Е.Е. Поморцева

СЕТЕВЫЕ ТЕХНОЛОГИИ. ОСНОВЫ ВЕБ-ДИЗАЙНА

Учебное пособие

Е.Е.Поморцева

СЕТЕВЫЕ ТЕХНОЛОГИИ. ОСНОВЫ ВЕБ-ДИЗАЙНА

Учебное пособие

Харьков
ХНУГХ им. А. Н. Бекетова
2021

УДК [004.7:004.92](075.8)
П55

Автор

Поморцева *Елена Евгеньевна*, кандидат технических наук, доцент

Рецензенты:

А. Б. Костенко, кандидат физико-математических наук, доцент кафедры компьютерных наук и информационных технологий ХНУГХ им. А. Н. Бекетова;

К. А. Метешкин, доктор технических наук, профессор кафедры земельного администрирования и ГИС ХНУГХ им. А. Н. Бекетова

Рекомендовано к печати на заседании
Ученого совета ХНУГХ им. А. Н. Бекетова,
протокол № 8 от 28 февраля 2020 г.

Поморцева Е. Е.

П55 Сетевые технологии. Основы веб-дизайна : учеб. пособие / Е. Е. Поморцева ; Харьков. нац. ун-т гор. хоз-ва им. А. Н. Бекетова. – Харьков : ХНУГХ им. А. Н. Бекетова, 2021. – 132 с.

ISBN 978-966-695-524-4

Материал пособия изложен на примере единого проекта по созданию сайта-визитки, что позволяет в достаточной мере овладеть средствами языка гипертекстовой разметки документов HTML и каскадными таблицами стилей CSS для разработки собственного сайта. Изложенный материал направлен на овладение инструментальными средствами, которые позволяют создавать веб-страницы, объединять их в сайты, оформлять в едином стиле, использовать языки программирования для придания динамических эффектов статическим сайтам-визиткам и размещать созданные сайты в сети Интернет. Полученный уровень компетентности позволит эффективно использовать возможности глобальной сети Интернет при решении задач в профессиональной деятельности, создаст основу для самостоятельного освоения новых языков программирования и новых версий HTML и CSS.

УДК [004.7:004.92](075.8)

ISBN 978-966-695-524-4

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ

ASCII (American Standart Code for Information Interchage) – американский стандартный код для обмена информацией.

CSS – это формальный язык, служащий для описания оформления внешнего вида документа, созданного с использованием языка разметки (HTML, XHTML, XML).

DNS – система доменных имен.

E-Mail – электронная почта.

FTP (File Transfer Protocol) – Протокол Передачи Файлов.

HTML (HyperText Markup Language) – стандартизированный язык разметки документов во Всемирной паутине.

IP – Интернет протокол.

TCP (Transmission Control Protocol) – протокол управления передачей.

URI (Universal Resource Identification) – унифицированный идентификатор ресурса. Символьная строка, позволяющая идентифицировать какой-либо ресурс.

W3C (World Wide Web Consortium) – организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины.

WWW (World Wide Web) – Всемирная паутина или распределенная система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключенных к Интернету.

ПК – персональный компьютер.

ВВЕДЕНИЕ

В данном учебном пособии в первом и втором разделах рассмотрены основные теоретические сведения о истории развития глобальной мировой сети Интернет, о структуре этой сети, основные понятия Интернет, а также история развития языка гипертекстовой разметки текста HTML и каскадных таблиц стилей CSS. Изложены также основные принципы верстки веб-страниц и оформления их с помощью каскадных таблиц стилей.

Третий раздел учебного пособия посвящен практической части и состоит из следующих работ:

– Основы веб-дизайна. Использование языка гипертекстовой разметки текста HTML для создания веб-документов.

– Создание контента веб-страницы. Списки, таблицы.

– Использование CSS для оформления веб-документов.

– Использование внешней таблицы стилей для оформления веб-документов.

– Технология создания динамических веб-страниц средствами языка JavaScript.

– Создание на сайте информеров. Размещение сайта в сети Интернет.

Каждая практическая работа предполагает предварительную подготовку к ней. Перед выполнением задания необходимо сначала изучить теоретический материал и только после этого приступать к выполнению работы. Для более продуктивного усвоения изложенного материала в конце каждой лабораторной работы приведены задания для самостоятельного выполнения.

Профессиональные компетентности, которые формируются в ходе освоения материала учебного пособия

В процессе обучения студенты получают необходимые знания, закрепляют и углубляют их, приобретая при этом практические навыки и умения при выполнении практических работ. Особое значение имеет индивидуальная работа студентов, при выполнении которой они самостоятельно шаг за шагом создают сначала отдельные веб-страницы своего, затем объединяют их с помощью гиперссылок в сайт-визитку,

оформляют страницы своего сайта в едином стиле и добавляют динамические элементы в оформление с помощью программирования на языке JavaScript. Это позволит студентам в дальнейшем решать прикладные задачи в профессиональной деятельности.

Принципы, заложенные в основу построения учебного пособия

В данном пособии используются принципы системности и практической направленности с использованием современного программного обеспечения.

Принцип системности предусматривает рассмотрение в первом и втором разделах общих сведений по истории развития глобальной мировой сети Интернет, ее структуре, а также истории развития языка гипертекстовой разметки веб-страниц и каскадных таблиц стилей. Теоретический материал и основные понятия излагаются на примере разработки проекта по созданию с «нуля» своего собственного статического сайта-визитки.

Принцип практической направленности предусматривает фундаментальную теоретическую подготовку и активное практическое обучение студентов. Учебное пособие обеспечивает формирование навыков и умений на основе использования практической формы обучения.

Практическое обучение в пособии реализовано на основе метода активной рефлексии. Применение этого метода предполагает изучение и осмысленное повторение студентами операций, которые необходимы для успешного усвоения материала пособия.

Во время изложения материала дается краткое описание теоретических основ, анализируются способы применения полученных знаний и подробно рассматривается процесс создания тех или иных элементов веб-страницы сайта. Вследствие этого студенты учатся самостоятельно создавать свои собственные веб-страницы и в дальнейшем объединять их в полноценные сайты с последующей публикацией в глобальной сети Интернет.

