Это соответственно по левому краю, по центру, по правому краю, и по ширине. По умолчанию имеет значение **left**.

Допустим нам нужно выровнять первую и третью строчки текста из предыдущего примера по правому краю, не выделяя при этом их в абзац. Как это сделать?

```
<body>
  <div align="right">
    <h1> Самый крупный заголовок H1 </h1>
    <h2 align="center"> Заголовок чуть поменьше H2 </h2>
    <h6> Самый мелкий заголовок H6 </h6>
  </div>
  <h0> Заголовок H0 </h0>
</body>
```

Что мы увидим:

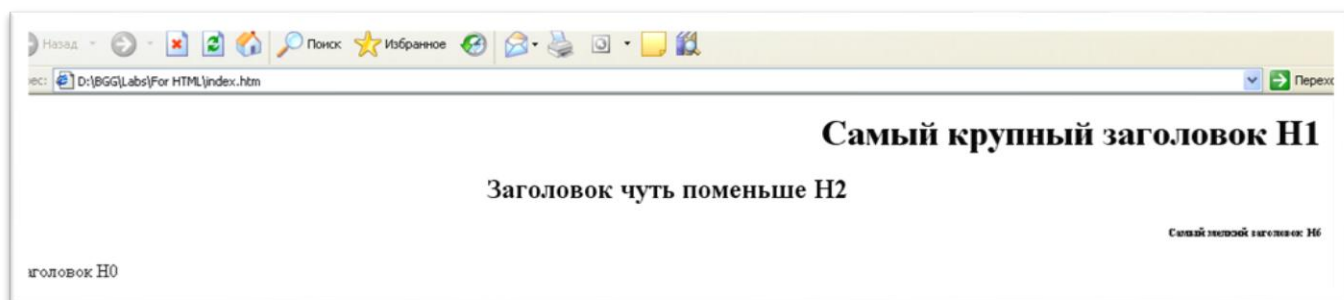


Рис. 5. Использование тега DIV

Так же само, можно заключить в контейнер и 20 и 50 и более строчек и манипулировать ими.

Примечание: Находящиеся между начальным и конечным тегам текст или HTML-элементы выделяются как бы в отдельный параграф (но отступы гораздо меньше).

SPAN - Используется для выделения части информации и придания ей различных стилей. Закрывающий тег обязателен! Сам по себе, без использования стилей, элемент SPAN никакого смысла не имеет!

Например, чтобы выделить фразу «Самый крупный» красным цветом, заключим эту фразу в тег SPAN: ` Самый крупный `

В результате увидим:

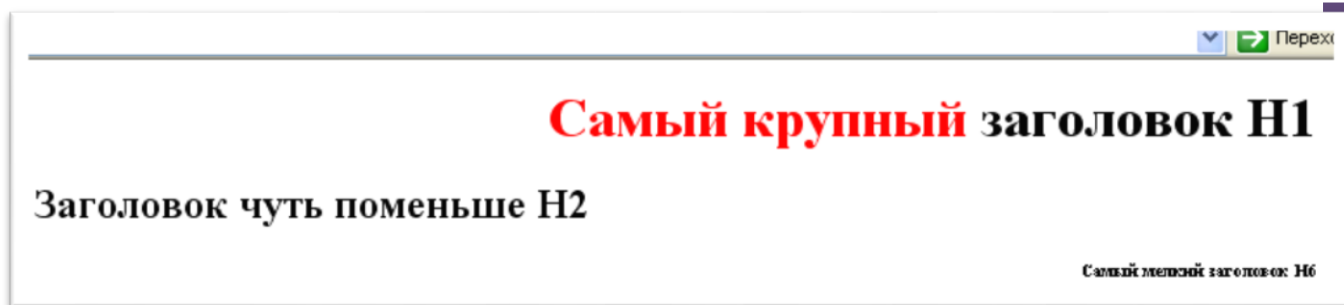


Рис. 6. Примерение тега SPAN

Таким образом с помощью элемента **SPAN** можно хоть каждой букве в тексте присвоить разный стиль (цвет, размер, и.т.д.)

Чтобы придать тексту ту или иную гарнитуру в html используются такие элементы:

- STRONG	Используется для выделения текста жирным
- EM	Используется для выделения текста курсивом
- U	Выделение текста подчеркиванием
- S	Перечеркивание текста
- SUP	Создание верхнего индекса
- SUB	Создание нижнего индекса
- FONT	Изменение цвета, типа, размера шрифта
- HR	Вставляет в текст горизонтальную разделительную линию

STRONG - Выделяет текст, заключенный между открывающим и закрывающим тегами, жирным шрифтом. Раньше везде использовался тег **<BOLD>** (или ****), в принципе его можно использовать и сейчас, но это не приветствуется поисковыми системами.

FONT - Позволяет изменять цвет, размер и тип шрифта текста, находящегося между открывающим **** и закрывающим **** тегами.

Атрибуты:

SIZE - Определяет размер шрифта. Возможные значение - 1 , 2 , 3 , 4 , 5 , 6 , 7.

COLOR - Определяет цвет текста. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

FACE - определяет используемый шрифт. Используйте Times New Roman, Arial, Tahoma, Courier, Courier New.

Если в теле документа написать:

`<body>`

```

    <div align="right">
    <h1> <span style="color:red;" > Самый крупный </span>
заголовок H1 </h1>
    <h2 align="center"> Заголовок чуть поменьше H2 </h2>
    <h6> Самый мелкий заголовок H6 </h6>
    </div>
    <h0> Заголовок H0 </h0>
    <!--Отделим следующий текст горизонтальной линией-->
    <hr width="50%" color="#AA0FHH" size ="5">
    <p>
    <strong> Этот текст жирный</strong><P>
    <EM> А это курсивный текст</EM><P>
    <U>Это пример подчеркивания</U><P>
    <S>Это пример перечеркнутого текста</S><BR>
    <SUP>Пример надстрочного (верхний индекс) тек-
ста</SUP>Обычный текст<BR>
    <SUB>Пример подстрочных (нижний ин-
декс) символов</SUB>Обычный текст<BR>
    <FONT SIZE="3" FACE="Courier New"
COLOR="Violet">Моноширинный (Courier) фиолетовый текст 3
размера</FONT><BR>
    <FONT SIZE="+5" FACE="Tahoma" COLOR="#0000FF">Tahoma,
синий, увеличенный на 5 пт по сравнению со стандартным (3
размер)</FONT>
    </body>

```

Получим следующий результат:

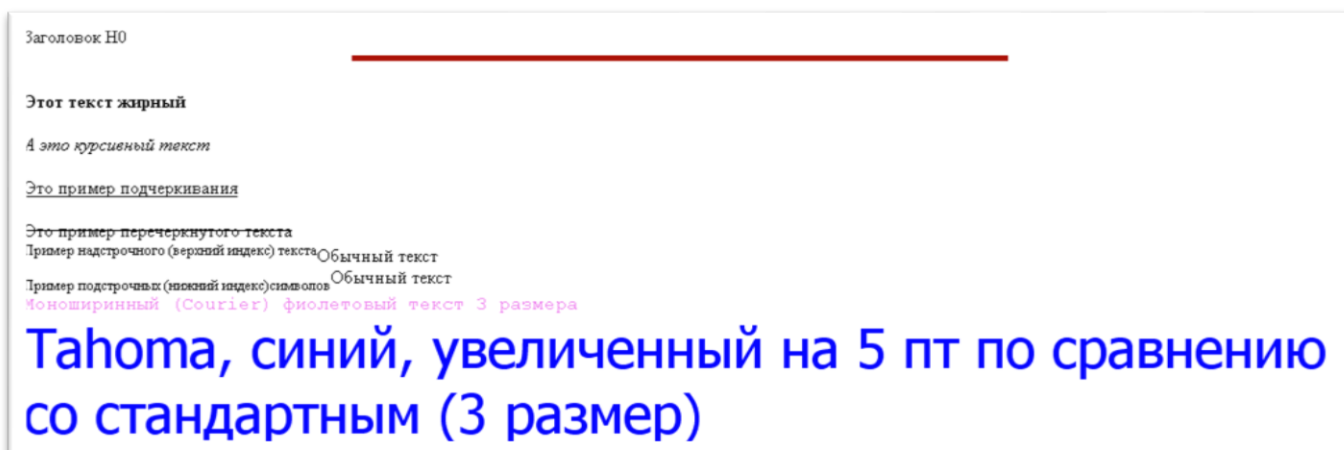


Рис. 7. Изменение начертания шрифта

Если написано **SIZE="+2"** то это означает увеличить на 2 единицы шрифт по сравнению со стандартным. Обычно стандартный размер шрифта равен 3 .

HR- служит для вставки в текст горизонтальной линии. **Закрывающего тега нет!**

Атрибуты:

WIDTH – определяет длину линии в пикселах или процентах от ширины окна браузера.

SIZE – толщина линии в пикселах.

ALIGN – определяет выравнивание горизонтальной линии. Атрибут может принимать следующие значения:

left – выравнивание по левому краю документа.

right – выравнивание по правому краю документа.

center – выравнивание по центру документа (используется по умолчанию).

NOSHADE – определяет способ закрашки линии как сплошной. Атрибут является флагом и не требует указания значения. Без данного атрибута линия отображается объемной.

COLOR – задает цвет линии. Можно использовать либо RGB-значение в шестнадцатиричной системе, либо один из 16 базовых цветов. Атрибут работает только в Internet Explorer.

Списки

Различают списки двух видов: упорядоченные (пронумерованные) и неупорядоченные (непронумерованные). Единственным отличием является то, что перед пунктами неупорядоченных списков обычно ставятся символы-буллеты (bullets), например, точки, ромбики и т.п., в то время как пунктам упорядоченных списков предшествуют их номера.

Элементы:

- UL	Создает маркированный список
- OL	Создает нумерованный список
- LI	Создает пункты списка внутри элементов OL или UL

UL (Unsorted List) - Создает маркированный список. Обязательно наличие закрывающего тега, причем между начальным и конечным тегами должны присутствовать один или несколько элементов **LI**, обозначающих отдельные пункты списка.

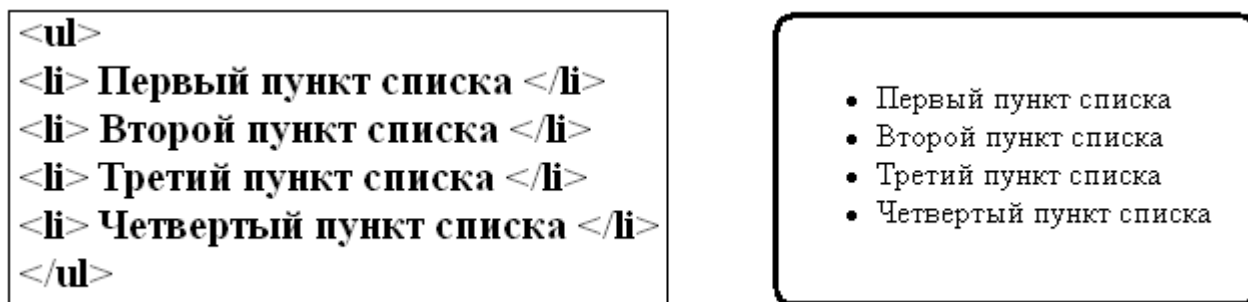


Рис. 8. Программный код маркированного списка (слева) и результат в браузере (справа)

OL (Ordered List)- Создает упорядоченный (нумерованный) список. Между начальным и конечным тегами должны присутствовать один или несколько элементов LI, обозначающих отдельные пункты списка.

