ЯЗЫК ГИПЕРССЫЛОК

Учитель математики и информатики

МБОУ «Кубянская сош»

Атнинского муниципального района РТ

Хакимзянова Нурания Идерисовна

2015 год

Пояснительная записка.

В связи с бурно развивающимися информационными технологиями и ресурсами большое значение приобрела проблема изучения сетевых технологий, чтобы каждый мог создавать личностно значимую для него образовательную продукцию. Такой продукцией в данном курсе является web-сайт.

Особенность изучаемого курса состоит в том, что он может использоваться во многих профилях, так как он относится ко всем сферам современного общества – гуманитарным, естественно - научным, социальным, экономическим и другим.

Курс рассчитан на 35 часов лекционно-практических занятий и проводится в течение учебного года по 1 часу в неделю. Данный курс рекомендован и составлен для обучающихся 10-11 классов.

Целью курса является научить обучающихся проектировать и конструировать сайты. Достижение цели реализуется с решением следующих *задач*:

- ✓ Познакомить с видами web-сайтов, их функциональными, структурными и технологическими особенностями;
- ✓ Сформировать навыки проектирования, конструирования, размещения web-сайта;
- ✓ Сформировать навыки написания html-кодов;
- ✓ Создать web-сайт по выбранной тематике.

Программа курса предусматривает проведение лекционных и практических занятий, лабораторных работ. Изучение нового материала носит сопровождающий характер. Освоение курса предполагает, помимо посещения коллективных занятий, выполнение внеурочных самостоятельных заданий. Курс завершается итоговым контролем в виде творческой работы. Обучающийся должен продемонстрировать уровень достижения минимально необходимых результатов, обозначенных в целях и задачах курса.

РАБОЧАЯ ПРОГРАММА

элективного курса

№	Наименование разделов и тем	Кол-во часов			
		Всего	Теория	Практи-ческих	Самостоя-тельных
1	Основы работы в глобальной сети Internet.	1	1		
2	Введение в HTML. Форматирование текста.	4	1	2	1
3	Графика.	4	1	2	1
4	Таблицы.	4	1	2	1
5	Формы.	4	1	2	1
6	Фреймы.	4	1	2	1
7	Ссылки.	4	1	2	1
8	Мультимедиа.	4	1	2	1
9	Каскадные таблицы стилей(CSS).	4	1	2	1
10	Проектирование сайта.	2		2	
	Итого	**35**	**9**	**18**	**8**

Содержание курса

1. Основы работы в глобальной сети Internet

Основные понятия. Работа с браузером. Поиск информации сети Internet.

2. Введение в HTML. Форматирование текста

Основные понятия. Структура HTML-страницы. Создание HTML-страницы. Параметры страницы.

Оформление текста. Выравнивание абзацев. Заголовки и подзаголовки. Параметры шрифта. Списки. Типы списков.

Обучающийся должен знать:

- структуру HTML-документа;

- основные теги для форматирования текста, создания фона и списков в различных вариантах.

Обучающийся должен уметь:

- пользоваться основными атрибутами для форматирования текста и создания списков в различных вариантах;
- создавать HTML-страницы.

3. Графика

Размещение графики на Web-странице. Форматы графических файлов. Выравнивание изображений. Карты изображений. Фон Web-страницы. Вставка бегущей строки на Web-страницу.

Обучающийся должен знать:

- основные теги для размещения изображений на Web-странице;
- форматы графических файлов;
- теги для вставки бегущей строки.

Обучающийся должен уметь:

- размещать изображения на Web-странице;
- вставлять бегущую строку на Web-страницу.

4. Таблицы

Основные теги таблиц. Создание таблиц. Создание строк и столбцов. Создание ячеек таблицы. Создание заголовка и подписи таблиц. Теги группирования элементов таблиц. Основные атрибуты элементов таблиц.

Обучающийся должен знать:

- теги для создания таблиц в html-документе;
- теги для создания строк и столбцов;
- теги для создания ячеек таблицы;
- теги для создания заголовка и подписи таблиц;
- теги группирования элементов таблиц.

Обучающийся должен уметь:

- создавать таблицы на Web-странице;
- пользоваться основными атрибутами для форматирования таблиц.

5. Формы

Создание форм. Размещение на форме элементов управления. Списки выбора. Многострочные текстовые поля.

Обучающийся должен знать:

- теги для создания форм в html-документе;
- теги для размещения на форме элементов управления.

Обучающийся должен уметь:

- создавать формы на Web-странице;
- пользоваться управляющими элементами разных типов.

6. Фреймы

Свойства фреймов. Наборы фреймов. Вставка фрейма в документ. Взаимодействие фреймов. Примеры использования и взаимодействия фреймов.

Обучающийся должен знать:

- понятие фрейма;
- свойства фрейма;
- наборы фреймов;
- теги для создания фреймов в html-документе.

Обучающийся должен уметь:

- вставлять фреймы на Web-странице;
- пользоваться основными атрибутами для преобразования фрейма;

7. Ссылки

Универсальный указатель ресурса. Вставка ссылок в документ. Внутренние ссылки. Внешние ссылки. Переходы внутри документа. Графические ссылки.

Обучающийся должен знать:

- понятие гиперссылки;
- теги для вставки ссылок в HTML-документ;
- теги внутренних и внешних ссылок;
- теги графических ссылок.

Обучающийся должен уметь:

- вставлять ссылки на текст и графические объекты.

8. Мультимедиа

Размещение объектов мультимедиа на Web-страницах. Анимационные файловые форматы. Звуковые форматы. Форматы видео.

Обучающийся должен знать:

- теги для вставки объектов мультимедиа на Web-странице;
- анимационные, звуковые форматы;
- форматы видео.

Обучающийся должен уметь:

- размещать звук, видео и flash-анимации на Web-странице.

9. Каскадные таблицы стилей(CSS)

Поддержка стилей. Уровни CSS. Синтаксис листа стиля. Группирование стилей и селекторы классов. Четыре варианта использования CSS. Примеры использования стилей.

Обучающийся должен знать:

- понятие каскадных таблиц стилей;
- синтаксис листа стиля;

Обучающийся должен уметь:

- использовать каскадные таблицы стилей.

Перечень практических работ

1. Создание простейшей HTML-страницы. Форматирование текста.
2. Вставка изображений в HTML-документ.
3. Оформление Web-страницы с таблицами.
4. Создание форм на Web-страницах.
5. Создание фреймов. Вставка фрейма в документ.
6. Создание HTML-документа с помощью редакторов гипертекста.
7. Размещение звука, видео и flash-анимаций на Web-странице.
8. Шрифтовое и абзацное форматирование. Форматирование списков. Размещение стилевой таблицы.

9. Проектирование сайта.

№	Наименование разделов и тем	Кол-во часов	Календарные сроки
1	***Основы работы в глобальной сети Internet*** Основные понятия. Работа с браузером. Поиск информации сети Internet.	1	сентябрь
2	***Введение в HTML. Форматирование текста*** Основные понятия. Структура HTML-страницы. Создание HTML-страницы. Параметры страницы. Оформление текста. Выравнивание абзацев. Заголовки и подзаголовки. Параметры шрифта. Списки. Типы списков.	1	сентябрь
3	Практическая работа № 1. Создание простейшей HTML-страницы. Форматирование текста.	2	сентябрь
4	Самостоятельная работа №1.	1	октябрь
5	***Графика*** Размещение графики на Web-странице. Форматы графических файлов. Выравнивание изображений. Карты изображений. Фон Web-страницы. Вставка бегущей строки на Web-страницу.	1	октябрь
6	Практическая работа № 2. Вставка изображений в HTML-документ.	2	октябрь
7	Самостоятельная работа №2.	1	октябрь
8	***Таблицы*** Основные теги таблиц. Теги группирования элементов таблиц. Основные атрибуты элементов таблиц.	1	ноябрь
9	Практическая работа № 3. Оформление Web-страницы с таблицами.	2	ноябрь
10	Самостоятельная работа №3.	1	декабрь
11	***Формы*** Создание форм. Размещение на форме элементов управления. Списки выбора. Многострочные текстовые поля.	1	декабрь
12	Практическая работа № 4. Создание форм на Web-	2	декабрь

	страницах.		
13	Самостоятельная работа №4.	1	январь
14	***Фреймы*** Свойства фреймов. Наборы фреймов. Вставка фрейма в документ. Взаимодействие фреймов. Примеры использования и взаимодействия фреймов.	1	январь
15	Практическая работа № 5. Вставка фрейма в документ.	2	январь
16	Самостоятельная работа №5.	1	февраль
17	***Ссылки*** Универсальный указатель ресурса. Вставка ссылок в документ. Внутренние ссылки. Внешние ссылки. Переходы внутри документа. Графические ссылки.	1	февраль
18	Практическая работа № 6. Создание HTML-документа с помощью редакторов гипертекста.	2	февраль
19	Самостоятельная работа №6.	1	март
20	***Мультимедиа*** Размещение объектов мультимедиа на Web-страницах. Анимационные файловые форматы. Звуковые форматы. Форматы видео.	1	март
21	Практическая работа № 7. Размещение звука, видео и flash-анимаций на Web-странице.	2	март
22	Самостоятельная работа №7.	1	апрель
23	***Каскадные таблицы стилей(CSS)*** Поддержка стилей. Уровни CSS. Синтаксис листа стиля. Группирование стилей и селекторы классов. Четыре варианта использования CSS. Примеры использования стилей.	1	апрель
24	Практическая работа № 8. Шрифтовое и абзацное форматирование. Форматирование списков. Размещение стилевой таблицы.	2	апрель
25	Самостоятельная работа №8.	1	май
26	Практическая работа № 9. Проектирование сайта.	2	май

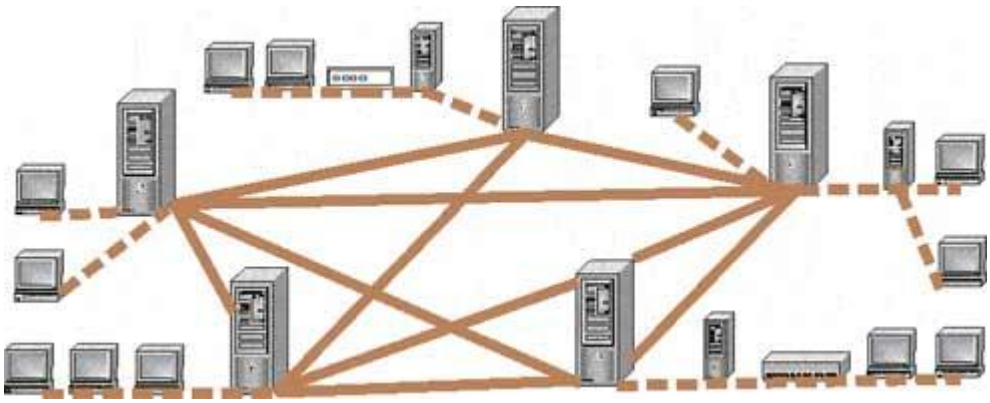
1. Глобальная сеть Интернет. Определение сети Интернет

Интернет – всемирная информационная компьютерная сеть, представляющая собой объединение множества региональных компьютерных сетей и компьютеров, обменивающихся друг с другом информацией по каналам общественных телекоммуникаций (выделенным телефонным аналоговым и цифровым линиям, оптическим каналам связи и радиоканалам, в том числе спутниковым линиям связи).

Интернет является одноранговой сетью, т.е. все компьютеры в сети по сути равноправны, и любой компьютер можно подключить к любому другому компьютеру. Любой компьютер, подключенный к сети, может предлагать свои услуги любому другому. Но Интернет – это не только каналы связи. В узлах этого всемирного соединения установлены компьютеры, которые содержат различные информационные ресурсы и предлагают различные информационные и коммуникационные услуги.

Информация в Интернет хранится на серверах. Серверы имеют свои адреса и управляются специализированными программами. Они позволяют пересылать почту и файлы, производить поиск в базах данных и выполнять другие задачи. Обмен информацией между серверами сети выполняется по высокоскоростным каналам связи. Доступ отдельных пользователей к информационным ресурсам Интернет обычно осуществляется через провайдера или корпоративную сеть. Провайдер - поставщик сетевых услуг – лицо или организация предоставляющие услуги по подключению к компьютерным сетям. В качестве провайдера выступает некоторая организация, имеющая модемный пул для соединения с клиентами и выхода во всемирную сеть. Существуют также компьютеры, которые непосредственно подключены к глобальной сети. Они называются хост - компьютерами (host - хозяин). Хост – это любой компьютер, являющийся постоянной частью Интернета, т.е. соединенный по Internet – протоколу с другим хостом, который в свою очередь, соединен с другим, и так далее.

Ниже представлена структура глобальной сети Интернет



Просмотр Web-страниц осуществляется *с* помощью специальных программ просмотра - браузеров (браузеров). В настоящее время существует множество Web -браузеров, но наиболее популярными считаются Internet Explorer от компании Microsoft .

Работу с браузером рассмотрим на примере программы Internet Explorer 5.0, входящей в состав операционной системы Windows*98* , Windows 2000, а также в состав пакета прикладных программ Microsoft 2000.

Программа Internet Explorer позволяет:

• загружать Web-страницы с удаленных серверов вместе со встроенными объектами;

• просматривать Web -страницы, принятые из Сети и воспроизводить встроенные в них объекты;

• сохранять принятые Web -документы на жестком диске компьютера в виде, пригодном для их последующего

• автономного просмотра без подключения к Сети;

• просматривать ранее принятые и сохраненные Web -документы;

• одновременно выполнять операции с несколькими Web -документами в разных окнах Обозревателя.

Найти Web-страницу в Интернете или сделать на нее ссылку можно с помощью *универсального указателя ресурсов* (адреса страниц), который включает в себя способ доступа к документу, имя сервера, на котором находится документ, а также путь к файлу. Способ доступа к документу определяется используемым протоколом передачи информации. Для доступа к Web -страницам используется протокол передачи гипертекста HTTP Например, для начальной Web-страницы Internet Explorer универсальный указатель ресурсов принимает вид http home.microsoft.com\intl\ru и состоит из трех частей: Вместе с браузером на компьютер обычно устанавливается программа для пользования службами электронной почты *(e-mail)* и новостей *(nesw).* По сути дела, браузер является основной программой для доступа к службам Сети. Через него можно получить доступ

практически к любой службе Интернст, даже если браузер не поддерживает работу с этой службой. Для этого используются специальным образом запрограммированные веб-сервсра, которые связывают Всемирную паутину с данной службой Сети. Примером Такого рода веб-серверов являются многочисленные бесплатные почтовые сервера с веб-интерфейсом

Навигация. Работа с браузером начинается с того, что пользователь набирает в адресной строке (Адрес) ***URL*** того ресурса, к которому он хочет получить доступ, и нажимает клавишу ***Enter***

Процесс загрузки Web -страницы происходит в несколько этапов, различающихся тем, что происходит в строке состояния:

• в системе DNS выполняется поиск IP — адреса узла по его доменному имени (Поиск узла...);

• устанавливается соединение с сервером (Узел найден. Устанавливается соединение. Ожидается ответ...);

• отправляется запрос на получение файла (Начало загрузки...);

• производится прием файла (Открытие страницы...). Содержащиеся в полученной веб-странице текстовые гиперссылки, как правило, выделяются цветом, отличным от цвета остального текста документа, и подчеркиваются. Ссылки, указывающие на ресурсы, которые пользователь еще не просматривал, и ссылки на уже посещенные ресурсы обычно имеют разный цвет. Изображения также могут функционировать как гиперссылки. Независимо от того, текстовая ссылка или графическая, если навести на нее курсор мыши, его форма изменится. Одновременно в статусной строке браузера появится адрес, на который указывает ссылка.

При нажатии на гиперссылку браузер открывает в рабочем окне ресурс, на который она указывает, при этом предыдущий ресурс из него выгружается. Браузер ведет список просматриваемых страниц и пользователь при необходимости может вернуться назад по цепочке просмотренных страниц. Для этого нужно щелкнуть мышкой на кнопке ***Назад (ВасК)*** **в** меню браузера, — и он вернется к странице, которую вы просматривали до того, как открыли текущий документ.

Каждый раз, когда вы будете нажимать на эту кнопку, браузер будет возвращаться на один документ назад в списке посещенных документов. Если вдруг вы вернулись слишком далеко назад, воспользуйтесь кнопкой ***Вперед (Forward*** меню браузера. Она поможет вам переместиться вперед по списку документов.

Работа с документом. Браузер позволяет производить над документом набор стандартных операций. Загруженную в него веб-страницу

можно распечатать (кнопка ***Печать (Print)*** или из меню ***Файл => Печать***), сохранить на диск ***(Файл => Сохранитъ***

Можно найти интересующий вас фрагмент текста в загруженной странице. Для этого используйте *Правка => Найти на этой странице....* А если вас интересует, как выглядит данный документ в исходном гипертексте, который обработал браузер, то выберите *Вид => просмотр HTML — кода.* Когда в процессе работы в Интернете пользователь находит особенно интересную для него страницу, он использует предусмотренную в браузерах возможность устанавливать закладки (по аналогии с закладками, отмечающими интересные места книги). Для этого необходимо выполнить: *Избранное => Добавить в избранное.* После этого новая закладка появляется в списке закладок, который можно просмотреть, нажав кнопку *Избранное* или через меню*Избранное.*

Существующие закладки можно удалять, изменять, организовывать в папки с помощью команды *Избранное =>Упорядочить избранное.*

Браузеры работают на компьютерах под управлением самых разных операционных систем. Это дает основание для того, чтобы говорить о независимости Всемирной паутины от типа применяемого пользователем компьютера и операционной системы.

Поиск информации в Интернете. Современная Сеть в состоянии предложить своему пользователю массу информации самого разного профиля. Здесь можно познакомиться с новостями, интересно провести время, получить доступ к разнообразной справочной, энциклопедической и учебной информации.

Однако у обилия информации есть и отрицательная сторона: с ростом количества информации становится все труднее и труднее найти ту информацию, которая нужна в данный момент. Поэтому самая главная проблема, возникающая при работе с Сетью - быстро найти нужную информацию и разобраться в ней, оценить информационную ценность того или иного ресурса для своих целей.

Для решения проблемы поиска нужной информации в Интернете существует отдельный вид сетевого сервиса. Речь идет о поисковых серверах или поисковых машинах.

Поисковые серверы достаточно многочисленны и разнообразны. Принято различать поисковые *индексы* и *каталоги. Серверы-индексы* работают следующим браузера : регулярно прочитывают содержание большинства веб-страниц Сети ("индексируют" их) и помещают их полностью или частичную общую базу данных. Пользователи поискового сервера имеют возможность осуществлять поиск по этой базе данных, используя ключевые слова, относящиеся к интересующей их теме. Выдача результатов поиска обычно состоит из выдержек рекомендуемых вниманию пользователя страницы их адресов, оформленных в ввде гиперссылок. Работать с поисковыми серверами этого типа удобно в том случае, если

имеется четкое представление о предмете поиска. ***Серверы-каталоги*** представляют собой многоуровневую классификацию ссылок, построенную по принципу "от общего к частому". Иногда ссылки сопровождаются кратким описанием ресурса. Как правило, возможен поиск в названиях рубрик (категориях) и описаниях ресурсов по ключевым словам. Каталогами пользуются тогда, когда не вполне четко знают, что именно ищут. Переходя от самых общих категорий к более частым, можно определить, с каким именно ресурсом Сети следует ознакомиться. Поисковые каталоги уместно сравнивать с тематическими библиотечными каталогами или классификаторами. Ведение поисковых каталогов частично автоматизировано, но до сих пор классификация ресурсов осуществляется главным образом вручную.

Поисковые каталога бывают общего назначения и специализированные. Поисковые каталоги общего назначения включают в себя ресурсы самого разного профиля. Специализированные каталоги объединяют только ресурсы, посвященные определенной тематике. Им часто удается достичь лучшего охвата ресурсов из своей области и построить более адекватную публикацию.

2. Введение в HTML

Для создания Web-страницы можно воспользоваться специальными программами редактирования документов Всемирной паутины. Другой способ подготовки Web-страниц заключается в «ручном» создании кода документов на языке HTML – HyperText Markup Language – Язык разметки гипертекста. Данный язык представляет собой довольно простой набор команд, описывающий структуру документа. Язык HTML позволяет выделить в документе отдельные элементы – заголовки, абзацы, таблицы и т.д. Файлы с текстом кода на языке HTML имеют расширение .html или .htm.

Элементы разметки состоят из заключённых в угловые скобки (< и >) дескрипторов — тэгов(tags) и их атрибутов. Совокупность открывающего (< >) и закрывающего (< / >) дескрипторов есть контейнер. Элементы HTML подразделяются на структурные, которые организуют текст и на форматирующие, которые задают его стиль.

Структура Web-страницы

Структура HTML-документа позволяет задействовать вложенные друг в друга контейнеры. Собственно, сам документ — это один большой контейнер, который начинается с тега <HTML> и заканчивается тегом </HTML>. Он указывает браузеру, что данный текст представляет собой HTML-документ и, содержит в себе теги, которые браузер должен выявить, распознать и правильно интерпретировать.

Типичная Интернет-страница состоит из двух частей: головная часть (HEAD) и тела (BODY). Эту базовую структуру в простейшем виде можно представить следующим образом:

<HTML>	Начало HTML-документа
<HEAD>	Начало головной части
<TITLE>	Начало строки названия страницы
…	Строка названия страницы
</TITLE>	Конец строки названия страницы
</HEAD>	Конец головной части
<BODY>	Начало тела документа
…	
</BODY>	Конец тела документа
</HTML>	Конец HTML-документа

Элемент <HEAD>

Область, обозначаемая тэгами <HEAD> и </HEAD> служит только для формирования общей структуры документа, задавая его глобальные свойства. Информация, находящаяся в этом разделе документа, является служебной и необходима программе-браузеру пользователя. Она допускает вложение между дескрипторами следующих элементов: <TITLE>,<BASE>,<ISINDEX>,<LINK>,<META>,<STYLE>.

Элемент <BODY>

Элемент <BODY> предназначается для выделения той части документа, которая будет визуализирована для пользователя. Он имеет как начальный, так и конечный теги. Начальный тег <BODY> может иметь несколько атрибутов.