Формирование навыков студентов может осуществляться как под руководством преподавателя в аудитории, так и дома путем самостоятельного создания индивидуальных статических сайтов с использованием рассматриваемой методики.

При выборе материала учитывались ограничения, которые обусловлены учебным процессом, количество отведенных кредитов и часов на изучение дисциплины. В пособие включен необходимый набор тем, без которых невозможна осмысленная и эффективная работа в процессе разработки и создания сайтов.

Для определения уровня освоения материала каждой практической работы предлагаются индивидуальные задания.

Освоенный в полном объеме материал данного учебного пособия поможет развить навыки дальнейшего обучения, самостоятельного развития и овладения постоянно обновляющимися Интернет-технологиями и в частности, позволяющими создавать сайты.

Материал учебного пособия апробирован во время проведения практических работ со студентами Харьковского национального университета городского хозяйства имени А. Н. Бекетова.

Автор высказывает глубокую благодарность профессору, доктору технических наук К. А. Метешкину за неоценимую помощь в рецензировании данного, и не только, учебного пособия, за проявленное терпение и доверие.

РАЗДЕЛ 1
ГЛОБАЛЬНАЯ МИРОВАЯ СЕТЬ ИНТЕРНЕТ

В данном разделе рассматриваются история развития глобальной мировой сети Интернет, структура этой сети, основные понятия, а также история развития языка гипертекстовой разметки текста и каскадных таблиц стилей.

1.1 История развития Интернет

В современном мире скорость обмена информацией играет первостепенную роль. Интенсификация развития науки и технологии, неуклонное формирование единой мировой экономической системы, развитие систем электронных банковских платежей, биржевых операций, сделок, торгов и закупок, задачи глобального управления, мониторинга и связи требуют качественно новых средств коммуникации. Всё большее количество людей испытывает потребность в быстрых и недорогих формах контакта для бизнеса, работы, образования и самообразования, отдыха и развлечений. Всему этому служит уникальная система, называемая сегодня информационной Супермагистралью – Интернет. Возникнув в начале 1970-х годов в результате слияния новых коммуникационных и компьютерных технологий, сеть Интернет сегодня сама даёт огромный импульс к развитию этих технологий. Более того, Интернет сегодня порождает не только новые виды деятельности человека, новые информационные каналы, новые технологии в бизнесе, но и даёт новое видение современного мира, ставит современного человека на новую ступень развития, даёт возможность всем людям Земли впервые реально почувствовать своё единство.

Если два компьютера соединить друг с другом, то они могут обмениваться данными. Если объединить большее количество компьютеров, то получится компьютерная сеть. Компьютерная сеть – это объединение автономных персональных компьютеров для совместного использования вычислительных ресурсов (процессора, памяти и периферии). Компьютерную сеть в пределах сравнительно небольшой территории обычно называют локальной (расстояние между взаимосвязанными компьютерами не превышает 300 м).

Локальная сеть обычно организуется и работает в пределах одной фирмы (организации) и объединяет компьютеры на рабочих местах для

более быстрого и качественного обмена информацией. Каждая организация, эксплуатирующая более десятка персональных компьютеров (ПК), старается объединить их в локальную сеть с целью уменьшения бумажного документооборота и повышения эффективности деятельности своих подразделений. Различают также и другие компьютерные сети (рис. 1.1).

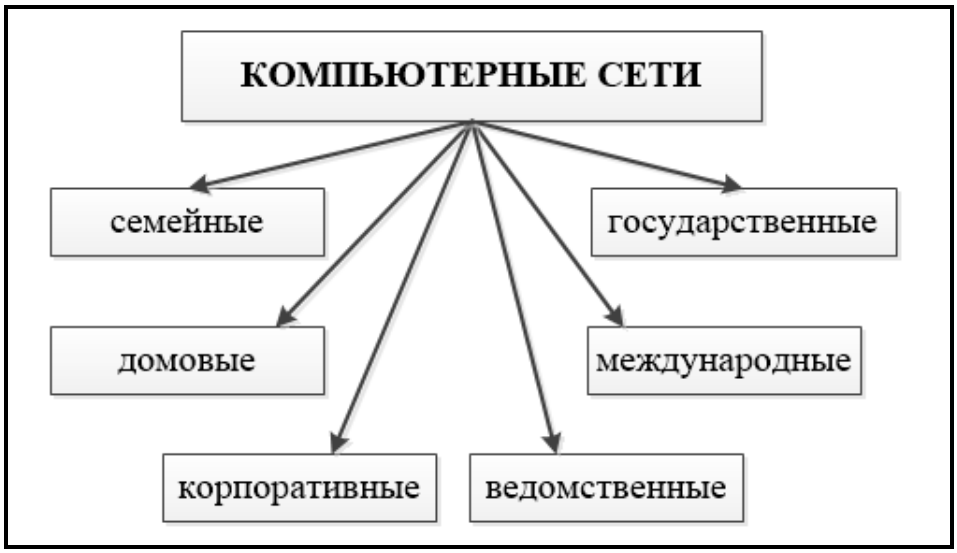

Рисунок 1.1 – Классификация компьютерных сетей

Сети, охватывающие большие пространства, а некоторые и весь земной шар, называются глобальными. Глобальная сеть – это чаще всего самостоятельная (в техническом и юридическом отношении) структура и другие фирмы подключаются к ней для работы за определённую плату.

Интернет – это глобальная компьютерная сеть, основным назначением которой является совместное использование информации. Интернет представляет собой совокупность множества региональных и ведомственных сетей, функционирующих на основе строго определенного протокола обмена данными IP (Internet Protocol). Более корректно под словом Интернет следует понимать не только множество соединенных между собой компьютеров, но и протокол обмена информацией, то есть принятые правила обмена данными и формат этих данных, а также совокупность информационных услуг, предоставляемых в сети.

В качестве линий связи могут выступать простые телефонные каналы, коаксиальные и оптоволоконные линии, всевозможные каналы радиосвязи в любом частотном диапазоне, мощные радиорелейные и спутниковые линии. Основное назначение Интернета: совместный доступ к ресурсам и совместное использование ресурсов.

История создания Интернета

Вокруг истории создания Интернета сложилось столько легенд и мифов, что восстановить истину не просто. Единственное, что ни у кого не вызывает сомнения – это то, что родина Интернет – Соединенные Штаты Америки.

