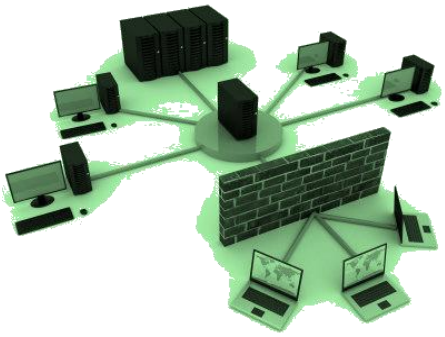
**ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ КОЛЛЕДЖ ЭЛЕКТРОНИКИ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ СТУДЕНТОВ ПО ПРОВЕДЕНИЮ
ПРАКТИЧЕСКИХ РАБОТ
ПО ДИСЦИПЛИНЕ**

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНЫХ СЕТЕЙ И WEB-СЕРВЕРОВ



для специальности
230111 Компьютерные сети
(№ специальности и её наименование)

Рязань 2013, октябрь

© Ларина Марина Евгеньевна, 2013 год, РКЭ

ОДОБРЕНО
Цикловой комиссией
ВТ и ИТ
ОГБОУ СПО «РКЭ»

Председатель ЦК
_____

Заместитель
Директора по УМиВР_____
/Е.И.Мирчетич/

Авторы:
Ларина Марина Евгеньевна, преподаватель высшей категории спецдисциплин, председатель предметно-цикловой комиссии вычислительной техники и информационных технологий

© Ларина Марина Евгеньевна, 2013 год, РКЭ

Введение

Настоящее руководство содержит описания 9 практических работ по курсу «Программное обеспечение компьютерных сетей и web-серверов» для студентов 3-го курса специальности 230111 – Компьютерные сети.

Каждая работа содержит теоретические пояснения необходимые для выполнения работы, порядок выполнения практической части, а также теоретические вопросы, ориентированные на понимание и анализ полученных результатов. Зачет за каждую работу выставляется при качественном выполнении практической работы, правильном и грамотном оформлении отчета и устной защите работы (заключающейся в ответе на вопросы преподавателя). Работы выполняются с разбиением студентов на бригады по 2 человека. Отчёт по каждой работе студент выполняет индивидуально на специальном бланке установленного образца.

В руководство включены практические работы:

«Изучение браузеров» 4
«Работа с почтовым клиентом Outlook Express» 6
«Утилиты ОС для диагностики и анализа сети» 7
«Изучение брандмауэра Windows XP» 10
«Создание простейшей WEB страницы. Форматирование текста в HTML. Создание ссылок» 15
«Таблицы на WEB странице» 21
«Фреймы» 23
«Каскадные таблицы стилей CSS (Cascading Style Sheets)» 26
«Разработка Java-скриптов и их внедрение с web-страницу» 30

© Ларина Марина Евгеньевна, 2013 год, РКЭ

Практическая работа №1
«Изучение браузеров»

Цель работы: Познакомиться с некоторыми популярными браузерами, научиться производить их тонкую настройку, выделить отличительные особенности, осуществлять настройку параметров безопасности.

Теоретические сведения:

При работе с системой WWW пользователи имеют дело с программами-клиентами системы, называемыми *браузерами*

Браузер (от англ. browse - листать, просматривать) - программа, с помощью которой пользователь организует диалог с системой WWW, просматривает web-страницы, взаимодействует с web-серверами и другими ресурсами в Internet.

Браузеры умеют взаимодействовать с любыми типами серверов, используя при этом их собственные протоколы. Информацию, полученную от любого сервера, браузер выводит на экран в стандартной, удобной для восприятия форме. Браузер может быть графическим или текстовым. Для того чтобы браузеры могли интерпретировать документы они должны быть оформлены специальным образом - размечены в соответствии со спецификацией языка html. В основе языка html лежит логическая разметка текста, это позволяет текстовым браузерам выделять заголовки либо строчными буквами, либо пустыми строками сверху и снизу. Графический браузер может выделить заголовки повышенным кеглем, курсивом.

Первым графическим браузером была программа Mosaic разработанная в национальном центре суперкомпьютерных приложений США (NCSA). В 1994 г. была образована корпорация Netscape Communications, а её основателями были создатели Mosaic. Вскоре появилась коммерческая версия графического браузера Netscape. Netscape Navigator 2.0 умел поддерживать Java-апплеты и язык JavaScript. К 1996 г. Netscape обладала полным господством в мире веб-браузеров.

В начале 1996 г. на рынке появляется браузер Microsoft Internet Explorer 2.0, а летом того же года Microsoft Internet Explorer 3.0. В дальнейшем появились версии Microsoft Internet Explorer 4.0, Microsoft Internet Explorer 5.0, Microsoft Internet Explorer 6.0 и т.д.

Основное назначение браузеров - это просмотр и сохранение web-страниц однако в настоящее время все больше альтернативных браузеров, обладающих гибкой настройкой различных параметров и расширенными возможностями. Кроме того есть специальные надстройки и оптимизаторы для разных типов браузеров, например для тех кто привык использовать определенный браузер и не хочет переходить на работу с другим.

Ход работы:

1. Запустите браузер Internet Explorer.
2. Ознакомьтесь с основными элементами:
   А) команды меню;
   Б) навигационная панель браузера;
   В) адресная строка;
   Г) справочная система;
3. Какие сервисы Интернета поддерживает данный браузер (только ли просмотр web-страниц или другие сервисы, например IRC, FTP, RSS).
4. Внимательно изучите меню и возможности настроек браузера. Ответьте на ряд вопросов:
   А) Как изменить кодировку текста?
   Б) Как отключить показ рисунков (звук, анимацию)?
   В) В каких видах можно сохранить загруженную страницу?
   Г) Как изменить домашнюю страницу?
   Д) Как изменить вид ссылки (цвет)?
   Е) Как добавить или удалить кнопки на панели?
5. Основные особенности настройки журнала посещений.
6. Временные файлы Интернета.
   А) Как указать путь к папке?
   Б) Как изменить время хранения файлов?

© Ларина Марина Евгеньевна, 2013 год, РКЭ

В) Удаление временных файлов Интернета.

7. В чем заключается настройка безопасности. Какие параметры можно изменить?
8. Настройка конфиденциальности. Основные параметры.
9. Как блокировать всплывающие окна?
10. Как сделать так чтобы при открытии HTML документов браузер автоматически осуществлял подключение к Интернету?
11. Как сделать, чтобы при окончании загрузки файла браузер делал уведомление?
12. Запустите браузер Opera. Повторите задания с данным браузером. Внимательно изучите меню и навигационную панель. (Обратите внимание, что аналогичные возможности в разных браузерах могут по-разному называться и по-другому настраиваться.)
13. После изучения всех элементов и настроек выделите отличительные особенности данного браузера от Internet Explorer.
14. Запустите браузер FireFox. Повторите указанные действия.

Отчет по работе должен содержать:

1. Краткое описание ответов на вопросы по каждому браузеру.
2. Указание отличительных особенностей.
3. Вывод о проделанной работе, удобстве и необходимости тех или иных функций браузера.

Рекомендации:

Отчет удобно оформить в виде таблицы, в которой указать ответы на вопросы и перечислить особенности. Например:

	Internet Explorer	Opera	FireFox	Google Chrome	Safari
Отключение рисунков анимации					
Настройка параметров безопасности					
Способ блокировки всплывающих окон					
Указание способа работы через Proxy					
Изменение кодировки текста					
Папка хранения временных файлов интернета.					
Способ очистки кэша браузера					
Добавление кнопок на панель инструментов					
Настройка журнала посещений					
Закладки или избранное для быстрого открытия страниц					
Способ задания домашней страницы					
Поддерживаемые сервисы (FTP, RSS)					
Способ блокирования адреса сайта					

Контрольные вопросы:

1. Как отключить рисунки и видео на загружаемых страницах?
2. Как добавить кнопки на инструментальную панель браузера?
3. Как запретить доступ к определенным узлам?

© Ларина Марина Евгеньевна, 2013 год, РКЭ

4. Как установить страницу, с которой по умолчанию будет начинаться загрузка браузера?
5. Оцените настройки параметров безопасности работы в сети у нескольких браузеров.

Практическая работа №2
«Работа с почтовым клиентом Outlook Express»

Цель работы: знакомство с почтовым клиентом Outlook Express, создание учетной записи, настройка программы, отправка и прием сообщений, работа с адресной книгой.

Теоретические сведения:

Электронная почта - средство обмена информацией, подготовленной в электронном виде, между людьми, имеющими доступ к компьютерной сети.

Почтовый сервер - программа, пересылающая сообщения из почтовых ящиков на другие серверы или на компьютер пользователя по запросу его почтового клиента.

На почтовом сервере создают почтовые ящики для пользователей с определенным именем и паролем для доступа. Клиентскую часть сервиса представляет почтовый клиент.

Почтовый клиент (мейлер) - программа, помогающая составлять и посылать электронные сообщения, получать и отображать письма на компьютере пользователя.

Работа с почтой может проводиться в режиме off-line. Это означает, что писать письма и читать полученные с сервера сообщения можно в автономном режиме, то есть, без подключения к Интернету.

Одним из наиболее популярных почтовых клиентов является программа Microsoft Outlook Express. Программа позволяет отправлять и принимать письма, имеет гибкие настройки как интерфейса программы, так и организации переписки. Составлять почтовые сообщения можно как в формате обычного текста, так и в формате HTML. Читать полученные сообщения можно и в окне Outlook Express, и в отдельном окне. Для организации почты Outlook Express предлагает средства создания почтовых папок и определения правил, по которым приходящие сообщения должны помещаться в эти папки. Кроме того, программа имеет встроенные средства для работы с группами новостей.

Почтовая программа Outlook поставляется в двух возможных конфигурациях: Outlook Express и MS Outlook (входит в пакет Microsoft Office).

Выполнение работы:

1. Запустите программу (с помощью значка на рабочем столе или из пуска). С помощью меню «Справка» получите сведения о версии программы и запишите их в отчет.

2. Познакомьтесь с основными элементами интерфейса:

А) основное меню

Б) панель инструментов (запишите элементы панели)

В) окно папок (запишите, какие папки здесь расположены, умейте охарактеризовать их назначение и содержимое)

Г) окно контактов

Д) окно заголовков

Е) окно отображения текста сообщения

3. Создайте новую папку, в которой будут сохраняться полученные вами сообщения. Она должна быть вложенной в папку «Входящие».

4. Создайте контакт, который будет занесен в адресную книгу. Запишите в отчет, какие сведения о контактах могут быть указаны.

5. Как изменить вид кодировки сообщения и размер шрифта отображения, полученного сообщения.

6. Измените, вид панели инструментов, и запишите, как это можно сделать.

7. Запишите, как сделать, чтобы область просмотра сообщений была расположена под заголовками сообщений (рядом с заголовками).

8. Настройка интерфейса программы. Как убрать панель инструментов (окно контактов).

9. Какими способами можно осуществить доставку почты. Запишите в отчет.

10. Как добавить отправителя в список блокируемых? Запишите в отчет.

© Ларина Марина Евгеньевна, 2013 год, РКЭ

11. По каким признакам можно произвести сортировку сообщений?
12. Создайте новую учетную запись с адресом почтового ящика, содержащим номер вашей рабочей машины (меню Сервис-Учетные записи). **Сетевое имя вашего компьютера будет первой частью адреса вашего почтового ящика** (например, macminiномер машины).
В качестве сервера входящей и исходящей почты укажите **imacteacher**. Адрес должен быть создан в форме **сетевое имя машины@domain.local**. После этого просмотрите свойства созданной учетной записи и на вкладке «Подключения» поставить галочку «Подключаться, используя локальную сеть». Кроме того на вкладке «серверы» поставьте галочку «Проверка подлинности пользователя» и галочка на вкладке «Запомнить пароль». На вкладке «Общие» должна стоить галочка «Использовать для получения почты или синхронизации». На вкладке «Дополнительно» обратите внимание на номера портов, их следует оставить по умолчанию (25 и 110).
Запишите в отчет, какие данные необходимы для создания учетной записи.
13. Внимательно изучите все вкладки в опции Сервис-Параметры.

А) Запишите, как настроить программу на получение уведомления о прочтении вашего письма.

Б) Как изменить формат отправляемых сообщений (текст, HTML).

В) Как изменить шрифт создаваемого сообщения.

Г) Внимательно изучите параметры настройки безопасности на соответствующей вкладке.

Д) Создайте свою подпись, которая будет автоматически добавляться ко всем исходящим сообщениям.

14. Создайте сообщение с использованием одного из существующих шаблонов оформления. Добавьте к своему письму статус высокой важности.
15. Отправьте сообщение на соседнюю машину.
16. Одно из сообщений из папки «Входящие» перешлите по двум адресам соседних машин.
17. Отправьте сообщение с вложенным файлом.
18. После выполненных заданий очистите список контактов, все папки. Запишите в отчет как это можно сделать.
19. Создание и изменение удостоверения. Что такое удостоверения и для чего оно используется?
20. Создание правил для сообщений. Назначение и особенности правил.