Вложенные атрибуты элемента <BODY>

BACKGROUND (задает графическое изображение для фона документа)

<BODY BACKGROUND="(URL)(путь) имя файла">

BGCOLOR (задает цвет фона документа)

<BODY BGCOLOR="#ff0000"> или <BODY BGCOLOR="RED">

TEXT (задает используемый по умолчанию цвет текста)

<BODY TEXT=" цвет ">

LINK (задает цвет гиперссылки)

<BODY LINK="цвет" >

ALINK (задает цвет активной гиперссылки)

<BODY ALINK="цвет" >

VLINK (задает цвет посещенной гиперссылки)

<BODY VLINK="цвет" >

BGPROPERTIES (задает свойства фонового изображения, в данный момент браузерами поддерживается единственное его значение fixed, запрещающее скроллинг изображения)

<BODY BGPROPERTIES="fixed" >

TOPMARGIN (задает верхнюю границу страницы в пикселях) <BODY TOPMARGIN=число >

BOTTOMMARGIN (задает нижнюю границу страницы в пикселях) <BODY BOTTOMMARGIN=число >

LEFTMARGIN (задает границу страницы в пикселях слева) <BODY LEFTMARGIN=число >

RIGHTMARGIN (задает границу страницы в пикселях справа) <BODY RIGHTMARGIN=число >

Форматирование текста

Текст – единственный объект Web-страницы, который не требует специального определения. Иными словами, произвольные символы интерпретируются по молчанию как текстовые данные. Но для форматирования текста существует большое количество элементов: выделение фрагментов курсивом, полужирным, выбирать шрифт и т.д.

1. Структурное форматирование.

Комментарии <COMMENT >

<COMMENT> Текст комментария </COMMENT>

Шесть уровней заголовков <Hn>

<Hn align=отступ> Текст заголовка </Hn>

Разделительные линии <HR>

Используется для проведения горизонтальной черты в документе, он может иметь атрибуты : color, задающий цвет линии, size высота в пикселях, width ширина в пикселях или процентах от ширины экрана, align режим выравнивания, и не имеет конечного тега.

<HR align="center" size=n width=n color="цвет">

Элемент <P>

Этот элемент задает один из способов разбиения текста на абзацы. Он может иметь вложенный атрибут align, который указывает отступ left, center или right. Каждый следу-

ющий абзац игнорирует, заданное для предыдущего абзаца значение align.

<P align=отступ> Текст абзаца </P>

**Элемент
**

Этот элемент задает разрыв текста с переходом на новую строку.

<BR clear=обтекание> Текст

Элемент <PRE>

Весь текст, заключенный в тэги <PRE> и </PRE> будет визуализирован браузером точно так, как он визуализирован в исходном коде документа, кроме того, текст выводится моноширинным шрифтом.

<PRE width=число символов >...текст.. .</PRE>

Элемент <DIV>

Элемент <DIV> позволяет выделить в структуре документа несколько разделов. Он является блочным элементом, функционирующим во многом подобно элементу <P>.

<DIV align=отступ> Текст раздела </DIV>

Элемент <ADDRESS>

Элемент <ADDRESS> используется для оформления контактной информации текущего документа, будь то адрес электронной почты или полный почтовый адрес с номером телефона.

ADDRESS>контактная информация </ADDRESS>

Элемент <BLOCKQUOTE>

Элемент <BLOCKQUOTE> позволяет выделить обьемный текст-цитату из общего текста.

<BLOCKQUOTE> Текст </BLOCKQUOTE>

2. Форматирование символов.

Элемент <I>

Используется с целью выделения курсивным шрифтом слова или текста.

<I> Текст </I>

Элемент <B>

Используется с целью выделения жирным шрифтом слова или текста.

<B> Текст </B>

Элемент <U>

Используется с целью выделения подчеркнутым шрифтом слова или текста.

<U> Текст </U>

Элемент <SUP>

Используется с целью выделения особым шрифтом слова или текста.

^{Текст}

Элемент <SUB>

Используется с целью выделения особым шрифтом слова или текста.

_{Текст}

Элемент <BIG>

Элемент <BIG> используется с целью выделения особым шрифтом слова или текста относительно основного текста.

<BIG> Текст </BIG>

Элемент <SMALL>

Элемент <SMALL> используется с целью выделения особым шрифтом слова или текста относительно основного текста.

<SMALL> Текст </SMALL>

Элемент <FONT>

Элемент <FONT> используется с целью выделения особым шрифтом слова или текста. С ним применяются два атрибута size и color. Некоторые браузеры поддерживают атрибут face, позволяющий задать любой из перечня шрифтов

<FONT size=n color="цвет"> Текст </FONT>

или <FONT face=имя color="цвет"> Текст </FONT>

Создание списков.

Элемент <OL>

Элемент <OL> используется с целью задания нумерованных списков, имеет атрибуты type=1, или A, или a, или I, или i для задания вида нумерации и start для указания, с какого индекса начинается нумерация списка. Элемент <OL> включает в себя дополнительный элемент <LI>, который задает элементы списка.

<OL type=1 start=1 >

<LI> элемент списка

<LI> элемент списка

</OL>

Элемент <UL>

Элемент <UL>, по сути, является аналогом <OL> без дополнительных элементов <LI>, он используется с целью задания маркированных списков, имеет атрибут

type=circle, square, или disc для задания вида маркера. Элемент <UL> включает в себя дополнительный элемент <LI>, который задает элементы списка.

<UL type=circle >

<LI> элемент списка

<LI> элемент списка

</UL>

Элемент <DL>

Элемент <DL>используется с целью задания словарей, глоссариев и прочих перечней. Элемент <DL> включает в себя дополнительные элементы <DT> и <DD>, которые обозначают соответственно термин и определение.

<DL >

<DT> термин 1

<DD>определение 1

<DT> термин 2

<DD>определение 2

</DL>

Практическая работа № 1

Создание простейшей HTML-страницы. Форматирование текста

1. Цель работы. Знакомство с основными HTML-тегами, получение навыков создания простейшей HTML-страницы. Изучить основные теги оформления текста и использовать их при создании web-страниц.

2. Литература. Конспект лекций.

3. Подготовка к работе. Изучить конспект лекций по теме "Введение в HTML. Форматирование текста".

4. Перечень оборудования. Компьютер.

5. Задание.

5.1. Ознакомиться со структурой HTML-документа и основными HTML-тегами.

5.2. Научиться создавать простейшие HTML-страницы.

5.3. Изучить основные теги оформления текста.

5.4. Научиться оформлять тексты для web-страниц.

6. Порядок выполнения работы.

6.1. **Создание пустой HTML-страницы.**

1**.** HTML - это язык тегов. Под тегами понимаются специальные управляющие коды, записываемые в тексте в угловых скобках. HTML -теги бывают парными. Первый тег указывает начало форматируемого текста, а второй, отличающийся от первого косой чертой "/" перед ключевым словом, указывает конец фрагмента текста. Такая пара тегов называется *контейнером.* Например, тег <html> начинает код всей web-страницы, а тег </html> - заканчивает его.

2. На диске С создайте вашу папку **Фамилия_Имя_Номер группы**.

3. Откройте текстовый редактор Блокнот и наберите текст HTML-кода (без комментария справа), который соответствует пустой странице:

<html>	начало HTML-документа
<head>	начало титульной части
<title> </title>	название документа
</head>	конец титульной части
<body>	начало тела документа
</body>	конец тела документа
</html>	**к**онец HTML-документа

4. Сохраните полученный документ в вашей папке, присвоив файлу имя **stih.html**. Здесь .html (либо .htm) – типовое расширение имени файла с HTML-кодом web-страницы.

5. Запустите браузер Internet Explorer. Просмотрите в нем полученную пустую страницу. Для этого откройте файл **stih.html**.

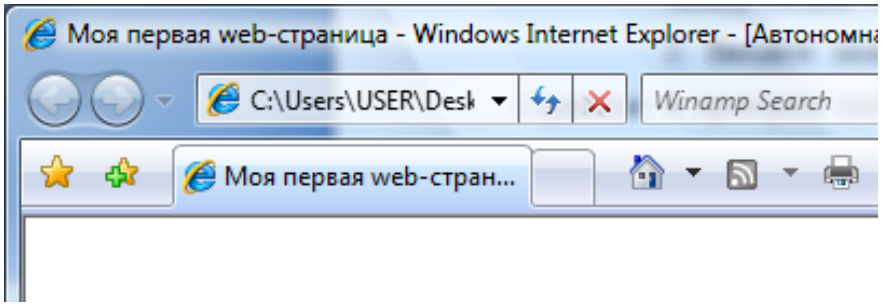
6.2. **Заголовок web-страницы. Использование тега <meta>.**

1. Откройте файл **stih.html.** С помощью команд **Вид → Просмотр HTML-кода** откройте окно редактора **Блокнот** для внесения изменений в HTML-код.

В контейнере **<head> </head>** описываются общие правила отображения HTML-документа и вспомогательная информация.

Между тегами **<title>** и **</title>** располагается заголовок web-страницы.

2. Введите между тегами **<title>** и **</title>** название «Моя первая web-страница» и проверьте отображение имени документа на полученной web-странице. Для этого сохраните изменения в HTML-файле и в браузере обновите web-страницу.



3. Чтобы созданный вами сайт могли легко найти поисковые системы (по ключевому слову, описанию, имени автора и.т.д.), в контейнере **<head>…</head>** размещается тег **<meta>,** в качестве значения параметра **name** которого указывается имя некоторого свойства, а в качестве значения параметра **content** – значение этого свойства:

- **Keywords**(ключевые слова) – слова, которые будут использованы для поиска вашей web-страницы;
- **Description**(описание) – описание вашей web-страницы;
- **Author**(имя автора web-страницы) – ваше имя.

4. Откройте файл **stih.html.** С помощью команд **Вид → Просмотр HTML-кода** откройте окно редактора **Блокнот** для внесения изменений в HTML-код.

5. Введите между тегами **<head>** и **</head>** следующий текст:

<meta name = «keywords» content = «Стихотворение»>

<meta name = «description» content = «Стихи Александра Сергеевича Пушкина >

< meta name = «author» content = «Ваше имя и фамилия»>

6. Сохраните изменения в HTML-файле.

6.3. **Тег <body>.**

1. Между тегами **<body> и </body>** располагаются отображаемый на web-странице текст и вставленные в него HTML-команды, согласно которым браузер выводит информацию в окне браузера.

2. Откройте файл **stih.html.** С помощью команд **Вид → Просмотр HTML-кода** откройте окно редактора **Блокнот** для внесения изменений в HTML-код.

3. Введите между тегами **<body>** и **</body>** следующий текст:

Зимнее утро

Мороз и солнце: день чудесный;

Еще ты дремлешь, друг прелестный,

Пора, красавица, проснись:

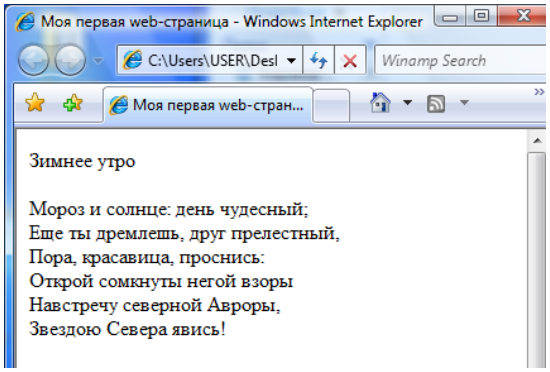
Открой сомкнуты негой взоры

Навстречу северной Авроры,

Звездою Севера явись!

А.С.Пушкин

4. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.



6.4**. Заголовки и подзаголовки.**

1. С помощью команд **Вид → Просмотр HTML-кода** откройте окно редактора **Блокнот** для внесения изменений в HTML-код.

2. Установите различные заголовки для стихотворения. Заголовок любого уровня может быть представлен в общем случае так:

<Hn>текст заголовка</Hn>

3. Для названия стихотворения установите заголовок первого уровня:

<H1>Зимнее утро</H1>

4. Для куплета установите заголовок третьего уровня, а для имени автора заголовок второго уровня.

5. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

6.5. **Выравнивание абзацев.**

1. Внесите изменения в выравнивание абзацев (по центру, по левому краю, по правому краю). Для этого используйте атрибут **align:**

<H1 align=center>Зимнее утро</H1 > - центрирование заголовка

2. Куплет стихотворения выровните по левому краю, а имя автора по правому. Для этого используйте атрибуты **align=right** и **align=left.**

3. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

6.6. **Задание начертания.**

1. Внесите изменения в начертание абзацев (полужирный, курсив). Для этого используйте следующие теги:

текст - полужирное начертание текста;

текст - курсивное начертание текста;

<u>текст</u> - подчеркнутое начертание текста.

2. Для названия стихотворения установите полужирное начертание, для куплета курсивное начертание, а для имени автора – подчеркнутое. Например,

<H1 align=center>Зимнее утро</H1>

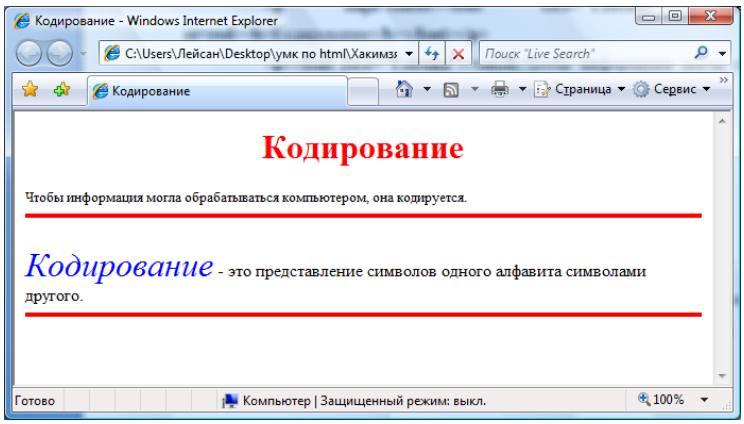
3. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

6.7. **Изменение параметров шрифта.**

1. Откройте текстовый редактор Блокнот и наберите текст HTML-кода:

```
<html>
   <head>
     <title>Кодирование</title>
   </head>
   <body>
<p align=center><font face=”Tahoma” size=6 color=red><b>Кодирование</b></font></p>
<p><font face=”Tahoma”><small>Чтобы информация могла обрабатываться компьютером, она кодируется.</small>
<hr color=red size=4>
<p><font size=6 color=blue><i>Кодирование</i></font> - это представление символов одного алфавита символами другого.
<hr color=red size=4>
   </body>
</html>
```

2. Сохраните полученный документ в вашей папке, присвоив файлу имя **kod.html**. Затем откройте этот файл с помощью браузера Internet Explorer.



3. Далее откройте HTML-код этого файла. Введите текст:

Для кодирования используется двоичная система счисления, в алфавите которой только два символа: 0 и 1. Символ 1 означает наличие сигнала, 0 - его отсутствие.

Один двоичный символ получил название бит, от английской аббревиатуры bit (binary digit), что означает «двоичная цифра».

4. Последнее предложение выровните по центру, установите полужирное начертание (для слова **bit (binary digit) –** курсивное), шрифт – Tahoma, размер шрифта – 4.

5. Далее введите таким же образом следующий текст:

Любой

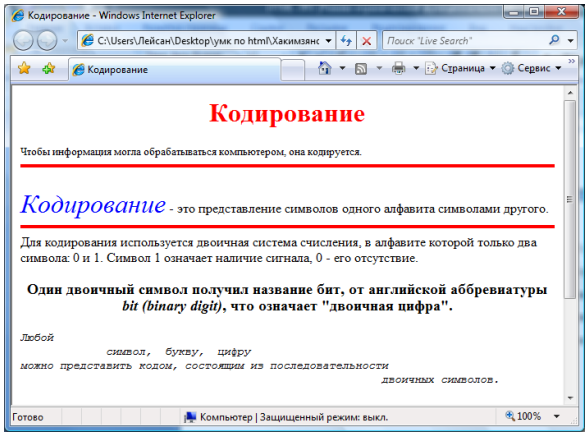
символ, букву, цифру

можно представить кодом, состоящим из последовательности

двоичных символов.

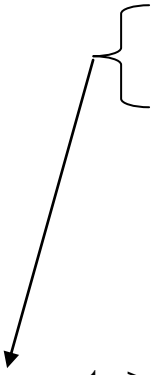
Для этого предложения установите курсивное начертание, а также теги **<pre>** и **</pre>,** которые выводят текст на экран так, как он записан в HTML-коде.

6. Сохраните изменения в HTML-файле и в браузере обновите web-страницу. Ваша web-страница должна выглядить следущим образом:



7. Добавьте текст в файл и отформатируйте его в соответствии с образцом (для надстрочного индекса используется пара тегов **[…]**).

Для определения количества информации введены следующие единицы измерения:



1 бит=**1 разряд** (может принимать значение 0 или 1)
1 байт=**8 бит**
1 кбайт=**1024 байта**(2^{10})*байт*
1 Мбайт=**1024 кбайта**(2^{20})*байт*
1 Гбайт=**1024 Мбайта**(2^{30})*байт*
1 Тбайт=**1024 Гбайта**(2^{40})*байт*

Фрагмент HTML-кода:

```
<p align=center><font color="blue">1 бит</font>= <b> 1 разряд </b> (может принимать значение
<font color="red">0 или 1</font>)<br>
<font color="blue">1 байт</font>=<b>8 бит</b><br>
<font color="blue">1 кбайт</font>=<b>1024 байта</b>
(2<sup>10</sup>)<i>байт</i><br>
```

8. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

6.8**. Списки. Нумерованный список.**

1. Откройте текстовый редактор Блокнот и наберите текст HTML-кода:

```
<html>
  <head>
    <title>Списки</title>
  </head>
  <body>
  </body>
</html>
```

2. Между тегами **<body>** и **</body>** введите следующий текст:

«В основу построения большинства ЭВМ положены принципы, сформулированные в 1945г. Джоном фон Нейманом

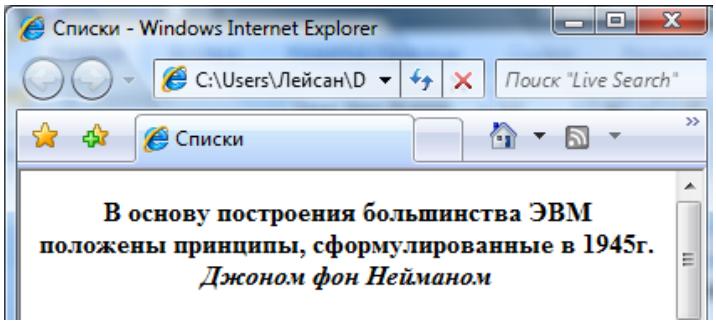
Принцип программного управления (работой компьютера управляет программа, состоящая из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности)

Принцип однородности памяти (программы и данные хранятся в одной и той же памяти; над командами можно выполнять такие же действия, как и над данными)

Принцип адресности (структура основной памяти: перенумерованные ячейки, номер ячейки - это ее адрес)

ЭВМ, построенные на этих принципах, имеют классическую архитектуру.»

3. Первое предложение выровните по центру, установите полужирное начертание, для некоторых слов установите курсивное начертание в соответствии с образцом.



4. Оставшийся фрагмент текста отформатировать в соответствии с образцом, применяя нумерованный список.

Для задания нумерованного списка используется пара тегов **
** и **.**

<ol>

<li>*первый элемент списка*

<li> *второй элемент списка*

…

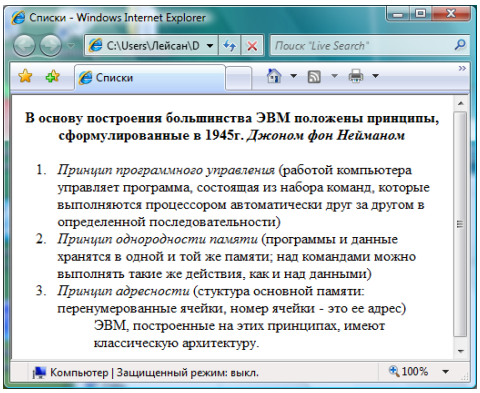
<li> *последний элемент списка*

**
**

Тег **<li>** - элемент списка

1. *Принцип программного управления* (работой компьютера управляет программа, состоящая из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности)
2. *Принцип однородности памяти* (программы и данные хранятся в одной и той же памяти; над командами можно выполнять такие же действия, как и над данными)
3. *Принцип адресности* (стуктура основной памяти: перенумерованные ячейки, номер ячейки - это ее адрес)

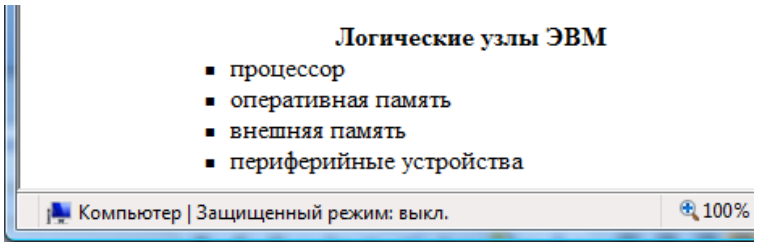
5. Сохраните полученный документ в вашей папке, присвоив файлу имя **spiski.html**. Затем откройте этот файл с помощью браузера Internet Explorer.



6.9. **Списки. Маркированный список.**

1. Откройте файл **spiski.html**. С помощью команд **Вид → Просмотр HTML-кода** откройте окно редактора **Блокнот** для внесения изменений в HTML-код.

2. Добавьте в него текст и отформатируйте его в соответствии с образцом (применяя маркированный список).



Для задания маркированного списка используется пара тегов **
** и **.**

<ul>

<li>*первый элемент списка*

<li> *второй элемент списка*

…

<li> *последний элемент списка*

**
**

Тип маркера – квадрат (**type=square)**

3. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

6.10. **Списки. Многоуровневый список.**

1. Откройте файл **spiski.html**. С помощью команд **Вид → Просмотр HTML-кода** откройте окно редактора **Блокнот** для внесения изменений в HTML-код.

2. Добавьте в него текст и отформатируйте его в соответствии с образцом (применяя маркированный список).

Память делят на:

основную

ОЗУ (оперативное запоминающее устройство, по английски RAM - Random Access Memory, в переводе - "память с произвольным доступом");

ПЗУ (постоянное запоминающее устройство, по английски ROM - Read Only Memory, в переводе - "память доступная только для чтения");

внешнюю (устройства внешней памяти позволяют длительно хранить информацию).

Носители внешней памяти: жесткие и гибкие магнитные диски, а также лазерные диски, флеш.

3. Фрагмент HTML-кода:

<dt>основную****

<dd>ОЗУ (оперативное запоминающее устройство, по английски RAM - Random Access Memory, в переводе - "память с произвольным доступом");

<dd>ПЗУ (постоянное запоминающее устройство, по английски ROM - Read Only Memory, в переводе - "память доступная только для чтения");

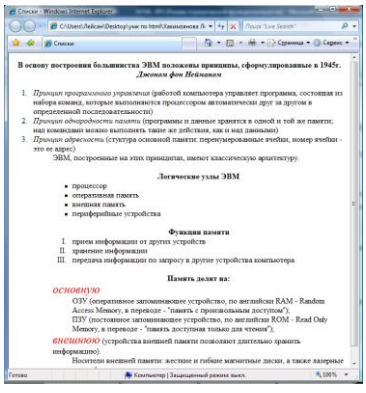
<**dt>**внешнюю****

(устройства внешней памяти позволяют длительно хранить информацию).