Одна из легенд гласит, что американское Министерство обороны решило на свои деньги объединить крупнейшие научные и университетские центры страны для совместной работы ученых над наиболее важными проектами. И этим занялось Управление перспективных разработок Пентагона – DARPA (Defence Advanced Research Projects Agency), и в сентябре 1969 года родилась первая очередь Интернет – ARPANET. Интернет – дитя «холодной войны», его истоки восходят к военным программам Пентагона. Корни Интернета уходят еще к середине 1940-х – началу 1950-х годов. Все началось в 1949 году, когда в СССР была испытана атомная бомба. Это был ответ на атомную бомбардировку Хиросимы и Нагасаки 6 и 9 августа 1945 г. Соединенными Штатами Америки. В 1952 году Советский Союз успешно провел испытание водородной бомбы, а в середине 1950-х годов под руководством С. П. Королева был развернут огромный комплекс работ по созданию средств их доставки. В 1957 году Советский Союз запустил первый искусственный спутник Земли, то есть в СССР появилось средство, способное доставить ядерное оружие в любую точку мира.

В 1956 году Пентагон обратился в правительство за деньгами на создание системы защиты от ракетного оружия, но тогда ему отказали. В 1958 году Пентагон вновь обратился за финансовой поддержкой и получил ее. Было принято решение о создании системы раннего оповещения. Поскольку траектории ракет, запущенных в СССР в направлении США, проходят через Северный полюс, систему оповещения пришлось строить на севере Канады. Эта система получила название NORAD. Станции NORAD протянулись от Аляски до Гренландии.

Разумеется, эта система не могла предотвратить достижение ракетами целей, но предупредить об их приближении и дать 15 минут на то, чтобы подготовиться, она могла. Единственное, что для этого требовалось – исключить человеческий фактор. Людям свойственно очень долго принимать решения, а здесь счет шел на секунды, поэтому все посты наблюдения и станции раннего оповещения надо было подключить к

единому центру управления, оснащенному компьютерами. Для решения этой задачи в 1964 году был создан подземный центр управления NORAD, расположился который в скальном массиве близ городка Колорадо-Спрингс. Когда подземный центр был запущен, его компьютеры начали обрабатывать информацию, поступающую с севера континента по каналам гигантской сети. В течение 1965–1966 годов к этой сети подключались многочисленные авиационные и метеорологические службы, в том числе и гражданские. Таким образом, уже к середине 1960-х годов в США действовала огромная компьютерная сеть национального масштаба, обслуживающая как гражданские сферы, так и множество служб Министерства обороны.

К середине 1960-х годов в Советском Союзе появились термоядерные заряды, мощность которых достигала 40–50 миллионов тонн в тротиловом эквиваленте. Расчеты показали, что одного такого заряда достаточно для уничтожения промежуточных центров передачи информации. Таким образом, выходило, что если эта гигантская глобальная система выйдет из строя после попадания одного единственного заряда, то ее эффективность мала. Тогда-то и было принято решение искать методы создания такой компьютерной сети, которая не выйдет из строя при поражении любого ее участка или нескольких любых участков.

Эксперименты в действующей сети, лежащей в основе всей системы национальной безопасности, практически невозможны. Пентагону потребовалась экспериментальная сеть, причем действующая в неустойчивой и ненадежной среде. Более ненадежной и неустойчивой среды, чем университетская найти сложно. На эту среду и обратили внимание. Под контролем Пентагона началось соединение университетских компьютеров и научных центров в единую сеть. Ученые обеспечили надежную работу своей сети, и решение было найдено. В конце 1969 г. четыре компьютера были объединены в первую сеть. Сеть получила то же название, что и весь проект – ARPANET. В 1973 году состоялось первое международное подключение к ARPANET – London University College (Великобритания) и Royal Radar Establishment (Норвегия). В 1983 году, когда в сети ARPANET работало более 550 компьютеров, были окончательно приняты и задействованы современные стандарты устройства сети – протокол TCP/IP и DNS – система доменных имен.

К 1990 году к Интернет было подключено около 200 000 узлов (хостов). В этом же году старая сеть ARPANET прекратила своё существование. В 1991 году были разработаны новые виды сервиса Интернет, такие как WAIS, GOPHER и World Wide Web. В 1992 году количество узловых компьютеров Интернет превысило 1 000 000. В 1996 году начинается взрывной рост Интернет – за год происходит пятикратный рост количества узлов. Появление World Wide Web революционно изменило отношение массового пользователя к сети Интернет. К настоящему времени в сети Интернет находится около 300 000 локальных сетей, к ней подключены более 180 стран мира (рис. 1.2).

Рисунок 1.2 – Изображение узлов, подключенных к сети Интернет

Идёт лавинообразное развитие новых технологий. Такими темпами, какими расширяется Интернет, не развивается ни одна другая отрасль человеческой деятельности.

Управление Интернет

В 1979 году под эгидой ARPA был учрежден ICCB (Internetwork Configuration and Control Board) – Совет по конфигурированию и управлению Сети, в 1983 году его сменил IAB (Internet Activities Board) – Совет по деятельности Интернет. IAB представляет собой группу приглашенных добровольцев, которая регулярно собирается, чтобы разработать стандарты и распределить ресурсы, такие, например, как адреса. Интернет работает, поскольку имеются стандартные способы коммуникации между компьютерами и прикладными программами. Это

позволяет компьютерам разного типа связываться без особых проблем. IAB ответственен за стандарты, он решает, когда стандарт необходим и каким ему следует быть. Когда требуется стандарт, совет рассматривает проблему, принимает стандарт и по сети оповещает о нем пользователей. IAB также следит за различными номерами, которые должны оставаться уникальными. Например, каждый компьютер в Интернет имеет свой уникальный 32-разрядный двоичный адрес, никакой другой компьютер не имеет такого же адреса. IAB разрабатывает стандарты для разрешения такого рода проблем. Он не присваивает адресов, но разрабатывает правила, как эти адреса присваивать.