Отчет по проделанной работе должен содержать:

1. Запись ответов на поставленные вопросы.
2. Вывод об основных возможностях почтового клиента.

Контрольные вопросы:

1. Как изменить окно программы, добавить и удалить кнопки на панель?
2. Алгоритм создания учетной записи.
3. Создание удостоверения. Что это и зачем нужно?
4. Как запретить получение почты с определенного адреса?
5. Создайте правило запрещающее получение почты со словом «купить».

Практическая работа №3
«Утилиты ОС для диагностики и анализа сети»

Цель работы: определить наличие встроенных в ОС Windows 2000 утилит для анализа и диагностики сетевых характеристик ПК, определить их назначение и особенности с помощью встроенной справки, провести диагностику ПК.

Теоретические сведения:

Утилиты запускаются из командной строки или из сеанса MS-DOS. Некоторые утилиты в разных вариантах операционных систем могут отсутствовать.

Набор средств для диагностики TCP/IP включает в себя:

- PING. Проверяет связь с IP-хостом.
- TRACERT (Traceroute). Отображает адреса всех маршрутизаторов на пути от клиента до удаленного хоста.

© Ларина Марина Евгеньевна, 2013 год, РКЭ

- NSLOOKUP. Сетевая утилита Windows NT.
- IPCONFIG. Отображает текущую конфигурацию сети TCP/IP.
- NETSTAT. Отображает статистику и текущие соединения по протоколу TCP/IP.
- ROUTE. Позволяет конфигурировать сетевые маршрутные таблицы.
- ARP (Address Resolution Protocol) - протокол преобразования адресов. Отображает IP-адреса в адреса Ethernet.
- FTP (File Transfer Protocol) - протокол передачи файлов. Позволяет выводить список файлов, осуществлять передачу файлов, а также управлять каталогами на удаленной системе.

Рассмотрим назначение и действие данных утилит чуть подробнее:

Утилита PING.

Используется для проверки коннективности с удаленным хостом. Действует посредством посылки IMCP пакетов и ожидания ответа в течение 1 секунды (значение по умолчанию). Посылается 4 одинаковых пакета (значение по умолчанию). На экран выводится время в миллисекундах, затраченное на ожидание отклика.

Формат команды:

ping [-n значение1][-w значение2][-t] IP-address или DNS-имя удаленного хоста

Пример. **C:\>ping -w 10000 bash.ac.ru**

Ключ -n используется для указания числа посылаемых пакетов. Если связь между хостами плохая, Вы увидите, сколько пакетов было потеряно. К сожалению, хосты некоторых организаций не принимают ICMP пакеты (которые используются утилитами, подобными ping) с помощью маршрутизаторов и брандмауэров. Например, ftp.microsoft.com не пингуется, хотя зайти на этот сервер не составляет труда. Так как некоторые вирусы атакуют удаленные хосты с помощью ping, возможно, именно по этой причине блокируется получение ping-пакетов.

Утилита TRACERT.

Утилита tracert используется для отслеживания маршрута пакета, посланного текущим хостом удаленному. Она может показаться более удобной и содержательной, чем ping, особенно в тех случаях, когда удаленный хост недостижим. Вы сможете определить район проблем со связью (у Вашего Интернет-провайдера, в опорной сети либо в сети удаленного хоста) по тому, насколько далеко будет отслежен маршрут. Если Вы увидите строку со зведочками (*) либо с сообщениями типа "Destination net unreachable" , "Destination host unreachable"или"Request time out", возможно, Вы обнаружили район проблем со связью.

Формат команды:

tracert [-d][-h количество ретрансляций][-j список_систем][-w тайм-аут] IP-address или DNS-имя удаленного хоста.

Утилита tracert срабатывает следующим образом: посылается по 3 пробных пакета на каждый хост, через который проходит маршрут до удаленного хоста. Утилита tracert использует параметр time-to-live (TTL) для ограничения времени прохождения пакета по маршруту, на котором каждый хост обнаруживается. TTL -это количество "скачков" или последовательных хостов, через которые разрешается пройти пакету. Стартуя со значения равного 1, TTL возрастает до тех пор пока, либо пакет не достигнет удаленного хоста либо не будет достигнуто максимальное значение "скачков" (30 по умолчанию).

C:\>tracert –d –h 16 ftp.microsoft.com

Данную утилиту также можно использовать для определения скорости действия путей.

Утилита NSLOOKUP.

Эта утилита поможет получить Вам много информации, определить причину возникших проблем. Эта утилита, например, может быть использована для определения DNS-серверов домена microsoft.com.

Утилита IPCONFIG .

По умолчанию данная утилита выводит только IP-адрес, маску подсети и шлюз по умолчанию для данного сетевого адаптера.

C:\WINDOWS>ipconfig

Также позволяет вручную продлевать или прекращать DHCP-аренду. Для более подробной информации используйте ключ /ALL. Этот ключ позволяет для каждого сетевого интерфейса, помимо информации по умолчанию, получить имя узла, адреса серверов DNS, тип запроса

© Ларина Марина Евгеньевна, 2013 год, РКЭ

NetBIOS, используется ли для данного сетевого интерфейса DHCP, а также аппаратный адрес интерфейса.

Утилита NETSTAT.

Утилита NETSTAT выводит статистику для протоколов (TCP, IP, ICMP, UDP) и информацию об IP-соединениях. Формат команды:

netstat [-a][-e][-n][-s][-p имя][-r][интервал]

-a Отображение всех подключений и портов, на которых компьютер ожидает соединения. (Подключения со стороны сервера обычно не отображаются).

-e Отображение статистики Ethernet. Этот ключ может применяться вместе с ключом -s.

-n Отображение адресов и номеров портов в числовом формате, без попыток определения имен.

-p протокол -- Отображение подключений для протокола "имя": tcp или udp. Используется вместе с ключом -s для отображения статистики по протоколам. Допустимые значения "имя": tcp, udp или ip.

-r Отображение содержимого таблицы маршрутов (таблица маршрутизации).

-s Отображение подробной статистики по протоколам. По умолчанию выводятся данные для TCP, UDP и IP. Ключ **-p** позволяет указать подмножество выводящихся данных по определенному протоколу.

интервал - повторный вывод статистических данных через указанный интервал в секундах. Для прекращения вывода данных нажмите клавиши CTRL+C. Если параметр не задан, сведения о текущей конфигурации выводятся один раз.

Утилита ROUTE.

Является утилитой TCP/IP, которая используется для создания или модификации статических таблиц маршрутизации на компьютере, работающем под управлением Windows NT Server.

Формат команды:

route [-f][-p][команда [адресат][маска][шлюз][метрика]]

Где:

-f -удаление всех записей для шлюзов. Если этот параметр используется в сочетании с другими, то сначала производится удаление записей для шлюзов.

-p-добавить (при помощи команды Add) постоянные записи. По умолчанию добавляемые записи не сохраняются при перезапуске системы. Все постоянные пути могут быть выведены на экран при помощи команды print.

команда - может быть указана одна из четырех команд: print (вывести список путей), add (добавить путь), delete (удалить путь), change (изменить существующий путь).

Адресат - определяет компьютер, которому нужно послать указанную команду.

маска - определяет маску подсети для указанного пути. По умолчанию используется 255.255.255.255

шлюз -шлюз для указанного пути.

метрика-установка поля метрика в таблице маршрутизации в указанное значение. Может быть задано любое значение от 1 до 9999.

Все произведенные изменения в статической таблице маршрутизации будут потеряны после перезапуска системы, чтобы произвести постоянные изменения, используйте команду **ROUTE** с параметром **-p**.

Утилита ARP.

Утилита просмотра и модификации ARP-кэша. Адреса можно вводить в ARP-кэш вручную при помощи утилиты ARP.exe. Служит для вывода и изменения записей кэша протокола ARP, который содержит одну или несколько таблиц, использующихся для хранения IP-адресов и соответствующих им физических адресов Ethernet или Token Ring. Для каждого сетевого адаптера Ethernet или Token Ring, установленного в компьютере, используется отдельная таблица. Запущенная без параметров, команда **arp** выводит справку.

Формат команды:

arp -s IP-адрес MAC-адрес.

-a-вывод записей, содержащихся в ARP-кэше.

-q-вывод записей, содержащихся в ARP-кэше (этот ключ недоступен в Windows for Workgroups).

© Ларина Марина Евгеньевна, 2013 год, РКЭ

-N-вывод записей, содержащихся в ARP-кэше и относящихся к определенному сетевому интерфейсу.
-s -добавление статической записи в ARP-кэш. Синтаксис: ARP -s IP-адрес MAC-адрес.
-d -удаление статической записи из ARP-кэша.

Утилита HOSTNAME.

Отображение имени узла, входящего в состав полного имени компьютера.
Синтаксис:
hostname
Параметры:
/?
Отображение справки в командной строке.

Ход работы:

1. Запустите командную строку.
2. Определите сетевое имя Вашего компьютера. Информацию о сетевом адаптере.
3. Определите имя сервера Вашей сети, маску подсети.
4. Проверьте связь с удаленным хостом. Измените количество посылаемых пакетов на 10 и время ожидания. Запишите формат команды с измененными параметрами и результат в отчет.
5. Определите MAC – адрес Вашего ПК.
6. Определите путь до сервера Вашей подсети.
7. Экспериментально проверьте работу команд: PING, TRACERT, NSLOOKUP, IPCONFIG, NETSTAT, ROUTE, HOSTNAME, **ARP.**
8. Запишите в тетрадь результат выполнения команд.

Отчет должен содержать краткий конспект с описанием команд (их назначение, формат записи), а также результат их выполнения полученный экспериментально.

Контрольные вопросы:

1. Какими способами можно определить маршруты к удаленным компьютерам в различных сетях? Приведите варианты команд.
2. Какими способами можно определить сетевые имена и DNS-адреса соседних ПК или сервера в локальной сети?
3. Как определить физические адреса серверов в локальной сети?
4. Какими способами можно определить IP-адреса удалённых ПК в Интернет?
5. Команды вида **arp** и **arp –a**. Содержательный смысл. Доставляемая информация.

Практическая работа №4
«Изучение брандмауэра Windows XP»

Цель работы: узнать параметры настройки стандартного брандмауэра, научиться создавать различные типы исключений

Брандмауэр позволяет обеспечить безопасность системы путем ограничения трафика, который поступает от других компьютеров, и предоставления пользователю возможности контролировать локальные данные. (запишите в отчет)

Кроме того, брандмауэр защищает систему от несанкционированного подключения других пользователей и программ (в т.ч. вирусов и компьютерных червей).

Брандмауэр Windows

Брандмауэр можно представить себе в виде защитного барьера, на котором осуществляется проверка поступающих из Интернета или другой сети данных. В зависимости от настроенных параметров брандмауэр блокирует трафик или пропускает его на компьютер.

© Ларина Марина Евгеньевна, 2013 год, РКЭ

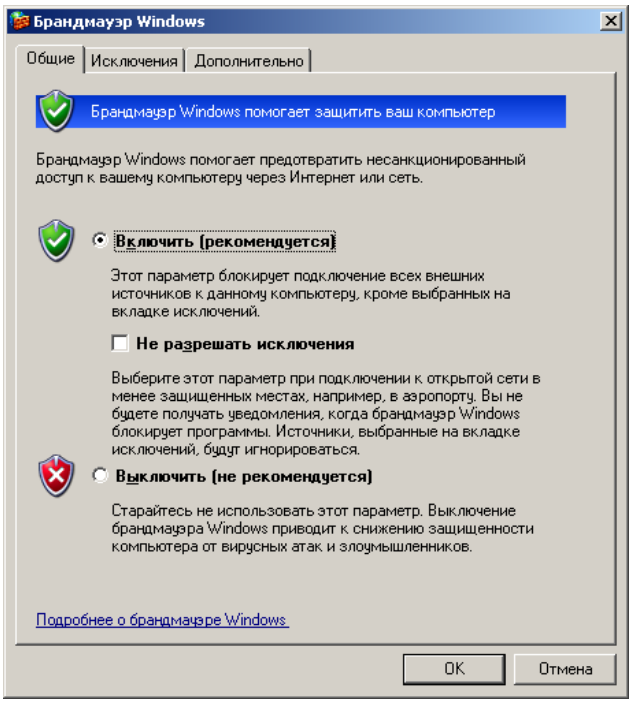
В Windows XP с пакетом обновления 2 (SP2) пользователь может включать и выключать брандмауэр. Несмотря на то, что брандмауэр Windows включен по умолчанию, он может быть отключен производителем компьютера или сетевым администратором. Вы не обязаны использовать на своем компьютере брандмауэр Windows, а можете установить брандмауэр любого стороннего производителя, предварительно взвесив преимущества каждого из них и выбрав наиболее подходящий. В случае использования брандмауэра стороннего производителя брандмауэр Windows следует отключить.