<dd>Носители внешней памяти: жесткие и гибкие магнитные диски, а также лазерные диски,флеш.

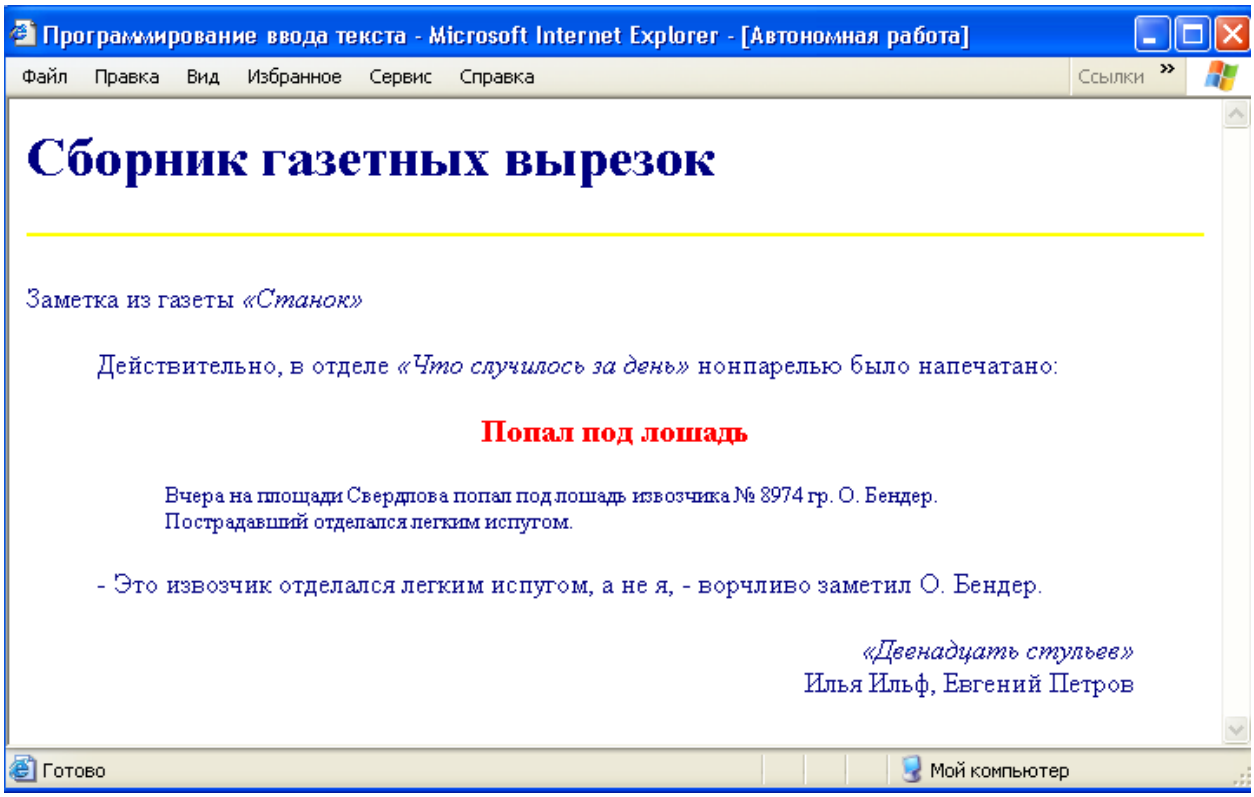
</dl>

4. Сохраните изменения в HTML-файле и в браузере обновите web-страницу. Ваша web-страница должна выглядить следущим образом:



7. Самостоятельная работа.

7.1. Создайте страничку в соответствии с образцом:



Сохраните полученный документ в вашей папке, присвоив файлу имя **sam.html.**

8. Содержание отчета.

8.1. Файлы **stih.html, kod.html** и **spiski.html, sam.html** в вашей папке.

9. Контрольные вопросы.

9.1. Что такое язык HTML? Для чего он нужен?

9.2. Из каких двух основных частей состоит любой HTML-документ?

9.3. Что такое тег? Что такое контейнер? Чем HTML-теги отличаются от остального текста HTML-документа?

9.4. Какие теги описывают общие правила отображения HTML-документа и содержат дополнительную информацию о нем?

9.5. Между какими тегами располагаются команды, согласно которым браузер выводит текст в своем окне?

9.6.Между какими тегами располагается имя HTML-документа?

9.7. Между какими тегами задается метаинформация?

9.8. Основное назначение горизонтальной линии. Какой тег создает эту линию?

9.9. Какими бывают заголовки по значению или по уровню?

9.10. Какой параметр определяет выравнивание абзаца? Какие возможны способы выравнивания?

9.11. Перечислить параметры тега <font>, позволяющие менять размер, цвет и гарнитуру шрифта.

9.12. С помощью какого тега можно задать нижний (верхний) индекс?

9.13. Какой тег позволяет создать маркированный список? Типы маркеров для маркированного списка.

9.14. Какой тег позволяет создать нумерованный список? Типы нумерации для нумерованного списка.

9.15. Сколько уровней может иметь вложенный список?

3. Графика

В Интернете наиболее популярны два графических формата: GIF и JPEG.

Обычно используют:

JPEG – для фотографий и очень сложных по цветовой гамме рисунков с плавными цветовыми переходами (в расширении записывается как JPG).

GIF – для логотипов, надписей, заголовков, рисунков, имеющих четкие цветовые граница.

Формат GIF

GIF-формат имеет три дополнительные возможности:

1. Мультипликация.
2. Прозрачная графика. GIF-формат позволяет один или несколько цветов в картинке объявить прозрачными. Это помогает избавиться от строго прямоугольных иллюстраций и вписывать рисунок в документ более привлекательным образом.
3. Чересстрочная развертка. Применяется для больших GIF. Иллюстрация разделяется на четыре части, размером с оригинальную картинку.

Формат JPEG

Этот формат был разработан специально для передачи фотографий. Он поддерживает миллионы цветов и позволяет получать изображения очень высокого качества.

Элемент <IMG>

Используется для вставки в тело документа графического изображения. Под графическим изображением подразумевают: маленькие пиктограммы, рисунки, графические объекты и карты изображений, занимающие большую часть окна браузера. Кроме того,

элемент <IMG> поддерживает различные атрибуты, определяющие расположение изображения относительно окружающего текста и содержания Web страницы в целом. Изображение может выравниваться по левой, правой, верхней или нижней границе строки или располагаться в центре окна.

<IMG src= URL атрибуты >

Атрибуты тега <IMG>

Src (указывает на файл графики, задавая его URL)

<IMG src=" URL ">

Alt (задает альтернативный текст типа здесь нарисован логотип фирмы)

<IMG alt=" описание ">

Align (задает расположение рисунка в окне и его выравнивание)

<IMG align=" значение ">

Width (задает ширину области в пикселях, отводимой в окне под изображение)

<IMG wigth= число >

Height (задает высоту области в пикселях, отводимой в окне под изображение)

<IMG height= число >

Hspace (задает пустое пространство в пикселях справа и слева от рисунка)

<IMG hspace= число >

Border (задает в пикселях толщину рамки вокруг рисунка)

<IMG border= число >

Vspace (задает пустое пространство в пикселях сверху и снизу от рисунка)

<IMG vspace= число >

Ismap (разрешает использовать изображения, отдельные части которого связаны со ссылками и позволяют выполнять переходы)

<A href=" URL карты "><IMG src=" URL изображения " ismap>

Usemap (разрешает использовать изображения, отдельные части которого связаны с картами, они используются совместно с элементом <MAP>)

<IMG src=" URL изображения " usemap= "URL" карты(#имя карты) >

Элемент <MAP>

Элемент <MAP> применяется для представления графического изображения в виде карты с активными областями. Он поддерживает атрибут name, который задает его имя, и включает внутри себя элемент <AREA>, который и задает собственно активные области карты, связанные ссылками с прочими ресурсами.

<MAP name=" имя "> <AREA атрибуты > </MAP>

Элемент <AREA>

Элемент <AREA> задает активные области карты, щелчком по которым можно осуществить ссылку. Элемент <AREA> поддерживает различные атрибуты:

Href (атрибут указывает URL ссылки)

<AREA href=" URL ">

Alt (задает альтернативный текст для браузеров, которые не поддерживают данный элемент)

<AREA alt=" текст подсказки ">

Title (задает альтернативный текст для браузеров, который всплывает при наведении курсора на данный элемент)

<AREA title=" текст подсказки ">

Shape (задает форму активной области на карте и её координаты, он может принимать значения: "circle" coords=X,Y,R, где X,Y,R - координаты центра круга и его радиус, "poly" coords=X1,Y1,X2,Y2,X3,Y3..., гдеX1,Y1,X2,Y2,X3,Y3... - координаты вершин многоугольника, если многоугольник - прямоугольник, то достаточно указать его верхнюю левую и правую нижнюю вершины "rect" coords=X1,Y1,X3,Y3)

<AREA " circle " coords= X,Y,R >

Пример изображения - карты:

<IMG src=" map.gif " usemap="# supermap " border= 0 >

<MAP name=" supermap ">

<AREA shape= circle coords=" 34,32,23 " href=" page1.html " title=" ссылка 1 ">

<AREA shape= poly coords=" 12,110,37,62,72,114 " href=" page2.html " title=" ссылка 2 ">

<AREA shape= rect coords=" 83,44,133,94 " href=" page3.html " title=" ссылка 3 ">

</MAP>

Элемент <MARQUEE>

Элемент <MARQUEE> используется с целью создания в документе бегущей строки.

<MARQUEE атрибуты> Текст строки </MARQUEE>

Он может иметь атрибуты:

Bgcolor (задает цвет фона бегущей строки)

<MARQUEE bgcolor="цвет">

Height (задает высоту бегущей строки в пикселях или процентах от всего окна)

<MARQUEE height=число>

Align (задает выравнивание бегущей строки по верхнему краю top, по середине middle или по нижнему краю bottom)

<MARQUEE align= ...>

Direction (задает направление движения бегущей строки: left влево, right вправо, up вверх, down вниз)

<MARQUEE direction="...">

Loop (задает количество проходов бегущей строки)

<MARQUEE loop=число>

Scrollamount (задает скорость движения бегущей строки, если его значение равно 1, то она будет еле ползти, если его значение больше 10, то она будет двигаться очень быстро)

<MARQUEE scrollamount=число>

Практическая работа № 2

Вставка изображений в HTML-документ

1. Цель работы. Знакомство с основными тегами для вставки изображений в HTML-документ, изучение атрибутов для преобразования изображений в документе.

2. Литература. Конспект лекций.

3. Подготовка к работе. Изучить конспект лекций по теме "Графика".

4. Перечень оборудования. Компьютер.

5. Задание.

5.1. Ознакомиться с основными тегами для вставки изображений в HTML-документ.

5.2. Изучить основные атрибуты для преобразования изображений.

5.4. Научиться оформлять изображения в web-страницах.

6. Порядок выполнения работы.

6.1. **Вставка изображений на web-страницу.**

1.Откройте программу Paint и создайте в нем три рисунка по образцу: каждый рисунок – в отдельном файле.

Рис.1

Рис.2

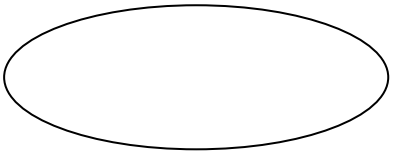
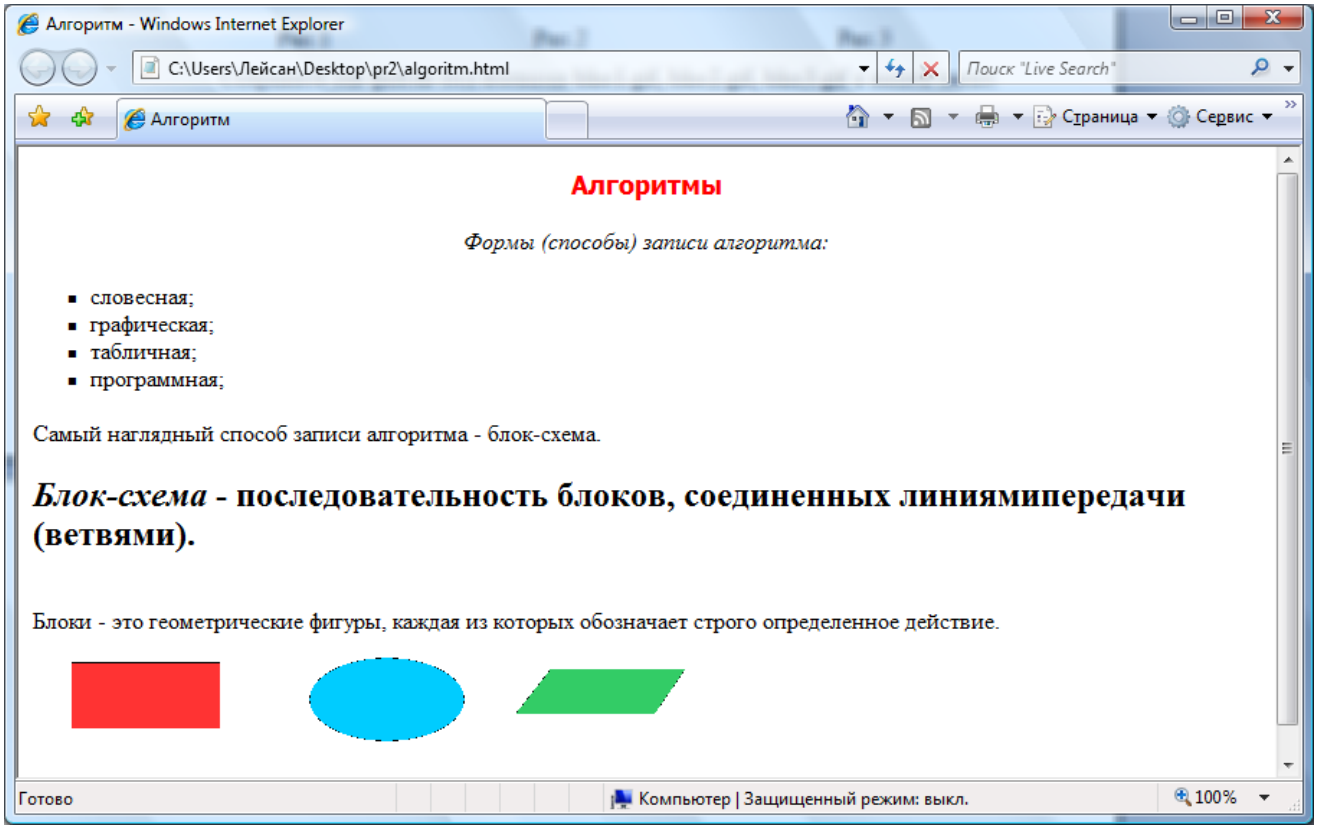


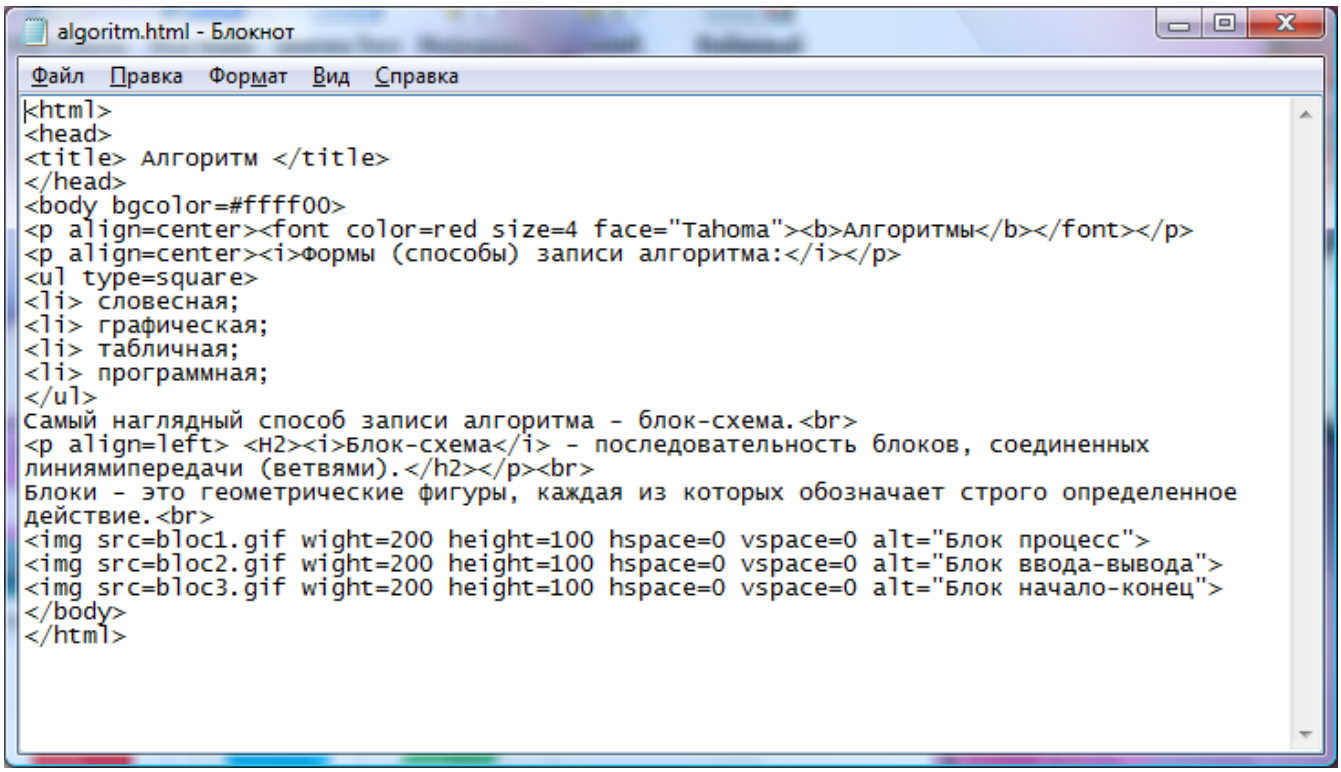
Рис.3

Сохраните эти файлы под именами **bloc1.gif, bloc2.gif, bloc3.gif** в вашей папке.

2. Откройте текстовый редактор Блокнот и создайте web-страницу в соответствии с образцом:



Код web-страницы:

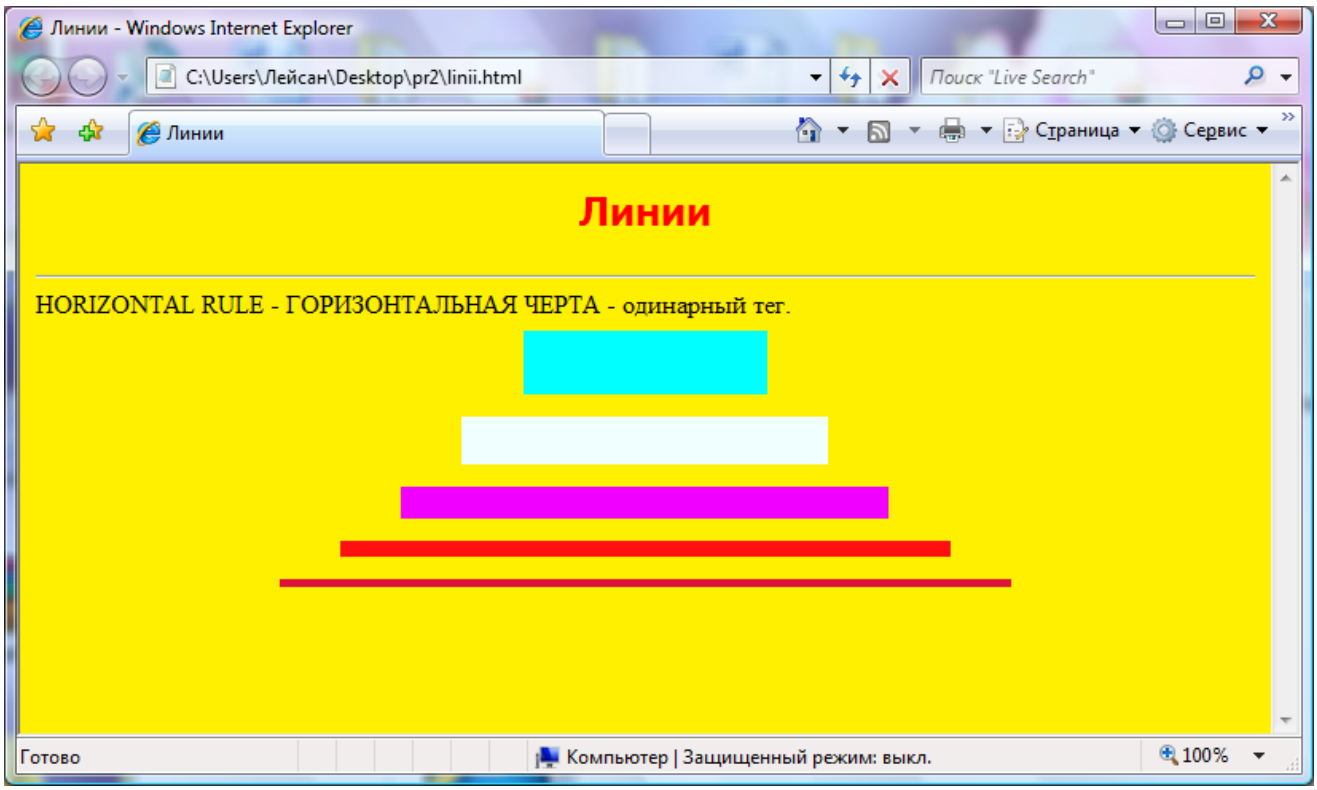


3. Сохраните полученный документ в вашей папке, присвоив файлу имя **algoritm.html**.

4. Запустите браузер Internet Explorer. Просмотрите в нем полученную web-страницу.