Пользователи Интернет высказывают свои жалобы и предложения на встречах IETF (Internet Enginering Task Force) – Инженерный оперативный комитет Интернет. IETF – это другая добровольная организация, которая также собирается регулярно, чтобы обсудить текущие эксплуатационные и назревающие технические проблемы. При обсуждении достаточно важной проблемы IETF создает рабочую группу для ее дальнейшего исследования. Рабочие группы имеют различные функции: это может быть выпуск документации, выработка стратегии действий при возникновении проблем, стратегические исследования, разработка новых стандартов и протоколов, доработка уже существующих. Рабочая группа обычно разрабатывает рекомендации. В зависимости от вида рекомендации, это может быть просто документация, или же это может быть послано в IAB, и быть объявлено стандартом. Многочисленные группы и организации разрабатывают новые проекты в помощь развитию Интернет, среди них наиболее известной является W3C (World Wide Web Consortium) – Консорциум по Всемирной паутине.

За Интернет никто централизовано не платит. Нет такой организации как Internet Inc., которая собирает плату со всех сетей Интернет или пользователей. Вместо этого все платят за свою часть. NSF платит за содержание NSFNET. NASA платит за Научную Сеть NASA (NASA Science Internet). Представители сетей собираются вместе и решают, как им соединяться друг с другом и содержать эти взаимосвязи. Колледж или корпорация платит за ее подключение к некоторой региональной сети, которая в свою очередь платит за свой доступ сетевому владельцу государственного масштаба.

То, что Интернет не сеть, а собрание сетей, мало как сказывается на конкретном пользователе. Для того чтобы сделать что-либо (запустить программу или добраться до каких-либо единственных в своем роде данных), пользователю не надо заботиться о том, как эти составляющие сети содержатся, как они взаимодействуют и поддерживают межсетевые связи. Каждая сеть имеет свой собственный сетевой эксплуатационный центр (NOC). Каждый такой рабочий центр связан с другими и знает, как разрешить различные возможные проблемы. Ваш регион имеет соглашение с одной из составляющих сетей Интернет и ее забота состоит в том, чтобы люди вашего региона были довольны работой сети. Так что, если что-то испортится, NOC и есть та самая организация, с кого за это спросят.

Архитектура сетевых протоколов TCP/IP, на базе которых построен Интернет, предназначена специально для объединенной сети. Сеть может состоять из совершенно разнородных подсетей, соединенных друг с другом шлюзами. В качестве подсетей могут выступать самые разные локальные сети, различные национальные, региональные и специализированные сети, а также другие глобальные сети (рис. 1.3).

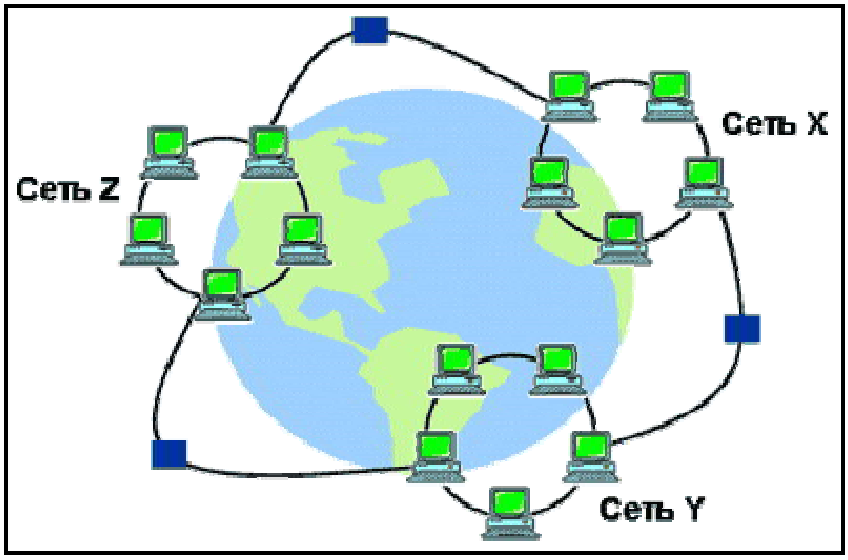

Рисунок 1.3 – Структура сети Интернет

Каждая из подсетей работает в соответствии со своими специфическими требованиями и имеет свою природу связи, сама разрешает свои внутренние проблемы. Однако предполагается, что каждая подсеть может принять пакет информации и доставить его по указанному адресу в этой конкретной подсети. Все же не требуется, чтобы подсеть

гарантировала доставку пакетов и имела надежный сквозной протокол (протокол работы сети в качестве посредника при передаче сообщений между двумя внешними сетями). Таким образом, два ПК, подключенные к одной подсети, могут напрямую обмениваться пакетами, а если возникает необходимость передать сообщение машине в другой подсети, то вступают в силу межсетевые соглашения (IP-протокол) и компьютеры передают сообщение по определенной цепочке шлюзов и подсетей, пока оно не достигнет нужной подсети, где оно и будет доставлено непосредственно получателю.

Значение Интернет

В сфере информационных технологий инновационный процесс происходит небывало высокими темпами. Специализированные издания назвали Интернет «Сетью сетей», а популярный журнал делового мира «Business week» определил ближайшее будущее как «эпоху Интернета».

Интернет открывает новый способ человеческого общения, который можно назвать горизонтальным. До его появления общение и распространение информации было в основном вертикальным: автор пишет книгу – читатели ее читают. Радио и телевидение что-то передают – зрители и слушатели это слушают и смотрят. Газета печатает новости – подписчики их читают. Обратная связь почти отсутствовала, хотя потребность в ней была исключительно высока. Обмен же информацией между самими читателями конкретной книги, слушателями конкретной передачи был практически неосуществим. Интернет обеспечивает распространение информации для практически неограниченного круга потребителей, причем они без всякого труда могут включиться в обсуждение.

Интернет предоставляет уникальные возможности и для вертикального информационного общения: между властью и гражданами, для обратной связи последних с первыми. За широким внедрением в нашу жизнь Интернета не стоит никакая организация. Всемирная Сеть как явление развивается самостоятельно, двигателем Интернет является все человечество. Главная идея Интернет – свободное распространение информации и установление связей между людьми. Это наиболее эффективный путь преодоления расовых, религиозных и идеологических барьеров между людьми, странами, народами.