Атрибуты:

START – определяет первое число, с которого начинается нумерация пунктов. (только целые числа). Если не указывать, начинается с начала.

TYPE – определяет стиль нумерации пунктов списка.

Возможные значения:

"A"	–	заглавные	буквы	A,	B,	C	...	
"a"	–	строчные	буквы	a,	b,	c	...	
"I"	–	большие	римские	числа	I,	II,	III	...
"i"	–	маленькие	римские	числа	i,	ii,	iii	...
"1"	– арабские числа 1, 2, 3 ...							

Значение по умолчанию **<OL TYPE="1">**.

Приведу пример:

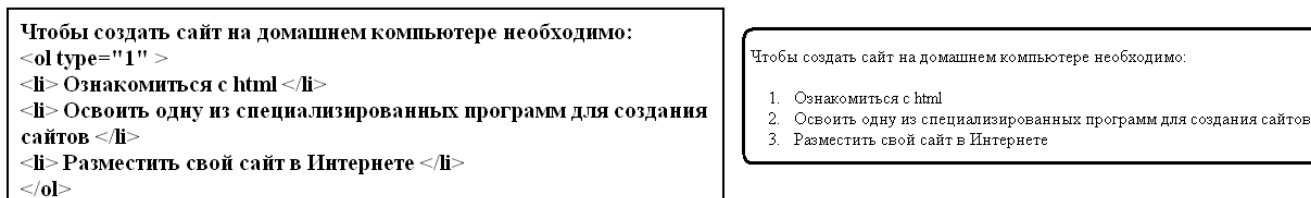


Рис. 9. Программный код нумерованного списка (слева) и результат в браузере (справа)

LI (List Item) - Создает пункт в списке. Располагается внутри элементов **OL** или **UL**. Закрывающий тег писать желательно, но не обязательно.

Атрибуты:

VALUE – изменяет порядок нумерации элементов списка. Используется только если элемент **LI** находится внутри элемента **OL**. В качестве значения указывается порядковый номер элемента. Например, приведенный ниже код позволяет присвоить данному элементу списка номер 5 независимо от того, какой порядковый номер списка был перед ним.

```
<li value="5"> Неплохо было бы освоить CSS </li>
```

Работа с изображениями

Есть три типа файлов изображений, которые можно вставить на ваши страницы:

- **GIF** (Graphics Interchange Format)
- **JPG / JPEG** (Joint Photographic Experts Group)
- **PNG** (Portable Network Graphics)

GIF - использует всего 256 цветов и соответственно лучше подходит для рисунков с малым кол-вом оттенков. Этот формат поддерживает прозрачность изображений. JPEG - формат изображений, который использует до миллиона цветов. Обычно используется для фотографий и качественной графики (с огромным количеством оттенков). PNG - сравнительно новый формат. По многим параметрам превосходит JPEG и GIF: миллионы цветов и эффективное сжатие. Также поддерживает прозрачность.

В любом случае нужно определить размеры изображения на странице сайта и выполнить необходимые изменения размеров и разрешения изображения с помощью графического редактора. В противном случае большие размеры файла изображения приведут к увеличению времени загрузки файла. Но не всегда ожидание приятно...

Воспользуемся элементом **IMG**, не имеющим закрывающего тега и атрибутом **SRC**⁶, который указывает, где находится изображение. Вы легко можете вставлять изображения, размещенные в других папках или на других сайтах, если сайт создается на вашем компьютере. Для размещения сайте во всемирной сети помещайте все изображения в папку, созданную в папке вашего сайта. Такой папкой у нас будет папка **images**, расположенная в папке **My Site**.

Вставить изображение на страницу очень просто. Вот пример если оно лежит в той же папке что и страница.

```

```



В результате мы увидим:

⁶ source - положение

Атрибуты

ALIGN - определяет способ выравнивания картинки по горизонтали. Очень полезное свойство при обтекании картинки текстом. Обычно используют **LEFT** (выравнивание по левому краю, текст будет обтекать справа) и **RIGHT** (выравнивание по правому краю, текст обтекает слева) .Если на странице есть текст, то это обязательное свойство.

HSPACE и **VSPACE** - отступы в пикселях по горизонтали и по вертикали от картинки до других объектов документа. Легко запомнить название если взять и просто перевести с английского. **HSPACE** - Horizontal Space - горизонтальный отступ и **VSPACE** - Vertical Space - вертикальный отступ.

HEIGHT и **WIDTH** - высота и ширина изображения в пикселях. Золотое правило web-мастера – всегда явно задавать размеры картинки в атрибутах **HEIGHT** и **WIDTH**, резервируя тем самым место в окне браузера еще до загрузки изображения. В противном случае документ при загрузке каждой картинки будет заново перерисовываться.

Рассмотрим пример:

```
<!-- первый пример с отступами и выравниванием по левому краю-->
```

```
<p align="justify"> 
```

ALIGN - определяет способ выравнивания картинки по горизонтали. **
Очень полезное свойство при обтекании картинки текстом.
** Обычно используют **LEFT** (выравнивание по левому краю, текст будет**
** обтекать справа) и **RIGHT** (выравнивание по правому краю,**
** текст обтекает слева. Если на странице есть текст, **
то это обязательное свойство.<P>**

```
<!-- второй пример с отступами и выравниванием по правому краю-->
```

```
<p align="justify"> 
```

HSPACE и **VSPACE** - отступы в пикселях по горизонтали и по вертикали от картинки до других объектов документа. Легко запомнить название если взять и просто перевести с английского. **HSPACE** - Horizontal Space - горизонтальный отступ и **VSPACE** - Vertical Space - вертикальный отступ.**<P>**

<!--третий пример без отступов, с выравниванием по левому краю-->

```
<p align="justify"> 
```

HEIGHT и WIDTH - высота и ширина изображения в пикселах. Золотое правило web-мастера - всегда явно задавать размеры картинки в атрибутах HEIGHT и WIDTH, резервируя тем самым место в окне браузера еще до загрузки изображения. В противном случае документ при загрузке каждой картинки будет заново перерисовываться.

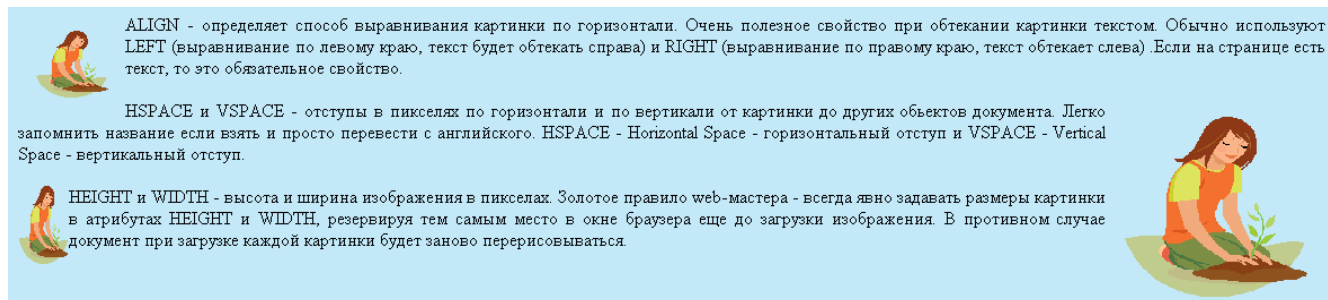


Рис. 10. Результат вставки рисунков в браузере

ALT - определяет текст, отображаемый браузером на месте изображения, если браузер не может найти файл с изображением или включен в текстовый режим. В качестве значения задается текст с описанием изображения.

```

```

В данном примере не указана папка с рисунками, поэтому браузер не может его найти и выводит на экран прямоугольник с надписью, указанной в атрибуте ALT.

TITLE - Определяет заголовок картинки, т.е. при наведении на картинку высветится надпись с текстом из атрибута title . Ну вот например:

```

```

Цвет фона и текста

Мы уже знаем как менять цвет текста, но для этого нам нужно было заключать его в теги **font** , а это не всегда удобно. Иногда, лучше задать цвет текста для всего документа. Также, можно задать и фоновое изображение.

Атрибуты

BACKGROUND – определяет изображение для "заливки" фона. Значение задается в виде полного URL или имени файла с картинкой в формате GIF или JPG.

BGCOLOR – определяет цвет фона документа.

TEXT – определяет цвет текста в документе.

Все они прописываются для элемента BODY. Значения цветов задаются либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

Примеры 1:

```
<!-- задаем фоновый цвет и цвет текста -->
<body bgcolor="#FFF8D2" text="red">
  <p> Этот текст будет красный, потому что мы изменили
цвет текста в теге БОДИ и теперь весь текст на странице по
умолчанию будет красный </p>
  <font color = "green">
    <p> В этом абзаце текст будет зеленый, потому что мы
заклучили его в теги font и придали соответсвующий цвет
</p>
  </font>
<p> Теперь текст снова будет красный </p>
</body>
```

Пример 2:

```
<!-- задаем фоновое изображение и цвет текста -->
<body background="images/bg1.jpg" text="red">
<p> Этот текст будет красный, потому что мы изменили цвет
текста в теге БОДИ и теперь весь текст на странице по умол-
чанию будет красный </p>
  <p>Теперь тут тоже красное и только <font color
="green"> эти слова зеленые </font> </p>
  <p> Тут как вы поняли текст тоже красный</p>
</body>
```

А как можно изображение сделать ссылкой?

Для этого просто вместо текста ссылки, вставляете изображение. Вот например:

```
<a href="http://www.zvirec.com">

</a>
```

Вот результат:



Что еще за **border="0"** спросите Вы, а дело в том, что когда делаешь изображение ссылкой, вокруг него появляется некрасивая рамка(border) и чтобы ее убрать пишут дополнительный атрибут border .

HTML ссылки

Элемент **a (anchor)** является как бы якорем, т.е. текст, заключенный между открывающим **<a>** и закрывающим **** тегом , будет текстом ссылки. **Атрибут href** это сокращение от "hypertext reference/гипертекстовая ссылка", специфицирует место, на которое выполняется переход по данной ссылке - обычно это internet-адрес и/или имя файла.

```
<a href="http://www.tut.by">Это ссылка на сайт tut.by</a>
```

[Это ссылка на сайт tut.by](http://www.tut.by)

Сайт – это совокупность страниц, поэтому все они должны быть связаны между собой ссылками. Пусть у нас есть главная (Index.htm) и две дополнительные (page1.htm и page2.htm), расположенные в одной папке. Код ссылки с **index** или **page1** на **page2** будет выглядеть так:

```
<a href="page2.htm"> Для перехода на page2 щелкни здесь! </a>
```

А если надо сделать ссылку внутри страницы?

Бывают случаи когда необходимо сделать ссылку с начала страницы в конец или наоборот, или к примеру с оглавления на главы и.т.д. Для этого необходимо пометить место документа (закладка в WORD) следующей конструкцией:

```
<!-- этот способ используется при пометке заголовков -->
```

```
<h2 id="razdel1">Раздел 1</h2>
```

```
<!-- а такую метку можно поставить везде где понадобится-->
```

```
<h2> <a name="razdel1"></a> Раздел1: </h2>
```

Теперь вы можете ссылаться на помеченную область (в данном случае Раздел1) простым указанием ее имени после значка #.