6.2. **Создание горизонтальных линий и других объектов.**

1. Откройте текстовый редактор Блокнот и создайте 2 web-страницы в соответствии с образцом:



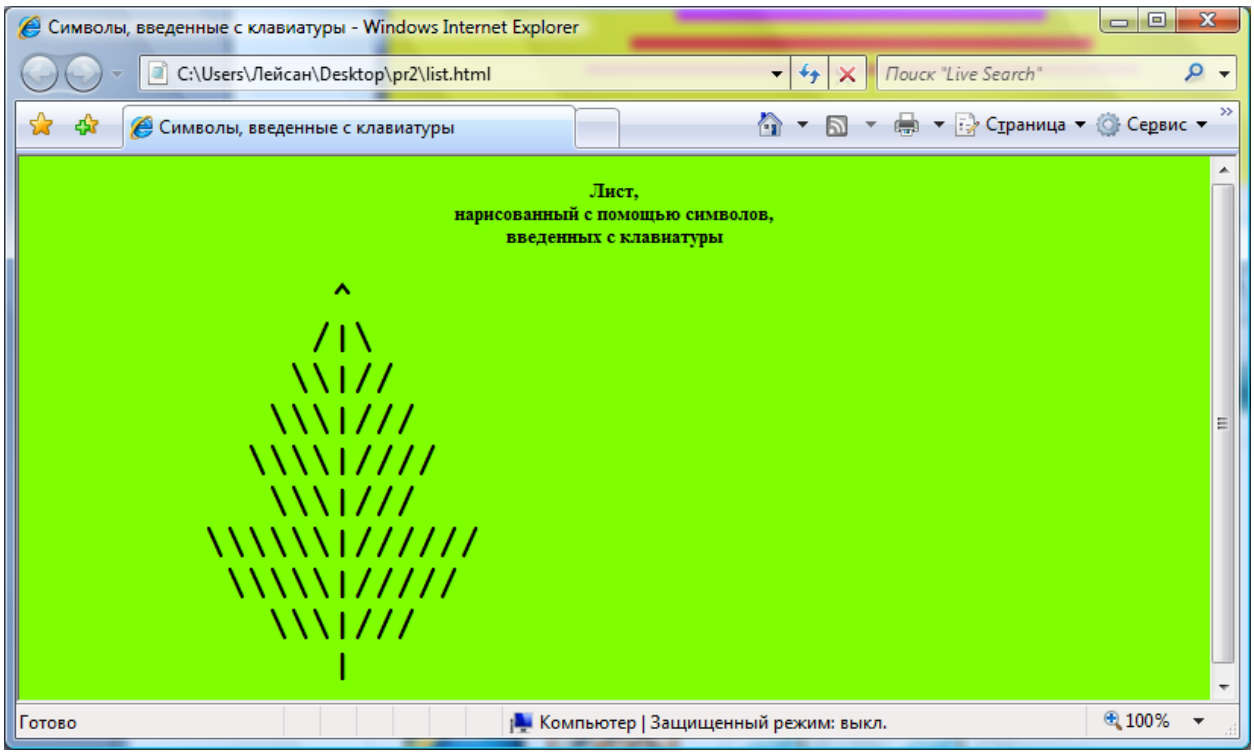
Фрагмент кода web-страницы:

```
<HR>
HORIZONTAL RULE - ГОРИЗОНТАЛЬНАЯ ЧЕРТА - одинарный тег.
<hr width="20%" noshade size=40 color=#00ffff>
<hr width="30%" noshade size=30 color=#f0ffff>
<hr width="40%" noshade size=20 color=#f000ff>
<hr width="50%" noshade size=10 color=#ff0f0f>
<hr width="60%" noshade size=5 color=#dc143c>
```

3. Сохраните полученный документ в вашей папке, присвоив файлу имя **linii.html**.

4. Запустите браузер Internet Explorer. Просмотрите в нем полученную web-страницу.

5. Создайте вторую web-страницу.



Фрагмент кода web-страницы:

```
<body bgcolor="#7fff00">
<h5 align=center> Лист,<br> нарисованный с помощью
символов,<br> введенных с клавиатуры</h5>
<h1><pre>
                    ^
                   /|\
                  \\|//
                 \\\|///
                \\\\|////
                 \\\|///
               \\\\\|//////
                \\\\\|/////
                 \\\|///
                    |
</h1></pre>
```

6. Сохраните полученный документ в вашей папке, присвоив файлу имя **list.html**.

7. Запустите браузер Internet Explorer. Просмотрите в нем полученную web-страницу.

6.3. Создание карты изображений и вставка бегущей строки.

7. Самостоятельная работа.

7.1. Создайте страницу, и назовите её "Моё любимое изображение", после чего разместите на странице ваши любимые изображения (фото, картинки, скриншоты) и подпишите ровно под изображениями.

8. Содержание отчета.

7.1. Файлы **algoritm.html**, **linii.html, list.html** в вашей папке.

9. Контрольные вопросы.

9.1. Какой тег служит для вставки изображений в HTML-документ?

9.2. Какой атрибут позволяет указать имя файла с изображением?

9.3. Какие атрибуты позволяют задать размеры картинки?

9.4. Какой атрибут позволяет задать положение изображения относительно окружающего текста?

9.5. Какой атрибут позволяет задать толщину рамки вокруг картинки?

9.6. Какие атрибуты определяют величину свободного пространства слева, справа, над и под картинкой?

9.7. Какие теги используются для создания карты изображений?

9.8. Какой тег используется для вставки бегущей строки?

4. Создание таблиц

Элемент <TABLE>

Элемент <TABLE> используется с целью внедрения таблиц в Web-страницу. Они удобны тем, что браузер сам прорисовывает рамку таблицы. Размер рамки может задаваться, как фиксировано, так и автоматически согласовываться с размерами окна просмотра браузера и с размерами, находящегося в ячейках текста и рисунков. Кроме того таблицы позволяют решать чисто дизайнерские задачи: выравнивать части таблицы друг относительно друга, размещать рядом рисунки и текст, управлять цветовым оформлением, разбивать текст на колонки и т.д.

При создании таблиц используется принцип вложения: между тэгами <TABLE> и </TABLE> задается заголовок вне рамки таблицы <CAPTION>, создается ряд элементов <TR>, определяющих начало строки, а внутри этих элементов размещаются элементы <TD>, описывающие ячейки и <TH>, описывающие заглавные ячейки.

Элемент <TABLE> может иметь атрибуты:

Bgcolor (задает фоновый цвет ячейкам)

<TABLE bgcolor=цвет>

Background (задает фоновый рисунок ячейкам)

<TABLE background=URL файла изображения>

Bordercolor (задает цвет рамки)

<TABLE border=число bordercolor=цвет >

Bordercolorlight (задает цвет рамки)

<TABLE border=число bordercolorlight=цвет>

Bordercolordark (задает цвет рамки)

<TABLE border=число bordercolordark=цвет>

Align (задает режим горизонтального)

<TABLE align= способ >

Width (задает ширину таблицы в пикселях или процентах от всего окна)

<TABLE width=число или %>

Border (задает ширину внешнего обрамления таблицы в пикселях)

<TABLE border=число >

Cellspacing (задает ширину внутреннего обрамления в пикселях)

<TABLE cellspacing=число >

Cellpadding (задает отступ между содержимым ячейки и обрамлением таблицы в пикселях)

<TABLE cellpadding=число >

Rules (задает линии между ячейками)

<TABLE rules=all >

Начало строки: элемент <TR>

Элемент <TR> открывает строку определений ячеек и не требует конечного тэга, хотя такой элемент можно использовать для наглядности обозначения каждой строки, поддерживает атрибуты, которые задают стиль оформления всех ячеек в строке. Отдельные ячейки могут быть отформатированы иначе.

Bgcolor (задает цвет фона ячеек)

<TR bgcolor= цвет >

Align (задает режим горизонтального выравнивания содержимого внутри ячейки)

TR align= способ >

Valign (задает режим вертикального выравнивания содержимого внутри ячейки)

<TR valign=значение>

Заголовок таблицы: элемент <CAPTION>

Элемент <CAPTION> задает заголовок вне рамки таблицы, имеет атрибут align, который может принимать значения top и bottom, left и right.

<CAPTION> текст </CAPTION>

Ячейки заголовка: элемент <TH>

Элемент <TH> задает ячейку.

<TH атрибуты > текст заголовка

Ячейки с данными <TD>

Элемент <TD> определяет ячейку с данными. Атрибуты:

Bgcolor (задает цвет фона ячейки)

<TD bgcolor= цвет >

Width (задает ширину ячейки в пикселях или в %)

<TD width= число или % >

Height (задает высоту ячейки в пикселях)

<TD height=значение>

Rowspan (задает объединение соседних ячеек столбца в одну большего размера)

<TD rowspan=количество строк >

Colspan (задает объединение соседних ячеек строки в одну большего размера)

<TD colspan=количество колонок >

Nowrap (блокирует автоматический перенос по словам в пределах ячейки в зависимости от других параметров таблицы.

Исходные коды некоторых таблиц.

1. Таблица с общим заголовком и заголовками в колонках.

```
<TABLE border=4 cellspacing=3>
<CAPTION> Заголовок таблицы </CAPTION>
<TR><TH bgcolor="white">Заголовок 1
      <TH bgcolor="white">Заголовок 2
<TR><TD>Ячейка 1
      <TD>Ячейка 2
<TR><TD>Ячейка 3
      <TD>Ячейка 4
</TABLE>
```

Заголовок таблицы	
Заголовок 1	Заголовок 2

Ячейка 1	Ячейка 2
Ячейка 3	Ячейка 4

2. Таблица с общим заголовком и заголовками в колонках, и в строках.

```
<TABLE border=4 cellspacing=3>
<CAPTION> Заголовок таблицы </caption>
<TR><TH bgcolor="white">
<TH bgcolor="white">Заголовок 1
<TH bgcolor="white">Заголовок 2
<TR<TH bgcolor="white">Заголовок 3
<TD>Ячейка 1
<TD>Ячейка 2
<TR><TH bgcolor="white">Заголовок 4
<TD>Ячейка 3
<TD>Ячейка 4
</table>
```

Заголовок таблицы		
	Заголовок 1	Заголовок 2
Заголовок 3	Ячейка 1	Ячейка 2
Заголовок 4	Ячейка 3	Ячейка 4

3. Таблица с объединенными ячейками в колонках, и в строках.

```
<TABLE border=4 cellspacing=0 >
<CAPTION>Заголовок таблицы </caption>
<TR><TD bgcolor="white">Заголовок 1
<TD bgcolor="white">Заголовок 2
<TR><TD rowspan=3 bgcolor="white">Ячейка 1
<TD>Ячейка 2
<TR><TD>Ячейка 3
<TR><TD>Ячейка 4
<TR><TD colspan=2 bgcolor="white" align="center">Ячейка 5
</table>
```

Заголовок таблицы

Заголовок 1	Заголовок 2
Ячейка 1	Ячейка 2
	Ячейка 3
	Ячейка 4
Ячейка 5	

4. Более сложное объединение ячеек в колонках, и в строках.

```
<TABLE border=4 cellspacing=0 >
<TR><TD bgcolor="white">Заголовок 1
<TD bgcolor="white">Заголовок 2
<TD bgcolor="white">Заголовок 3
<TR><TD rowspan=4 colspan=2 bgcolor="white">Ячейка 1
<TR><TD>Ячейка 2
<TR><TD>Ячейка 3
<TR><TD>Ячейка 4
<TR><TD colspan=3 bgcolor="white" align="center">Ячейка 5
</table>
```

Заголовок 1	Заголовок 2	Заголовок 3
Ячейка 1		Ячейка 2
		Ячейка 3
		Ячейка 4
Ячейка 5		

5. Таблица в ячейке таблицы.

```
<TABLE border=4 cellspacing=0>
<TR><TD bgcolor="white">Ячейка 1
<TD>
<TABLE border=2>
<TR><TD>Ячейка 2-1
<TD>Ячейка 2-2
<TR><TD>Ячейка 3-2
<TD>Ячейка 4-2
```

</table>

<TR><TD bgcolor="white">Ячейка 3-1

<TD bgcolor="white">Ячейка 4-1

</table>

<table>
<tr><td rowspan="2">Ячейка 1</td><td>Ячейка 2-1</td><td>Ячейка 2-2</td></tr>
<tr><td>Ячейка 3-2</td><td>Ячейка 4-2</td></tr>
<tr><td>Ячейка 3-1</td><td colspan="2">Ячейка 4-1</td></tr>
</table>

Практическая работа № 3

Создание таблиц в Web-страницах

1. Цель работы. Знакомство с основными тегами для создания таблиц в Web-страницах, изучение атрибутов для форматирования таблиц.

2. Литература. Конспект лекций.

3. Подготовка к работе. Изучить конспект лекций по теме "Таблицы".

4. Перечень оборудования. Компьютер.

5. Задание.

5.1. Изучить основные теги для создания таблиц в Web-страницах.

5.2. Изучить основные атрибуты для форматирования таблиц.

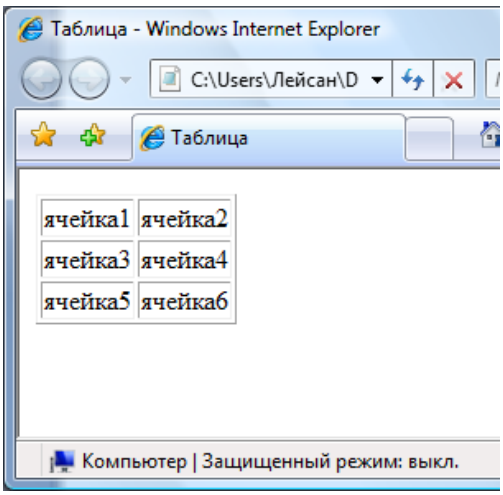
5.3. Научиться создавать таблицы в web-страницах.

6. Порядок выполнения работы.

6.1. **Создание таблицы без объединения ячеек.**

1. Откройте текстовый редактор Блокнот и создайте web-страницу, которая содержит следующую таблицу:

<table>
<tr><td>ячейка1</td><td>ячейка2</td></tr>
<tr><td>ячейка3</td><td>ячейка4</td></tr>
<tr><td>ячейка5</td><td>ячейка6</td></tr>
</table>



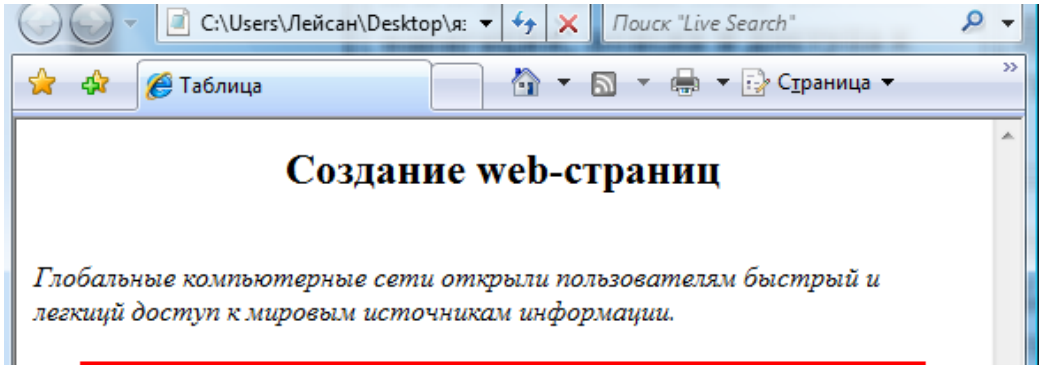
2. Таблица задается тегом: **<table>…</table>**. Для задания строк используйте тег **<tr>…</tr>,** а для ячеек - **<td>…</td>.**

3. Сохраните полученный документ в вашей папке, присвоив файлу имя **tab1.html**.

4. Запустите браузер Internet Explorer. Просмотрите в нем полученную web-страницу.

6.2. Внесение информации в ячейки таблицы.

1. Откройте только что созданный файл **tab1.html** в текстовом редакторе Блокнот и внесите новую информацию в соответствии с образцом:



2. Для этого введите следующий текст:

Создание web-страниц
Глобальные компьютерные сети открыли пользователям быстрый и легкий доступ к мировым источникам информации.

Для первого предложения установите заголовок второго уровня, полужирное начертание и выровните по центру. Для второго предложения установите курсивное начертание.

3. Далее в первую ячейку таблицы введите следующее предложение

WWW(Word Wide Web) - система навигации, поиска и доступа к мультимедейным ресусам с помощью средств гипертекста.

4. Сохраните файл присвоив ему имя **tab2.html** и просмотрите полученную web-страницу.

5. Таким же образом введите по одному предложению в остальные ячейки таблицы:

Браузер - программа просмотра гипертекстовых страниц WWW.
Гипертекст - текст, содержащий связи с другими текстами, графической, видео- или звуковой информацией.
HTML(Hyper Text Markup Language) - язык разметки гипертекста.
HTML - это текстовые файлы, в которые встроены специальные теги.
Теги(tags) - команды языка HTML.

Сохраните изменения.

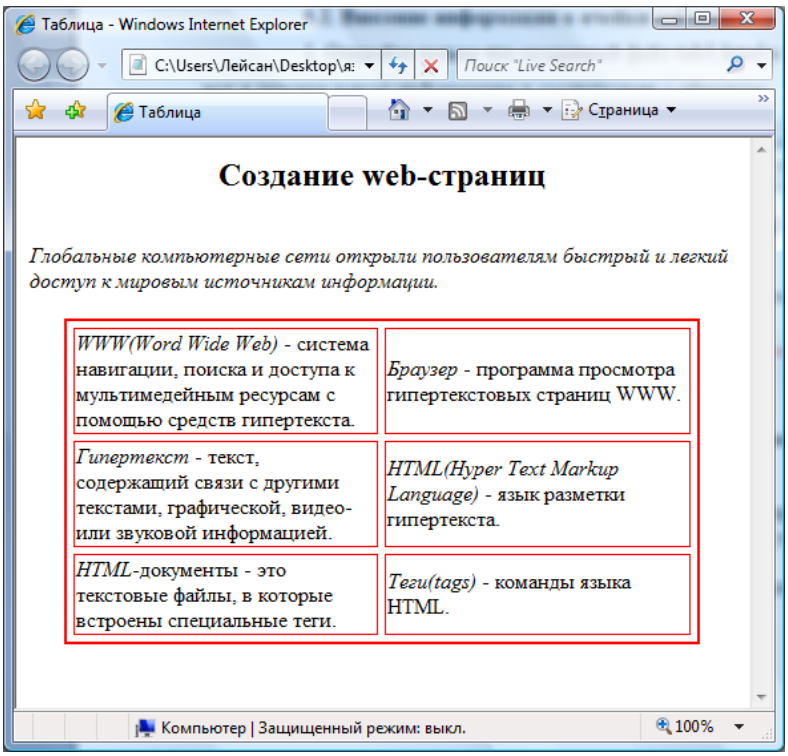
6. Отформатируйте таблицу. Для этого задайте в теге **<table>** следующие параметры:

- ширина таблицы – 90%;
- расстояние между содержимым ячейки и рамкой – 5;
- расстояние между ячейками таблицы – 5;
- цвет линий таблицы – красный;
- толщина линий – 2
- выравнивание во всех ячейках – по центру.

Далее в теге **<td>** задайте ширину ячейки равной 45%.

7. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

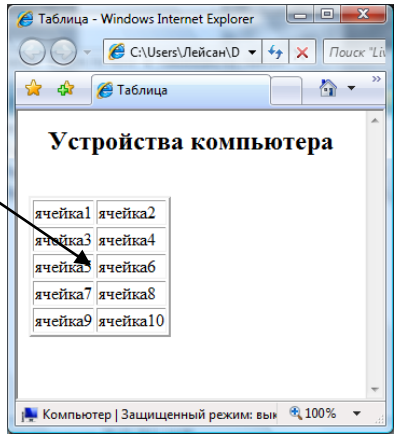
Ваша web-страница должна выглядить следущим образом:



6.3. Форматирование ячеек таблицы.

1. Откройте текстовый редактор Блокнот и создайте web-страницу, которая содержит следующую таблицу:

ячейка1	ячейка2
ячейка3	ячейка4
ячейка5	ячейка6
ячейка7	ячейка8
ячейка9	ячейка10



2. Задайте в теге **<table>** следующие параметры:

- ширина таблицы – 90%;
- расстояние между содержимым ячейки и рамкой – 3;
- расстояние между ячейками таблицы – 3;
- цвет линий таблицы – красный;
- цвет фона – зеленый или желтый;
- толщина линий – 2;
- выравнивание во всех ячейках – по центру.

3. Задайте в теге **<tr>** высоту строк равной 120 и выравнивание содержимого ячеек по центру, а в теге **<td>** - ширину ячейки 50%.

4. Сохраните полученный документ в вашей папке, присвоив файлу имя **tab3.html**.

5. Запустите браузер Internet Explorer. Просмотрите в нем полученную web-страницу.

6. Скопируйте папку **КАРТИНКИ**, которая находится на рабочем столе в свою папку.

7. Далее откройте только что созданный файл **tab3.html** в текстовом редакторе Блокнот и внесите новую информацию: в **Ячейке1** наберите слово **Компьютер.** Для него установите заголовок 2 уровня и полужирное начертание. В **Ячейку2** вставьте картинку **comp.gif,** которая находится в папке **КАРТИНКИ.**

Для вставки картинки используйте следующий код:

<img src=картинки\comp.gif>

8. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

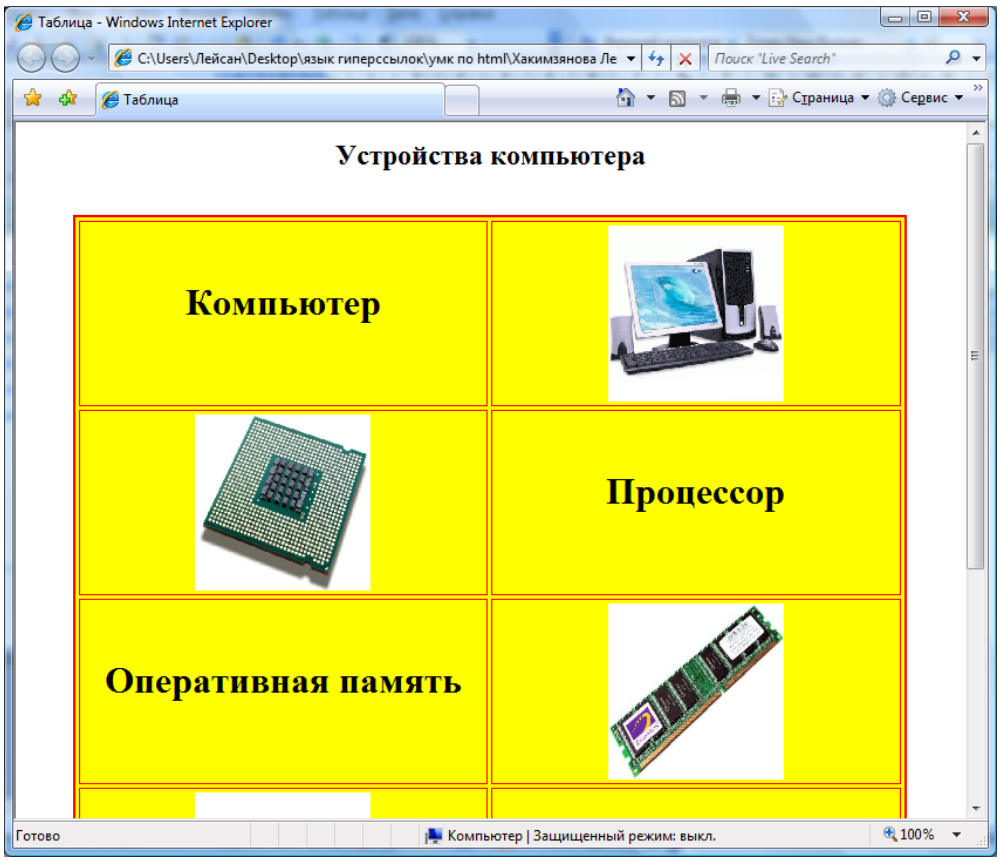
9. Таким же образом внесите информацию в другие ячейки:

- в **Ячейку3** вставьте картинку **pros.jpg;**
- в **Ячейке4** наберите слово **Процессор**;
- в **Ячейке5** наберите слово **Оперативная память**;
- в **Ячейку6** вставьте картинку **oper.jpg;**
- в **Ячейку7** вставьте картинку **clav.png;**
- в **Ячейке8** наберите слово **Клавиатура**;
- в **Ячейке9** наберите слово **Монитор**;
- в **Ячейку10** вставьте картинку **monitor.jpg.**

Для всех картинок установите ширину и высоту равной 150.