Интернет – одно из самых значительных демократических достижений технологического процесса. С его появлением информация становится потенциальным достоянием большинства жителей планеты. Все глобальные коммуникации, связанные с телеграфом, телефоном, радио, телевидением и компьютерной техникой, ныне интегрируются в единое целое – Интернет. Речь идет о механизме распространения информации, объединения людей и их взаимодействия вне зависимости от расстояния, временных, государственных и многих других границ.

Полезность Интернет повышалась вместе с развитием вычислительной техники с запаздыванием примерно в пять – десять лет. В конце 80-х годов появление персональных компьютеров перенесло информатику от узких специалистов в этой области к широкой публике. Интернет в ходе своего развития и повсеместного распространения занимается именно таким переносом. Сеть постепенно становилась проще в использовании, частично потому что оборудование стало лучше, а частично потому, что сама стала быстрее и надежнее. Улучшались старые средства, появлялись новые, предназначенные для доступа к новым ресурсам, что облегчало использование сети. Этот процесс продолжается и развивается и по сей день.

Инструмент принятия решений

Интернет объединяет всю информацию в организации. Нет больше необходимости собирать по крупицам разрозненные данные, исходящие из разных источников. Вместо этого есть единый интерактивный интерфейс, включающий в себя удобные средства просмотра, стандартные запросы и средства поиска. Актуальная информация о движении денежных средств, рыночные тренды, результаты общения с коллегами и партнерами по бизнесу позволяет вовремя принимать ответственные деловые решения.

Инструмент бизнеса

Если обмен информацией происходит мгновенно, то те, кто ответственны за принятия решений, способны анализировать деловые процессы, деловые возможности, и деловые цели намного быстрее. Это связано с тем, что большее количество служащих могут принять участие в обсуждении решения. Проекты реализовываются более эффективно. Требования клиентов регистрируются и их твердо придерживаются. Развитие событий происходит в общедоступном электронном поле данных – гораздо быстрее, чем посредством встреч и телефонных

переговоров. Компания, которая управляет информацией подобным образом, узнает все практически мгновенно, одновременно совершенствуется и в итоге принимает верное управленческое решение.

Совершенный инструмент связи

Интернет обеспечивает интеграцию всех подразделений корпорации: компаний, отделов, рабочих групп, индивидуальных лиц. Поддерживает мгновенную передачу информации между любыми точками в организации, всякий раз, когда это необходимо, где бы человек не находился.

Инструмент сотрудничества

Интернет – это простой в освоении, мощный инструмент для сотрудничества, проектного управления, сбора данных, управляющий знаниями и информацией, доступен каждому в организации. Все лучшие действия и достижения людей помещаются на передний план: лучшая продукция, высокие продажи, отличное качество. Полная информация о прогрессивных технологиях, услугах для клиентов, технических процедурах, советы, предупреждения, ответы на часто задаваемые вопросы. Все это частично заменяет форумы, семинары и конференции по обмену опытом.

Инструмент эксперта

Эксперты – это люди, знающие свою работу лучше других. Они ответственны за распространение результатов своего труда среди всего персонала, а также за то, чтобы информация, исходящая от них, была правильно воспринята. Благодаря Интернет, можно быть постоянно на связи с экспертами и пользоваться их знаниями ежеминутно. Эти знания можно накапливать с тем, чтобы пользоваться ими и в дальнейшем.

Телефон XXI века

Трудно себе вообразить затраты на многочисленные служебные телефонные переговоры между разными странами и континентами. Интернет предоставляет уникальные возможности дешевой, надежной и конфиденциальной глобальной связи по всему миру. Это оказывается очень удобным для фирм, имеющих свои филиалы по всему миру, транснациональных корпораций и структур управления. Обычно использование инфраструктуры Интернет для международной связи обходится значительно дешевле прямой компьютерной связи через спутниковый канал или через телефон.

Инструмент контроля и совершенствования производственного цикла

Наверняка многие сотрудники больших компаний, специализирующиеся на каком-то узком виде деятельности, задавались вопросом: а как в самом деле работает весь производственный и коммерческий процесс? Интернет дает визуальное представление процессов, происходящих внутри организации: сделок, движения ресурсов, взаимодействия подразделений.

Инструмент маркетинга

Наверное, нет уже такой компании, которая не создала бы себе страничку в Интернет. Посредством Интернет есть возможность обмениваться информацией с партнерами по бизнесу относительно изделий, услуг, технологий, стандартов, новостей.

Инструмент клиента

Точно так же, как партнерам, клиентам можно сообщать необходимую информацию: новости, рекламу. Кроме того, всегда можно узнать, что клиенты думают об организации, их предложения и критику.

Инструмент маркетинга

Элементы традиционного делового маркетинга и коммерческих программ могут быть интегрированы внутри среды Web, чтобы создать целевой маркетинг, который удовлетворяет разнообразным запросам клиентов и обслуживает их в процессе продажи и сервиса. Результатом могут быть более выгодные, долгосрочные связи. Это взаимодействие сдвигает акценты от однонаправленного информационного потока до двустороннего диалога и сотрудничества, и от массовых рынков до рыночных сегментов.

Инструмент человеческих ресурсов

С недавних времен возникла необходимость постоянно переосмысливать характер связей человек-организация и человек-человек внутри нее. Теперь каждый служащий может немедленно отчитаться о работе, сообщить о своих решениях, передать опыт любому человеку.

Если ранее сеть использовалась исключительно в качестве среды передачи файлов и сообщений электронной почты, то сегодня решаются более сложные задачи распределенного доступа к ресурсам. Интернет, служивший когда-то исключительно исследовательским и учебным группам, чьи интересы простирались вплоть до доступа к суперкомпьютерам, становится все более популярным в деловом мире.

Компании соблазняют быстрота, дешевая глобальная связь, удобство для проведения совместных работ, доступные программы, уникальная база данных сети Интернет. Они рассматривают глобальную Сеть, как дополнение к своим собственным локальным сетям. При низкой стоимости услуг пользователи могут получить доступ к коммерческим и некоммерческим информационным службам США, Канады, Австралии и многих европейских стран. В архивах свободного доступа сети Интернет можно найти информацию практически по всем сферам человеческой деятельности, начиная с новых научных открытий до прогноза погоды на завтра.