Вот как будет выглядеть ссылка на раздел1:

```
<a href="#razdel1"> Ссылка на Раздел 1 </a>
```

Пример1.

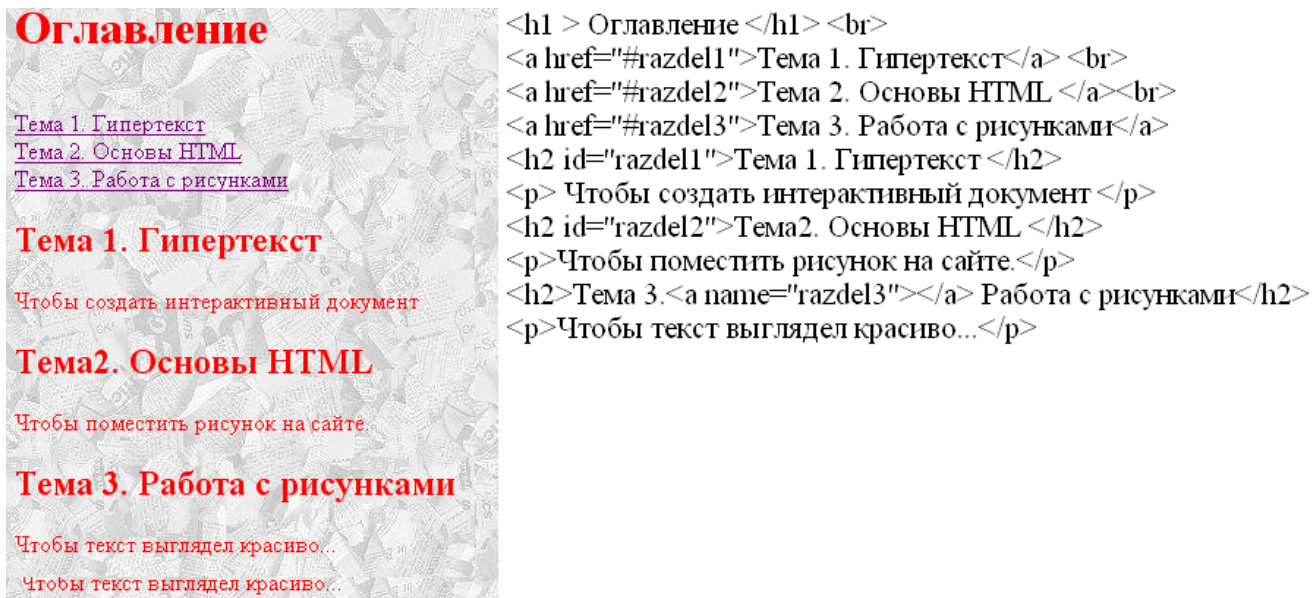


Рис. 11. Пример создание оглавления

Пример 2.

Ссылка на e-mail сайта tut.by.

```
<a href="mailto:admin@tut.by">Написать письмо на
tut.by </a>
```

В теге используется mailto и почтовый адрес получателя, разделенные двоеточием. При нажатии на ссылку запустится почтовая программа-клиент с заполненным полем имени получателя.

Дополнительные атрибуты:

TARGET - указывает где открывать страницу на которую ведет html ссылка. По умолчанию она открывается в том же окне. Если хочется чтобы открывалась в новом, следует написать **target = " blank"**.

TITLE - указывает заголовок ссылки, который появляется при наведении на нее.

Пример 3

```
<a href="http://www.bspu.unibel.by/moodle"
target=" blank">
```

Это ссылка на сайт дистанционного обучения БПУ, откроется в новом окне

```
</a> <br> <br>
```

```
<a href="http://www.bspu.unibel.by/moodle "
title="Сайт дистанционного обучения БПУ ">
```

Эта - тоже на bspu.unibel.by/moodle. При наведении появится заголовок.

```
</a>
```

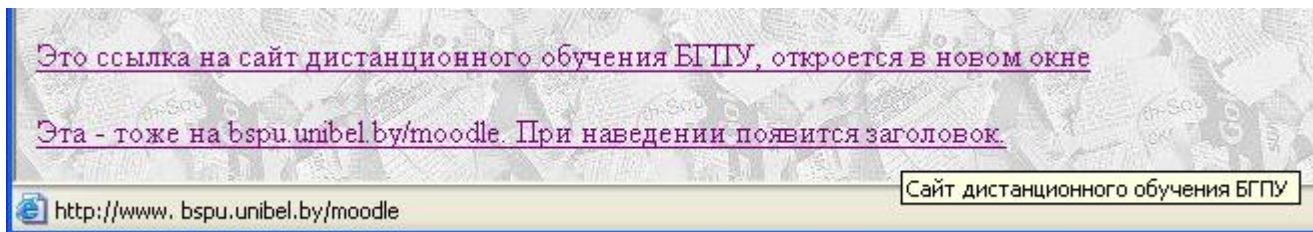


Рис. 12. Результат выполнения кода

Изменение цвета ссылок

Чтобы изменять цвет ссылок во всем документе, существуют специальные атрибуты элемента BODY .

LINK - цвет просто ссылок.

ALINK(Active Link) - цвет активных ссылок (активная означает в момент нажатия на нее)

VLINK(Visited Link) - цвет уже посещенных ссылок

Все цвета задаются либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

К примеру если при открытии тела документа, элементу body прописать:

`<body link="red" vlink="green" alink="white">`

то все ссылки в данном документе станут красными, уже посещенные ссылки станут зелеными а ссылки в момент нажатия будут белыми.

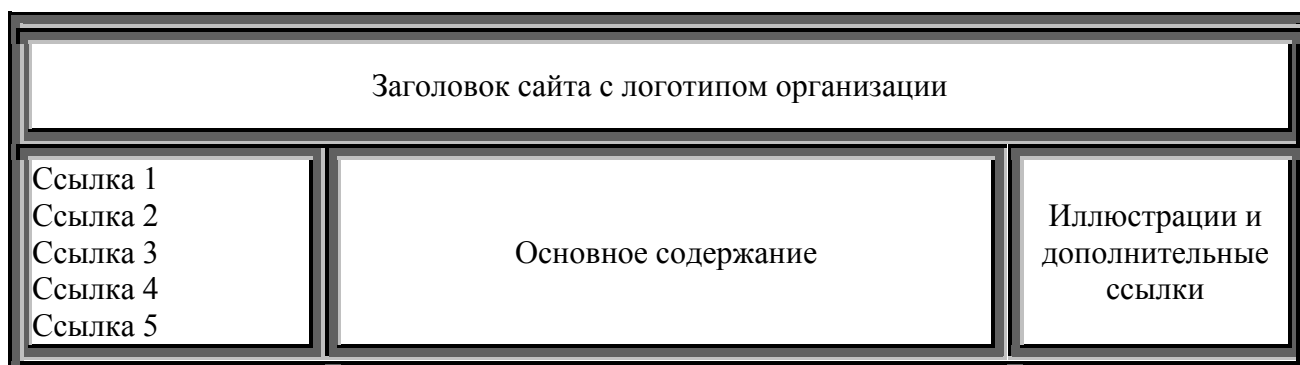
А если нужно чтобы в каком -то месте ссылка имела другой цвет? Например по всему документу ссылки красные, а именно в одном месте черная?

Тогда нужно внутри html ссылки прописать элемент font с атрибутом цвета:

`Черная ссылка`

Разберемся с таблицами в html

Html таблицы используют не только для отображения таблиц как таковых, но и для создания невидимого каркаса страницы, помогающего расположить текст и изображения должным образом.



Чтобы построить простейшую таблицу понадобятся как минимум три элемента:

TABLE - Элемент для создания *html таблицы*. Обязательно должен иметь начальный и конечный теги. По умолчанию html таблица печатается без рамки, а разметка осуществляется автоматически в зависимости от объема содержащейся в ней информации. **Атрибут BORDER** задает толщину рамки таблицы в пикселах.

TR (Table Row) - создает новый ряд таблицы. Закрывающий тег обязателен.

TD (Table Data) - начинает и заканчивает каждую ячейку ряда html таблицы. Обязателен закрывающий тег.

Создадим таблицу из двух строк и двух столбцов

Таблица начинается с `<table>`, затем идёт `<tr>`, указывающий на начало нового ряда. В ряду вставлены две ячейки: `<td>ряд 1 ячейка1</td>` и `<td>ряд1 ячейка2</td>`. Ряд закрывается `</tr>`, и сразу начинается новый ряд `<tr>`. В новом ряду также две ячейки. Таблица закрывается `</table>`.

Пример 1

```
<table border="1">
<tr>
<td>ряд 1 ячейка1</td> <td>ряд1 ячейка2</td>
</tr>
<tr>
<td>ряд 2 ячейка 1</td> <td>ряд 2 ячейка 2</td>
</tr>
</table>
```

Вид таблицы в браузере:

ряд 1 ячейка1	ряд1 ячейка2
ряд 2 ячейка 1	ряд 2 ячейка 2

Для объединения ячеек существуют атрибуты. **COLSPAN** и **ROWSPAN**

COLSPAN – определяет количество столбцов, на которые простирается данная ячейка. По умолчанию имеет значение 1.

ROWSPAN – определяет количество рядов, на которые простирается данная ячейка. По умолчанию имеет значение 1.

Пример 2

```
<table border="1">
<tr>
<td colspan="2">ряд 1 ячейка 1+2</td>
</tr>
```

```
<tr>
<td>ряд 2 ячейка 1</td> <td>ряд 2 ячейка 2</td>
</tr>
</table>
```

Вид таблицы в браузере:

ряд 1 ячейка 1+2	
ряд 2 ячейка 1	ряд 2 ячейка 2

Пример 3

```
<table border="1">
<tr>
<td rowspan="2">Ячейка 1, ряд 1+2</td>
<td>ряд1 ячейка2</td>
</tr>
<tr>
<td>ряд 2 ячейка 2</td>
</tr>
</table>
```

Вид таблицы в браузере:

Ячейка 1, ряд 1+2	ряд1 ячейка2
	ряд 2 ячейка 2

Дополнительные атрибуты:

CELLPADDING – определяет расстояние (в пикселах) между рамкой каждой ячейки html таблицы и содержащимся в ней материалом.

CELLSPACING – определяет расстояние (в пикселах) между границами соседних ячеек таблицы html.

WIDTH – определяет ширину таблицы html. Ширина задается либо в пикселах, либо в процентном отношении к ширине окна браузера. По умолчанию этот атрибут определяется автоматически в зависимости от объема содержащегося в таблице материала.

HEIGHT – определяет высоту таблицы. Высота задается либо в пикселах, либо в процентном отношении к высоте окна браузера. По умолчанию этот атрибут определяется автоматически в зависимости от объема содержащегося в html таблице материала.