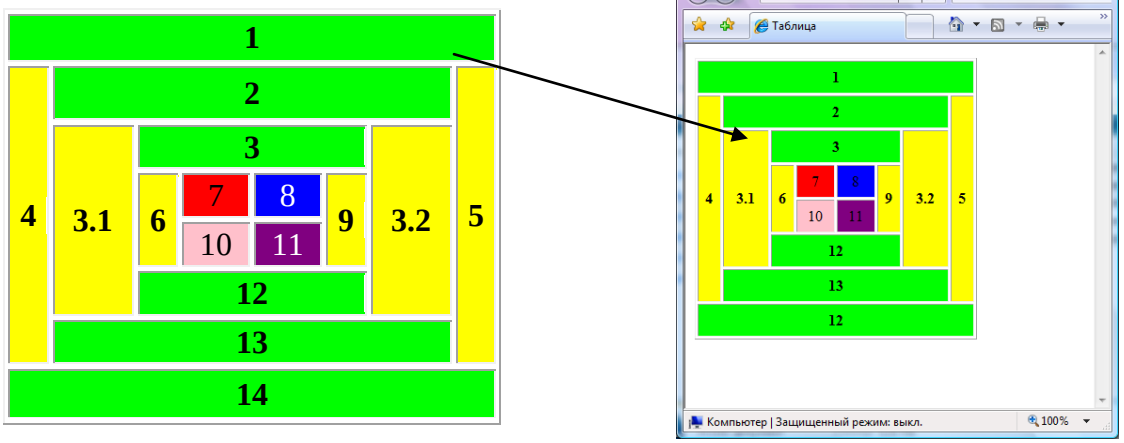
10. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

Ваша web-страница должна выглядить следущим образом:



6.4. Создание таблицы с объединенными ячейками.

1. Откройте текстовый редактор Блокнот и создайте web-страницу, которая содержит следующую таблицу:



2. Для создания данной таблицы используйте атрибуты **colspan="..." и rowspan="...",** которые объединяют соответственно столбцы и строки в один.

3. Задайте в теге **<table>** следующие параметры:

- ширина таблицы – 300;
- высота таблицы – 300;

- толщина линий – 2.

4. Фрагмент HTML-кода:

```
Файл  Правка  Формат  Вид  Справка
</head>
<body>
<table border=2 width='300' height='300'>
<tr>
  <td colspan='8' bgcolor='lime' align='center'> 1 </td>

</tr>
<tr>
  <td rowspan='6' bgcolor='yellow' align='center'> 4 </td>
  <td colspan='6' bgcolor='lime' align='center'> 2 </td>
  <td rowspan='6' bgcolor='yellow' align='center'> 5 </td>
</tr>
<tr>
  <td rowspan='4' bgcolor='yellow' align='center'> 3.1 </td>
  <td colspan='4' bgcolor='lime' align='center'> 3 </td>
  <td rowspan='4' bgcolor='yellow' align='center'> 3.2 </td>
```

5. Для ячеек **1, 2, 3, 12, 13, 14** установите цвет **“lime”, 4, 3.1, 6, 9, 3.2, 5 – “yellow”, 7 – “red”, 8 – “blue”, 10 – “pink”, 11 – “purple”.**

6. Сохраните полученный документ в вашей папке, присвоив файлу имя **tab4.html**.

5. Запустите браузер Internet Explorer. Просмотрите в нем полученную web-страницу.

7. Самостоятельная работа.

7.1. Создайте страницу **tab5.html**, которая содержит таблицу следующего вида:

основная таблица заголовок 1	**основная таблица заголовок 2**
основная таблица ячейка 1	основная таблица ячейка 2
основная таблица ячейка 3	**Вложенная таблица заголовок 1** / **Вложенная таблица заголовок 2** вложенная таблица ячейка 1 / вложенная таблица ячейка 2 вложенная таблица ячейка 3 / вложенная таблица ячейка 4

8. Содержание отчета.

7.1. Файлы **tab1.html, tab2.html, tab3.html, tab4.html, tab5.html** в вашей папке.

9. Контрольные вопросы.

9.1. Какой тег служит для создания таблицы?

9.2. Как задать в таблице определенное количество строк и столбцов?

9.3. Какой атрибут задает ширину таблицы, строки, ячейки?

9.4. Какой атрибут задает расстояние между содержимым ячейки и ее обрамлением?

9.5. Какой атрибут задает расстояние между ячейками?

9.6. Какие атрибуты задают цвет фона и фоновую картинку для таблицы, строки, ячейки?

9.7. Какие атрибуты задают толщину и цвет линий для таблицы?

9.8. Какие атрибуты позволяют объединять ячейки таблицы?

5. Формы

Создание форм.

Формы представляют собой наиболее важные интерактивные элементы HTML, позволяющие разработчикам страниц интерактивно взаимодействовать с посетителями. С их помощью пользователь может возвращать комментарии по поводу посещения определенного узла, пересылать запросы или регистрироваться. Разработчик задает вопросы, создавая форму, а пользователь отвечает на них заполняя её.

Форма создается при помощи различных тэгов и атрибутов, заключенных в пару <FORM></FORM>

Элемент <FORM>.

Элемент <FORM> является необходимым условием для всех форм. Он может иметь следующие атрибуты:

Method (определяет способ пересылки данных)

Action (определяет путь к сценарию или адрес электронной почты)

Enctype (определяет способ кодирования содержимого формы)

Пример:

FORM action="mailto:имя@домен.ru?subject="Comment from Home Page" method="post" enctype="text/plain">

Комментарии:

<TEXTAREA type="text" NAME="комментарий" ROWS=5 COLS=50>

</TEXTAREA>

Имя:

<INPUT type="text" NAME="имя" SIZE="57">

E-mail:

<INPUT type="text" name="e-mail" size="52" maxlength="360">

<INPUT type="reset" VALUE=Сброс >

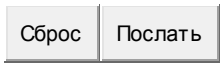
<INPUT type=SUBMIT VALUE=Послать>

</FORM>

Комментарии:

Имя:

E-mail:



Элемент <INPUT>.

Элемент <INPUT> является базовым для всех элементов формы. Он используется для внедрения в форму кнопок, графических изображений, флажков, переключателей, паролей и текстовых полей.

Элемент может иметь атрибуты обозначаемые как type:

1. TEXT

Однострочное текстовое поле, используется для ввода информации, которую нельзя ввести ни в одном из остальных элементов формы. Сюда вводятся имена, адреса, должности, телефоны, хобби, и данные практически любого типа. Элемент может иметь атрибуты:

Maxlength (задаёт максимально допустимую длину вписываемого значения в символах)

Size (задаёт максимально допустимую длину поля в символах)

Value (задаёт значение по умолчанию, которое можно менять)

<INPUT type="TEXT" name="Hobby" maxlength="35" size="20" value="Shopping">

2. PASSWORD

Однострочное поле, в котором вместо вводимых символов отображаются звездочки. Элемент может иметь атрибуты:

Maxlength (задаёт максимально допустимую длину вписываемого значения в символах)

Size (задаёт максимально допустимую длину поля в символах)

value (задаёт значение по умолчанию, которое можно менять)

<INPUT type="PASSWORD" name="PASSWORD_BOX" maxlength="35" size="20">

3. HIDDEN

Еще один тип скрытого ввода информации. Позволяет пересылать сценариям информацию, которая не может быть изменена пользователем.

<INPUT type="HIDDEN" name="file" value="anyfile.html">

4. CHECKBOX

Флажки используются для предоставления возможности пользователю ответить односложно: да/нет истина/ложь больше/меньше и т.д. Выглядит обычно в виде крестика или птички. Элемент может иметь атрибуты:

Checked (задаёт начальный статус флажка по умолчанию)

value (задаёт значение по умолчанию, которое можно менять)

<INPUT type="checkbox" name="send_mail" value="yes" checked>

5. RADIO

Переключатели во многом напоминают флажки, отличаясь лишь более широкими функциональными возможностями выбора. В группе переключателей может быть выбран лишь один. Для каждого переключателя указывается отдельный элемент INPUT.

6. SUBMIT

Щелчок на этой кнопке приводит к пересылке содержимого формы сценарию, который был задан атрибутом action в элементе <FORM>. С помощью кнопок можно вычислять сумму, загружать страницы, пересылать данные, сбрасывать значения.

7. RESET

Кнопка используется для восстановления значений, заданных по умолчанию. Если значение по умолчанию не предусмотрено, то оно просто обнулится. Ширина кнопки может меняться в зависимости от других элементов. Имеет так же атрибут value.

<INPUT type="reset" value="очистка">

8. IMAGE

Во многом похож на кнопку SUBMIT, только в качестве кнопки используется изображение. Элемент может иметь атрибуты:

Src (задаёт URL файла с изображением)

Align (задаёт выравнивание изображения относительно текста при помощи значений TOP, MIDDLE и BOTTOM)

Name (задаёт имя карты, которое так же пересылается сценарию вместе с координатами)

<INPUT type="image" src="knopka.gif">

9. BUTTON

Создает другую кнопку, браузеры пользователей могут использовать значение атрибута value в качестве исходного имени файла.

<INPUT type="button"value="кнопка">

Элемент <TEXTAREA>.

При помощи этого элемента создается область для ввода и просмотра текста. Может использоваться и не в составе формы, а как самостоятельные детали страницы. Область ввода помогает сэкономить место благодаря полосам прокрутки. Может иметь атрибуты:

Name (задаёт ключевое слово, по которому сценарий может обращаться к его содержимому)

Rows (задаёт высоту области в строках)

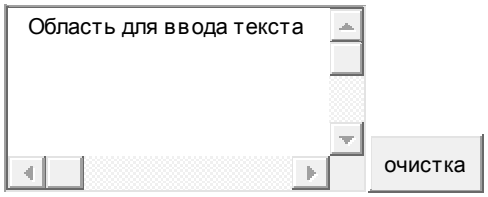
Cols (Задаёт ширину области в символах)

<FORM><H3>Введи текст

<TEXTAREA name="ключевое слово" rows=5 cols=30>Область для ввода текста </TEXTAREA></H3>

<INPUT type="reset" value="очистка">

</FORM>



Элемент <SELECT>.

Элемент <SELECT> может принимать форму раскрывающегося списка или меню элементов. Имеет вложенный тэг <OPTION> и атрибуты:

Name (задаёт имя)

Size (задаёт максимальное количество элементов списка, одновременно отображаемых на экране)

Multiple (задаёт возможность одновременного выбора нескольких значений)

Элемент <OPTIONS>.

Элемент же <OPTIONS> задает возможные варианты выбора меню <SELECT>

<OPTION value="n" selected>значение

Имеет атрибуты:

Selected (задаёт изначально выбранное слово)

Value (задаёт значение выбранного слова для сценария)

Практическая работа № 4

Создание форм в Web-страницах

1. Цель работы. Изучение основных тегов и атрибутов для создания формы в Web-страницах.

2. Литература. Конспект лекций.

3. Подготовка к работе. Изучить конспект лекций по теме "Формы".

4. Перечень оборудования. Компьютер.

5. Задание.

5.1. Изучить основные теги для создания форм в Web-страницах.

5.2. Изучить основные атрибуты для размещения на форме элементов управления.

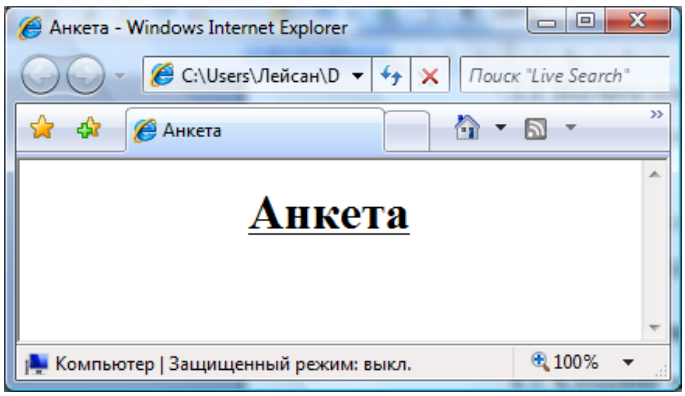
5.3. Научиться создавать формы в web-страницах.

5.4. Научиться размещать на форме элементы управления.

6. Порядок выполнения работы.

6.1. **Создание web-страницы, используя текстовые поля ввода, поле ввода пароля.**

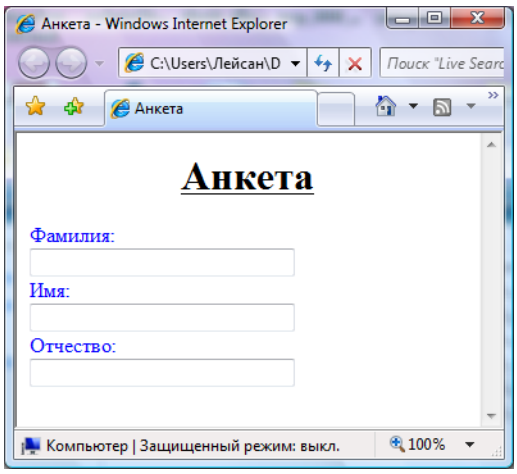
1. Откройте текстовый редактор Блокнот и создайте web-страницу в соответствии с образцом:



2. Сохраните полученный документ в вашей папке, присвоив файлу имя **anketa.html**.

3. Запустите браузер Internet Explorer. Просмотрите в нем полученную web-страницу.

4. Откройте только что созданный файл **anketa.html** в текстовом редакторе Блокнот и внесите новую информацию в соответствии с образцом:



5. Форма (формуляр) в HTML-документе определяется в виде блока **<FORM>…</FORM>**, внутри которого располагаются теги, задающие те или иные формы**.**

6. Для того, чтобы создать текстовое поле ввода используется тег **<INPUT type=text>** с атрибутами **name=name, size=30**.

Таким образом фрагмент HTML-кода для одного текстового поля ввода имеет вид:

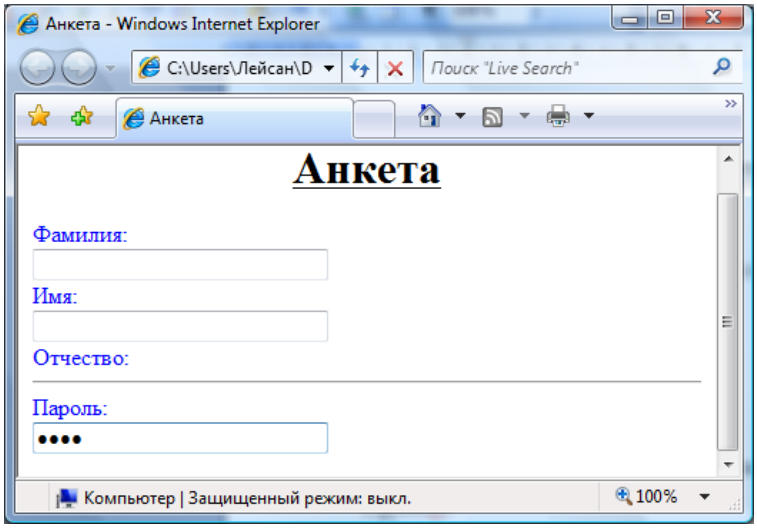
```
<form>
<font color=blue>
Фамилия:<br>
<input type=text name=name size=30><br>
Имя:<br>
</font>
</form>
```

7. Таким же образом создайте остальные текстовые поля ввода. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

8. Далее на этой странице создайте поле ввода пароля. Для этого применяется тег <**INPUT type= password >** с атрибутами **size=30, maxlenght=40**.

9. Текстовые поля ввода и поле ввода пароля должны отделятся друг от друга горизонтальной линией.

10. Сохраните изменения в HTML-файле и в браузере обновите web-страницу. Ваша web-страница должна выглядить следущим образом:



6.2. **Создание раскрывающегося списка.**

1. Откройте файл **anketa.html** в текстовом редакторе Блокнот. На этой странице создайте раскрывающийся список.

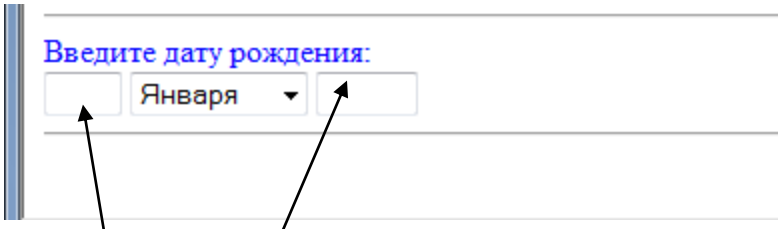
2. Для создания раскрывающегося списка используется тег **<SELECT>…</SELECT> с** атрибутом **name=mounth.**

Значение отдельных пунктов списка определяется при помощи тега **<OPTION>** с атрибутом **selected.**

3. Фрагмент HTML-кода имеет вид:

```
<select name=mounth>
<option selected> Января
<option> Февраля
<option> Марта
<option> Апреля
<option> Мая
<option> Июня
<option> Июля
<option> Августа
<option> Сентября
<option> Октября
<option> Ноября
<option> Декабря
</select>
```

4. Кроме этого в этом разделе создайте два текстовых поля ввода с размерами 2 и 4 в соответствии с образцом:

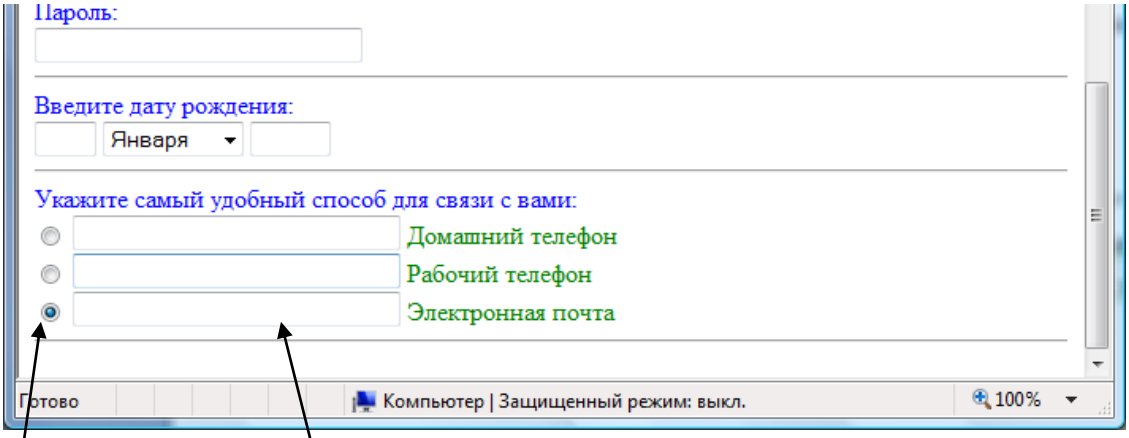


Текстовые поля ввода

5. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

6.3. **Создание радиокнопок.**

1. На странице **anketa.html** создайте радиокнопки в соответствии с образцом:



Радиокнопка Текстовое поле ввода

2. Радиоктопка задается командой: **<INPUT type=radio>** с атрибутами **name=group, value=domtel.** (Атрибут **name** принимает одинаковое имя **group** для всех кнопок, а **value –** разные. Для первой кнопки **value=domtel,** для второй - **value=rabtel,** для третьей - **value=elpost)**

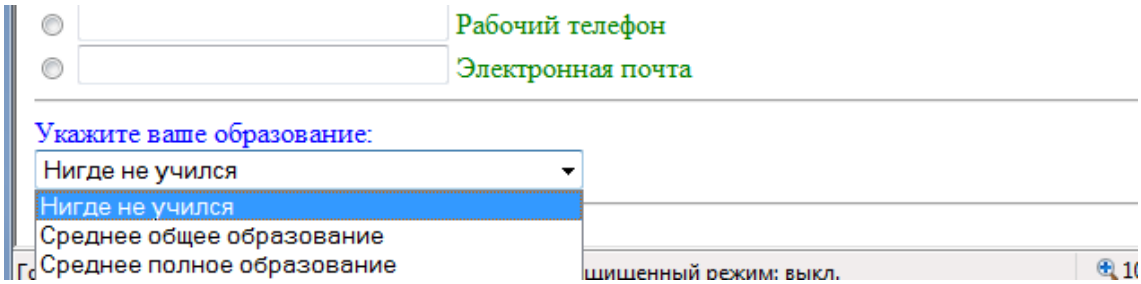
3. Фрагмент HTML-кода для одной радиокнопки имеет вид:

<input type=radio name=group Value=domtel>

4. Таким же образом создайте еще две радиокнопки и необходимые текстовые поля ввода. Не забудьте ввести рядом необходимую текстовую информацию. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

6.4. **Создание второго раскрывающегося списка и многострочного поля ввода.**

1. На странице **anketa.html** создайте еще один раскрывающийся список вида:



Для создания раскрывающегося списка используется тег **<SELECT>…</SELECT> с** атрибутом **name= education.**

Пункты списка:

Нигде не учился

Среднее общее образование

Среднее полное образование

Среднее специальное образование

Среднее профессиональное образование

Незаконченное высшее образование

Высшее образование

Два высших образования

2. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

3. Далее на этой странице создайте многострочное поле ввода. Для этого используется тег **<TEXTAREA>…</TEXTAREA> с** атрибутами **name=osebe, rows=4, cols=30.** Сохраните изменения.

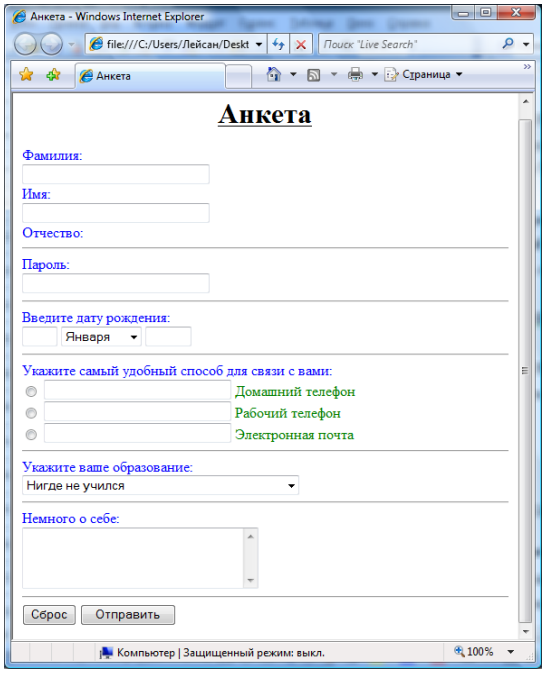
6.5. **Создание кнопок Сброс и Отправить.**

1. На странице **anketa.html** создайте кнопки **Сброс** и **Отправить**.