Чтобы настроить параметры брандмауэра Windows, выполните следующие действия.

1. Выберите в меню **Пуск** пункт **Выполнить**, введите команду **wscui.cpl** и нажмите кнопку **ОК**.

2. В диалоговом окне **Центр обеспечения безопасности Windows** щелкните ссылку **Брандмауэр Windows**.

Запишите в отчет способы запуска брандмауэра.



Диалоговое окно **Брандмауэр Windows** содержит следующие вкладки.

- **Общие**
- **Исключения**
- **Дополнительно**

На вкладке **Общие** расположены следующие элементы управления.

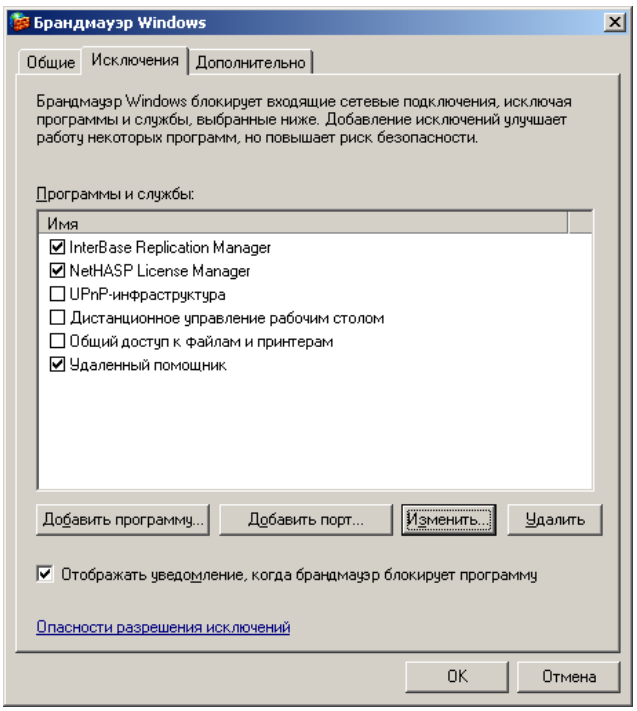
- **Включить (рекомендуется)**
- **Не разрешать исключения**
- **Выключить (не рекомендуется)**

Запишите в отчет, какие вкладки и элементы управления находятся в настройках брандмауэра.

Если установлен флажок **Не разрешать исключения**, брандмауэр Windows блокирует все запросы на подключение к компьютеру, включая запросы от программ и служб, перечисленных на вкладке **Исключения**. Кроме того, брандмауэр блокирует совместный доступ к файлам и принтерам, а также обнаружение сетевых устройств.

Использовать брандмауэр Windows в этом режиме целесообразно при подключении к сетям общего пользования, например сетям в гостиницах и аэропортах. Защита компьютера обеспечивается путем блокирования всех попыток установить к нему подключение.

Брандмауэр Windows в режиме «Не разрешать исключения» позволяет просматривать веб-страницы, обмениваться сообщениями электронной почты или немедленными сообщениями.



На вкладке **Исключения** можно создать исключение ***для порта или программы***, чтобы разрешить получение определенных видов входящего трафика. Каждому исключению можно назначить область.

Для домашних и небольших офисных сетей рекомендуется ограничивать такую область пределами локальной сети. В этом случае установленные на компьютере программы доступны другим компьютерам в составе той же подсети, а трафик из удаленных сетей блокируется.

Вкладка **Дополнительно** служит для выполнения следующих действий:

© Ларина Марина Евгеньевна, 2013 год, РКЭ

- настройка правил для подключений, которые применяются к каждому сетевому интерфейсу;
- настройка параметров ведения журнала безопасности;
- настройка глобальных правил для трафика по протоколу ICMP (Internet Control Message Protocol), который служит для передачи сведений об ошибках и состоянии;
- восстановление параметров по умолчанию.

Настройка исключений

Брандмауэр Windows содержит несколько предопределенных исключений, которые разрешают сетевой доступ для некоторых общих системных задач***, таких как удаленное управление и совместное использование файлов и принтеров.*** В таблице перечислены включенные по умолчанию исключения брандмауэра Windows, а также открытые порты и использующие их программы.

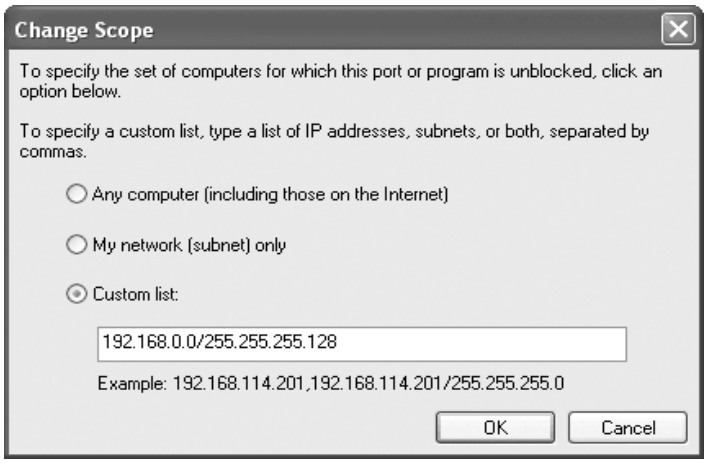
Исключение	Открываемые порты	Разрешающая программа	Ограничить источник по
Разрешить удаленное управление	TCP 135, TCP 445,		Подсети
Разрешить совместное использование файлов и принтеров	UDP 137, UDP 138, TCP 139, TCP 445,		Подсети
Удаленный рабочий стол	TCP 3389		Любой адрес
Allow UPnP Framework	UDP 1900, TCP 2869		Подсети
Удаленный помощник		Sessmgr.exe	Любой адрес
Параметры ICMP	Разрешить входящий запрос эхо		

Запишите в отчет, что такое исключение и какие исключения заданы в работе брандмауэра по умолчанию. Проверьте настройки на соответствующей вкладке. Запишите таблицу в отчет.

Администратор может также задать собственный набор исключений локально через приложение "Брандмауэр Windows" в панели управления или удаленно с использованием механизма групповых политик. Для этого достаточно просто указать имя программы, которая генерирует трафик или сетевые параметры (номер порта TCP или UDP) и адрес источника, а затем разрешить сформированное исключение в брандмауэре Windows.

Предопределенные исключения предоставляют более гибкие возможности настройки, чем пользовательские, поскольку в них допускается задание нескольких портов в одном правиле. Например, исключение для совместного доступа к файлам и принтерам разрешает использование портов TCP 139, TCP 445, UDP 137 и UDP 138.

При создании пользовательского исключения система позволяет указать только один порт, так что для открытия диапазона портов необходимо создать набор исключений для каждого порта. При этом допускается создание пользовательских областей, т.е. адресов IP или диапазонов адресов IP, для которых разрешен обмен. Это позволяет задавать наборы портов для предопределенных и для пользовательских исключений.



Администраторы корпоративных сетей для настройки брандмауэра Windows могут использовать групповые политики. При этом необходимо указать порт (например, 80), транспорт

© Ларина Марина Евгеньевна, 2013 год, РКЭ

(TCP или UDP), область, статус (разрешен или запрещен), и имя соединения. Конструкция имеет вид Порт:Транспорт:Область:Статус:Имя (Port:Transport:Scope:Status:Name).

Ведение журнала

Можно настроить брандмауэр Windows для ведения журнала в виде текстового файла на локальном компьютере или сетевом диске. Брандмауэр Windows может записывать события блокирования пакетов и успешных подключений. Журнал содержит необходимые сведения для устранения неисправностей и ошибок при невозможности подключиться к необходимым ресурсам или просмотре разрешенных соединений.

По умолчанию брандмауэр Windows выдает пользователю предупреждение о том, что некоторая программа пытается использовать указанный порт. При централизованной настройке брандмауэра Windows через групповые политики администратор может по своему желанию отключить выдачу предупреждений пользователям.

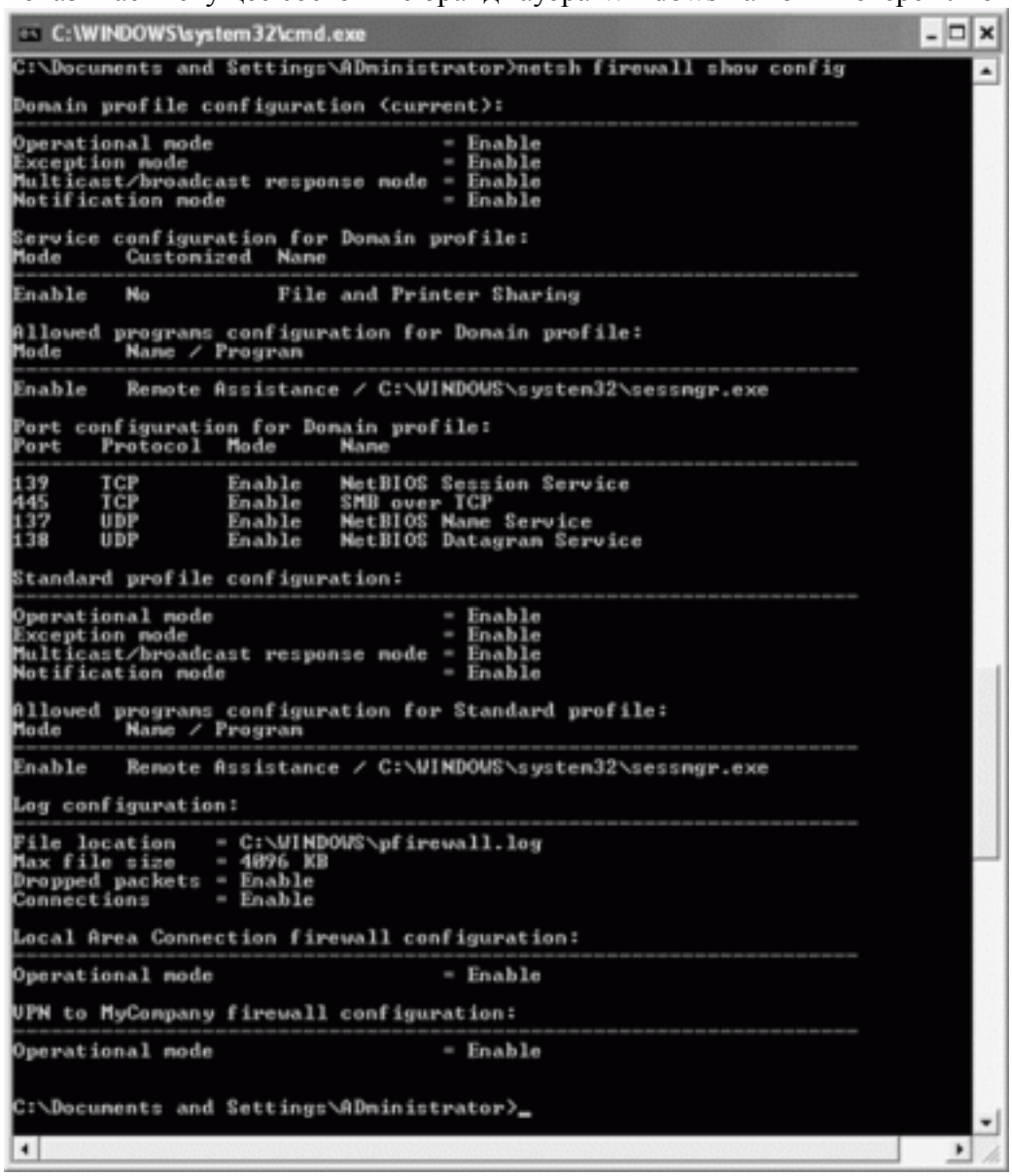
Запишите в отчет, как создать файл журнала и указать путь для его хранения. После его создания, просмотрите его содержимое, предварительно осуществив какое-либо сетевое подключение.

Управление брандмауэром Windows из командной строки

Программа Netsh позволяет выполнять настройку клиентов брандмауэра Windows из командной строки напрямую или через пакетный файл. Например, команда

netsh firewall show config

показывает текущее состояние брандмауэра Windows на компьютере клиента.



Выполните команду, запишите команду и ее назначение в отчет.

© Ларина Марина Евгеньевна, 2013 год, РКЭ

Netsh позволяет управлять почти всеми параметрами брандмауэра Windows, так что администраторы, активно использующие в работе сценарии, могут использовать сценарии подключения к домену для настройки и проверки работы брандмауэра Windows.