Пример 4

```
<table border="1" cellpadding="10">
<tr>
<td>ряд 1 ячейка1</td><td>ряд1 ячейка2</td>
</tr>
```

```
<tr>
<td>ряд 2 ячейка 1</td> <td>ряд 2 ячейка 2</td>
</tr>
</table>
```

Вид таблицы в браузере:

ряд 1 ячейка1	ряд1 ячейка2
ряд 2 ячейка 1	ряд 2 ячейка 2

Пример 5

```
<table border="1" cellspacing="10">
<tr>
<td>ряд 1 ячейка1</td><td>ряд1 ячейка2</td>
</tr>
<tr>
<td>ряд 2 ячейка 1</td> <td>ряд 2 ячейка 2</td>
</tr>
</table>
```

Вид таблицы в браузере:

ряд 1 ячейка1	ряд1 ячейка2
ряд 2 ячейка 1	ряд 2 ячейка 2

Пример 6

```
<table border="1" width="400" height="100">
<tr>
<td>ряд 1 ячейка1</td><td>ряд1 ячейка2</td>
</tr>
<tr>
<td>ряд 2 ячейка 1</td> <td>ряд 2 ячейка 2</td>
</tr>
</table>
```

Вид таблицы в браузере:

ряд 1 ячейка1	ряд1 ячейка2
ряд 2 ячейка 1	ряд 2 ячейка 2

ALIGN – определяет способ горизонтального выравнивания html таблицы или содержимого ячеек. Возможные значения: left, center, right. Значение по умолчанию – left.

VALIGN – должен определять способ вертикального выравнивания таблицы или содержимого ячеек таблицы. Возможные значения: top, bottom, middle. (прижать к верху, прижать к низу, и установить посередине).

BGCOLOR – определяет цвет фона ячеек таблицы. Задается либо RGB-значением в шестнадцатичной системе, либо одним из 16 базовых цветов.

BACKGROUND – позволяет заполнить фон таблицы рисунком. В качестве значения необходимо указать URL рисунка.

Примечание: можно хоть каждой ячейке таблицы задать свой цвет фона или фоновый рисунок. Если вы зададите фоновый цвет (или рисунок) в базовом элементе TABLE то во всех ячейках будет этот фон (рисунок), и если вы захотите в какой-то ячейке поменять, пропишите соответствующий атрибут именно для этой ячейки.

Пример 7

```
<!-- задаем ширину, высоту, рамку, выравнивание по
центру и фоновый цвет всей таблицы-->
<table width="400" height="100" border="1" align="center"
bgcolor="#FFF8D2">
<tr>
<!-- эту ячейку оставляем по умолчанию-->
<td>ряд 1 ячейка1</td>
<!-- содержимое горизонтально выравниваем по центру, верти-
кально - прижимаем к верху-->
<td align="center" valign="top">ряд1 ячейка2</td>
</tr>
<tr>
<!-- содержимое горизонтально выравниваем по центру, верти-
кально - прижимаем к низу-->
<td align="center" valign="bottom">ряд 2 ячейка 1</td>
<!-- содержимое горизонтально выравниваем по правому краю,
вертикально - посередине, и меняем фоновый цвет-->
<td align="right" valign="middle" bgcolor="#33FF99">ряд 2
ячейка 2</td>
</tr>
</table>
```

Результат в браузере:

ряд 1 ячейка1	ряд1 ячейка2
---------------	--------------

ряд 2 ячейка 1	ряд 2 ячейка 2
----------------	----------------

Пример 7

<!-- задаем ширину , высоту, рамку, фоновый цвет всей таблицы, выравнивание оставляем по умолчанию (по левому краю) -->

```
<table width="400" height="100" border="1" bgcolor="#FFF8D2">
```

```
<tr>
```

<!-- горизонтальное-по центру, вертикально по умолчанию (по центру) -->

```
<td align="center"> ряд 1 ячейка1 </td>
```

<!-- горизонтальное-по центру, вертикально по умолчанию (по центру) и делаем фоновый рисунок -->

```
<td align="center" background="pchela.jpg"> ряд1 ячейка2 </td>
```

```
</tr>
```

```
<tr>
```

<!-- горизонтальное-по центру, вертикально по умолчанию (по центру) -->

```
<td align="center"> ряд 2 ячейка 1 </td>
```

<!-- горизонтальное-по центру, вертикально по умолчанию (по центру) и меняем фоновый цвет -->

```
<td align="center" bgcolor="#33FF99">ряд 2 ячейка 2</td>
```

```
</tr>
```

```
</table>
```

Результат в браузере:

ряд 1 ячейка1	ряд1 ячейка2
ряд 2 ячейка 1	ряд 2 ячейка 2

Если картинка меньше чем ячейка, то она будет дублироваться, если больше - то будет отображаться та часть которая помещается в ячейке.

Веб-ресурсы образовательного назначения (обзор основных классов)

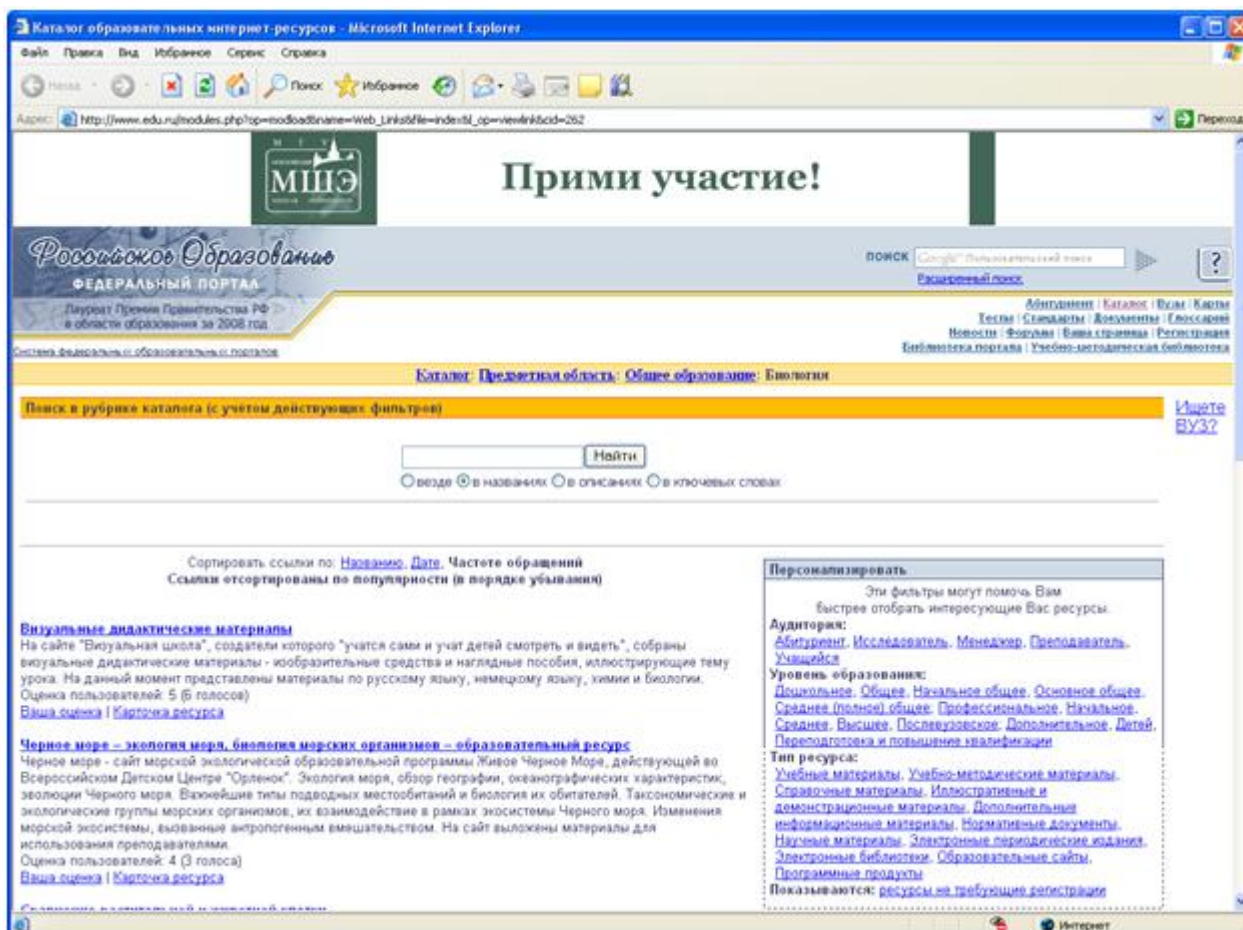


Рис. 13. Российский каталог образовательных ресурсов

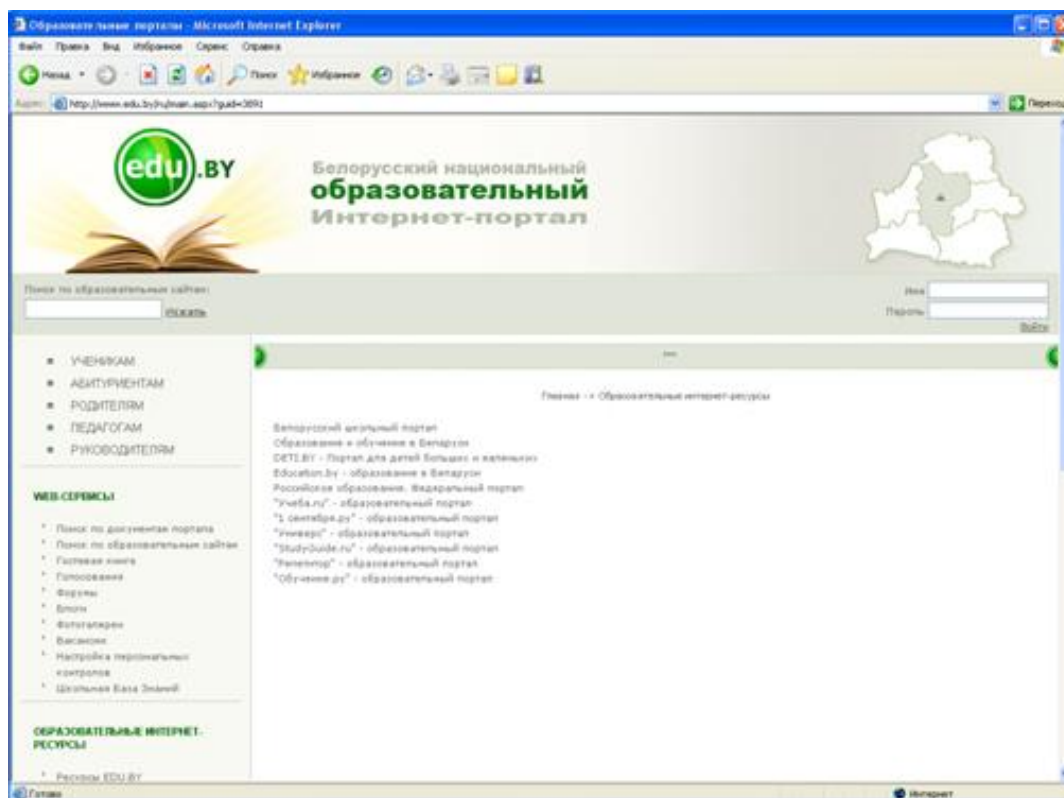


Рис. 14. Белорусский национальный образовательный портал

Технология разработки гипертекстовых сред на основе формата HTML.