2. Для того, чтобы создать кнопку **Сброс** применяется тег **<INPUT type=reset> с** атрибутом **value=Сброс.**

3. А для создания кнопки **Отправить** применяется тег **<INPUT type=submit> с** атрибутом **value=Отправить.**

4. Сохраните изменения в HTML-файле и в браузере обновите web-страницу. Ваша web-страница должна выглядить следущим образом:



7. Самостоятельная работа

7.1. Создайте страницу **zakaz.html**, которая содержит следующую форму:

8. Содержание отчета.

7.1. Файлы **anketa.html, zakaz.html** в вашей папке.

9. Контрольные вопросы.

9.1. Что такое форма? Какие дополнительные возможности она предоставляет пользователю?

9.2. С помощью какого тега создается форма?

9.3. Какой тег позволяет создать текстовое поле ввода?

9.4. Какое значение атрибута type позволяет создать поле для ввода пароля?

9.5. С помощью какого тега создается раскрывающийся список? Какие атрибуты имеет тег <select>?

9.6. Какой тег позволяет задать элементы списка?

9.7. С помощью какого тега создаются радиокнопки? Какие атрибуты имеет тег <INPUT type=radio>?

9.8. Как создать кнопку для очистки данных?

9.9. Как создать кнопку для отправки данных?

6. Фреймы

Использование тега <FRAME> позволяет помещать в окна одной страницы несколько отдельных страниц, произвольно менять их размеры и организовывать изменение содержимого одного окна после выполнения пользователем действий в другом окне. Это позволяет использовать их в качестве инструмента навигации. Тэги <FRAMESET> и </FRAMESET> в данном случае заменяют тэги <BODY> и </BODY> соответственно.

Внутри пары <FRAMESET> и </FRAMESET> могут быть использованы только тэги <FRAME>, <FRAMESET> и <NOFRAMES>. По сути, создается несколько отдельных страниц фреймов, которые выводятся на экран одновременно в виде нескольких окошек одного документа. Другими словами, прежде чем организовывать отдельные документы в виде фреймов одного документа, необходимо прежде создать сами документы. Для пользователей, чей браузер не поддерживает фреймы, необходимо создать альтернативный документ или сообщение о том, каким браузером нужно воспользоваться для просмотра данного документа и заключить его в тэги <NOFRAME> и </NOFRAME>.

Тэги <FRAMESET> и </FRAMESET> позволяют задать относительный и абсолютный размеры фреймов, <FRAMESET> может иметь атрибуты:

Border (задает толщину обрамления в пикселях для всех производных фреймов)

<FRAMESET border="число" >

Cols (задает количество и размер колонок в создаваемом наборе кадров в пикселях, процентах или *пропорционально другим кадрам)

<FRAMESET cols="число, число* или % ">

Rows (задает количество и размер строк в создаваемом наборе кадров в пикселях, процентах или *пропорционально другим кадрам)

<FRAMESET rows="число, число* или % ">

Frameborder (задает наличие или отсутствие обрамления у фреймов, значение 1 соответствует наличию, а 0 - отсутствию обрамления)

<FRAMESET frameborder="1 или 0" >

Использование <FRAME>

Элемент <FRAME> определяет содержимое заданных фреймов. Он может иметь атрибуты:

Src (задает документ, который должен быть отображен в фрейме)

<FRAME src="URL" >

Frameborder (задает наличие или отсутствие обрамления у фреймов)

<FRAME frameborder="1 или 0 ">

Marginheight (задает толщину верхнего и нижнего обрамления в пикселях)

<FRAME marginheight="число ">

Marginwidth (задает толщину правого и левого обрамления в пикселях)

<FRAME marginwidth="число ">

Name (задает любое имя фрейма, по которому можно будет обращаться к нему с помощью атрибута target в ссылках <A href>)

Но существует четыре зарезервированных имени:

_blank - имя открывает в новом окне содержимое указанного URL.

_parent - имя открывает содержимое указанного URL в родительском, относительно текущего, фрейме.

_self - если так назвать фрейм, то содержимое указанного URL заменит первоначально находившуюся в этом фрейме ссылку.

_top - имя отображает содержимое указанного URL в развёрнутом на всё окно фрейме.

Noresize (лишает пользователя возможности изменить размеры текущего и смежного фреймов с помощью мыши)

<FRAME noresize >

Scrolling (задает наличие у кадра полос прокрутки, принимает значения YES, NO и AUTO)

<FRAME scrolling="значение ">

Использование <IFRAME>.

Этот тэг позволяет вставить в тело BODY окно, в котором визуализируется другая страница, при этом тэг IFRAME, в отличие от тэга FRAME вставляется не между тэгами FRAMESET и /FRAMESET, а между тэгами BODY и /BODY. Атрибуты:

Src (задает документ, который должен быть отображен в фрейме)

<IFRAME src="URL" >

Frameborder (задает наличие или отсутствие обрамления у фреймов)

<IFRAME frameborder="1 или 0 ">

Marginheight (задает толщину верхнего и нижнего обрамления в пикселях)

<IFRAME marginheight="число ">

Marginwidth (задает толщину правого и левого обрамления в пикселях)

<IFRAME marginwidth="число ">

Name (задает любое имя фрейма, по которому можно будет обращаться к нему в ссылках и направлять в него содержимое)

Align (позволяет позиционировать кадр по отношению к тексту, принимает значения left, right, middle, top и bottom)

<IFRAME align="left" >

Scrolling (задает наличие у кадра полос прокрутки, принимает значения YES, NO и AUTO)

<FRAME scrolling="значение ">

Width (определяет ширину кадра в пикселях)

<IFRAME width="число ">

Height (определяет высоту кадра в пикселях)

<IFRAME height="число ">

Практическая работа № 5

Создание фреймов в Web-страницах

1. Цель работы. Изучение основных тегов и атрибутов для создания фреймов в Web-страницах.

2. Литература. Конспект лекций.

3. Подготовка к работе. Изучить конспект лекций по теме "Фреймы".

4. Перечень оборудования. Компьютер.

5. Задание.

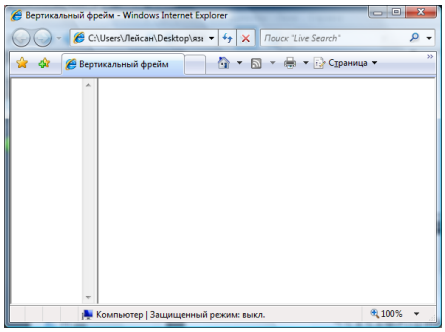
5.1. Изучить основные теги и атрибуты для создания фреймов в Web-страницах.

5.2. Научиться создавать фреймы в web-страницах.

6. Порядок выполнения работы.

6.1. **Создание вертикальных и горизонтальных фреймов.**

1. Откройте текстовый редактор Блокнот и создайте web-страницу с вертикальными фреймами в соответствии с образцом:



2. Для создания вертикальных фреймов используется тег **<frameset>…</frameset>** с атрибутом **cols (**вертикальное деление экрана**).**

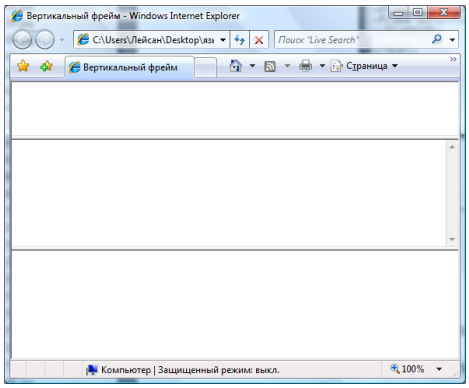
3. Для данной страницы установите следующие параметры: количество столбцов – 2, соотношение ширины столбцов – 20% и 80% соответственно. В первом фрейме разрешите прокрутку, а во втором – запретите. Запретите изменение размеров фреймов.

4. Фрагмент HTML-кода:

```
<frameset cols="20%,80%">
<frame noresize scrolling=yes>
<frame scrolling=no>
</frameset>
```

5. Сохраните файл присвоив ему имя **frame1.html** и просмотрите полученную web-страницу.

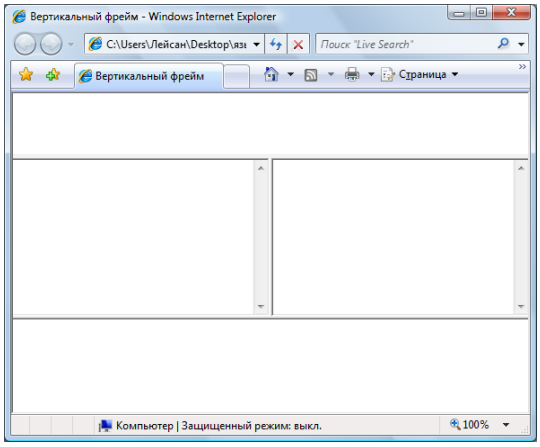
6. Таким же образом создайте еще одну web-страницу с горизонтальными фреймами в соответствии с образцом:



7. Для данной страницы установите следующие параметры: количество строк – 3, соотношение ширины строк – 20%, 40%, 40% соответственно. Во втором фрейме разрешите прокрутку, а в первом и третьем – запретите. Запретите изменения размеров фреймов.

8. Сохраните файл присвоив ему имя **frame2.html** и просмотрите полученную web-страницу.

9. Далее создайте третью web-страницу по образцу:



10. Для этой страницы установите следующие параметры: соотношение высот горизонтальных фреймов – 20%, 50%, 30% соответственно. Во втором фрейме разрешите прокрутку, а в первом и третьем – запретите. Второй горизонтальный фрейм разделите по ширине в пропорции 50:50%. Запретите изменения размеров фреймов.

11. Сохраните файл присвоив ему имя **frame3.html** и просмотрите полученную web-страницу.

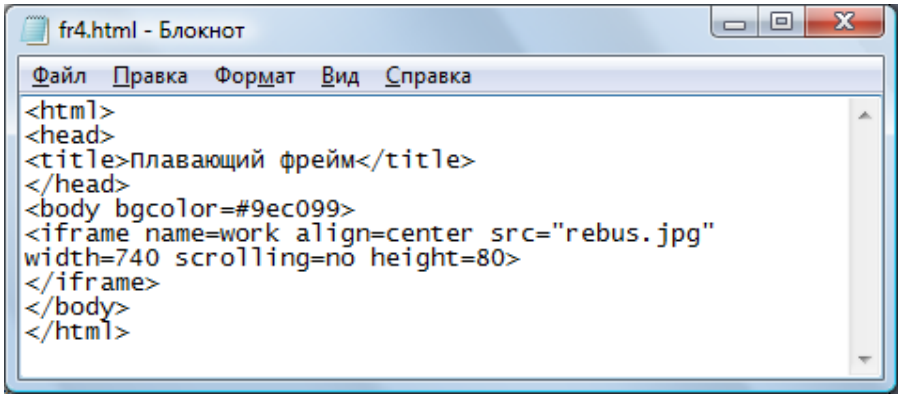
6.2. **Создание плавающих фреймов.**

1. Откройте текстовый редактор Блокнот и создайте web-страницу, который состоит из трех плавающих фреймов. Для этого используются теги **<IFRAME> … </IFRAME>.**

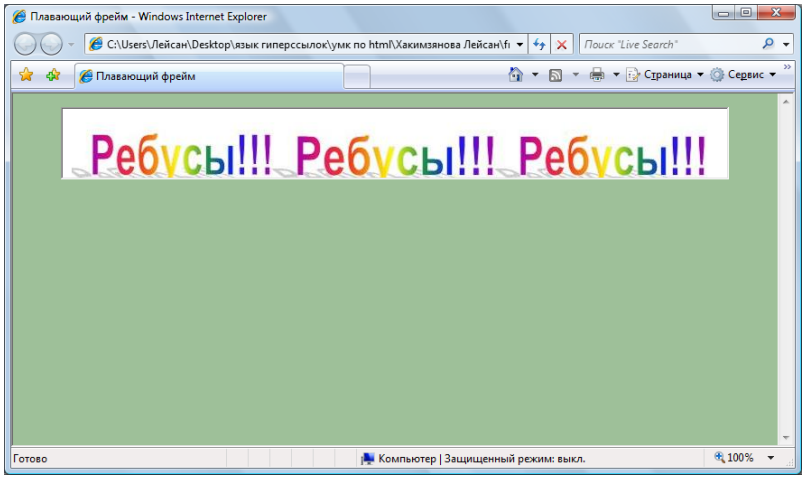
2. Для того, чтобы создать первый фрейм используйте теги **<IFRAME> … </IFRAME>** с атрибутами **name=work, align=center, src="rebus.jpg", width=740, scrolling=no, height=80.**

Картинка **rebus.jpg** и все остальные находятся в папке Картинки на рабочем столе. Скопируйте их в свою папку.

3. Фрагмент HTML-кода:



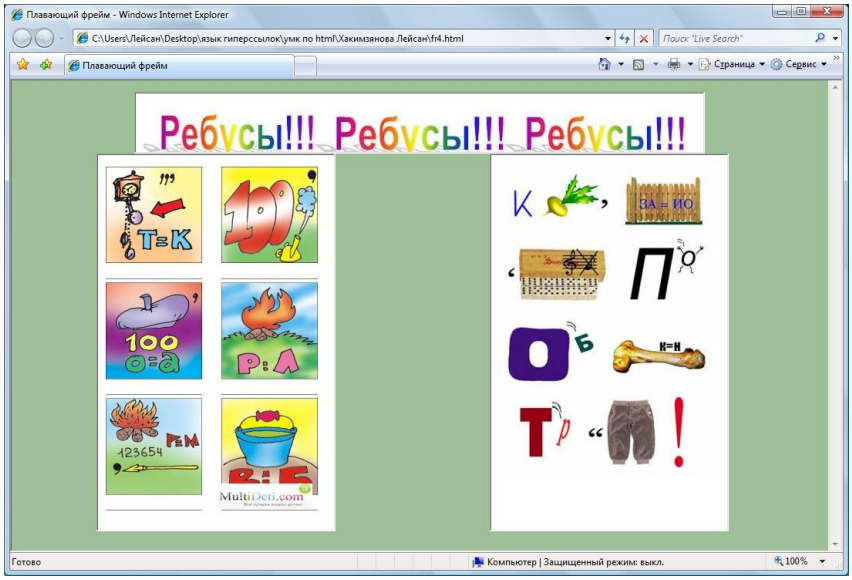
4. Сохраните файл присвоив ему имя **frame4.html** и просмотрите полученную web-страницу. Ваша web-страница должна выглядеть следующим образом:



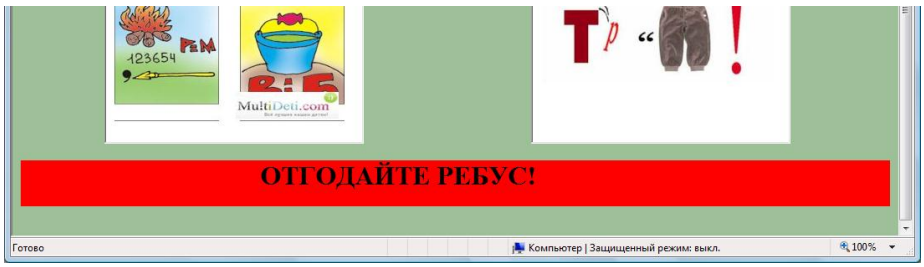
5. Далее таким же образом создайте еще два фрейма. Для первого фрейма используйте картинку **rebus2.jpg,** а для второго **rebus3.jpg** и установите следующие параметры:

- имя фрейма – name;
- ширина фрейма – 310;
- высота фрейма – 490;
- поля слева и справа от фрейма – 100;
- запретите прокрутку.

6. Сохраните изменения в HTML-файле и в браузере обновите web-страницу. Ваша web-страница должна выглядить следущим образом:



7. Далее в нижней части страницы поместите «бегущую строку», движущегося слева нараво.

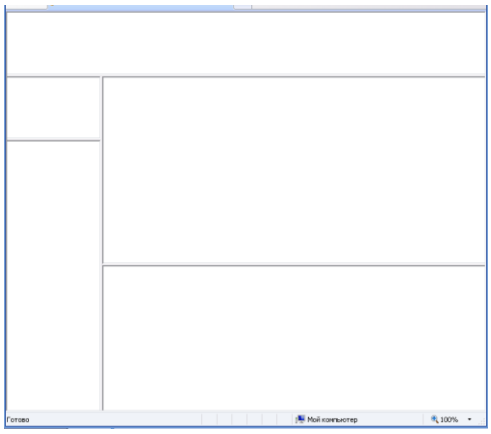


Для создания бегущей строки используйте тег **<marquee>…</marquee>** с атрибутами **bgcolor=red, height=15, vspace=20, align=middle.**

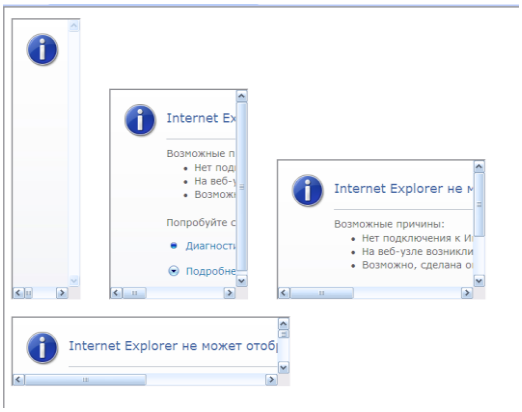
8. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

7. Самостоятельная работа.

7.1. Создайте страницу **frame5.html** с 5-ю фреймами:



7.2. Создайте страницу с 4-мя плавающими фреймами:



8. Содержание отчета.

8.1. Файлы **frame1.html**, **frame2.html, frame3.html, frame4.html** в вашей папке.

9. Контрольные вопросы.

9.1. Каково основное назначение фреймов?

9.2. С помощью какого тега можно разбить окно браузера на вертикальные или горизонтальные фреймы?

9.3. Какой атрибут тега <frameset> позволяет установить количество вертикальных(горизонтальных) фреймов?

9.4. Как можно задавать высоту и ширину фреймов?

9.5. С помощью какого атрибута тега <frameset> можно отключить вывод границы между фреймами?

9.6. С помощью какого атрибута тега <frameset> можно задавать ширину рамки между фреймами?

9.7. Какой атрибут позволяет запретить изменения размеров фрейма?

9.8. Каково назначение плавающего фрейма?

9.9. Какой тег используется для создания плавающих фреймов?

9.10. С помощью какого атрибута тега <iframe> задаются поля слева и справа от фрейма?

7. Ссылки

Элемент <A> используется с целью создания ссылок на другие элементы документа, или даже на другие документы, такие ссылки являют собой основную причину ошеломляющей популярности пространства World Wide Web , где пользователь может легко перескочить с одного фрагмента текста на другой или со страницы на страницу, не задавая явным образом URL последних. Элемент выполняет два действия: задает имя ссылки и задает ссылку на имя. Имя ссылки браузер автоматически выделяет другим цветом и подчеркивает. Атрибуты:

Name (задает привязку ссылки в тексте, на которую и будет производиться ссылка)

<A name=имя> необязательный текст </A >

Href (задает адрес ссылки. Он может указывать или на имя ссылки в тексте, или на URL и имя файла)

<A href="URL"> текст, для щелчка </A >

или же в тексте: <A href="#имя"> текст для щелчка

methods (указывает метод извлечения документа, например, FTP, Gopher и т. Д)

<A methods="название"> необязательный текст </A >

Frame (используется для указания названия кадра, в котором должен быть визуализирован целевой документ)

<A frame="имя фрейма"> необязательный текст </A >

Accesskey (используется для указания горячей клавиши, при нажатии на которую осуществляется переход по ссылке)

<A accesskey=" имя клавиши"> Ссылка</A >

Вместо текста между тегами <A> и </A> можно вставить изображение, и изображение будет ссылкой.

<img src="1.png">

У изображения, являющегося ссылкой, автоматически появляется рамка тощиной 1 пиксел такого же цвета, как текстовые ссылки.

Цвет ссылок можно задать с помощью параметров link, alink, vlink тега body.

Изменение цвета ссылки:

Текст ссылки

Ссылки внутри страницы.

Бывают еще и внутренние ссылки, т.е. ссылки на какое-то место внутри текущего документа. Внутренние ссылки удобно использовать в больших документах для разбиения на главы.

Итак, чтобы создать внутреннюю ссылку, надо выполнить 2 действия:

Сделать закладку в том месте документа, куда Вы будете ссылаться, и дать этой закладке имя. Для этого в нужном Вам месте вставьте контейнер <a> и его парметру name присвойте строковое значение:

Между тегами <a> и текст можно не писать, т.к. Вам нужно всего лишь пометить место в документе, куда ведет ссылка.

Ну и собственно сделать ссылку, которая будет ссылаться на эту закладку. Для этого Вы делаете обычную ссылку, только вместо адреса документа пишете знак # и имя закладки:

Ссылка на закладку

Вот пример использования внутренних ссылок:

А еще можно делать ссылку на закладку в другом документе. Например, в документе 1.html есть закладка с именем qqq. Чтобы сделать на нее ссылку из текущего документа, надо сделать обычную ссылку на документ1.html и добавить к адресу знак # и имя закладки qqq:

Ссылка на закладку

Ссылка на новое окно.

Все ссылки, заданные выше описанными способами, по умолчанию открываются в текущем окне браузера. Для изменения этого свойства ссылки у тега <a> существует параметр target.

Если Вы хотите, чтобы документ, на который Вы ссылаетесь открывался в новом окне браузера, присвойте параметру target значение _blank:

Ссылка

Если Вы хотите конкретно указать, что ссылка должна открыться в текущем окне, присвойте параметру target значение _self:

Ссылка

Ссылки во фреймах.

А теперь предположим, что Вы используете на своей страничке фреймы. Тогда по умолчанию все ссылки будут открываться в текущем фрейме.

Присвоив параметру target значение _parent, Вы укажете браузеру открывать ссылку во фрейме-родителе:

Ссылка

Присвоив параметру target значение _top, Вы укажете браузеру отменить все фреймы и открыть ссылку в полном окне браузера:

Ссылка

А если Вы хотите, чтобы ссылка открывалась в каком-то конкретном фрейме, присвойте параметру target имя нужного фрейма:

Ссылка

Ссылка на адрес электронной почты.