Например, приведенная ниже команда Netsh создает локальное правило брандмауэра, разрешающее доступ по Telnet к компьютеру, защищенному брандмауэром Windows, с указанных адресов.

```
netsh firewall add portopening
 protocol = TCP port = 23
 name = Telnet mode = ENABLE
 scope = CUSTOM addresses =
 192.168.0.0/255.255.255.0,
 10.0.0.0/255.255.240.0
```

Эту команду можно было сократить, исключив имена атрибутов:

```
netsh firewall add portopening
 TCP 23 Telnet ENABLE CUSTOM
 192.168.0.0/255.255.255.0,
 10.0.0.0/255.255.240.0
```

Результат исполнения приведенных выше команд можно проверить с помощью команды

```
netsh firewall show portopening
```

Как работает брандмауэр Windows в режиме повышенной безопасности

Появился, только начиная с Windows Vista.

Брандмауэр Windows в режиме повышенной безопасности – это брандмауэр, регистрирующий состояние сети, для рабочих станций. В отличие от брандмауэров для маршрутизаторов, которые развертывают на шлюзе между локальной сетью и Интернетом, брандмауэр Windows создан для работы на отдельных компьютерах. Он отслеживает только трафик рабочей станции: трафик, приходящий на IP-адрес данного компьютера, и исходящий трафик самого компьютера. Брандмауэр Windows в режиме повышенной безопасности выполняет следующие основные операции:

- Входящий пакет проверяется и сравнивается со списком разрешенного трафика. Если пакет соответствует одному из значений списка, брандмауэр Windows передает пакет протоколу TCP/IP для дальнейшей обработки. Если пакет не соответствует ни одному из значений списка, брандмауэр Windows блокирует пакет, и в том случае, если включено протоколирование, создает запись в файле журнала.

Список разрешенного трафика формируется двумя путями:

- Когда подключение, контролируемое брандмауэром Windows в режиме повышенной безопасности, отправляет пакет, брандмауэр создает значение в списке разрешающее прием ответного трафика. Для соответствующего входящего трафика потребуется дополнительное разрешение.
- Когда Вы создаете разрешающее правило брандмауэра Windows в режиме повышенной безопасности, трафик, для которого создано соответствующее правило, будет разрешен на компьютере с работающим брандмауэром Windows. Этот компьютер будет принимать явно разрешенный входящий трафик в режимах работы в качестве сервера, клиентского компьютера или узла однораноговой сети.

Первым шагом по решению проблем, связанных с Брандмауэром Windows, является проверка того, какой профиль является активным. Брандмауэр Windows в режиме повышенной безопасности является приложением, отслеживающим сетевое окружение. Профиль брандмауэра Windows меняется при изменении сетевого окружения. Профиль представляет собой набор настроек и правил, который применяется в зависимости от сетевого окружения и действующих сетевых подключений.

Брандмауэр различает три типа сетевого окружения: домен, публичная и частная сети. Доменом является сетевое окружение, в котором подключения проходят аутентификацию на контроллере домена. По умолчанию все другие типы сетевых подключений рассматриваются как публичные сети. При обнаружении нового подключения Windows Vista предлагает пользователю указать, является ли данная сеть частной или публичной. Общий профиль предназначен для

© Ларина Марина Евгеньевна, 2013 год, РКЭ

использования в общественных местах, например, в аэропортах или кафе. Частный профиль предназначен для использования дома или в офисе, а также в защищенной сети. Чтобы определить сеть как частную, пользователь должен обладать соответствующими административными полномочиями.

Хотя компьютер может быть подключен одновременно к сетям разного типа, активным может быть только один профиль.

Оформите отчет по проделанной работе, который должен содержать:

1. Запишите в отчет все, что указано по тексту работы.

Контрольные вопросы:

1. Зачем на вкладке «Дополнительно» используется настройка протокола ICMP?
2. В каком случае исключения создаются автоматически (то есть мы не заносим программу вручную в список запрещенных)?
3. Какая информация сохраняется в файле журнала брандмауэра. Чем она может помочь?
4. Какие типы сетевого окружения поддерживает брандмауэр в режиме повышенной безопасности.

Практическая работа №5

«Создание простейшей WEB страницы. Форматирование текста в HTML. Создание ссылок»

Цель работы: ознакомление с языком HTML и создание простейшего документа; приобретение навыков форматирования текста. познакомиться с тэгом для создания гиперссылок, научиться создавать локальные и глобальные ссылки, метки, графическую карту ссылок (сенсорное изображение).

Язык HTML основан на использовании тэгов. **Тэги - это команды, которые указывают браузеру, где и как расположить на странице текст и картинки, а также позволяют организовывать ссылки на другие документы.** Для того, чтобы браузер не путал тэги с текстом, их отмечают треугольными скобками < и >. *Большинство тэгов, как правило, **используются парами**. Т.е. первым идет открывающий тэг, он объясняет браузеру, что делать с последующим текстом. Затем следует закрывающий тэг, который указывает область действия первого. Отличается закрывающий тэг от открывающего наличием в нем косой черты - слэша.*

Тэги не чувствительны к регистру. Это означает, что <TITLE>, <title> и <TitLe> для браузера абсолютно одинаковы.

Теперь рассмотрим, из чего, в основном, состоит HTML документ. Сам документ ограничивается тэгами <HTML> и </HTML> - браузер должен знать, где страница начинается, и где заканчивается. Состоит из заголовка (теги <HEAD> и </HEAD>), содержащего различную информацию о документе, и, собственно, тела документа (теги <BODY></BODY>), т.е. того, что вы видите в окне браузера. В заголовке находится заглавие (или название) страницы, рекомендуется не более 64 символов, теги < TITLE> и <TITLE>.

1. Создание шаблона Web-страницы

1.1. Запустите текстовый редактор <Блокнот>.

1.2. Наберите или создайте шаблон Web-страницы:

```
<HTML>
   <head>
       <title>Моя Web-страница</title>
  </head>
  <body>
        текст страницы
  </body>
</HTML>
```

1.3. Сохраните этот файл под именем lab1 с расширением .html.

© Ларина Марина Евгеньевна, 2013 год, РКЭ

1.4. Просмотрите результат. Для этого запустите имеющийся Web-браузер. Выберите команду Файл -> Открыть -> Найдите ваш файл. В окне браузера должен появиться текст страницы, а сама страница иметь заголовок.

<u>!!!! Далее все задания делайте по порядку, располагая их внутри тэга <BODY> </BODY>!!!!!</u>

У каждого тега может присутствовать определенное свойство, присваиваемое ему с помощью атрибута! Атрибуты записываются внутри тега (внутри треугольных скобок) после самого тега и с помощью знака = присваивается значение атрибута!!!

2. Заголовки

HTML имеет 6 уровней заголовков разделов страницы, имеющих номера от 1 до 6. Заголовок высшего уровня имеет номер 1. Отличаются заголовки от текста и между собой толщиной и высотой букв, в заголовке <H1> самые крупные символы. Тэги <H1>..</H1>, <H2>..</H2>,..., <H6>..</H6>.

2.1. Откройте ваш текстовый файл. Используя тэг <H1>…</H1>, создайте заголовок для вашего текста.

Текст заголовка: Основные сведения о языке HTML

Сохраните изменения.

2.2. Чтобы браузер отобразил модифицированный документ, перезагрузите его в браузере. Для этого достаточно щелкнуть на кнопке <Обновить>.

2.3. Используя ***атрибут ALIGN***, выровняйте заголовок по центру. Формат заголовка должен выглядеть следующим образом:

<H1 ALIGN=CENTER>Заголовок</H1>.

3. Форматирование текста с помощью символов разрыва абзаца и строк.

3.1. Тэги <P> </P> и
 позволяют добавить в документ пробелы, чтобы отделить один блок текста от другого. Тэг <P> используется для вставки символа разрыва абзаца, а
 - символа разрыва строки. Вставляя символ разрыва абзаца, вы тем самым даете браузеру команду завершить текущий абзац и вставить пустую строку перед следующим абзацем. Вставляя символ разрыва строки, вы даете браузеру команду завершить текущую строку и перейти на следующую.

3.2. Добавьте на страницу текст практической работы и отформатируйте его используя указанные теги.

3.3. Текст абзаца выровняйте по левому краю. Для этого используется атрибут ALIGN, например:

<P ALIGN=LEFT> Этот текст будет выровнен по левому краю. </P>

4. Создание линий

4.1. Тэг <HR> интерпретируется и отображается браузером как горизонтальная линия, протянувшаяся вдоль всего окна браузера.

4.2. Добавьте этот тэг перед заголовками текста.

4.3. В конце документа, используя дополнительные атрибуты SIZE, WIDTH, ALIGN, COLOR создайте линию с такими параметрами: создайте линию синего цвета, расположенную посередине экрана, толщина = 30 пикселей, длина = 70% от ширины окна браузера.

5. Создание списков

Список – это набор абзацев, которые либо пронумерованы, либо сгруппированы как-то иначе, например, начинаются с одинакового символа (маркера).

Простейший вид списков – *<u>ul (ненумерованный).</u>* Второй популярный вид списка – *<u>нумерованный – ol</u>*. Данный тег имеет атрибут type, который позволяет указать способ нумерации.

type=	Результат
A a	A, B, C a, b, c
I i	I, II, III i, ii, iii
1	1, 2, 3

Создайте на своей странице несколько видов списков.

6. Создание иллюстраций

В свою страничку можно вставить рисунок. Для этого нужен сам рисунок - обычно используют изображения в формате GIF (файлы с расширениями .gif) и JPEG (файлы с

© Ларина Марина Евгеньевна, 2013 год, РКЭ

расширениями .jpg и .jpeg) и строчка HTML-тексте, указывающая браузеру, где этот рисунок находится:

<IMG SRC="picture.gif"> - рисунок "picture.gif" находится в том же каталоге, что и текущий документ;

<IMG SRC="images/picture.gif"> - перейти в подкаталог "images" текущего каталога и взять файл оттуда;

<IMG SRC="../images/picture.gif"> - подняться вверх на один шаг, перейти в каталог "images" и взять рисунок оттуда;

<IMG SRC="http://www.fortunecity.com/business/fax/339/artics/practic/myphoto.gif"> - указание полного пути к файлу.

Обычно применяется, если рисунок находится на другом сервере, или используется картинка с другого сайта.

У данного тэга могут быть следующие атрибуты:

ALT="какой-то текст" - альтернативный текст. *Текст, который появляется вместо картинок при просмотре странички браузером с отключенной автоматической загрузкой изображений (отключают многие - для экономии времени на медленных линиях связи).*

BORDER="n" - рамка вокруг рисунка, где n - ее толщина в пикселах. При n="0" рамка не рисуется.

WIDTH="n" - задается ширина изображения в пикселах или в процентах от ширины экрана браузера (тогда после n следует знак %).

HEIGHT="n" - высота изображения в пикселах или в процентах от высоты экрана.

HSPACE="n" - отступ слева и справа от картинки шириной в n пикселов (т.е. свободное пространство между рисунком и текстом или чем-то иным).

VSPACE="n" - вертикальный отступ от картинки в пикселах.

ALIGN=CENTER - по умолчанию сопровождающий текст выравнивается по нижнему краю рисунка. Применяя атрибут ALIGN можно выравнивать текст по верхнему краю (TOP) или по центру рисунка (CENTER или MIDDLE). Чтобы текст обтекал рисунок, используется значение атрибута LEFT или RIGHT, рисунок прижимается соответственно к левому или правому краю экрана.

Значения n перечисленных выше атрибутов по умолчанию равны нулю.

7. Цвет фона на WEB странице

Цвет фона определяется значением атрибута BGCOLOR - шестнадцатиричным числом вида #hhhhhh, где первая пара символов является значением насыщенности для красного цвета, вторая - для зеленого, и третья пара - для синего цвета. #00 - минимальная насыщенность, #FF - максимальная насыщенность цвета. Можно использовать текстовое определение цвета: BLUE, RED, FUCHSIA и т.п - одно из шестнадцати допустимых значений.

BLACK	#000000	GREEN	#008000
SILVER	#C0C0C0	LIME	#00FF00
GRAY	#808080	OLIVE	#808000
WHITE	#FFFFFF	YELLOW	#FFFF00
MAROON	#800000	NAVY	#000080
RED	#FF0000	BLUE	#0000FF
PURPLE	#800080	TEAL	#008080
FUCHSIA	#FF00FF	AQUA	#00FFFF

Например, чтобы сделать цвет фона страницы фиолетовым, можно написать
<BODY BGCOLOR="#FF00FF">, а можно и так:
<BODY BGCOLOR="PURPLE">.

Фоном может служить и какое-то изображение. При этом заданная область заполняется множественными копиями этого изображения (как кафельной плиткой). Для этого используется атрибут BACKGROUND с указанием пути к рисунку. Например

© Ларина Марина Евгеньевна, 2013 год, РКЭ

<BODY BACKGROUND="graph/picture.gif">

Атрибуты BGCOLOR и BACKGROUND могут находиться внутри метки <BODY>, тогда цвет фона устанавливается для всей страницы, либо в метках <TABLE>, <TR> и <TD> - устанавливается цвет фона во всей таблице, ее строке и ее ячейке.