Как правило, большинство электронных средств обучения, создаваемых педагогами, основываются не только на использовании разнотипной наглядной информации, но и строятся нелинейным образом за счет введения ссылок, связывающих между собой отдельные содержательные элементы, входящие в средство обучения.

Большинство мультимедиа средств обучения имеет систему навигации по содержательному наполнению, основанную на механизме ссылок. В связи с этим такие понятия как мультимедиа, гипертекст и гипермедиа оказываются тесно связанными. Благодаря широкому распространению WWW (всемирная компьютерная сеть, совокупность документов и мультимедиа-ресурсов, опубликованных в сети), гипертекстовую технологию знают, или, по крайней мере, используют все, кто работает на компьютере.

Формальное определение гипертекста определяет его через представление текстовой информации как сети, в которой читатели получают свободу перемещаться нелинейным образом. Однако можно выдвинуть и следующее определение: *гипертекст* - это расширение традиционного понятия текста, путем введения нелинейного текста, в котором между выделенными фрагментами текста устанавливаются перекрестные ссылки и правила перехода от одного фрагмента к другому.

У гипертекста и созданных на его основе мультимедиа-средств есть много преимуществ. Часть из них носит общий характер, другая часть имеет непосредственное отношение к обучению школьников. Гипертекст представляет собой крайне расплывчатую и вместе с тем широко используемую концепцию. Гипертекстом называют Интернет, мультимедиа-энциклопедию, справочник, книгу с содержанием и предметным указателем, а также любой текст, в котором обнаруживаются какие-либо ссылки (указания) на другие фрагменты. Гипертекст как новая информационная парадигма может рассматриваться как способ коммуникации в обществе, ориентированном на множественные, одновременные потоки разнотипной информации, которые не могут быть восприняты и усвоены субъектом. Усвоение всей суммы знаний становится невозможным, более того, жесткое структурирование такого знания становится труднодостижимой задачей. Знание организуется в гипертекст, в сеть относительно свободных сообщений, которые могут объединяться и распадаться в процессе производства и потребления знания. Мыслительный процесс не порождает идеи строго друг за другом, начиная с пустого места, и не производит их в готовом, отшлифованном виде. Представляется, что процесс мышления идет сразу по нескольким направлениям, что идеи развиваются и отбрасываются одновременно на разных уровнях и с разных позиций, а

также зависят одна от другой, взаимно обогащают друг друга. Есть необходимость во внешней фиксации таких переплетающихся линий мыслительного процесса, представления, например, того, как несколько нитей изложения или аргументации переплетаются вместе. В соответствии с этим, хотя традиционный текст и вынуждает писать и читать параграфы (абзацы), придерживаясь в основном линейной последовательности, он насыщен внутренними связями. Всякий, выполнявший научную работу, знает, что значительная доля труда уходит на добывание литературы, на которую даны ссылки, выискивание перекрестных ссылок, розыск терминов в словаре или глоссарии, составление заметок на записных карточках. Даже при обычном чтении человек постоянно "преодолеывает" внутри текста ссылки на другие главы и разделы. Кроме того, идут ссылки на предметный указатель, делаются сноски, ссылки на библиографию, рисунки или таблицы.

Гипертекст же позволяет автору делать ссылки (устанавливать связи), а читателям гипертекста дает возможность выбирать, каким ссылочным связям следовать, и в каком порядке. Тем самым гипертекст уменьшает ограничения, налагаемые на думающего и пишущего. Он не принуждает к однозначному решению относительно того, принадлежит ли высказывание ходу мыслей написанного или оно находится в ответвлении от основного русла. Гипертекст позволяет хранить аннотации к тексту отдельно от этого текста (документа), однако в тесной привязке к адресатам. Значительный вклад в преимущества гипертекста вносит его насыщенность связями, движение по которым поддерживается компьютером, благодаря чему текст на практике становится нелинейным. В частности, когда гипертекст используется как инструмент мыслительного процесса, инструмент письма или разработки каких-то схем, может возникнуть естественное соответствие между объектами реального мира и узлами гипертекста. Гипертекст предлагает и новые возможности для доступа к большим и сложным источникам мультимедиа-информации. Вводится понятие *гипермедиа* как технологии представления информации разных типов, основанной на принципах гипертекста. В одном гипермедиа-ресурсе сочетаются и возможности перехода по гиперссылкам, и преимущества использования разнотипной информации. При разработке электронных средств обучения определение точного перечня всех структурных элементов содержательного материала - модулей (или блоков), частей (или разделов), глав (или тем), параграфов (или пунктов темы), подпараграфов (или подпунктов) является одной из важнейших задач автора. Усвоение учебного текста школьниками во многом зависит от его композиционной структуры. В настоящее время широко используется модульный принцип организации содержания электронных средств обучения, где основным элементом является так называемый *учебный модуль*. Учебный модуль следует понимать, с одной стороны, как форму организации

обучения, а с другой стороны, как основную структурную единицу учебного текста. К числу определяющих признаков учебного модуля можно отнести следующие признаки:

- элемент содержания учебного курса, представляющий собой логически целостный фрагмент курса (раздел, тема, параграф и т.п.);
- соотнесённые с элементом содержания дидактические средства управления процессом познания; дидактические средства контроля и стимулирования познавательной деятельности;
- фиксированное время работы школьника с модулем (как правило, 35-45 минут).

Таким образом, в соответствии с модульным принципом каждый курс разбивается на крупные тематические модули. В свою очередь эти модули состоят из других структурных единиц учебного текста - основных тем (параграфов), а последние - из пунктов тем (подпараграфов). Рассмотрим более мелкие структурные единицы текста как содержания ЭСО (глава или тема, параграф или пункт темы, подпараграф или подпункт темы). **Глава (тема), параграф и подпараграф.** Среди специалистов, занимающихся разработкой средств обучения для наименования структурных единиц текста используются разные термины (см. табл.). В настоящем Интернет-издании будут употребляться термины: глава (тема), параграф (пункт темы), подпараграф (подпункт темы).

Таблица

Возможные наименования структурных единиц содержательного наполнения электронных средств обучения		
Вариант 1	Вариант 2	Вариант 3
Раздел	Раздел или часть	Раздел
Подраздел (тема)	Глава	Модуль
Пункт темы	Параграф	Тема (или пункт)
Подпункт	Подпараграф	Номер занятия

Рекомендуется включать в состав глав-тем не более 10-12 параграфов-пунктов. Такое же количество пунктов является максимальным и для списков-перечислений. В оглавлении ЭСО перечисляются номера и названия всех разделов, глав (тем) и всех параграфов (пунктов), но не приводится никаких данных о подпараграфах (подпунктах). Подпараграфы номеров не имеют и не включаются в содержание ЭСО. Увеличение числа параграфов ведёт к увеличению числа заголовков, что в свою очередь приводит к изменению их дидактической роли в структуре ОЭИ. Осмыслить такую структуру, - значит сделать первый шаг к пониманию материала. Заголовок должен актуализировать имеющиеся у обучаемого

представления по заданной теме. **Заголовки** структурных элементов содержательного наполнения ЭСО должны быть разнообразными и интересными, создавать у школьников повышенный интерес и повышенную мотивацию к изучению трудного материала, оказывать ему максимально возможную помощь. Авторам необходимо отказаться от общих и неопределённых в содержательном плане заголовков. Заголовок небольшого параграфа должен быть тщательно продуманным и четко отражать содержание параграфа. Традиционные заголовки необходимо чередовать с заголовками нового типа. Оптимальное сочетание традиционных и нестандартных заголовков в структуре учебника способствует достижению максимального дидактического эффекта.

Внешним выражением композиционной структуры текста является **рубрикация**. Степень сложности рубрикации зависит от содержания текста. Простейшей рубрикой является абзац (отступ вправо в начале первой строки). Абзац служит показателем перехода от одной мысли (темы) к другой. Разделение текста на абзацы позволяет читателю делать небольшие остановки, мысленно возвращаться к прочитанному, сосредоточиться перед дальнейшим чтением.

Абзац всегда является внутренне замкнутой смысловой единицей. Нарушение этого требования затрудняет процесс чтения. Абзацное членение текста нередко сочетается с нумерацией - числовым или буквенным обозначением последовательности расположения составных частей текста. Нумерация подчёркивает необходимость самостоятельного рассмотрения каждого из нумеруемых элементов перечислений. При создании ЭСО наряду с нумерацией широко используются графические маркеры различной конфигурации. Рекомендуются их активное использование. Система нумерации может дополнить абзацное членение текста. Выбор того или иного варианта нумерации зависит от содержания ЭСО, его объёма, состава, композиционной структуры. Обычно используются только однотипные знаки - чаще арабские цифры или большие и строчные буквы. При использовании нумерации важно учитывать, что нумерация возможна лишь в том случае, если имеется, по крайней мере, два однородных элемента перечисления. Кроме того, однотипные средства нумерации (цифры, буквы, маркеры) могут применяться только по отношению к однотипным (по назначению, по месту в структуре) частям. Создание гипертекстовых электронных средств обучения требует от педагогов и знания основных правил организации гиперссылок и системы навигации по содержательному наполнению ЭСО. В числе таких правил можно отметить следующие:

- гиперссылки в тексте должны быть выделены;
- должна просматриваться чёткая логическая обусловленность каждого последующего шага в цепочке гиперссылок;

- в каждой главе (теме), параграфе и подпараграфе должны быть указатели гиперссылок, с помощью которых можно вернуться к началу темы (страницы), оглавлению, перейти к параграфу или подпараграфу, пункту или подпункту;
- гиперссылка в поле текста должна быть предельно короткой;
- количество гиперссылок в тексте определяется целями и задачами создания ЭСО, а также спецификой его содержания;
- гиперссылка в тексте должна быть максимально полезной, ставить её стоит только в том случае, если она ведёт на ресурс, тема которого затронута в тексте, но не раскрыта.
- не рекомендуется делать гиперссылки ссылки со слов "тут", "здесь" и т.п.;
- при компоновке гипертекста не следует перегружать текст гиперссылками.

Таким образом, гипертекстовые и гипермедиа-технологии открывают широкие возможности перед педагогами, занимающимися разработкой электронных средств обучения. В то же время создание таких средств требует от учителей, как знаний правил компоновки гипертекстовых документов, так и умений работать с основными инструментальными средствами и системами, предназначенными для создания гипермедиа-ресурсов.

Создание web-сайта средствами web-редакторов.

Планируем веб-узел.

Создадим сайт для ознакомления всех желающих с писателями и поэтами Белорусии. Подготовим изображения в формате JPG. Размер изображения должен составлять несколько килобайт для уменьшения времени загрузки сайта. Подготовим тексты в любом текстовом редакторе для их быстрого переноса на веб-странички нашего сайта. Все материалы поместим в папку Images, расположенной в папке сайта.

Создайте папку Site в своей личной папке.

В ней создайте папки для рисунков Images и с текстами Texts. Скопируйте из сетевой папки с материалами для лабораторной работы рисунки и тексты.

Изготовление надписей для веб-страниц. Откройте программу Hara WebStyle. Выберите пункт меню Заголовки (Heading, см. рис 258). Откроется выбора стиля надписи (рис. 259). Выберите стиль надписи и перейдите к следующему пункту меню **Text**, введите текст надписи, выберите гарнитуру и начертание шрифта (рис.260).