И это только начало. Несомненно, в конечном счете, все придут к пониманию того, что наступает эра информации. Потребность в ней возрастает и будет возрастать лавинообразно, количество потребителей тоже. Никуда от этого не деться. Без надежной и оперативной информации нельзя идти в ногу со временем, развивать науку и технику на уровне лучших мировых образцов. И все мы – потенциальные пользователи глобальной информационной сети.

Интернет коммерческий

Прежде всего бизнес ждет колоссального роста продаж товаров и услуг через Интернет. Подобный «ненавязчивый» сервис предлагается на веб-сайтах многих производителей. Со временем на коммерческую основу могут перейти такие виды услуг, как трехмерные экскурсии по лучшим музеям мира, консультации специалистов по ремонту автомобилей, видеоконференции с врачами, беседы с адвокатами, онлайновые библиотеки. Кроме того, контроль за банковским счетом, осуществление страховых и прочих платежей – все это становится нормой.

Инструмент развлечений и потоковых технологий

Разнообразные потоковые технологии завоевывают все большую популярность у пользователей. Интернет-камеры, расставленные в удаленных точках мира, покажут вам картинку с «места события», а онлайновый политический обозреватель расскажет о последствиях кризиса в отдельно взятой стране. Серьезное внимание на потоковые технологии обращают такие монстры индустрии, как Intel, Siemens и Microsoft.

Инструмент издательской деятельности

Ни одна бумажная газета не может выйти так быстро, как сетевая. Онлайновая статья может создаваться практически на ваших глазах. К вашим услугам мгновенная обратная связь, актуальность, отсутствие 24-часового интервала для печати и доставки, накладных расходов.

Интернет как политический и идеологический инструмент

С его помощью можно устраивать соцопросы населения, а не проводить дорогостоящие референдумы при принятии каких-либо политических решений. Проще с помощью глобальной сети провести онлайновое голосование по животрепещущим вопросам общественной жизни.

Инструмент в образовании

Студент получает новую степень свободы, позволяющую ему гибко распоряжаться своим временем. Если не хватает семинарских занятий и «живого» общения – можно воспользоваться специально разработанными дистанционными курсами.

1.2 Основные понятия Интернет

Ресурс

В Интернете под словом «ресурс» понимается очень много объектов. Это могут быть документы самых разных типов (текстовые, звуковые, графические, видео), а также программы, данные и аппаратные средства.

Сервер сети

Сеть Интернет – это миллионы компьютеров, разбросанных по всему миру и соединенных друг с другом средствами связи в единую сеть. Эти компьютеры называют **серверами.**

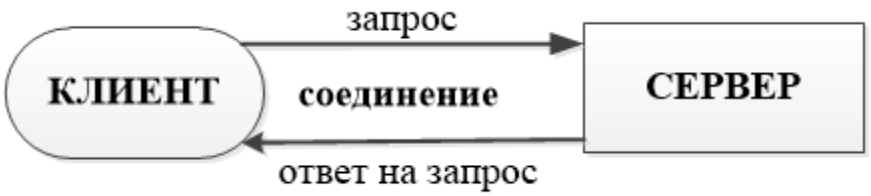

Рисунок 1.4 – Взаимодействие компьютера-сервера и компьютера-клиента

Владельцами этих серверов могут быть государственные организации, учебные заведения, крупные коммерческие организации и частные лица. Таким образом, «сервер» – это компьютер сети Интернет. Компьютер, к которому было осуществлено подключение, именуется сервером удаленного доступа. Компьютер, который присоединился, называется клиентом удаленного доступа (рис. 1.4).

Программа-клиент

Программа, которая работает на компьютере пользователя и предназначена для взаимодействия с сетью путем обращения к различным сетевым серверам (рис. 1.5). Программа, которая выполняется на удаленном компьютере и обрабатывает запросы программ-клиентов на выполнение определенных операций, также именуется сервером.

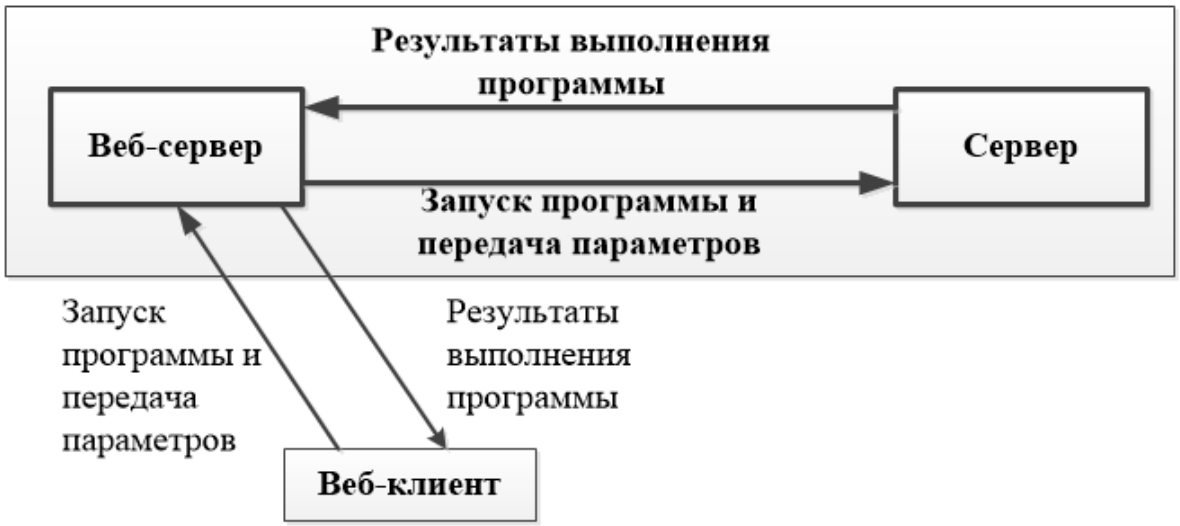

Рисунок 1.5 – Взаимодействие клиента с программой, выполняющейся на сервере

Онлайновый режим работы (online)

Когда пользователь выходит в Интернет, например, получает информацию из какого-либо банка данных, говорят, что он работает в онлайновом режиме.

Автономный режим работы (offline)

Если пользователь не выходит в сеть, а имеет дело лишь с банком данных, хранящимся, например, на CD-ROM его персонального компьютера, говорят, что он работает в автономном режиме.