Ссылка на адрес электронной почты делается как и обычная ссылка, только вместо адреса документа пишетсямailto:адрес электронной почты:

Пишите письма

При нажатии на такую ссылку запустится почтовая программа, установленная по умолчанию, и откроется окно создания нового письма.

Практическая работа № 6

Создание ссылок

1. Цель работы. Изучение основных тегов и атрибутов для создания ссылок в Web-страницах.

2. Литература. Конспект лекций.

3. Подготовка к работе. Изучить конспект лекций по теме "Ссылки".

4. Перечень оборудования. Компьютер.

5. Задание.

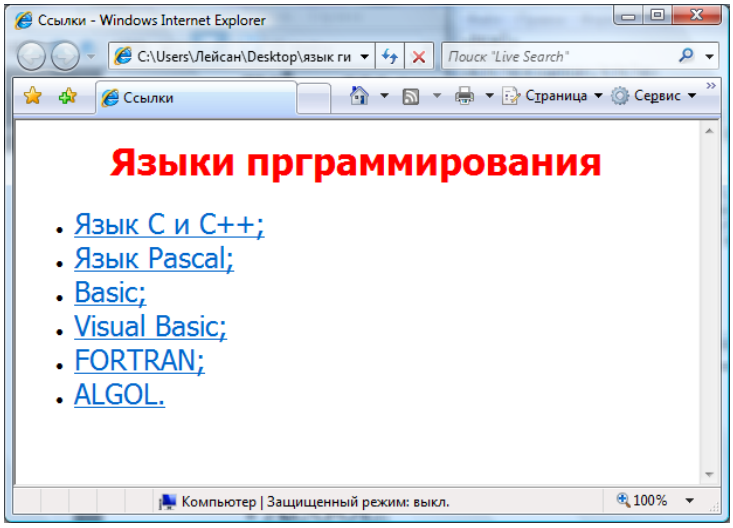
5.1. Изучить основные теги и атрибуты для создания внешних и внутренних ссылок в Web-страницах.

5.2. Научиться создавать ссылки в web-страницах.

6. Порядок выполнения работы.

6.1. **Создание внутренних ссылок.**

1. Откройте текстовый редактор Блокнот и создайте web-страницу в соответствии с образцом:



2. Для создания внутренней ссылки используется тег **….**

3. Задайте цвета гиперссылок при помощи тега <body>: синий цвет для непосещённой ссылки, красный – для активной ссылки и пурпурный для ранее посещенной ссылки.

4. Фрагмент HTML-кода:

```
<ul>
<li><a href=#one>Язык C и C++;</a>
<li><a href=#two>Язык Pascal;</a>
<li><a href=#three>Basic;</a>
<li><a href=#four>Visual Basic;</a>
<li><a href=#five>FORTRAN;</a>
<li><a href=#six>ALGOL.</a>
```

5. Сохраните файл присвоив ему имя **ssilka1.html** и просмотрите полученную web-страницу.

6. Откройте только что созданный файл **ssilka1.html** и добавьте в него текст. Текст находиться в папке **Тексты для сайтов** на рабочем столе.

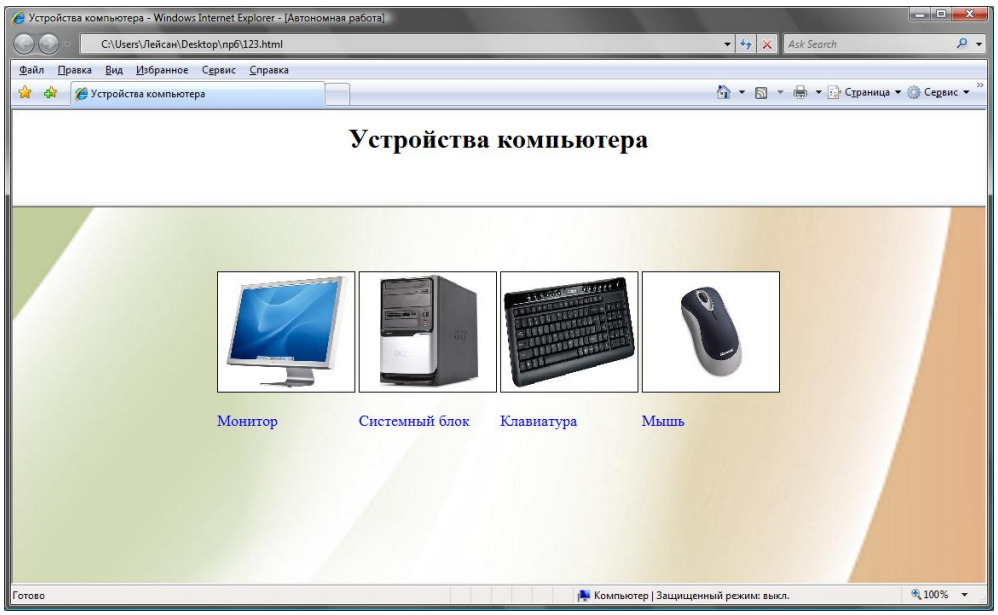
7. Используя метки создайте ссылки на разделы о каждом языке программирования. Проверьте работу созданных ссылок.

8. Сохраните изменения в HTML-файле и в браузере обновите web-страницу.

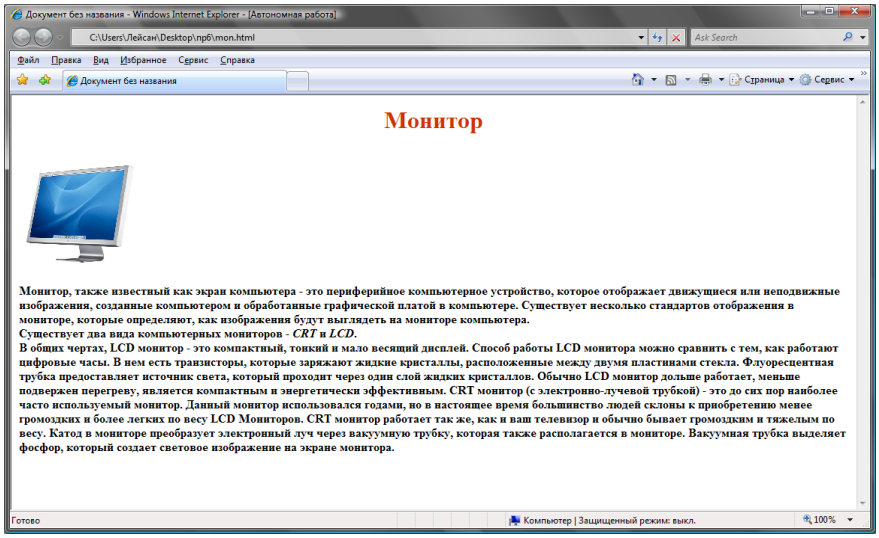
6.2. **Создание внешних ссылок. Создание ссылки на локальный файл.**

1. Создайте страницу **comp.html** в соответствии с образцом:

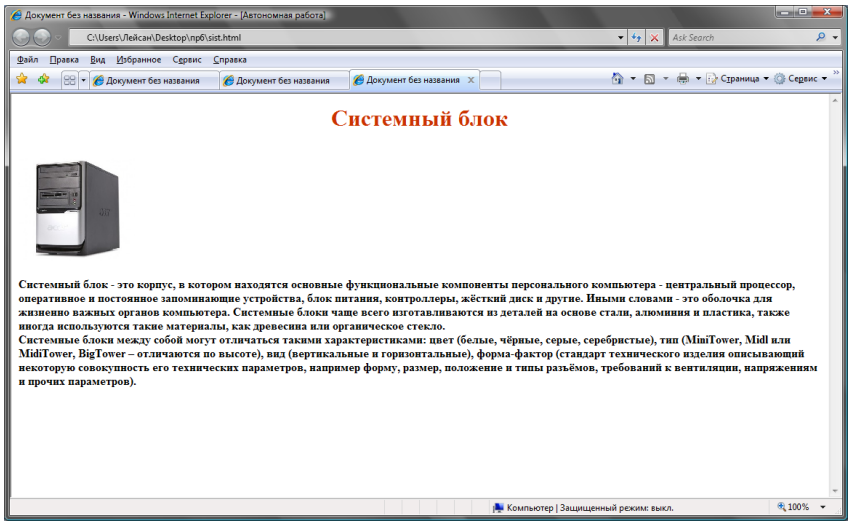
Все необходимые картинки находятся в папке Лабораторные работы по HTML → Картинки на Рабочем столе.



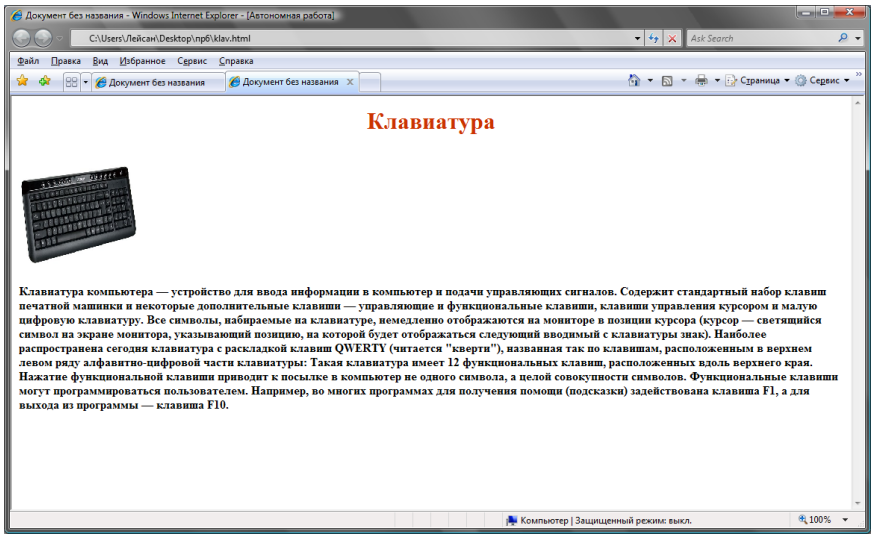
2. Далее создайте еще три страницы **mon.html**, **sist.html, klav.html,** которые содержат информацию о каждом устройстве компьютера.



mon.html



sist.html



klav.html

3. Установите ссылки между этими страницами и страницей **comp.html.** Для этого откройте страницу **comp.html** с помощью блокнота и создайте ссылки для каждой картинки.

4. Для изображения монитора создайте ссылку на файл **mon.html** таким образом, чтобы этот файл открывался в новом окне браузер. Для этого присвойте параметру **target** значение **_blank.**

5. Для изображения системного блока создайте ссылку на файл **sist.html** таким образом, чтобы этот файл открывался в фрейме. Для этого присвойте параметру **target** имя того фрейма, где вы хотите отобразить страницу**.**

6. Для изображения клавиатуры создайте ссылку на файл **klav.html** таким образом, чтобы этот файл открывался в текущем окне. Для этого присвойте параметру **target** значение **_self.**

7. Для изображения мыши создайте ссылку на файл **мышь.doc.**

8. Далее создайте ссылки со страниц **mon.html**, **sist.html, klav.html** на главную страницу **comp.html.**

7. Самостоятельная работа.

7.1. Создайте небольшой сайт (4-5 страниц) с информацией о себе и своей семье (или своей учебной группе, друзьях и пр.).

Обоснуйте выбор структуры сайта и способа навигации по нему.

8. Содержание отчета.

8.1. Файлы **ssilka1.html**, **mon.html**, **sist.html, klav.html,comp.html** в вашей папке.

9. Контрольные вопросы.

9.1. Какой тег предназначен для создания гиперссылки?

9.2. Какой атрибут обеспечивает ссылку на метку в текущем HTML-документе?

9.3. Какой атрибут формирует метку на ссылки?

9.4. Почему в имени метки используется символ #?

9.5. Какой параметр и какого тега определяет цвет А) еще ни разу не использованной ссылки; Б)ссылки, которой уже хотя бы один раз воспользовались; С) ссылки в момент щелчка на ней мышью?

9.6. Какой тег служит для задания ссылок на другие web-документы?

9.7. Как отображаются на экране текстовые и графические гиперссылки?

8. Размещение элементов мультимедиа на web-странице.

Элемент <OBJECT> вставляет в документ графическое изображение, видео клип, апплет JAVA, или элемент управления ActiveX. Единый для всех объектов и модулей элемент <OBJECT> функционально замещает элемент IMG и при необходимости может расширяться. Его атрибуты:

Id (определяет идентификатор документа)

Declare (позволяет *подразумевать* документ, не создавая его)

Classid (определяет URL идентификатора класса или экземпляра объекта)

Codebase (определяет URL местоположения кода)

Data (определяет URL, который указывает на необходимые данные или объект)

Type (обозначает тип содержимого Internet данных в соответствии с атрибутом DATA)

Codetype (обозначает тип содержимого Internet данных в соответствии с атрибутом CLASSID)

Standby (позволяет выводить окна с коротким сообщением в процессе загрузки объекта)

Align (определяет режим выравнивания объекта относительно строки текущего текста или как отдельного элемента)

Width (определяет ширину объекта в окне браузера)

Height (определяет высоту объекта в окне браузера)

Border (задает наличие или отсутствие обрамления объекта при помощи значений 1 и 0 соответственно)

Hspace (позволяет задать дополнительное пространство вокруг объекта)

Vspace (позволяет задать дополнительное пространство вокруг объекта)

Usemap (определяет URL объекта, если тот является картой)

Shapes (используется вместе с объектами, которые обладают точками привязки или активными зонами)

Name (используется с формами HTML, которые могут содержать объект, и определяет, должен ли тот передавать данные серверу при запуске формы)

Alt (определяет альтернативное содержимое, которое будет визуализировано в случае, если пользователь не может или не хочет отобразить объект)

Title (определяет заголовок объекта)

<OBJECT атрибуты>параметры и альтернативное содержание</OBJECT>

Элемент <EMBED> является популярным расширением к языку HTML от Netscape пока он сам и его синтаксис не описаны в спецификации HTML 4.0, но т.к. Microsoft обес-

печила некоторую поддержку этого элемента своими браузерами, то довольно часто применяется при внедрении в страницы мультимедиа содержимого и других файлов. Официально для этих целей Консорциум W3C в спецификации HTML 4,0 рекомендует применять дескриптор <OBJECT>, однако Web-мастера довольно часто используют тэги <EMBED> и </EMBED>, и в зависимости от возможностей браузера элемент обрабатывается либо браузером, либо в специально запущенном приложении.

<EMBED атрибуты> </EMBED>

Элемент может поддерживать следующие атрибуты:

height (задает вертикальный размер, вставляемого объекта)

width (задает горизонтальный размер, вставляемого объекта)

autostart (задает возможность запуска при загрузке, принимает значения true или false)

loop (задает количество повторений, принимает значения true или false)

hidden (позволяет скрыть панель управления, принимает значения true или false)

src (указывает на URL мультимедиа файла)

pluginspage (указывает на URL плагина для проигрывания мультимедиа файла)

bgcolor (задает фон объекта)

type (указывает на тип мультимедиа файла)

quality (указывает на качество мультимедиа файла)

alt (задает альтернативное содержание)

Практическая работа № 7

Размещение звука, видео и flash-анимаций на Web-странице

1. Цель работы. Изучение основных тегов и атрибутов для размещения мультимедиа объектов на Web-страницах.

2. Литература. Конспект лекций.

3. Подготовка к работе. Изучить конспект лекций по теме "Мультимедиа".

4. Перечень оборудования. Компьютер.

5. Задание.

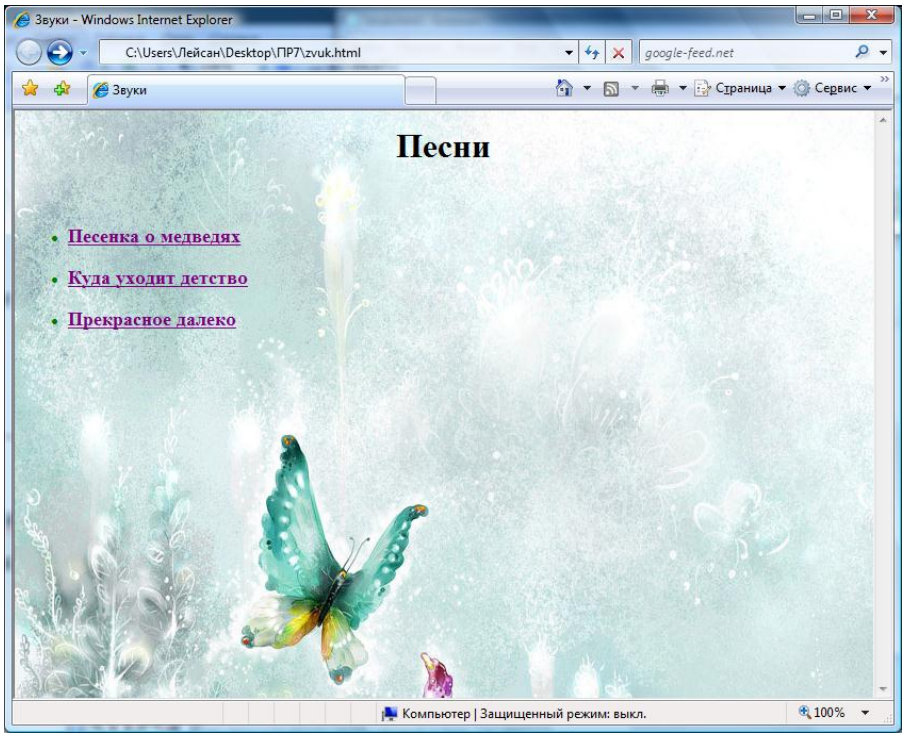
5.1. Изучить основные теги и атрибуты для размещения звука, видео и flash-анимаций на Web-страницах.

5.2. Научиться создавать web-страницы, содержащие мультимедиа объекты.

6. Порядок выполнения работы.

6.1. **Размещение звука на Web-странице.**

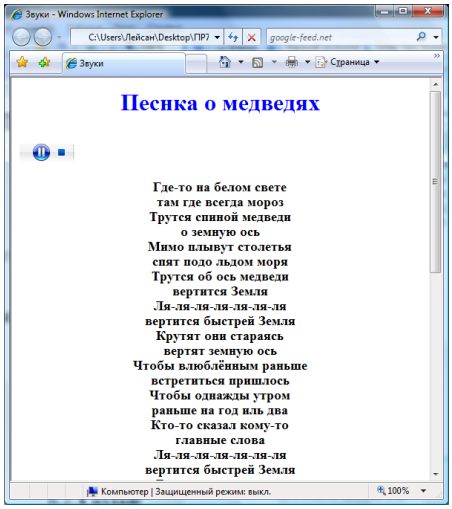
1. Откройте текстовый редактор Блокнот и создайте web-страницу в соответствии с образцом:



2. Сохраните файл присвоив ему имя **zvuk.html** и просмотрите полученную web-страницу.

3. Далее вам необходимо создать еще 3 страницы **zv1.html, zv2.html и zv3.html,** при открытии которых будет играть мелодия**.**

4. Образец файла **zv1.html:**



5. Для внедрения в web-страницу звукового файла или видеоизображения используется тег **<embed src="имя файла"…<embed>.** Чтобы установить ширину и высоту медиапанели используются атрибуты **width** и **height.**

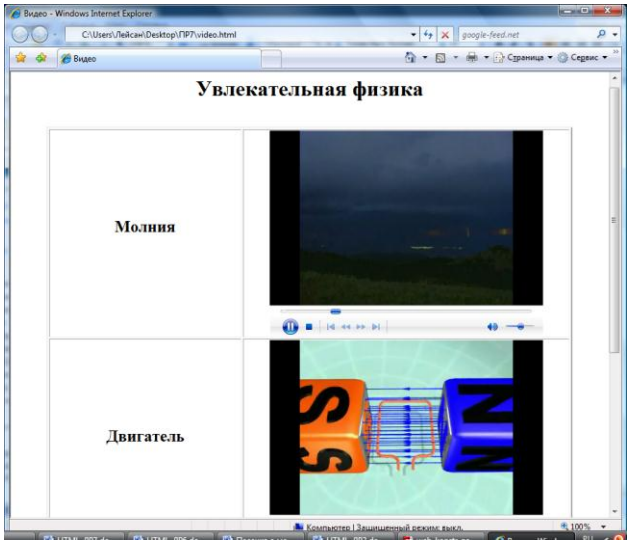
6. Таким же образом создайте файлы **zv2.html и zv3.html.** Вся необходимая информация для этих файлов находится в папках Тексты и Звуки.

7. На странице **zvuk.html** установите ссылки на эти файлы и в браузере обновите web-страницу.

6.2. **Размещение видео на Web-странице.**

1. Откройте текстовый редактор Блокнот и создайте web-страницу с именем **video.html.**

Образец:



2. Оформите ее в виде таблицы, где в первом столбце располагается название видеоролика, а во втором сам видеоролик.

Молния	moln.avi
Двигатель	dvig.avi
Иллюзия	illuz.avi

3. Вставка видеоизображения в ячейку таблицы осуществляется с помощью тега:
<embed src="имя файла" width=500 height=300 autostart=false></embed>

4. Вся необходимая информация находится в папке Видео.

5. Сохраните изменения и обновите страницу.

6.3. **Размещение flash-анимаций на Web-странице.**

1. Создайте страницу **flash.html,** которая содержит различные анимации. Все необходимые файлы находятся в папке Лабораторные работы по HTML→Анимации.

7. Самостоятельная работа.

7.1. Создайте страницу, которая содержит объекты мультимедиа.

8. Содержание отчета.

8.1. Файлы **zvuk.html, zv1.html, zv2.html, zv3.html, video.html, flash.html** в вашей папке.

8. Контрольные вопросы.

9.1. Какой тег предназначен для вставки мультимедиа объектов на web-страницу?

9.2. Какие атрибуты имеет элемент <object>?

9.3. Для чего предназначен элемент <embed>?

9.4. Перечислить атрибуты элемента <embed>?

9. Каскадные таблицы стилей

CSS – Cascading Style Sheets(каскадные таблицы стилей). CSS реализует механизм добавления стилевого оформления к веб-страницам. Основная идея приведшая к появлению CSS, - разделить сами данные и их оформление. Технология CSS похожа на HTML - здесь также есть набор «команд», управляющих оформлением страницы. Но если в HTML «команды» всегда идут вместе с отображаемыми данными (обычно с текстом), то CSS предписывает сначала описать внешний вид структуры документа, а в самом документе размещать только указание на соответствующий тип форматирования – стиль.

Синтаксис стиля

Синтаксис:

селектор {атрибут: значение}

Здесь под селектором подразумевается элемент, для которого задается стиль, далее в фигурных скобках, через ";" записываются пары атрибут:значение, где ":" является заменителем знака "=".