Рис. 15. Меню программы: 1 – заголовки; 2- объемные заголовки; 3 – подложки; 4 – кнопки с надписями; 5 – кнопки; 6- линии.



Рис. 16. Окно изготовления заголовков: 1 – стиль надписи; 2 – стиль текста; 3 – цвет; 4 – текстура подложки; 5 – стиль тени; 6 – темы, сохраненные ранее; 7 – размер изображения; 8 – сохранение рисунка; 9 – окно выбора (стиль надписи); 10 – окно просмотра.

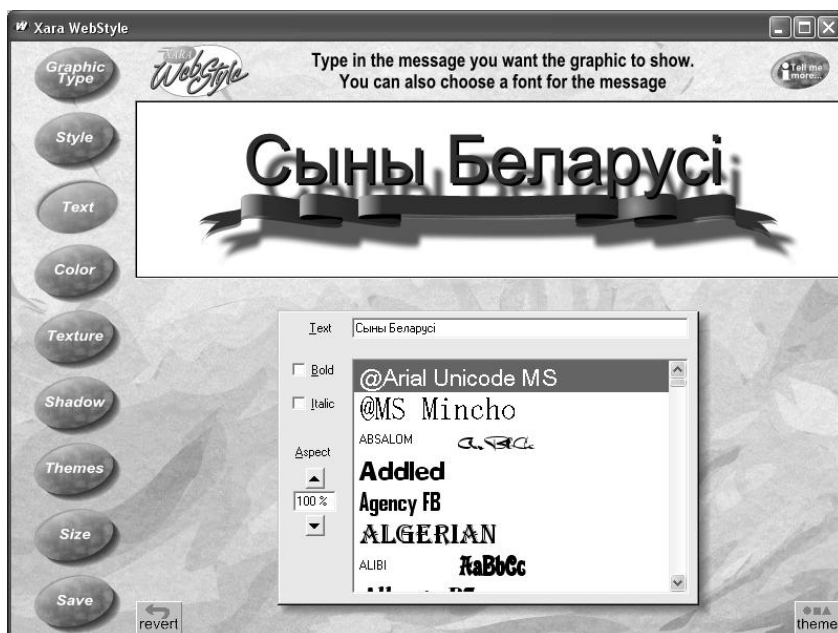


Рис. 17. Работа с текстом

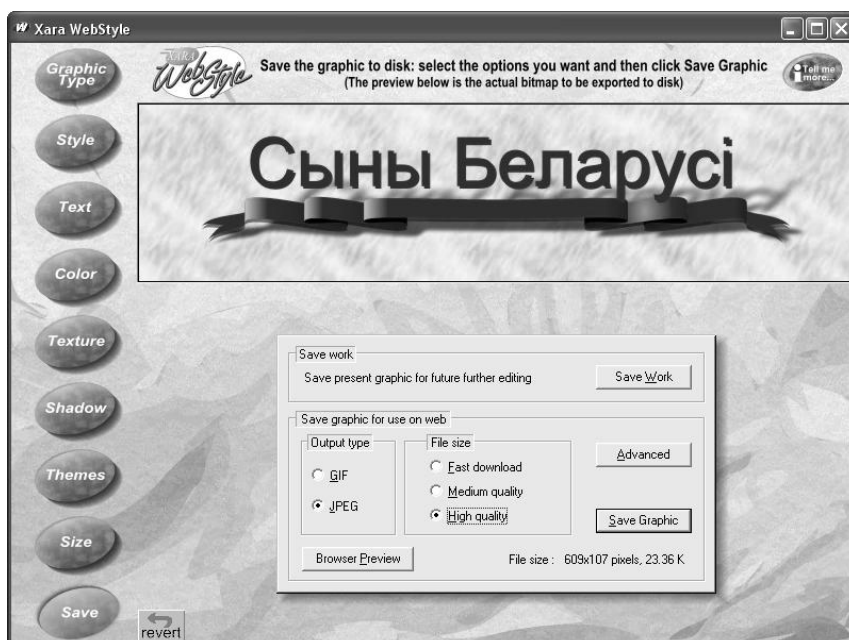


Рис. 18. Сохранение рисунков

Перейдите на вкладку **Color**, измените цвет букв и тени, на вкладке **Texture** выберите фон для надписи. Советуем оставить его прозрачным, тогда надпись подойдет на любой фон вашей страницы.

На вкладке **Shadow** измените характер тени; следующий пункт, сохранение и выбор темы, можно пропустить. Размер изображения устанавливается на вкладке **Size**. Сделайте небольшой запас в сторону увеличения размеров изображения, так как изображения (jpg) в дальнейшем не поддаются сильному масштабированию, как и все растровые изображения.

При сохранении изображения выберите формат сохранения файла (Gif или Jpg), качество контролируйте по размеру файла (рис. 261, File size). Сохраняйте файлы в папку с рисунками (Images) внутри папки сайта.

Подготовка фотографий для публикации на веб-страницах.

Откройте одну из фотографий из папки с материалами для лабораторной работы с помощью графического редактора. Выполните действия по уменьшению размера изображения (рис.262).

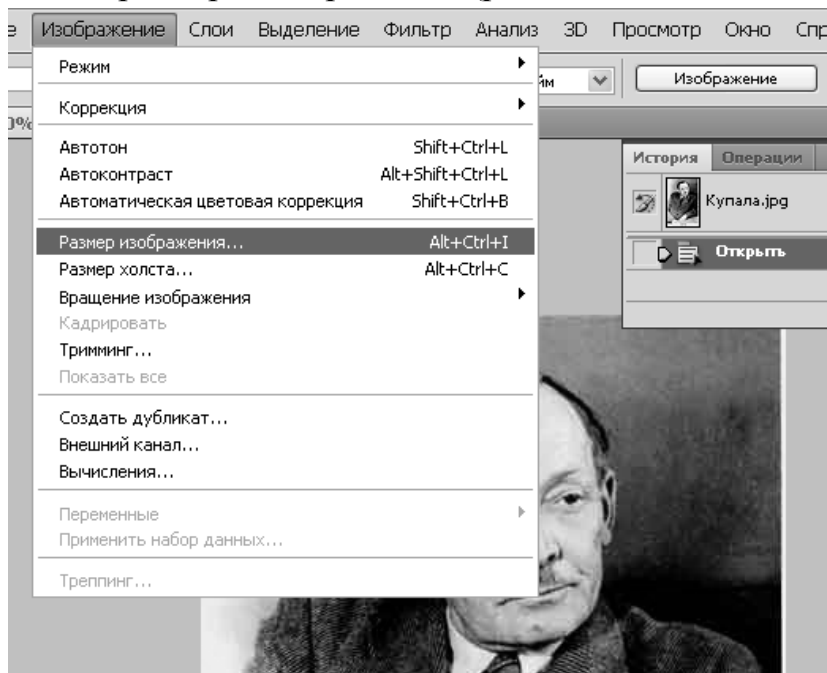


Рис. 19. Изменение размера изображения

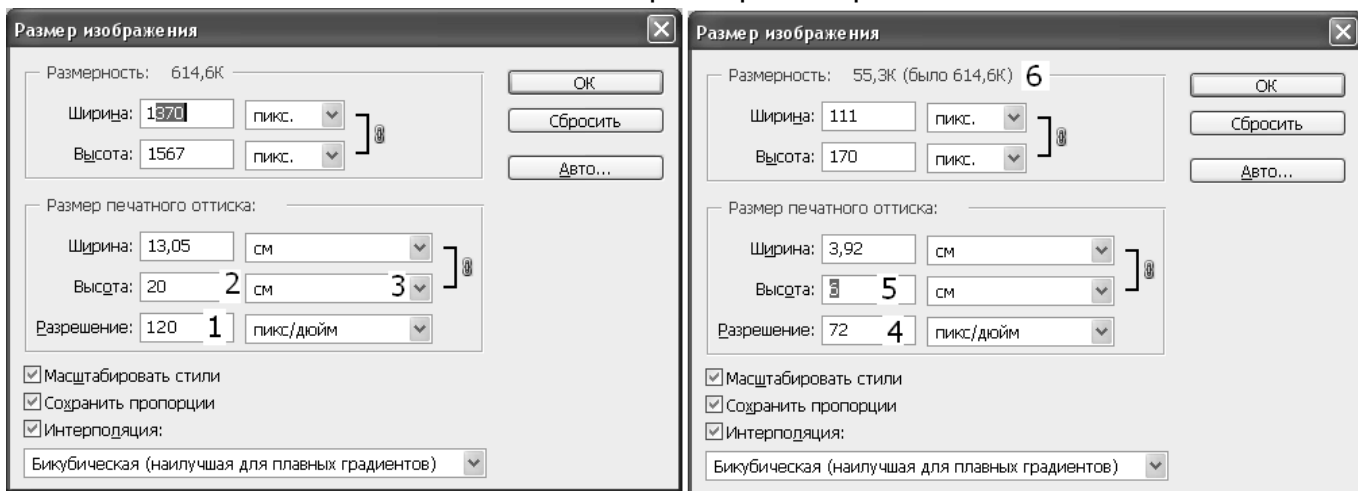


Рис. 20. Первоначальный (слева) и конечный размер изображения

В окне **Размер изображения** (рис.263) обратите внимание на высоту (ширину) изображения (2) и единицы их измерения (3) и разрешение изображения (1). Установите высоту изображения 6 см (5), а разрешение - достаточное для экранного воспроизведения изображений - 72 пикс/дюйм (4). Проследите за изменением размера полученного файла (6).

Выполните операции **Файл – Сохранить для Web**. Установите формат файла, способ и степень сжатия изображения (3): **JPG, Низкое, Оптимизация**. Как видно из рис. 264, размер конечного изображения не превышает 10 Кб (4). Загрузка изображения будет происходить в течение 2 секунд.



Рис. 21. Оригинал (слева) и конечный файл для веб-страницы (справа) : 1 – оригинал; 2 – фото для веб; 3 – параметры сжатия; 4- параметры сжатого изображения.

Сохраните изображение в папку Image внутри своей папки Site.

Создание сайта

Запустите на выполнение программу Front Page. Закройте открывшуюся по умолчанию страницу. Выполните действия **Файл – Создать – Другие шаблоны страниц...** для создания многостраничного сайта с использованием кадров (рис.265).