По умолчанию фоновый рисунок прокручивается вместе с текстом документа. Если зафиксировать фоновый рисунок, то можно добиться эффекта водяных знаков. Для этого используется атрибут BGPROPERTIES=«fixed».

8. Размер и цвет текста на WEB странице

Для изменения цвета и размера шрифта используется тэг <FONT></FONT>. Ее атрибуты:

COLOR="#hhhhhh" - цвет текста;

SIZE="n" - размер шрифта;

FACE="имя_шрифта" - смена шрифта, где имя_шрифта может быть arial, verdana, helvetica, sans-serif и другие.

Пример использования тэга <FONT>:

<FONT SIZE="5" COLOR="RED" FACE="SANS-SERIF">Крупный шрифт красного цвета без засечек</FONT>

9. Создание бегущей строки

Тэг marquee предназначен для вывода в окне браузера так называемой "бегущей строки", или "бегущего текста".

Введем простейшие тэги

<marquee>Это бегущая строка</marquee>

На статичном экране появится движущийся текст.

К тексту, который заключен в тэги <marquee>, применима **любая** разметка HTML. Сделаем, например, его красным и жирным:

<marquee>
**
Это бегущая строка с размеченным текстом
**
</marquee>

Тэг marquee имеет много атрибутов, которые позволяют управлять представлением бегущего текста. Например, атрибутом bgcolor можно определить фоновый цвет:

<marquee bgcolor="yellow">Это бегущая строка с желтым фоновым цветом </marquee>

Атрибут behavior позволяет определить общее поведение бегущей строки. В приведенных ранее примерах для этого атрибута принято значение по умолчанию scroll: текст вдвигается с правой стороны, продолжается до достижения другой стороны, полностью выдвигается, а затем процесс повторяется. Если определить атрибут behavior="alternate", то текст, при достижении любой границы начинает двигаться в противоположном направлении:

<marquee behavior="alternate">Это бегущая строка с чередующимся направлением движения</marquee>

Атрибут direction позволяет определить направление движения текста. Значением по умолчанию для этого атрибута является left, но можно задать и значение right:

<marquee direction="right">Это бегущая строка с направлением движения вправо</marquee>

Это бегущая строка с направлением движения вправо

Атрибуту direction можно также присваивать значения up или down, например

<marquee direction="up">Это бегущая строка с направлением движения вверх</marquee>

Это бегущая строка с направлением движения вверх

С помощью атрибутов ширины width и высоты height можно определить размеры области (в пикселах) для бегущего текста, например:

<marquee bgcolor="yellow" width="200" height="30">Это бегущая строка с определенными размерами области</marquee>

Это бегущая строка с определенными размерами области

Атрибуты scrollamount и scrolldelay предназначены для управления скоростью и плавностью движения текста. Значение атрибута scrollamount задает число пикселов, на которое смещается текст на каждом шаге (по умолчанию принимается 10 пикселов), а значение атрибута scrolldelay

© Ларина Марина Евгеньевна, 2013 год, РКЭ

задает число миллисекунд между последовательными шагами движения текста (по умолчанию принимается около 10 миллисекунд). Для иллюстрации быстрого движения определим scrollamount="30":

<marquee scrollamount="30">Это бегущая строка с более быстрым движением</marquee>

10. Создание ссылок

Ссылка (*links*) может быть на локальный ресурс (файл, страницу на данной машине) или глобальный (адрес в Интернете). Создается ссылка с помощью тега <a> у которого имеется специальный атрибут, указывающий куда ссылаться. Пример:

<body bgcolor="#ffffff">
Переходи к Yahoo!
</body>

При наведении на ссылку (синее и подчеркнутое слово Yahoo) указатель мыши превращается в поднятый указательный палец (обычно такая форма указателя называется hand - рука). В качестве ссылки можно использовать рисунок или даже части рисунка (графическая карта ссылок).

Ссылка **<u>может содержать атрибут TITLE, описывающий ссылку. Если пользователь наведет указатель мыши на ссылку, то появится ее описание - содержимое этого атрибута</u>**.

11.Внутренняя ссылка на часть того же документа

Можно делать ссылки внутри документа. Это удобно для перехода к разным главам одной большой статьи. Для этого необходимо создать некую опорную точку - анкер (anchor - якорь, англ.), на которую и будет происходить переход. Разберем это на примере. Допустим, нам надо перейти на начало урока в электронном учебнике. Для чего создадим такой анкер:

<A NAME=**lesson4**>Ссылки</A>

При этом слово "Ссылки" <u>в тексте никак выделяться</u> не будет. Теперь в тексте можем делать ссылку на анкер (и не один раз). Например, так:

Нажмите <A HREF="**#lesson4**">здесь</A> для возврата к началу урока

К этому анкеру можно перейти и из другого файла.

12. Графическая карта ссылок или изображение карта

Изображения-карты позволяют выделить отдельные области изображений и определить для каждой из них свое действие. Скажем, какие-то части картинки будут служить ссылками на другие страницы, после клика по другим областям рисунка будет загружен и проигран музыкальный фрагмент, или выполнен JavaScript. Существует два типа изображений-карт:

- **клиентские (client-side)** - когда пользователь кликает по рисунку, браузер сам интерпретирует координаты клика. Он выбирает ссылку, определенную для данной области, и переходит по ней (либо выполняет заданное действие);
- **серверные (server-side)** - координаты клика передаются для интерпретации на сервер и уже он делает соответствующие действия (например, возвращает браузеру URL для перехода).

Первый тип проще и доступнее, поэтому **<u>далее рассматривать будем только его</u>**.

Для создания изображения-карты используются элементы **<MAP>** и **<AREA>.**

Элемент **MAP** (так и переводится - "карта") должен обязательно иметь атрибут **NAME**. Это позволяет указать браузеру, к какому именно рисунку на странице относится данная карта.

Элемент **AREA** имеет следующие атрибуты:

1. **SHAPE** - описывает форму выделяемой области, возможные значения:
   a) **RECT** - прямоугольник;
   b) **CIRCLE** - круг;
   c) **POLY** - многоугольник;
   d) **DEFAULT** - определяет всю область.
2. **COORDS** - координаты, определяющие размеры и положение области на изображении. Количество и порядок значений зависит от значения атрибута SHAPE:
   a) **RECT:** - левый-X, верхний-Y, правый-X, нижний-Y (т.е. сначала координаты левого верхнего угла, затем правого нижнего);
   b) **CIRCLE:** - центр-X, центр-Y, радиус (т.е. горизонтальная и вертикальная координаты центра круга и радиус);
   c) **POLY:** - X1, Y1, X2, Y2, ..., Xn, Yn (просто перечисляются координаты всех вершин

© Ларина Марина Евгеньевна, 2013 год, РКЭ

многоугольника).

3. **NOHREF** - этот атрибут определяет, что данной области не соответствует никакая ссылка. Где это может пригодиться? Ну, например, если вы хотите сделать ссылку не в виде круга, а в виде кольца.
4. **ALT** - альтернативный текст для выделенной области, используется невизуальными браузерами.
5. **TITLE** - название выделенной области, выводится в виде подсказки, всплывающей при наведении курсора на область рисунка.
6. **TARGET** - значение этого атрибута ("_top", "_blank", "_self" или "_parent") определяет, в каком окне будет открыт документ (в случае если окно разбито на фреймы).

Если описываемые области пересекаются (накладываются друг на друга), то приоритет имеет область, описанная первой.

Атрибут **USEMAP** используется для того, чтобы соотнести какое-то изображение с определенной картой. Он может использоваться с элементами IMG (чаще всего!), OBJECT и INPUT.

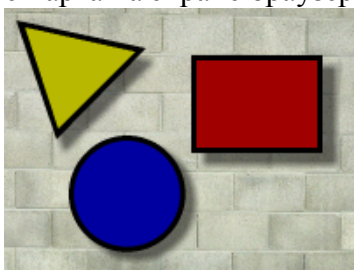
Пример использования графической карты ссылок

Приведем практический пример создания изображения-карты. Выберем изображение безымянный.gif и определим его размеры - ширина 200 пикселов и высота 150 пикселов. Определим три фигуры: прямоугольник с вершинами (110,30) и (178,79); круг с центром в точке (70,106) и радиусом 30; треугольник с вершинами (75,25), (31,69) и (12,11). Вам рекомендуется сделать поясняющий рисунок. Пусть первая фигура должна загружать файл lesson1.htm из каталога с текущим документом, вторая - вызывать переход в начало документа, а третья - загружать файл dolphin.jpg в новый экземпляр броузера (для этого в соответствующем тэге AREA нам потребуется атрибут target=_blank).

Вот как выглядит описание изображения-карты:

```
<center><img src="hotstou3.gif" width="200" height="150" usemap="#mymap"></center>
<map name="mymap">
  <area shape="rect" coords="110,30,178,79" href="lesson1.htm" title="Урок 1">
  <area shape="circle" coords="70,106,30" href="#totop" title="К началу этого урока">
  <area shape="poly" coords="75,25,31,69,12,11" href="dolphin.jpg"
     title="Рисунок" target=_blank>
</map>
```

Вот как будет выглядеть изображение-карта на экране браузера:



Выбранный рисунок с четко видимыми геометрическими фигурами служит наглядным примером изображения-карты. Как уже говорилось, на практике для изображения-карты можно использовать **любой** графический файл, изображение которого соответствует назначению изображения-карты.

Задания:

1. Сделайте на странице меню, со ссылками на методические рекомендации и поисковые системы.
2. Сделайте внизу страницы ссылку, возвращающую пользователя в начало (вверх страницы).
3. Измените стандартные цвета, принятые для ссылок в своем документе (ответ на данный вопрос найдите самостоятельно в учебниках!)
4. Создайте на своей странице графическую карту ссылок.

Оформите отчет по проделанной работе, который должен содержать:

Формат записи рассмотренных на лабораторной работе тегов, их атрибуты и назначение. Для этого начертите и заполните таблицу, в которой в одном столбце указывается тэг, в другом описывается его назначение, а далее все возможные для него атрибуты.

Контрольные вопросы:

© Ларина Марина Евгеньевна, 2013 год, РКЭ

1. Рисунок в качестве фона. Какой тег для этого используется, какие атрибуты у него могут быть? Как зафиксировать фоновый рисунок под текстом, чтобы при прокручивании он не перемещался?
2. В какой форме можно задавать цвет в HTML?
3. Как изменить стандартные цвета для активной и посещенной ссылок?
4. Создание ссылок на другую часть текущей страницы (например, вверх страницы).
5. Чем отличаются клиентские и серверные графические карты ссылок?
6. Как создать графическую ссылку-миниатюру (то есть маленькое изображение является ссылкой, при нажатии на которую открывается тоже изображение, но в полный размер)?
7. Как задать ссылку на e-mail автора страницы.

Практическая работа №6
«Таблицы на WEB странице»

Цель работы: познакомиться с тэгами для создания таблиц, строк, ячеек, с их атрибутами, а также создавать простые и сложные таблицы, создавать макет страницы с помощью таблиц

1. Создание таблиц

Для создания таблиц используются теги table, tr и td.

<TABLE ...> ... </TABLE>

Эта тэговая пара определяет начало и конец таблицы. Все элементы между тэгами <TABLE ...> ... </TABLE> являются частью таблицы. Все элементы таблицы вне этих тэгов игнорируются.

<TR ...> ... </TR>

Определяет одну строку в таблице.

<TD ...> ... </TD>

Определяет данные (table data) для ячейки в строке таблицы. Каждая пара <TD ...> ... </TD> соответствует одной ячейке в строке таблицы <TR ...> ... </TR>. Каждая строка в таблице необязательно должна иметь одно и то же число пар <TD ...> ... </TD>.

Между тэгами <TD ...> ... </TD> могут находиться любые HTML-тэги тела документа.

<TH ...> ... </TH>

Определяет ячейку таблицы как ячейку заголовка - ячейку, которая находится вверху столбца. Текст в таких ячейках по умолчанию отображается жирным шрифтом и центрируется. Атрибуты для элемента <TH ...> ... </TH> такие же как для элемента <TD...> ... </TD>.