Например, для окрашивания всех заголовков второго уровня в красный цвет достаточно просто указать в описании стиля следующее:

H2 {color: red}

Селекторы могут описывать следующие элементы:

1. **Тэги**

Пример: P {color: red} стиль применяется к тэгу <P>

2. Тэги потомков

Пример: H1 B {color: red} стиль применяется к тэгу <B>, находящемуся внутри заголовка первого уровня, т.е. являющимся его потомком

3. Дочерние тэги

Пример: OL>LI {color: red} стиль применяется к тэгу <LI>, который является дочерним элементом для тэга <OL>, т.е. на тэги <LI>, которые являются дочерними элементами для тэга <UL> стиль не распространяется

4. Атрибуты тэгов

Пример: DIV[class=red] {color: red } стиль применяется ко всем элементам, включенным в <DIV>, у которых атрибут class равен red

5. ID

Пример: P#12 {color: red } стиль применяется к тэгу <P>, у которого идентификатор ID равен 12

6.Классы

Пример: .red {color: red} стиль применяется к любомым тэгам, у которых атрибут class равен red

Включенные таблицы стилей

Пример: P {color:blue} стиль применяется к тэгу <P>, данный единичный абзац будет иметь синий текст.

Внедренные таблицы стилей

Пример:

```
<HTML>
<HEAD>
</HEAD>
<STYLE>
BODY {color: #000000;}
H3 {color: #0000ff;}
A {color: #ff0000;}
</STYLE>
<BODY>
Содержимое
</BODY>
</HTML>
```

Внешние таблицы стилей

```
<LINK rel=stylesheet href="файл.css" type="text/css">
```

либо

<STYLE>

@import URL("_strong_файл.css_/strong_/default.htm");

</STYLE>

Свойства шрифта:

font-family - определяет тип шрифта

font-size - определяет размер шрифта

font-style - определяет стиль начертания шрифта

font-variant - определяет размер начертания шрифта

font-weight - определяет насыщенность начертания шрифта

font-face - определяет загружаемый шрифт с расширением .eot

Свойства текста:

word-spacing - определяет оформление интервала между слов

letter-spacing - определяет оформление интервала между символами

white-space - определяет оформление пробелов в тексте

line-height - определяет межстрочный интервал

text-decoration - определяет оформление начертания шрифта

text-indent - определяет отступ текста

text-align - позволяет изменить режим выравнивания текста text-transform - позволяет изменить регистр шрифта текста

Свойства цвета и фона:

background-color - позволяет задать цвет фона абзаца background-image - позволяет задать фоновый рисунок страницы, указав его URL

background-position - позволяет задать расположение фонового рисунка страницы

background-attachment

background-repeat - атрибут позволяет задать будет ли фоновый рисунок страницы фиксированным

color - атрибут позволяет задать цвет текста страницы, присвоив ему шестнадцатеричное значение.

Существуют дополнительные методы, которые расширяют функциональные возможности таблиц стилей. Стили можно сгруппировать, что позволяет уменьшить количество необходимых атрибутов и аргументов, путем создания логических групп. Группирование таблиц стилей может происходить по одинаковым тэгам, или по атрибутам одного семейства.

Пример группирования по тэгам:

До группирования:

H1 {font-family: arial;font-size: 14pt}

H2 {font-family: arial;font-size: 14pt}

H3 {font-family: arial;font-size: 14pt}

После группирования:

H1,H2,H3 {font-family: arial;font-size: 14pt}

Назначение классов

Для того, чтобы сделать работу более универсальной, можно создать несколько вариантов - классов свойств стилей и применять их к элементам поочередно. Сгруппировав стили по свойствам, необходимо дать им названия в виде расширения к элементу.

Синтаксис:

H1.left {text-align: left}

H1.right {text-align: right}

Далее в документе указывается либо

<H1 class=left>, либо <H1 class=right>

Аналогично можно создать универсальный класс и применять его потом к различным элементам:

Синтаксис:

.left {text-align: left}

.right {text-align: right}

Далее в документе указывается либо

<тэг class=left>, либо <тэг class=right>

Практическая работа № 8

Создание Web-страниц с использованием каскадных таблиц стилей

1. Цель работы. Изучение основ каскадных таблиц стилей и способов их использования.

2. Литература. Конспект лекций.

3. Подготовка к работе. Изучить конспект лекций по теме "Каскадные таблицы стилей".

4. Перечень оборудования. Компьютер.

5. Задание.

5.1. Изучить способы создания и использования каскадных таблиц стилей.

5.2. Научиться создавать стили.

6. Порядок выполнения работы.

6.1. **Стили для шрифтового и абзацного форматирования.**

1. Откройте текстовый редактор Блокнот и создайте web-страницу **st1.html**. Введите описание стилей **h1,h2,p** . Примените их к тексту так, чтобы отформатировать информацию по образцу.

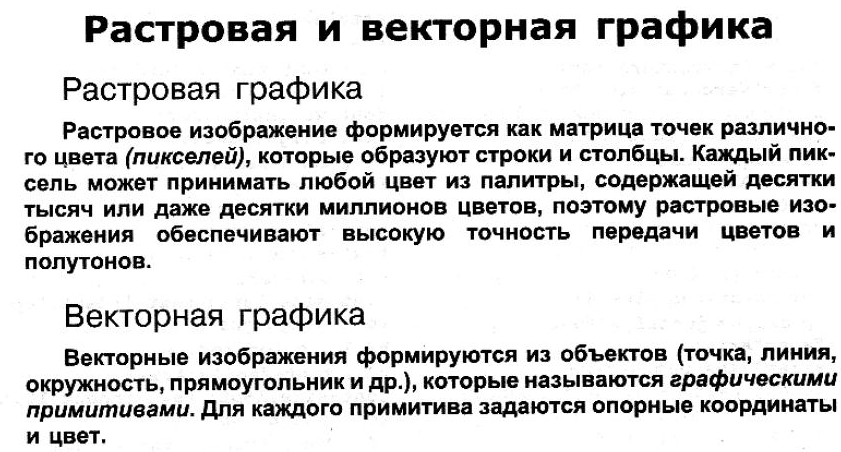
```
<style>
<!--
h1{font-family:Verdana;
  font-size:24pt;
  font-weight:bolder;
  text-align:center;
  }
h2{font-family:Sans-serif;
  font-size:14pt;
  font-weight:bolder;
  text-align:justify;
  }
p {font-family:Arial;
  font-size:18pt;
  text-align:left;
  }
-->
</style>
```

Для стиля **h1** задан шрифт **Verdana**, размер — 24 пункта, жирный шрифт, выравнивание по центру

Для стиля **h2** задан шрифт **Sans-serif**, размер — 14 пунктов, жирный шрифт, выравнивание по ширине

Для стиля **p** задан шрифт **Arial**, размер — 18 пунктов, выравнивание по левому краю

Образец:



Растровая и векторная графика

Растровая графика

Растровое изображение формируется как матрица точек различного цвета *(пикселей)*, которые образуют строки и столбцы. Каждый пиксель может принимать любой цвет из палитры, содержащей десятки тысяч или даже десятки миллионов цветов, поэтому растровые изображения обеспечивают высокую точность передачи цветов и полутонов.

Векторная графика

Векторные изображения формируются из объектов (точка, линия, окружность, прямоугольник и др.), которые называются *графическими примитивами*. Для каждого примитива задаются опорные координаты и цвет.

6.2. **Стили для форматирования списков, для определения цвета элемента и фона.**

1. Откройте текстовый редактор Блокнот и создайте web-страницу **st2.html.** Для этой странице создайте отдельный файл **st2.css,** который должен содержать описание стилей для элементов **body, h1, h2, p ,ol.**

Стиль **body:** цвет фона – светло-голубой.

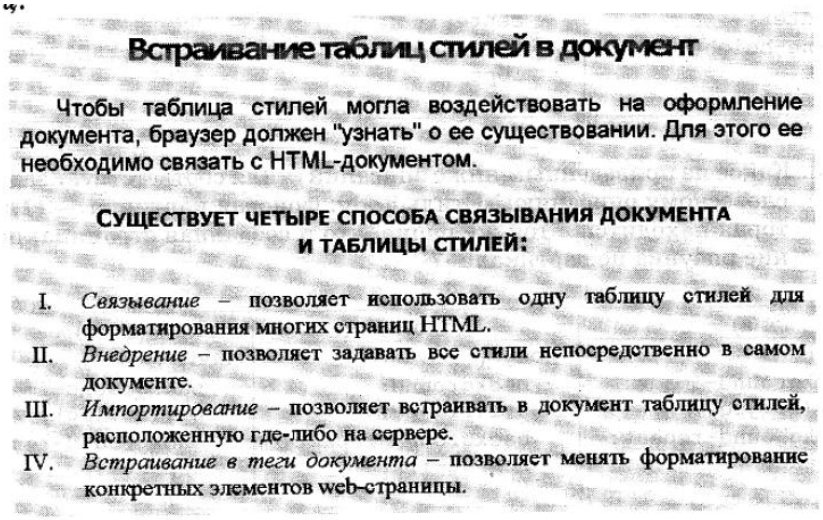
Стиль **h1:** тип шрифта – Verdana, размер шрифта – 20 пунктов, полужирный шрифт, выравнивание по центру, расстояние между буквами – 2 пункта, красный цвет.

Стиль **h2:** тип шрифта – Verdana, размер шрифта – 16 пунктов, полужирный шрифт, выравнивание по центру, расстояние между буквами – 1пункт, синий цвет.

Стиль абзаца **p**: тип шрифта – Arial, размер шрифта – 16 пунктов, отступ первой строки – 13 пунктов, выравнивание по ширине.

Стиль нумерованного списка **ol:** тип нумерации в списке – римские заглавные цифры, размер шрифта – 18 пунктов, расстояние между строками – 90%, выравнивание по ширине.

Образец:



Встраивание таблиц стилей в документ

Чтобы таблица стилей могла воздействовать на оформление документа, браузер должен "узнать" о ее существовании. Для этого ее необходимо связать с HTML-документом.

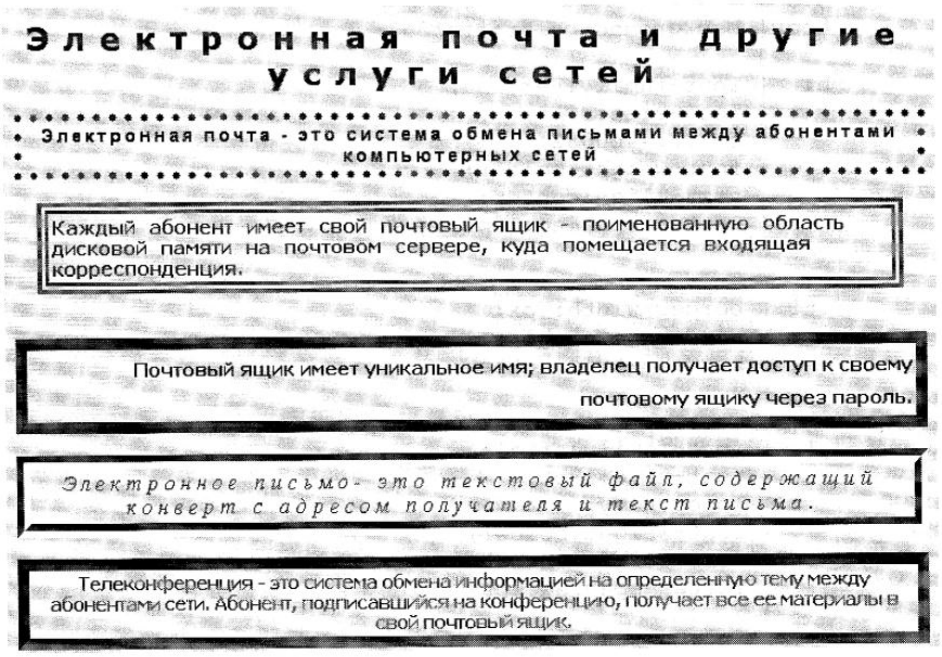
СУЩЕСТВУЕТ ЧЕТЫРЕ СПОСОБА СВЯЗЫВАНИЯ ДОКУМЕНТА И ТАБЛИЦЫ СТИЛЕЙ:

I. *Связывание* – позволяет использовать одну таблицу стилей для форматирования многих страниц HTML.
II. *Внедрение* – позволяет задавать все стили непосредственно в самом документе.
III. *Импортирование* – позволяет встраивать в документ таблицу стилей, расположенную где-либо на сервере.
IV. *Встраивание в теги документа* – позволяет менять форматирование конкретных элементов web-страницы.

6.3. **Стили для форматирования рамок и отступов.**

1. Создайте страницу **st3.html,** с использованием стилей текста, списков, фона. Введите описание стилей **body, h1, .t1, .1, .t2, .2, .t3, .3, .t4, .4, .t5, .5.**

Образец:



Фрагмент HTML-кода:

```
<style>
<!--
body{ background-color:#faf3d2;
    }
h1  { font-family:Verdana;
      font-size:20pt;
      color:#4b0082;
      font-weight:bolder;
      text-align:center;
      letter-spacing:10pt
     }

.t1 { border-width:7px;
      border-color:blue;
      border-style:dotted;
    }

.1  { font-family:arial;
      font-size:12pt;
      color:navy;
      font-weight:bolder;
      text-align:center;
      letter-spacing:2pt
    }
```

7. Самостоятельная работа.

7.1. Создайте web-страницу **1.html** с использованием стилей шрифтового текста, списка, фона и цвета.

7.2. Создайте web-страницу **2.html** с использованием размещения стилевой таблицы на отдельном файле.

Образцы страниц и все необходимые файлы находятся в папке Лабораторные работы по HTML→CSS.

8. Содержание отчета.

8.1. Файлы **st1.html, st2.html, st3.html, 1.html, 2.html, st2.css** в вашей папке.

9. Контрольные вопросы.

9.1. Как расшифровывается аббревиатура CSS?

9.2. Каково назначение технологии CSS?

9.3. Как задаются параметры стиля?

9.4. Какой HTML-контейнер служит для размещения стилевых описаний?

9.5. С помощью, каких параметров стиля определяется абзацное и шрифтовое форматирование?

9.6. Каким образом определяется цвет элемента и фона?

9.7. Как можно задать стиль для рамок?

9.8. Какие возможны способы внедрения стилей в html-документ?

10. Основные этапы создания сайта

Разработка проекта

Первый и один из самых важных этапов при создании сайта, разработка проекта. В проекте ставится техническое задание, определяется целевая аудитория, количество разделов сайта, наполнение (текст, фото и т.д.) продумывается дизайн, при необходимости определяется система управления и многие другие особенности будущего сайта. Для чего нужен проект? В процессе создания сайта участвуют специалисты совершенно разных направлений, на основе проекта каждый участник процесса точно определяет объем и поставленные перед ним задачи. Только после разработки проекта можно приступать к работе по созданию сайта.

Наполнение

К этому этапу нужно подойти очень серьезно, так как именно наполнение в дальнейшем будут просматривать пользователи, и индексировать поисковые системы. Как

правило, контент (наполнение сайта) делится на несколько разновидностей: текстовое наполнение именно его индексируют поисковые роботы, обязательно должно быть уникальным и seo-оптимизированным (определенное количество ключевых слов в тексте). Фото (изображения, картинки) видео, аудио наполнение очень часто используется для оформления, или демонстрации продукта или услуг, можно сказать что это одежка по которой встречают. Главные критерии контента: уникальность и качество.

Дизайн

Принято считать, что дизайн это только визуальная часть, но при разработке дизайна сайта кроме его оформления должны быть правильно расставлены элементы управления, с учетом правил изабилити и других технических особенностей необходимых для быстрой и качественной работы сайта так же для правильной индексации поисковыми системами. Дизайн сайта имеет ряд особенностей и кардинально отличается от всех остальных направлений визуального оформления. Для разработки качественного дизайна обязательно нужно знать основные языки, используемые при написании сайтов и технологии продвижения в поисковых системах, только при соблюдении всех правил можно рассчитывать на успешное развитие и качественную работу веб-ресурса.

Программирование

После того как поставлено техническое задание (разработан проект), и готов дизайн сайта начинается верстка сайта (программирование). Очень важный, трудоемкий этап разработки веб-ресурса, именно от качества программирования зависит на сколько хорошо он будет работать, и индексироваться поисковыми системами. Как правило, сроки программирования на прямую зависят от технических особенностей и функциональных возможностей сайта (чем больше функциональные возможности сайта, тем сложнее его программировать). Качество программирование полностью зависит от уровня специалиста, который будет осуществлять верстку, на первый взгляд абсолютно одинаковые части сайта могут быть написаны на совершенно разных языках, это определяет человек, который будет программировать веб-сайт.

Домен

Домен - имя сайта, состоящее только из латинских букв или цифр (идентификатор сайт). Именно по доменному имени находят Ваш сайт в интернете, введя его в адресную строку браузера. Домены делятся на разные виды, по географическому признаку, международные и т.д. соответственно для корректной и максимально эффективной работы сайта, нужно правильно его выбрать. Доменное имя должно быть не длинным, хорошо запоминаемым, ассоциироваться с товаром или услугой предоставленной на сайте, если сайт

работает на территории города или одной страны, то лучше выбрать домен соответствующего региона.

Хостинг

Хостинг - это удаленный компьютер, на котором располагается сайт. Для эффективной работы сайта нужно обеспечить круглосуточный и бесперебойный доступ к нему, своими силами это сделать очень сложно, так как нужен компьютер который будет подключен к сети 24 часа в сутки, и соответствующее программное обеспечение, которое требует настройки и постоянной поддержки. Вопрос с расположением сайта решают хостинг провайдеры, которые дают в аренду определенное дисковое пространство на своих серверах. Не рекомендую экономить на хостинге, так как от него зависит качество работы Вашего сайта, желательно оплачивать услуги по аренде хостинга и доменного имени не менее чем на год.

Поддержка

После того как сайт разработан и начал свою работу требуется его дальнейшая поддержка в продвижении и развитии. Все зависит от типа сайта, например: сайт визитка, как правило, после разработки и размещении на хостинге не требует особой поддержки, так как информация на нем обновляется не очень часто. Интернет магазин наоборот, требует ежедневной работы над ним, ежедневно нужно заниматься его продвижением, и удержанием позиций в поисковых системах. Так же под поддержка иногда подразумевается несущественное изменение (усовершенствование) дизайна или технической части сайта. Простыми словами к поддержке относятся все вопросы по содержанию сайта после его разработки и первичной настройки.

Продвижение

После того как сайт полностью готов, для того чтоб его могли находить пользователи по определенным запросам его необходимо внести в поисковые системы и каталоги сайтов. Но не все так просто как кажется на первый взгляд, на сегодняшний день конкуренция в интернете очень большая, и для того чтобы Ваш сайт занял наивысшие места в поисковиках ему необходимо поисковое продвижение. На основе многочисленных исследований можно сделать вывод, что основная масса пользователей не просматривает сайты, которые находятся на второй странице выдачи поисковых систем, соответственно чтоб сайт был максимально эффективен, он должен находиться в первой десятке выдачи. Даже если сайт очень хорошо seo-оптимизировать то все равно маловероятно, что он займет нужные позиции в поисковых системах, в этом случае выход только один дополнительный комплекс мер по его продвижению.

Практическая работа № 9

Создание сайта с использованием языка разметки гипертекста HTML и каскадных таблиц стилей.

1. Цель работы. Изучение основных этапов создания Web-сайтов.

2. Литература. Конспект лекций.

3. Подготовка к работе. Изучить конспект лекций по теме "Основы работы в сети Интернет".

4. Перечень оборудования. Компьютер.

5. Задание.

5.1. Изучить основные этапы создания web-сайтов и размещение их в сети Интернет.

5.2. Научиться создавать статические web-сайты.

6. Порядок выполнения работы.

6.1. **Разработка структуры сайта.**

Структуру сайта можно условно разделить на внешнюю и внутреннюю. Внутренняя структура зависит от того, какую информацию вы будете размещать, какие у вас есть материалы. Вам предстоит решить, какие у вас будут на сайте разделы, подразделы,т.е. вам предстоит продумать древо сайта.

Внутренняя структура на примере домашней страницы:

1. **Главная страница**
2. **Фотографии**
3. **Контактная информация**

На этом можно остановиться, а можно создать более сложную внутреннюю структуру.

1.Главная страница

2.Фотографии

2.1.На море

2.2.У бабушки в деревне

2.3.Мои друзья

3.Мое творчество

3.1.Картины

3.2.Музыка

3.3.Проза

4.Гостевая книга

5.Контактная информация

Внешней структура определяет расположение основных значимых элементов на каждой странице.

Например:

При разработке внешней и внутренней структуры ваша главная задача сделать так, чтобы в будущем посетителю было легко ориентироваться на вашем сайте, чтобы важная и нужная информация легко находилась.

6.2. **Оформление внешнего вида сайта.**

Создайте Главную страницу сайта, продумайте дизайн. Для оформления страницы создайте таблицу стилей.

6.3. **Разработка остальных страниц сайта.**

Создайте другие страницы Вашего сайта. Все они должны быть одного стиля. Создайте ссылки между Главной страницей и остальными.

6.3. **Размещение сайта в сети Интернет.**

Когда сайт готов, его выкладывают в сеть (интернет). Для этого вам нужно завести для сайта доменное имя (адрес), и хостинг (место для вашей страницы).

Разметить ваш сайт в интернете, а именно **в narod.ru**

7. Самостоятельная работа.

Перечень тем для создания сайта:

1. **Компьютерный магазин.**
2. **Магазин "Одежда".**
3. **Автотранспортное предприятие.**
4. **Студия WEB-дизайна.**
5. **Строительная компания.**
6. **Магазин "Продукты".**
7. **Косметический салон.**
8. **Туристическая компания.**
9. **Аптека.**
10. **Мир моих увлечений.**

8. Содержание отчета.

8.1. Web-сайт по определенной теме.

8. Контрольные вопросы.

9.1. Какие этапы включает в себя процесс создания сайта?

9.2. Какие существуют способы и методы создания сайтов?

9.3. Классификация сайтов.

9.4. Какие программы используются для создания сайтов?

9.5. Как создается структура сайта?

9.6. Каким образом можно создать дизайн сайта?

9.7. Что такое хостинг и доменное имя?

9.8. Как разместить разработанный сайт в интернете?

9.9. Каким образом организовать поддержку и продвижение сайта?

Литература.

1. О.Б. Богомолова Web-конструирование на HTML: практикум БИНОМ Лаборатория знаний, 2008 – 192с.
2. Н.В. Комолова Самоучитель HTML – СПб.:Питер,2008. – 268с.
3. Гончаров А. HTML в примерах. С.-Пб.: Питер, 2003

Содержание

Из опыта работы учителя математики и информатики

МБОУ «Кубянская сош» Атнинского муниципального района РТ

Хакимзяновой Н.И.

422740

РТ, Атнинский район

с. Кубян, ул. Школьная, д.2

kubjan2@mail.ru

8(843)6936311

2015 год