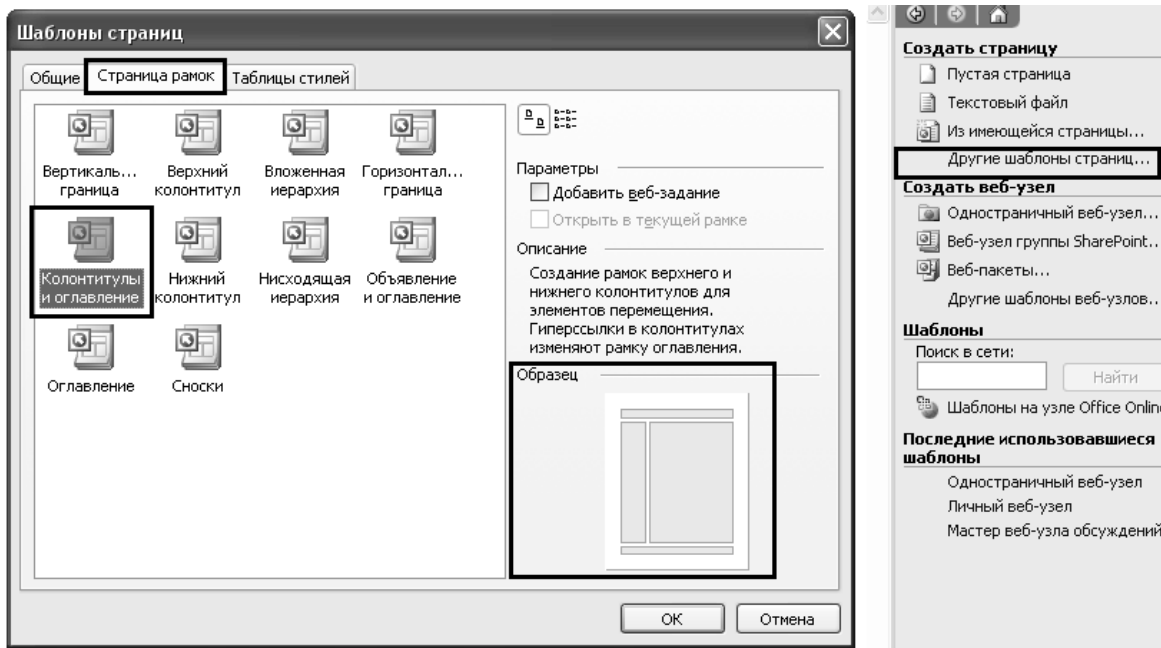


Рис. 22. Выбор шаблона веб-страниц

После нажатия на ОК окно Front Page примет вид, представленный на рис.

266.

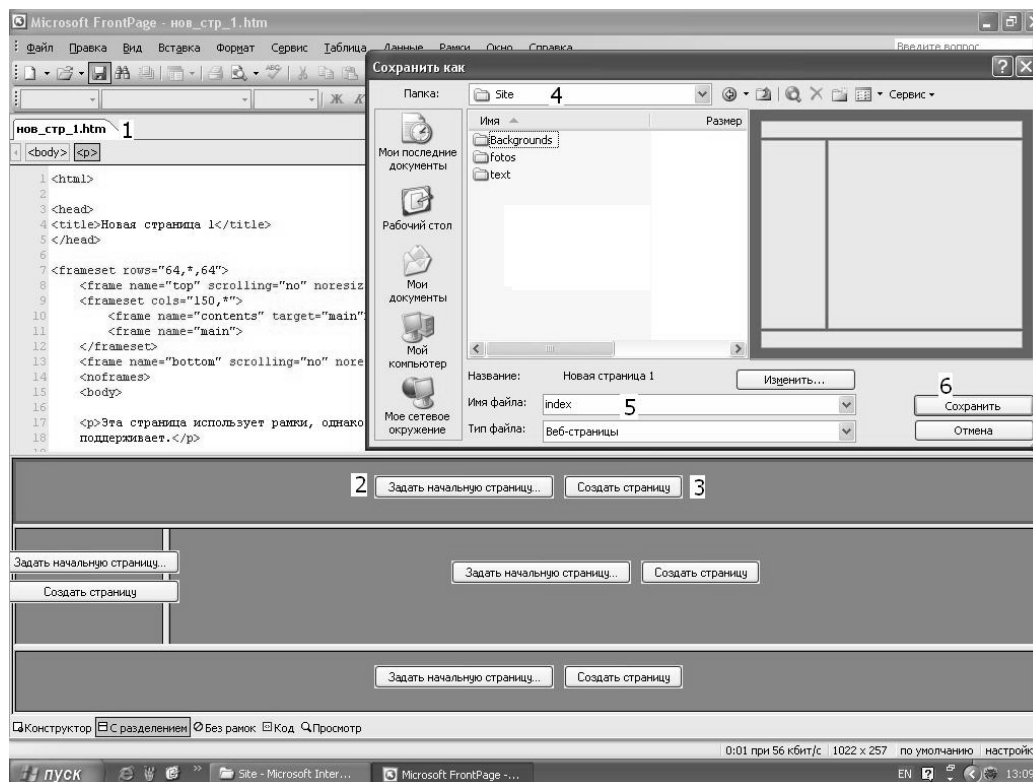


Рис. 23. Сохранение главного файла сайта.

Главный файл сайта сейчас назван **нов_стр_1.htm** (1). Сохраним (6) его в свою папку (4) Site под именем **index.htm** (5, рис.266).

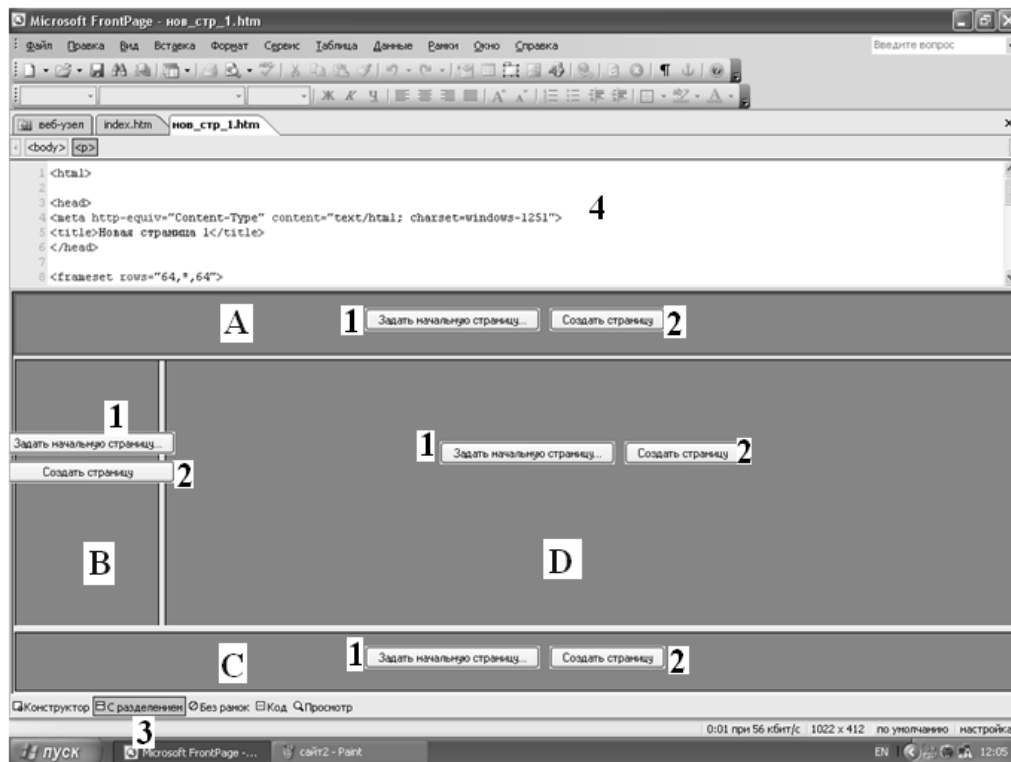


Рис. 24. Редактирование веб-страницы: А – окно заголовка страницы; В – окно ссылок (содержание сайта); С – окно дополнительных ссылок; D – основное окно-кадр с меняющейся информацией.

Главная страница (рис.267) состоит из четырех кадров (фреймов). Каждый кадр (А-Д) веб-страницы содержит 2 кнопки: 1 – Задать начальную страницу (используется для выбора имеющейся страницы, которая будет загружаться и отображаться в кадре первой после загрузки страницы в браузер. 2 – кнопка для создания новых страниц.

Щелкните по кнопке **Создать страницу** последовательно во всех кадрах веб-страницы. В результате будут созданы чистые страницы.

Щелчок по любому кадру вызывает появление в окне кода текста страницы.

Обратите **внимание** на кодировку страницы (4): если она отличается от принятой в нашей стране, измените ее на «**windows-1251**». Сохраните все четыре файла в папку со своим сайтом, переименовав станицы в «Page1 – Page4» (рис.268).

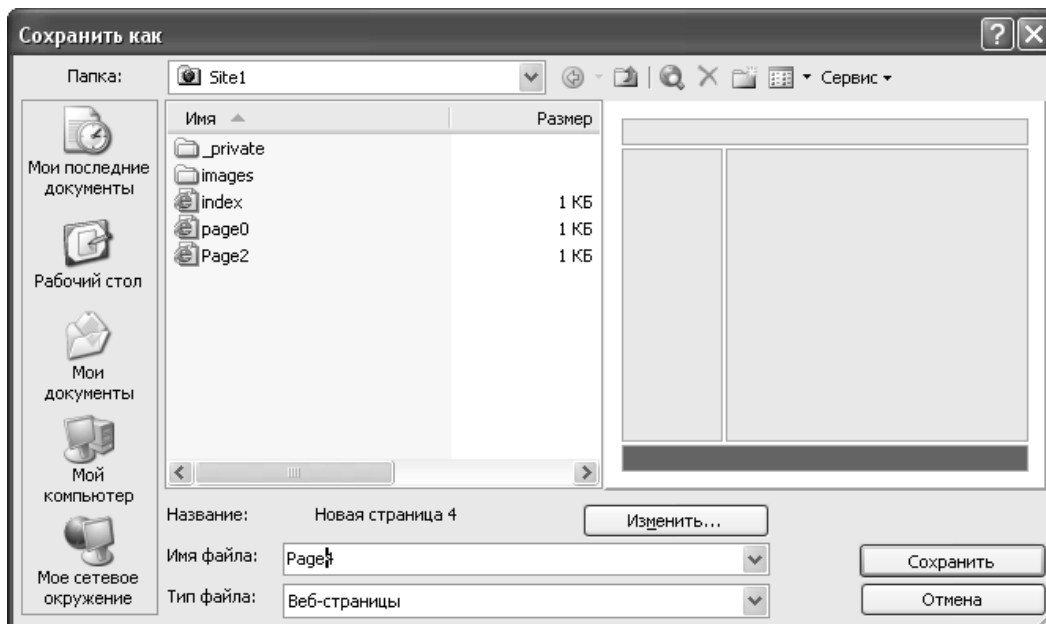


Рис. 25. Сохранение страниц сайта.

Изменим фон каждого кадров: **Формат – Фон – Фоновый рисунок – Сделать подложкой** (рис.269). Выберите фон кадра из своей папки:\Site\Backgrounds (рис.270).