Рассмотрим пример:

```
<table>
<tr>
   <td>1-1</td>
   <td>1-2</td>
   <td>1-3</td>
</tr>
<tr>
   <td>2-1</td>
   <td>2-2</td>
   <td>2-3</td>
</tr>
</table>
```

1-1	1-2	1-3
2-1	2-2	2-3

***<u>Обратите внимание, что по умолчанию таблица отображается без границ, если они явно не заданы</u>*!**

Таблица может иметь и другой вид:

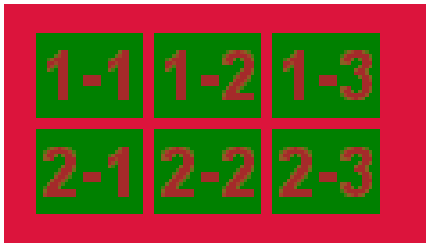
1-1	1-2	1-3
2-1	2-2	2-3

1-1	1-2	1-3
2-1	2-2	2-3

© Ларина Марина Евгеньевна, 2013 год, РКЭ

С помощью различных атрибутов можно задать толщину границ ячеек, цвет, фон ячеек и т.д.Комбинируя значения цветов фона и границы для таблицы и ячеек, можно добиваться различных оформительских эффектов:

```
<table border=2 bordercolor=red bgcolor=green>
<tr> <td>1-1</td>
     <td>1-2</td>
     <td>1-3</td>
</tr>
<tr> <td>2-1</td>
     <td>2-2</td>
     <td>2-3</td>
</tr>
</table>
```



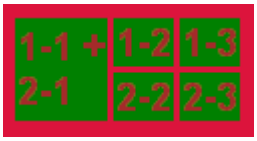
<u>Запишите в отчет теги и атрибуты для создания таблицы. Добавьте на созданную ранее свою страницу, табличку (например, с вашим расписанием занятий или любую другую).</u>

2. Объединение ячеек в таблице

Атрибуты тега ячейки td позволяют объединять ячейки по горизонтали (colspan) и вертикали (rowspan):

Обратите внимание: из элемента tr, описывающего вторую строку таблицы, удален элемент, создававший ячейку «2-1», т.к. его место теперь занято первой ячейкой из первой строки. Та же ситуация наблюдалась бы, если бы ячейка «1-2» стала занимать место двух ячеек (ячейку «1-3» удалили бы).

```
<table>
<tr>
    <td  rowspan="2">1-1 +<br> 2-1</td>
    <td>1-2</td>
    <td>1-3</td>
</tr>
<tr>
    <td>2-2</td>
    <td>2-3</td>
</tr>
</table>
```

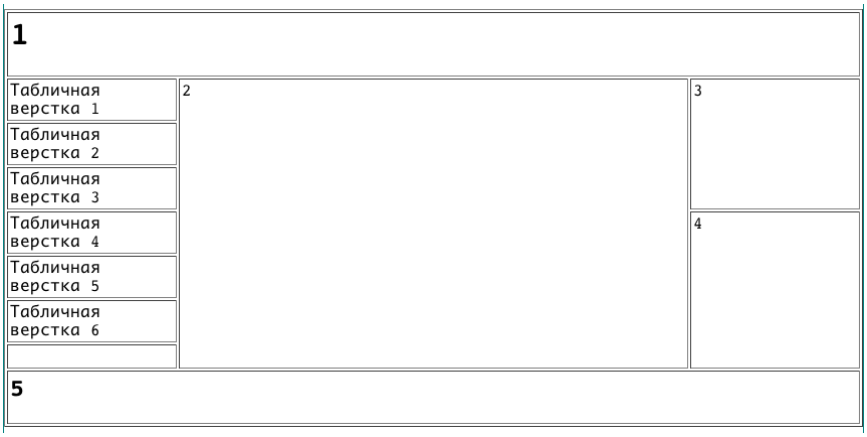


3. Табличная верстка страницы

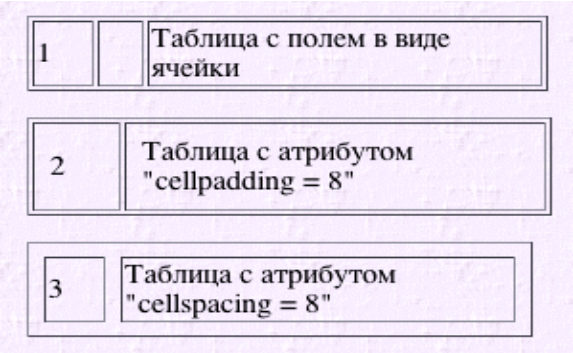
Таблицы используются не только для представления табличных данных, но и для верстки макета страницы, когда текст необходимо разместить в несколько колонок.

С использованием таблиц можно сверстать, например следующую страницу:

© Ларина Марина Евгеньевна, 2013 год, РКЭ



4.Установка внутренних полей в таблице



При использовании таблиц возникает необходимость установки промежутков между ячейками или внутренних полей.

Внутренние поля - В полиграфии пустой промежуток между колонками называется внутренним полем (gutter). Для создания внутренних полей можно использовать таблицы:

- ввести ячейку, которая служит внутренним полем (как содержание ячейки можно указать несколько неразрывных пробелов),
- воспользоваться атрибутом CELLPADDING (промежуток между содержанием и рамкой ячейки) или атрибутом CELLSPACING (промежуток между ячейками), как показано на приведенных примерах:

Задайте при верстке своей страницы внутренние поля различными способами.

<u>Оформите отчет по проделанной работе, который должен содержать:</u>

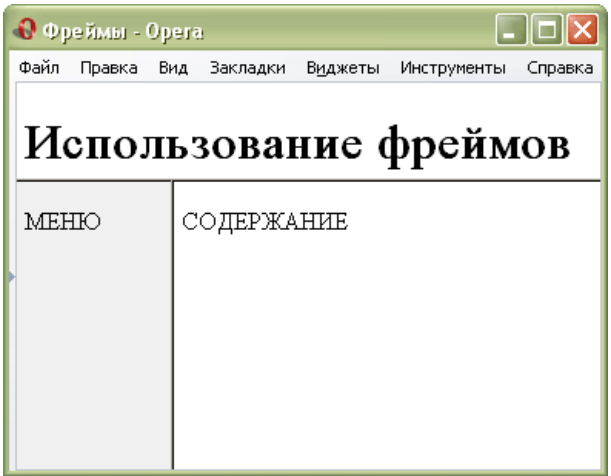
1. Запишите в отчет все теги, которые применяются при создании таблиц и их атрибуты.

<u>Контрольные вопросы:</u>

1. Что такое статическая компоновка страницы и в чем её достоинства/недостатки. Как она осуществляется?
2. Как объединить две ячейки в таблице по горизонтали?
3. Как задать толщину и цвет границ таблицы?
4. Как задать внутренние поля таблицы (внутренние отступы от текста)?
5. Как задать рисунок в качестве фона таблицы?

Практическая работа №7
«Фреймы»

Цель работы: научиться создавать фреймы различных типов, познакомиться с тегами для создания фреймов, создавать плавающий фрейм на странице.



1. Создание фреймов

Основная идея создания фреймов состоит в том, что каждый фрейм представляет собой область окна браузера, в которой отображается законченный HTML-документ. Если разбить окно браузера по вертикали ("столбцам") на два фрейма, можно

© Ларина Марина Евгеньевна, 2013 год, РКЭ

поместить один HTML-документ в левый фрейм и второй HTML-документ в правый фрейм. Однако при этом вам необходимо создать и **третий** HTML-документ. Он называется **мастер-страницей** (*master page*) и содержит специальные тэги `<frameset>`, определяющие, как разбивается окно браузера на фреймы, а также тэги `<frame>`, которые определяют, какие документы отображаются в каждом фрейме. *По существу, в этом и состоит основная функция мастер-страницы.*

До сих пор рассматривалась структура веб-сайта, которая реализуется сменяющими друг друга страницами или же верстка с помощью таблиц. Например, код для панели навигации приходилось помещать в каждую страницу сайта. *Размножение одинакового кода ведет к увеличению размера сайта и усложнению его преобразования.*

Существует способ зафиксировать одну или несколько страниц в окне (например, для размещения панели навигации), тем самым разделив окно на независимые области. Это способ называется *фреймы*.

Для создания фреймового веб-сайта необходимо создание

а) фреймов (страниц, которые будут отображаться в окне). Для этого используются отдельные HTML документы, загружаемые в отдельные фреймы.

б) контейнера (документа, содержащего объединяющий фреймы код). Фактически это и есть мастер страница.

В качестве фреймов можно использовать любые веб-страницы. В качестве контейнера используется специфический документ, который не является веб-страницей, т.е. его содержимое нигде не отображается, и элемента **body** он не содержит.

Предположим, что мы имеем две веб-страницы (1.html и 2.html) с содержимым и одну веб-страницу с панелью навигации (menu.html). Тогда контейнер (index.html) будет иметь следующий код:

```
<frameset cols="90%,*" border="0">
<frame name="main" src="1.html">
<frame name="menu" src="menu.html">
</frameset>
```

Этот код находится в документе index.html, который является мастер-страницей, то есть контейнером для фреймов

Атрибут cols=”90%,*” означает, что используется вертикальное деление на фреймы, левый будет занимать 90% окна, правый – остальное место. Соответственно, указание rows вместо cols даст нам горизонтальное деление.

Примечание: если опустить знак процента, значение берется в пикселах.

Следует обратить особое внимание на атрибут **name**. Именно его наличие позволяет управлять отображением содержимого в «сменном» фрейме. *Заданное в этом атрибуте имя указывается затем в гиперссылках панели навигации в атрибуте target, например:*

```
<a target="main" href="1.html">1</a> <br>
<a target="main" href="2.html">2</a> <br>
```

Этот код – в документе menu.html

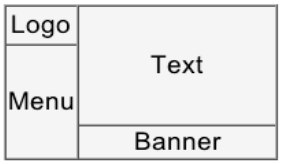
Примечание: чтобы ссылки из внутренних страниц во фреймовом сайте открывались в новом окне (например, для отображения внешних ресурсов), нужно указывать атрибут target=”_blank”.

Обратите внимание на атрибуты имеющие отношение к разделителям окон фреймов, заданию их толщины, цвета, а также наличия полос прокрутов и т.д.

2. Верстка страницы фреймами

Предположим, мы хотим сделать такую страницу: окно, состоящее из левого фрейма шириной 200 пикселов, содержащего меню, и правого с каким-то текстом. Причем в левом фрейме вверху должен быть логотип высотой в 150 пикселов, размещенный в отдельном фрейме, а в правом внизу - баннер, который при скроллинге правого окна все время будет оставаться на месте. Должно получиться следующее:

© Ларина Марина Евгеньевна, 2013 год, РКЭ




```
<frameset cols="200,*">
  <frameset rows="150,*">
    <frame src="logo.html">
    <frame src="menu.html">
  </frameset>
  <frameset rows="*,70">
    <frame src="text.html">
    <frame src="banner.html">
  </frameset>
</frameset>
```

Рассмотрим атрибуты тегов **<frameset>** и **<frame>.** Первый из них, **<frameset>** - парный тег, т.е. он обязательно должен иметь закрывающий тег. Может включать следующие атрибуты:

- **cols** и **rows** - применение и возможные значения этих аттрибутов были рассмотрены выше. Внимание! Не рекомендуется одновременное использование аттрибутов **cols** и **rows**, т.е. в некоторых браузерах конструкция **<frameset cols="50%,50%" rows="30%,70%">** может привести к непредсказуемым результатам;
- **border="x"** - толщина рамок фреймов в пикселах;
- **frameborder="no|yes" или "0|1"** - определяет отсутствие (no или 0) или наличие (yes или 1) рамок в определенном фреймсете. Например, таким образом в предыдущем примере можно было бы убрать только рамки между баннером и текстом и между логотипом и меню, а между левым и правым фреймами оставить;
- **bordercolor="#hhhhhh"** - цвет рамок. Значением может быть шестнадцатиричное число или зарезервированное слово (red, blue, teal, olive и т.д.);

Тег **<frame>** может включать следующие атрибуты:

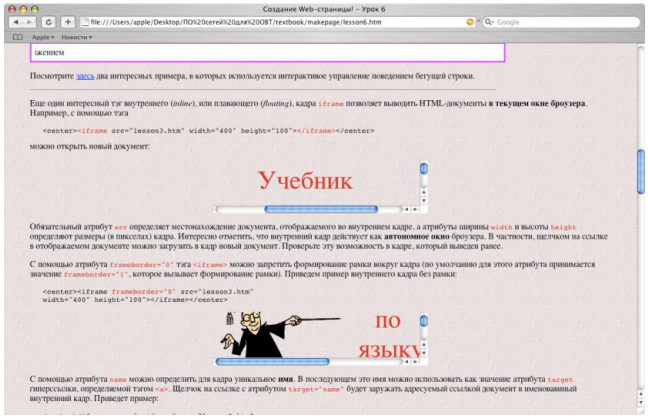
- **src="имя"** - указывает браузеру, что и откуда надо загружать в данный фрейм;
- **name="имя"** - присваивает имя фрейму. Позволяет управлять данным фреймом из других. Например, клик по ссылке в одном фрейме будет вызывать загрузку страницы в другом фрейме (подробнее в следующем абзаце);
- **scrolling="yes"** - определяет наличие полос прокрутки у данного фрейма. Возможные значения атрибута:
  - **yes** - полосы прокрутки присутствуют обязательно;
  - **no** - полосы прокрутки отсутствуют всегда - независимо от того, вмещается ли документ в фрейм целиком или нет;
  - **auto** - автоматический режим (значение по умолчанию): полосы прокрутки появляются при необходимости, т.е. если документ не вмещается в фрейм;
- **noresize** - наличие этого атрибута запрещает изменение размеров фрейма пользователем (обычно любую границу фрейма можно с легкостью переместить мышкой).