Службы Интернета

Служба Интернет – это сетевая подсистема, обеспечивающая использование (распространение) определенной информации определенным образом.

Представьте себе железнодорожную сеть Украины. Все мы по ней ездили, и, наверное, каждый подсознательно понимает, что железнодорожная сеть – это система для перевозки пассажиров. Однако это не совсем так. Пассажирские перевозки – это только одна из служб железной дороги. Есть еще почтовые перевозки, багажные перевозки, службы, занимающиеся перевозкой промышленных грузов. Все эти службы разные, хотя и используют одну и ту же железнодорожную сеть. И человек и его багаж могут добраться по железной дороге на другой конец страны. Но, согласитесь, условия и сроки их путешествия будут разными. При этом обслуживать их будут разные люди и разные технические устройства, а сопровождать в пути их будут разные документы. Итак, в одной сети могут действовать разные службы.

В Интернете таких служб несколько. Наиболее популярными службами являются: электронная почта, World Wide Web, телеконференции, File Transfer Protocol (FTP), Internet Relay Chat (IRC) – программа трансляции разговора.

Гиперссылка

Это фрагмент текста, который является указателем на другой файл или объект. Обеспечивает переход от одного документа к другому.

Гипертекст

Это такой вид текста, в котором имеются гиперссылки.

HTML (Hyper Text Marcup Language)

Это язык разметки текста (независимый от платформ). Основное назначение языка HTML – определение логической структуры текста, обозначение в тексте заголовков, начала и окончания абзацев, авторских ударений на слова и так далее.

Веб-документы (веб-страницы)

Веб-страница – это отдельный гипертекстовый документ службы WWW. Все документы службы WWW имеют одинаковый

формат – так называемый формат HTML. Их называют веб-документами, HTML – документами или веб-страницами. Эти термины равнозначны.

Протоколы Интернета

Протокол – набор правил, согласно которому происходит взаимодействие компонентов сети. У каждой службы – свой протокол. Протокол – это набор правил, которые необходимо соблюдать, чтобы обе стороны (отправитель и получатель) могли четко взаимодействовать друг с другом. Протокол поездки по железной дороге предполагает, что надо заранее купить пассажирский билет, прийти точно в назначенное время к нужному поезду и нужному вагону, предъявить билет проводнику и в пути соблюдать правила, действующие на железной дороге. Несоблюдение любого из этих пунктов может не позволить совершить поездку. Со своей стороны, железная дорога должна обеспечить прибытие в пункт назначения в сроки, предусмотренные расписанием. Протокол отправки, например, багажа с личными вещами несколько иной. Его сдают на грузо-товарной станции под квитанцию. Здесь железная дорога тоже гарантирует его доставку, но она не может заранее назвать ни время, ни дату его прибытия. Зато она обещает известить получателя по почте или по телефону, когда багаж прибудет. Протокол также предполагает, что получатель будет платить штраф за хранение, если, получив извещение о прибытии багажа, не поторопится его забрать.

Получается, у разных служб могут быть разные протоколы. Точно так же и в Интернете. У электронной почты – свой протокол, а у WWW – свой, а у службы, предназначенной для передачи и приема файлов – свой. Соблюдением протоколов при передаче информации занимаются программы, работающие у отправителей и получателей.

IP-адрес

Это уникальный числовой адрес компьютера, используемый при передаче информации между компьютерами сети Интернет. Это 32-разрядное двоичное число, разделенное на группы по 8 бит, называемых октетами. Обычно IP-адрес записывается в виде четырех десятичных октетов и разделяется точками. Например: 123.45.67.89.

Протокол HTTP

Протокол передачи гипертекста HTTP (Hypertext Transfer Protocol) –

набор правил и процедур, регулирующих взаимодействие между веб-серверами и компьютером пользователя.

Протокол TCP/IP

Протокол передачи данных TCP/IP это своего рода единый свод правил, который понимают все компьютеры, подключенные к сети Интернет. Включает протоколы IP (Internet Protocol), задача которого – правильно адресовать пакеты данных, и TCP (Transmission Control Protocol), используемый для помещения данных в такие пакеты. Когда они доходят до получателя, протокол TCP вновь собирает из них сообщение.

Протокол FTP

Протокол передачи файлов FTP (File Transfer Protocol) является широко используемым протоколом для обмена файлами по любой сети, поддерживающей протокол TCP/IP (Интернет или интранет). В FTP-передаче участвуют два компьютера: серверный и клиент. FTP-сервер, который работает на программном обеспечении FTP, ожидает сигнала от сети для запроса подключений с других компьютеров. Клиентский компьютер, который работает на клиентском программном обеспечении FTP, инициирует подключение к серверу. Непосредственно после установки подключения клиент сможет выполнять ряд операций по обработке файлов, например, загружать файлы на сервер или с сервера, переименовывать, удалять файлы на сервере (рис. 1.6). Практически каждая компьютерная платформа поддерживает FTP-протокол. Это позволяет любому компьютеру, подключенному к сети, базирующейся на TCP/IP, провести операции с файлами на другом компьютере той же сети независимо от типа операционных систем (если компьютеры позволяют произвести FTP-доступ).

Рисунок 1.6 – Взаимодействие клиента с сервером по FTP-протоколу

URL (Uniform Resourse Locator)

Это универсальный указатель источника – точный адрес в сети Интернет, служащий для определения местонахождения документа и доступа к нему.

Интернет-провайдер

Так называется фирма, которая обеспечивает (provide) доступ в Интернет. Провайдер обычно не предоставляет клиентам никаких собственных материалов, а имеет лишь свою страницу во Всемирной паутине.

Это может быть компания, организация или бизнес структура, которая предоставляет своим потребителям и клиентам (как юридическим, так и физическим лицам) подключение к Интернету за определенную плату, как правило, ежемесячную. Различными провайдерами обслуживаются разные географические регионы и области. По своей сути это большая сеть, которая обслуживается различными организациями.

Условно провайдеры делятся на организации первого и второго звеньев:

– Мирового уровня. У провайдеров есть межконтинентальные узлы, обеспечивающие связь между материками. Это гиганты, которые держат всю сеть Интернет.