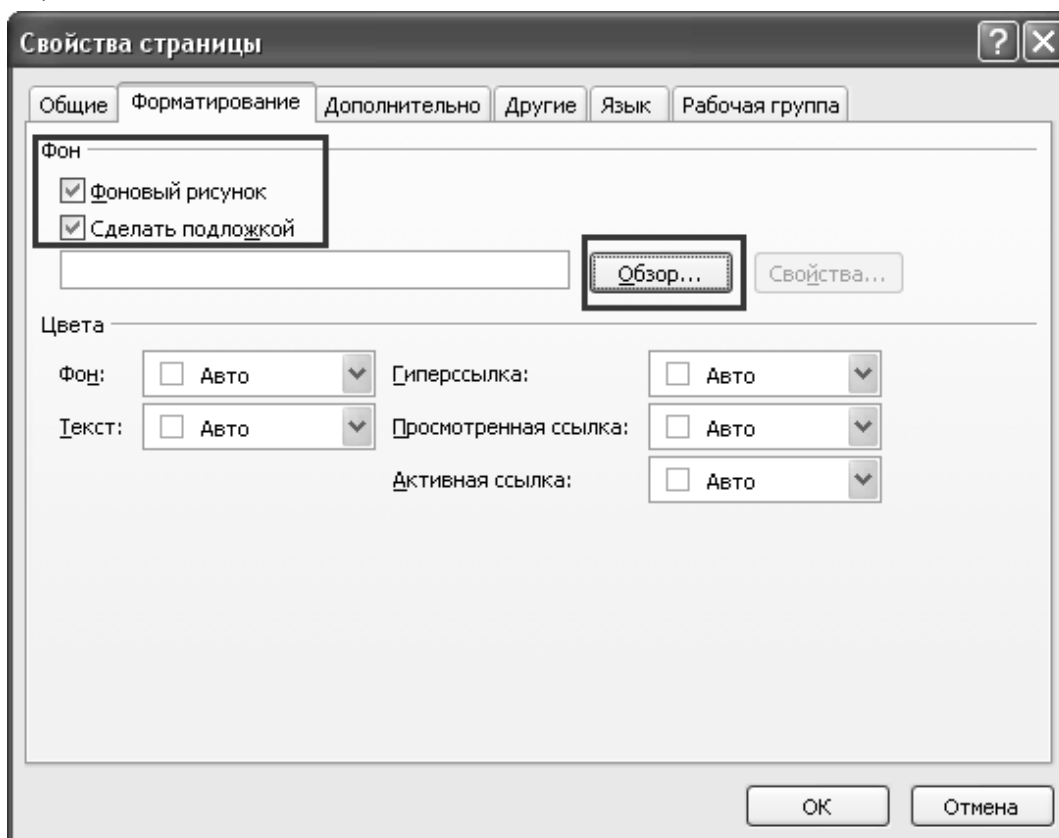


Рис. 26. Выбор подложки (фона) страницы

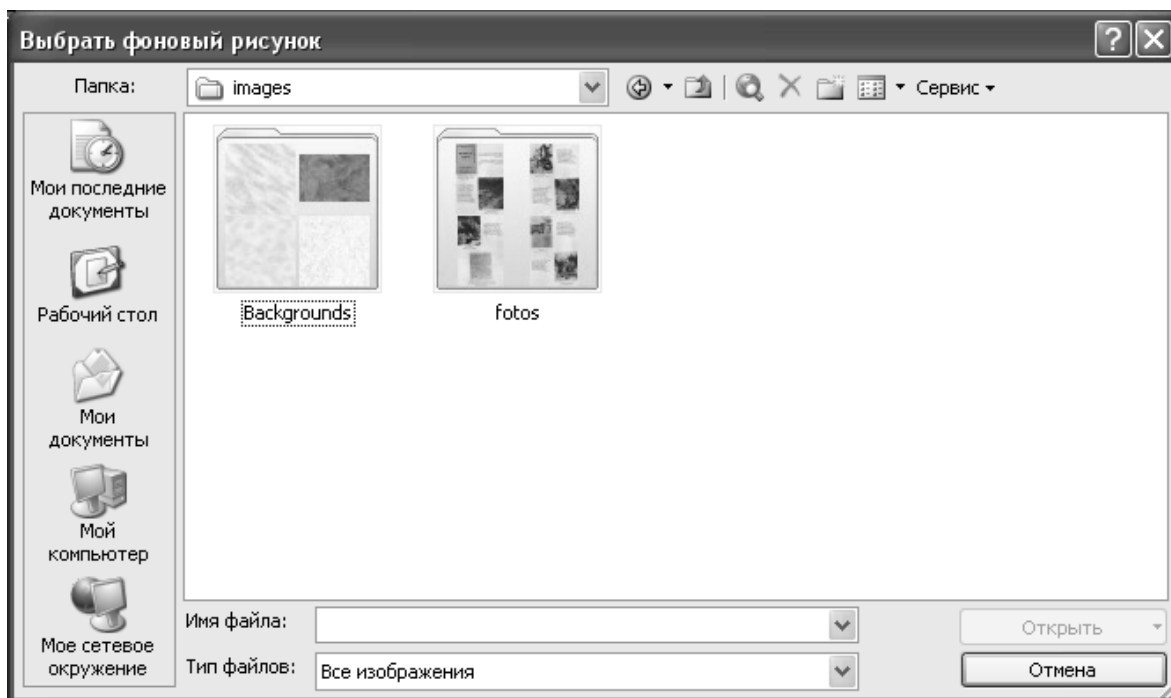


Рис. 27. Выбор рисунка для обоев

Повторите операцию для всех кадров страницы.

Вставьте рисунок “Сыны Беларуси” в кадр А (рис.271).

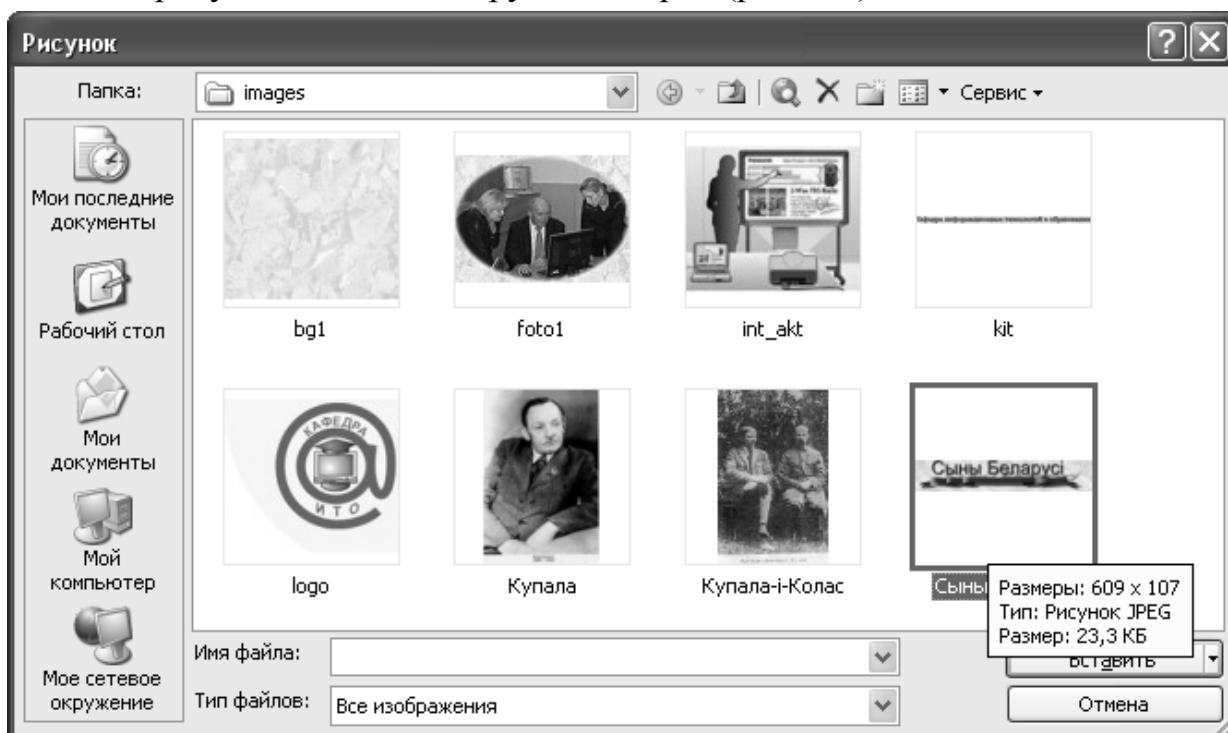


Рис. 28. Вставка рисунков

Измените подложку, если она не гармонирует со вставленным рисунком. Перейдите на кадр D. Вставьте таблицу 2x2. Поместите в первой строке фото писателей, во второй строке – пояснения (фамилия, и.о., псевдоним). Для проверки результатов изменений переключите редактор в режим просмотра страниц.

Сохраните файлы и закройте редактор. Откройте файл index из своей папки Site, проверьте работоспособность сайта.

Откройте файл `index` с помощью Front Page и продолжите редактирование сайта.

Разбейте ячейки верхней строки на 2 столбца и вставьте еще 2 фотографии. Добавьте еще 2 строки в таблицу и дополните ее фотографиями и текстами. В кадре В веб-страницы перечислите фамилии писателей и поэтов Беларуси. Создадим веб-страницу с информацией о Я.Купале и сделаем ссылку на нее с кадра В.

Под словом Файл меню редактора выберите пиктограмму чистого листа. В результате будет создана новая веб-страница («нов_стр_1»). Сохраните ее, в открывшемся меню измените имя страницы на «Купала». Поместите на странице текстовую и графическую информацию. Сохраните файл.

Переключите редактор на страницу. «`index`» и вставьте ссылку со слова «Я. Купала» на одноименный файл. Сохраните и проверьте работу сайта. Перенесите сайт в свою личную папку и проверьте его работоспособность.



Рис. 29. Результат создания сайта

Создайте по аналогии со страничкой «Купала» страницы о всех писателях, фото которых находятся на главной странице. Изготовьте ссылки на данные информационные страницы с кадра В и с фотографий кадра D.

Размещение сайта в сети.

Выполните следующие действия по поиску сервера в сети ПК:

Пуск – Поиск – Компьютеры или людей – Компьютер в сети – 10.200.200.241 – Найти.

Откройте папку **htdocs**. Создайте свою папку. Скопируйте в нее все файлы своего сайта.

Запустите браузер. В строке адреса введите **http://10.200.200.241** и нажмите **Enter**.

Откроется страница с папками имеющихся сайтов. Откройте сайт щелчком по имени своей папки.

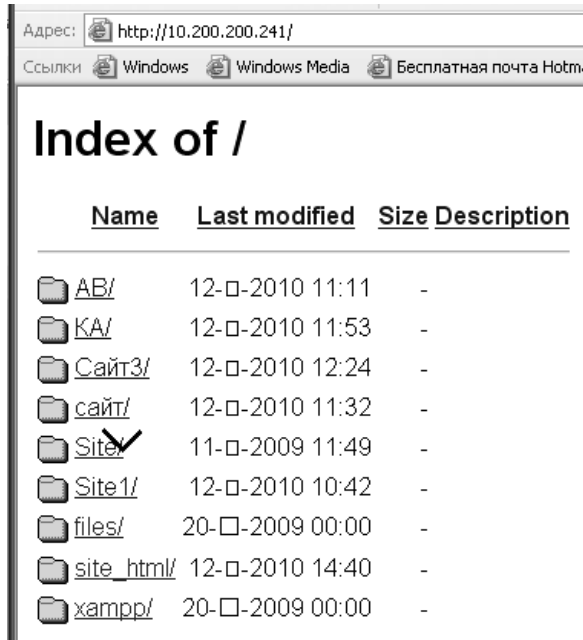


Рис. 30. Запуск сайта с сервера

Литература:

Г.Рябов Сете- или - тура? - URL: <http://www.litera.ru/slova/ryabov/setetura.html>
Точность цитирования – дословно

Руднев В. Словарь культуры XX века
Точность цитирования - почти дословно

[Википедия](#)