3.Плавающий фрейм

Еще один интересный тэг внутреннего (*inline*), или плавающего (*floating*), кадра iframe позволяет выводить HTML-документы **в текущем окне браузера**. Например, с помощью тэга

```
<center><iframe src="lesson3.htm" width="400" height="100"></iframe></center>
```

Ниже приведен пример, иллюстрирующий открытие плавающего фрейма внутри WEB-страницы.



© Ларина Марина Евгеньевна, 2013 год, РКЭ

Обязательный атрибут src определяет местонахождение документа, отображаемого во внутреннем кадре, а атрибуты ширины width и высоты height определяют размеры (в пикселах) кадра. Интересно отметить, что внутренний кадр действует как **автономное окно** браузера. В частности, щелчком на ссылке в отображаемом документе можно загрузить в кадр новый документ. Проверьте эту возможность в кадре, который выведен ранее.

С помощью атрибута frameborder="0" тэга <iframe> можно запретить формирование рамки вокруг кадра (по умолчанию для этого атрибута принимается значение frameborder="1", которое вызывает формирование рамки). Приведем пример внутреннего кадра без рамки:

```
<center><iframe frameborder="0" src="lesson3.htm"
width="400" height="100"></iframe></center>
```

Задание к лабораторной работе.

Создайте на вашей странице несколько фреймов, например напоминающих структуру словаря, статьи по определенному слову загружаются на той же странице в определенной части окна. Кроме того разместите на вашей странице плавающий фрейм, в котором, например будет загружаться информация об авторских правах и т.д.

Оформите отчет по проделанной работе, который должен содержать:

Запишите шаблон мастер страницы.

Запишите все атрибуты тегов для создания страницы с фреймами.

Запишите ответы на контрольные вопросы.

Контрольные вопросы:

1. Как создать плавающий или встроенный фрейм?
2. Можно ли запретить или разрешить передвигать пользователю границу разделения фреймов, меняя тем самым ширину окон? Каким образом?
3. Почему не рекомендуется использовать фреймы для верстки сайтов? Какие с ними могут быть связаны проблемы?
4. Как сделать разделители фреймов невидимыми?
5. Как задать величину отступа (внутренние границы) содержимого внутри фрейма?

Практическая работа №8
«Каскадные таблицы стилей CSS (Cascading Style Sheets)»

Цель работы: изучение и применение Каскадных Таблиц Стилей (CSS) при оформлении (разметке) Web-страниц.

Теоретические сведения:

Вы создавали предыдущие страницы, так как их создавали раньше до появления каскадных таблиц стилей или CSS (Cascading Style Sheets).

Основные проблемы, с которыми сталкивались разработчики сайтов до появления CSS:

- *Проблема позиционирования (разметка страниц). Хотя для этого стали использовать таблицы (они не были для этого предназначены изначально), они не решили всех проблем. Невозможно осуществить следующее:*

- Нельзя задать фиксированные размеры какого-нибудь блока. Например, текстовый блок будет разъезжаться при переполнении (даже если он в таблице).

- Нельзя задать фиксированные координаты положения блока на странице.

- Нельзя наложить один блок на другой. Например, попробуйте надвинуть картинку поверх таблицы или другой картинки.

- *Вторая проблема заключалась в том, что приходилось каждый раз задавать на страницах размер и цвет шрифта, свойства ячеек таблицы и т.д. Это сильно увеличивало размеры страницы, а значит, она медленнее загружается. В случае если вам захотелось изменить цвет или размер шрифта, вам пришлось бы редактировать все страницы.*
- *Третья проблема заключалась в том, что все браузеры имеют индивидуальные настройки. Например, в браузере можно увеличить шрифт, что приведет к искажению всей страницы.*

© Ларина Марина Евгеньевна, 2013 год, РКЭ

С помощью CSS эти проблемы можно решить.

Обычный HTML позволяет задавать цвет и размер текста с помощью тегов форматирования. Если понадобится изменить параметры однотипных элементов на сайте, придется просматривать все страницы, чтобы найти и поменять теги.

Каскадные таблицы стилей (Cascading Style Sheets, CSS) позволяют хранить цвет, размеры текста и другие параметры в стилях. **Стилем называется набор правил форматирования, который применяется к элементу документа, чтобы быстро изменить его внешний вид.**

Стили позволяют одним действием применить сразу всю группу атрибутов форматирования. С их помощью можно, например, изменить вид всех заголовков. Вместо форматирования заголовка в три приема, когда сначала задается его размер, затем шрифт "Arial" и, наконец, выравнивание по центру, то же самое можно сделать одновременно, применив стиль к тегу <H1>. Если требуется быстро изменить внешний вид текста, созданного с помощью одного из стилей, достаточно изменить параметры стиля во всех документах, где он используется, и вид текста поменяется автоматически.

Другое преимущество CSS состоит в том, что стили предлагают намного больше возможностей для форматирования, нежели простой HTML. Кроме того, стили могут храниться во внешнем файле, браузер кэширует такие документы, поэтому загрузка сайта будет происходить чуть быстрее.

Способы применения CSS

Существует три способа применения таблиц стилей:

- **Внутренние таблицы стилей** (Inline Style Sheets) - при помощи специального атрибута помещаются прямо в HTML теги.

Пример HTML:

 Пример

Пример CSS:

 Пример

Как можно заметить, код Inline Style Sheet получился больше чем HTML. *Поэтому ISS следует использовать, только если необходимо задать определенному элементу свой индивидуальный стиль.*

При помощи дополнительного атрибута style мы можем определить нужные нам стилевые параметры в любом теге. Это самый легкий способ, и действует он в пределах лишь **одного тега.** Но представьте, насколько вырастет размер файла, и насколько неудобно будет его исправлять, если мы будем указывать стиль у каждого тега.

- **Глобальные таблицы стилей** (Global Style Sheets) - определяют стиль элементов во всем документе.

Для этого используется тег <STYLE type="text/css">.

Он размещается в заголовке документа <head>.

Пример:

<html>
<head>
<STYLE type="text/css">
h1 {color:red; font-style:italic; font-size:32px} - переопределение стандартного тега.
.blue {color:blue} - определение нового класса
</STYLE>
</head>
<body>

Теперь эти стили можно применять в любом месте html-кода. Для этого используются следующие конструкции:

<h1> Этот заголовок написан крупным красным курсивом </h1>
Вот это слово - синие.
</body>
</html>

В данном примере все элементы H1 будут написаны крупным красным курсивом, все элементы с указанным классом BLUE будут синими.

© Ларина Марина Евгеньевна, 2013 год, РКЭ

• **Связанные таблицы стилей** (Linked Style Sheets) - могут быть использованы для нескольких документов сразу и хранятся во внешнем файле. Самый мощный и удобный способ определения стилей и правил для сайта. *Стили хранятся в отдельном файле, который может быть использован для любых веб-страниц. Для подключения таблицы связанных стилей используется тег <LINK> в заголовке страницы*

Пример внешнего файла:

Файл main.css

body {background:black; font-size:9pt; color:red; font-family:Arial Black}
.base{color:blue; font-style:italic}
h1 {color:white}
#bold {font-weight:bold}

В самих же HTML документах делается ссылка на этот файл при помощи тега <LINK>. Выглядит это так:

<LINK rel="STYLESHEET" TYPE="text/css" HREF="путь до файла">

Пример:

Файл Index.html

<html>
<head>
<LINK rel="stylesheet" type="text/css" href="main.css">
</head>
<body>
Содержание Документа
</body>
</html>

Это самый удобный способ, и для основной таблицы стилей рекомендуется пользоваться именно им.

Синтаксис CSS

Все объявления CSS (официально они называются "селекторы") записываются в фигурных скобках:

ТЕГ:псевдокласс.класс { характеристика1: значение1; характеристика2: значение2}

Описание каждого класса делается при помощи конструкции, подобной этой: .small { font-size: 9pt; }

Сначала указывается имя класса - оно может быть произвольным, но желательно все-таки давать осмысленное название. Далее, в фигурных скобках {} перечисляются все необходимые параметры для данного класса. Параметры отделяются друг от друга точкой с запятой. Вот еще один пример, в котором используется более длинное описание:

.small { font-size: 9pt; color: #eeeeee; text-align: center; }

Универсальный класс *- может быть применен к любому тегу, имя класса начинается с точки.*

Пример:

Накладываем стиль на этот текст
<td class=small>Накладываем стиль на этот текст</td>

Теговые классы *- может быть применен к конкретному тегу, имя класса начинается с указания тега, после точки записывается имя класса.*

Пример:

p.small { font-size: 9pt; }

Класс, определенный таким образом, сработает только в том теге, для которого он предназначен, а для всех остальных будет проигнорирован:

Этот текст будет выведен стилем small
<td class=small>А этот останется неизменным</td>

Мы можем определять параметры не только для одного тега, но и сразу для нескольких. Для этого в определении стиля достаточно перечислить их через запятую:
P, TD, LI { font-size: 9pt; color:green;}

© Ларина Марина Евгеньевна, 2013 год, РКЭ

В случае переопределения существующих тегов, в описании стиля можно указывать не все параметры, а лишь те из них, которые мы хотим изменить. Все остальные параметры примут значения по умолчанию, которые для разных тегов различны.

Псевдоклассы

В CSS есть такое понятие как псевдокласс. В отличие от обычного класса, действие псевдокласса распространяется не на весь текст, к которому применен данный стиль, а лишь на его часть и возможно лишь в определенном состоянии. Чтобы было понятнее, давайте разберем эффект, при котором ссылки подчеркиваются лишь при наведении на них курсора. Эффект достаточно распространенный. Вот фрагмент таблицы стилей, который позволяет достигать вышеописанного эффекта:

a { text-decoration: none; }

a:hover { text-decoration: underline; }

Верхняя строчка - это переопределение стандартного тега <a>, которое запрещает подчеркивать ссылки, а вот нижняя - это определение стиля для псевдокласса hover, который описывает стиль ссылки в момент, когда курсор находится над ней.

А вот и другой пример псевдокласса - определение буквицы вначале абзаца: *p:first-letter { font-size: 200%; font-weight: bold; }*

Заметьте, что и в том, и в другом случае действие стиля распространяется либо на определенное состояние (пользователь собирается щелкнуть по ссылке), либо на фрагмент текста (изменяется только первая буква абзаца). В этом и заключается смысл псевдоклассов.

Выполнение работы:

Задание 1

С помощью **внутренних** таблиц стилей, задайте оформление шрифта на вашей странице.

Сделайте, чтобы текст был написан полужирным шрифтом Arial, размером 12 пкс. Цвет шрифта задайте согласно выбранной ранее цветовой схеме.

Задание 2

Оформите свои страницы с помощью **внешней таблицы** стилей.

1.Создайте каталог /css/ в своей папке. Создайте в нем файл main.css

2.Подключите его ко всем страницам разработанного ранее сайта с помощью тега <LINK>.

3.Задайте по умолчанию следующие параметры для всех страниц (переопределив, тег <body>):

- цвет фона
- размер шрифта
- цвет шрифта

4.Проверьте работоспособность внесенных изменений.

Задание 3

1.Задайте по умолчанию следующие параметры для всех абзацев с **помощью глобальной таблицы стилей** (переопределив, тег <p> и псевдоклассы тега <p>):

- выравнивание абзаца
- отступ красной строки
- размер и цвет первой буквы
- оформление всех заголовков третьего уровня

2.Проверьте работоспособность изменений.

3.В комментариях (в файле main.css) поясните эти параметры.

Все проделанные изменения **запишите в отчет** (то есть опишите, каким образом выполняется то или иное задание).

Запишите что такое внутренние, глобальные и каскадные таблицы стилей.

Контрольные вопросы:

1. Что такое классы и селекторы?
2. Какие типы таблиц стилей вы знаете?
3. Синтаксис записи CSS.
4. В чем заключаются достоинства каскадных таблиц стилей?
5. Запишите пример задания стиля для ссылок (активных, посещенных …) в документе всеми способами.

© Ларина Марина Евгеньевна, 2013 год, РКЭ

Практическая работа № 9
«Разработка Java-скриптов и их внедрение с web-страницу»

Цель работы: знакомство с основными понятиями языка Java-Script, создание скриптов и их внедрение в web-страницу.

Теоретические сведения:

Чтобы «оживить» веб-страницы используют специальные скрипты.

Язык JScript является интерпретируемым объектно-ориентированным скриптовым языком (*scripting language*). Хотя язык JScript обладает меньшими возможностями, чем полномасштабные объектно-ориентированные языки типа C++ и Java, он оказывается достаточно мощным и гибким для разработки скриптов.

JScript не является "урезанной" версией какого-либо другого языка, например, он только отдаленно и косвенно связан с языком Java, и не слишком упрощен, хотя, конечно, его возможности ограничены. Например, на нем нельзя разрабатывать автономные приложения и имеются слабые средства работы с файлами. Более того, ***JScript-скрипты можно выполнить только при наличии интерпретатора в web-сервере или web-браузере.***

Относительно языка JavaSript необходимо отчетливо представлять себе следующие основные моменты:

- JavaScript представляет собой **скриптовый язык** (*scripting language*)
- Скриптовый язык является упрощенным (*lightweight*) языком программирования
- Программа (**скрипт** - *script*) на языке JavaScript состоит из строк исполняемого компьютерного кода
- JavaScript-скрипты можно вставлять в HTML-страницы

Когда JavaScript-код вставляется в HTML-документ, браузер считывает документ и **интерпретирует** JavaScript-код. Этот код может выполняться немедленно или спустя некоторое время, реагируя на последующие события.

Скрипты в секции HEAD: ***Скрипты, которые должны выполняться при их вызове или при возникновении события, размещаются в секции HEAD. Это гарантирует загрузку скрипта до его использования.***

```
<html>
<head>
  <script language="JavaScript">
    некоторые операторы
  </script>
</head>
```

Скрипты в секции BODY: *Скрипты, **подлежащие выполнению при загрузке** страницы, помещаются в секцию **BODY**. Когда скрипт находится в секции **BODY**, он генерирует содержание на странице.*

```
<html>
<head>
</head>
<body>
  <script language="JavaScript">
    некоторые операторы
  </script>
</body>
```

пример (вставьте данный скрипт в свою страницу, объясните в отчете назначение оператора *document.write*):

```
<html>
<head>
</head>
<body>
<script language="JavaScript">
```

© Ларина Марина Евгеньевна, 2013 год, РКЭ

```
  document.write("Hello World!")
</script>
</body>
</html>
```

Скрипты в секциях HEAD и BODY: В документ можно поместить любое число скриптов, размещая их в секциях **HEAD** и **BODY**.

```
<html>
<head>
  <script language="JavaScript">
   некоторые операторы
  </script>
</head>
<body>
  <script language="JavaScript">
   некоторые операторы
  </script>
</body>
```

Как использовать внешние скрипты

Иногда может потребоваться выполнить один и тот же скрипт на нескольких страницах, не записывая скрипт на каждой странице. Для этого можно написать скрипт во внешнем файле и сохранить файл с расширением **.js**, например:

document.write("Этот скрипт является внешним")

Сохраните внешний файл как **xxx.js**

- Имя файла не может иметь более восьми символов
- Внешний скрипт не должен содержать тэг **<script>.**

А таким образом в документе будет создано обращение к вашему внешнему скрипту:

```
<html>
<head>
</head>
<body>
  <script src="xxx.js">
  </script>
</body>
</html>
```

Добавьте на свою страницу.

Достаточно часто используют готовые скрипты из библиотеки скриптов, вставить их с помощью редактора или скачать с некоторых сайтов (http://javascript.internet.com/ **и другие).**

Переменные языка

Переменные применяются для хранения данных. Переменная служит как бы "контейнером" для данных, которые необходимо сохранить. Значение переменной в ходе выполнения скрипта можно изменять. Можно обращаться к переменной по ее имени для получения ее значения или изменения его.

Переменная создается с помощью оператора var:

var strname = некоторое значение

Можно также создать переменную и без оператора var:

strname = некоторое значение

Присваивание переменной значения производится следующим образом:

var strname = "Елена"

Можно также присвоить переменной значение и таким способом:

strname = "Елена"

Приведем простой пример создания переменной с присваиванием ей значения, а затем отобразим значение переменной на странице обычным образом и в виде заголовка третьего уровня.

```
<script language="JavaScript">
var name = "ЕЛЕНА"
```

© Ларина Марина Евгеньевна, 2013 год, РКЭ

document.write(name)
document.write("<h3>" + name + "</h3>")
</script>

Функции языка JavaScript

Функция содержит некоторый код, который выполняется при возникновении события или вызова данной функции. Функция состоит из набора операторов. Можно повторно использовать функции в одном и том же скрипте или в других документах. Функции определяются в начале HTML-файла (в секции HEAD) и вызываются впоследствии из документа. Приведем теперь простейший пример с информационным боксом (alert box), который является методом языка JavaScript для извещения пользователя.

alert("здесь находится сообщение")

Также как и в других языках программирования в JavaScript при написании кода используются **условные операторы и циклы.** Подробнее с правилами их записи познакомьтесь с помощью электронного учебника.

Написание скриптов неразрывно связано с понятием события и обработчика события.

События (event) и обработчики событий (event handler) относятся к JavaScript, но они скорее «встроены» в HTML-код, а не существуют самостоятельно, как те скрипты, которые мы с вами разбирали. Они входят в структуру документа HTML, не требуя команд <SCRIPT> и </SCRIPT>. **Сами они не скрипты, а скорее область взаимодействия между вашей страницей и читателем. События — это то, что происходит.** Они добавят динамики вашему сайту. Среди разнообразных обработчиков событий для начала мы выберем один, самый популярный, — **onMouseOver (навести мышь).**

Скрипт

<A HREF="http://www.newmail.ru" onMouseOver="window.status='Бесплатный хостинг'; return true">Ссылка</A>

Очень простым примером служит событие **Click**, т.е. щелчок левой кнопкой мыши. Например, можно написать JavaScript-код, который вычисляет сумму платежа с учетом выбранных пользователем товаров. Важным фактором становится определение того, когда этот код выполнять? Логичным ответом на этот вопрос является выполнение кода тогда, когда это потребуется самому пользователю. Совсем нетрудно представить себе web-страницу с кнопкой "Найти общую сумму". Тогда код для вычисления платежа нужно выполнять тогда, когда пользователь щелкнет на этой кнопке. Таким образом, щелчок на кнопке является событием. Фактически это событие и называется "Click"!

Под событием понимается определенный сигнал, возникающий в результате действия пользователя, например щелчка левой кнопкой мыши (событие Click), двойного щелчка левой кнопкой мыши (событие DblClick), наведения указателя мыши на конкретный элемент документа (событие MouseOver), нажатия клавиши (событие KeyPress) и др., или автоматически формируемый системой при изменении ее состояния, например при окончании загрузки документа (событие Load), обнаружении ошибки (событие Error) и др. В ответ на все эти события система реагирует определенным образом, обычно вызовом специальной функции. Пример события Load и его обработки вы наблюдали при загрузке этого документа. Нетрудно заметить, что события называются английскими словами (регистр букв не играет роли), передающими содержательный смысл события.

Под обработчиком события (event handler) понимается код (скрипт - script), написанный на скриптовом языке (scripting language); этот скрипт перехватывает событие и выполняет некоторые ответные действия. Например, при щелчке на кнопке активизируется ассоциированный с этим событием обработчик события и реагирует на событие Click. Названия (имена) обработчиков событий образуются добавлением префикса on к названию события, например обработчик события Click называется onClick (часто названия обработчиков событий даются на нижнем регистре, т.е onclick).

Задание!

1. *Выпишите и поясните некоторые события* из рассмотренного выше материала.

© Ларина Марина Евгеньевна, 2013 год, РКЭ

2. Вставьте на страницу и запишите в отчет скрипт из примеров ниже

Пример № 1 - определение типа браузера

Свойство **appName** объекта **navigator** позволяет определить тип браузера в компьютере пользователя.

Исходный код:

```
<script language="JavaScript">
document.write("Вы пользуетесь браузером: "+ navigator.appName)
</script>
```

Пример № 2 - получение характеристик монитора

Объект **screen** имеет несколько свойств, значения которых показывают характеристики монитора в компьютере пользователя.

Исходный код:

```
<script language="JavaScript">
document.write("Разрешающая способность экрана: ")
document.write(screen.width + "*")
document.write(screen.height + "<br>")
document.write("Доступная область просмотра: ")
document.write(window.screen.availWidth + "*")
document.write(window.screen.availHeight + "<br>")
document.write("Глубина цвета: ")
document.write(window.screen.colorDepth + "<br>")
</script>
```

Пример № 3 - вывод информационного бокса

Метод **alert()** позволяет вывести на экран информационный бокс (*alert box*), содержащий текстовую цепочку, которая передается ему как параметр. Действие метода **alert()** вы наблюдали при открытии этого документа.

Исходный код:

```
<script language="JavaScript">
alert("Это пример информационного бокса")
</script>
```

Пример № 4 - вывод стимулирующего бокса

Метод **prompt()** позволяет вывести на экран стимулирующий бокс (*prompt box*), который требует от пользователя ввода некоторых данных. Метод имеет два параметра в виде текстовых цепочек - первая цепочка определят вводимые данные, а вторая задает значение по умолчанию. Метод возвращает в скрипт ввод пользователя или значение по умолчанию (после щелчка на кнопке **OK**) или значение **null** (после щелчка на кнопке **Cancel**).

Исходный код:

```
<script language="JavaScript">
var name = prompt("Пожалуйста, введите ваше имя","Елена")
if (name != null && name != "") {
document.write("Привет, " + name)
}
</script>
```

Контрольные вопросы:

1. Какие типы ошибок встречаются при использовании скриптов?
2. Что такое события и обработчики событий.
3. В каких частях HTML документа можно размещать скрипты и почему?
4. Использование метода alert().
5. Использование метода prompt().

© Ларина Марина Евгеньевна, 2013 год, РКЭ

Рекомендуемая литература

Основная:

1. Баканов В.М. Программное обеспечение компьютерных сетей и информационных систем. - М.: МГАПИ, 2003, -104 С.
2. Ватаманюк А. Создание, обслуживание и администрирование сетей на 100%. Издательство: Питер. Год издания: 2010, С: 288.
3. Дронов В.А. Название: HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. Издательство: БХВ-Петербург, Год издания: 2011, С: 416.
4. Исаченко О. В. Программное обеспечение компьютерных сетей. Издательство: Инфра-М. Год издания: 2012.
5. Сетевые операционные системы: учебник / В.Г. Олифер, Н.А. Олифер -СПб.: Питер, 2001.

© Ларина Марина Евгеньевна, 2013 год, РКЭ

Дополнительная:

1. Байрон Анжела, Эддисон Берри, Натан Хог, Джефф Итон, Джеймс Уокер, Джефф Роббинс. Drupal: создание и управление сайтом. Издательство: Символ-Плюс. Год издания: 2010, С: 575.
2. Билл Скотт, Тереза Нейл. Проектирование веб-интерфейсов. Издательство: Символ-Плюс, Год издания: 2010, С: 352.
3. Данилов И. Все об электронной почте. – К.: ОАО «Дом печати – ВЯТКА», 2005.
4. Казарин О.В. Безопасность программного обеспечения компьютерных систем. Москва, МГУЛ, 2003, 212 с.
5. Кожемякин А.А. HTML и CSS в примерах. Создание Web-страниц. 2004.
6. Коровченко Э. Энциклопедия Internet. CD, ООО «Бизнессофт», 2005.
7. Кучерявый Е.А. Управление трафиком и качество обслуживания в сети Интернет. Издательство: Наука и Техника. Год издания: 2004, С: 336.
8. Леонтьев В. Самоучитель: осваиваем Интернет. – М.: Олма-пресс, 2005.
9. Мак-Федрис Пол. Развертывание безопасных сетей в Windows Vista. Издательство: Вильямс. Год издания: 2009, С: 528.
10. Михасев В. Г., Прончев Г. Б. Компьютерные сети, Интернет и мультимедиа технологии. Издательство: МИПК им. И. Федорова. Год издания: 2010.
11. Олифер В.Г., Н.А. Олифер Н.А. Компьютерные сети: принципы, технологии, протоколы.- С.-Петербург: ПИТЕР, 2000.
12. Рева О. HTML. Просто как дважды два. – М.: Изд-во Эксмо, 2008.
13. Смит К., Уотерс К. WEB-дизайн: Photoshop&Dreamweawer. Пер. С англ. – М.: КУДИЦ-ОБРАЗ, 2004
14. Таненбаум Э., Уэзеролл Д. Компьютерные сети. Издательство: Питер, Год издания: 2011.
15. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. Издательство: Питер, Год издания: 2012, С: 272.
16. Холмогоров В. Основы WEB-мастерства. Учебный курс.- СПб: Питер, 2001

© Ларина Марина Евгеньевна, 2013 год, РКЭ

© Рязанский колледж электроники
© Цикловая комиссия вычислительной техники и компьютерных сетей
© Ларина Марина Евгеньевна