– Вторичные. К таким провайдерам относятся все остальные, распределяющие Интернет по городам и странам.

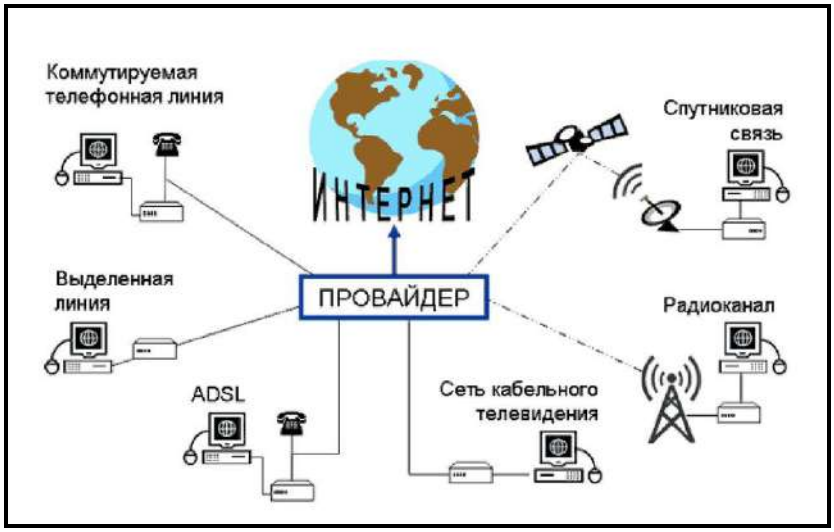

Рисунок 1.7 – Изображение провайдера, предоставляющего подключение к Интернет

В последние годы провайдеры часто объединяются, чтобы предлагать абонентам полный комплект услуг: кабельное телевидение, Интернет (рис. 1.7). Это операторы универсальных услуг связи.

Веб-браузер

Это специальная программа, которая позволяет путешествовать по Всемирной паутине. С помощью меню веб-браузера можно попасть практически на любой сервер в Интернет.

Веб-узел или веб-сайт

Отдельный сервер службы WWW вместе с теми документами, к которым он обеспечивает доступ. Это также может быть собрание логически и тематически связанных между собой документов.

В любом случае число документов, составляющих веб-узел, не менее двух. Это основной элемент WWW – определенное место или адрес, обратившись к которому, можно найти материалы по какой-либо конкретной теме. Существуют узлы, насчитывающие сотни и даже тысячи страниц. Связанные между собою веб-сайты и образуют Всемирную паутину.

1.3 Службы Интернет

Поиск и потребление информации и есть сама суть деятельности в сети Интернет. Взрывообразное развитие Интернет и популярность путешествий по WWW привело к тому, что многие пользователи отождествляют Интернет и World Wide Web, употребляя эти термины в одном и том же смысле. Это не совсем правильно – Интернет и Всемирная паутина не одно и то же. Интернет – термин собирательный, служащий для обозначения совокупности всемирных компьютерных сетей.

С технической точки зрения Интернет – это объединение транснациональных компьютерных сетей, работающих по различным протоколам, связывающих всевозможные типы компьютеров, физически передающих данные по всем доступным типам линий – от витой пары и телефонных проводов до оптоволокна и спутниковых каналов. Можно сказать, что Интернет – это сеть сетей, опутывающая весь земной шар. В Интернет присутствуют самые различные компьютерные службы, такие как электронная почта (E-Mail), система удаленного терминального доступа Telnet, система передачи файлов FTP и Всемирная паутина – WWW (World Wide Web).

Самое распространенное направление Интернет – **World Wide Web** (всемирная паутина или WWW или Web или W3) изначально придуманная для обмена исследовательской информацией, стала теперь частью повседневной жизни множества людей. Каждый может подключиться к ней, чтобы поработать над темой научных исследований, или изучить, что делают конкуренты.

В ноябре 1990 г. Тим Бернес-Ли из Европейского центра ядерных исследований (CERN) создал прототип первого сервера Всемирной паутины (WWW-сервера), который впоследствии коренным образом изменил облик Интернет.

Можно также создать свою домашнюю страничку и разместить её в WWW для всеобщего обозрения. Это очень удобный способ работы с информацией. Именно за счет WWW Сеть растёт так стремительно. Пользуясь несложным языком описания, можно составлять гипертекстовые документы для их последующей публикации в Интернет (под гипертекстом подразумевается документ, который может содержать все виды информации – от простого текста до мультимедийных роликов, а также ссылки на другие документы во всем мире). Чтобы увидеть содержание документа так, как его представляет себе его автор, нужно иметь на компьютере-клиенте программу просмотра – браузер. Эти гипертекстовые страницы содержат в себе множество самой разнообразной информации, которую может обработать и предоставить для восприятия компьютер: текстовой, графической, звуковой или видео.

В основе всей системы WWW лежат четыре понятия:

- единый формат документов (HTML);
- программы-клиенты для просмотра документов (браузеры);
- гипертекст;
- единая система адресации (URL).

Другими словами, основой WWW являются веб-страницы, которые содержат, кроме текста, графику, звук, видеозаписи и гипертекстовые ссылки Эти ссылки позволяют легко переходить от страницы к странице вне зависимости от географического положения веб-серверов, на которых эти страницы расположены. WWW работает по принципу клиент-сервер: существует множество серверов, которые по запросу клиента возвращают ему гипермедийный документ – документ, состоящий из частей с разнообразным представлением информации (текст, звук, графика,

трехмерные объекты), в котором каждый элемент может являться ссылкой на другой документ или его часть. Чтобы использовать WWW, пользователь должен иметь специальное программное обеспечение, которое, как правило, распространяется по сети бесплатно или поставляется в комплекте с большинством других программ и услуг Интернет.

Система электронной почты (E-mail). Весьма удобная и быстрая система общения людей, находящихся как в соседних зданиях, так и на разных концах земного шара. Многие компании предоставляют услуги передачи по E-mail новостей, рекламы и прочей информации. Отсутствие же электронного адреса на визитке представителя какой-либо компании сегодня вызывает усмешку: как если бы у него не было телефона.

Общемировой «чат», видеоконференции и социальные сети. Любой человек, имеющий выход в Интернет, может подключиться к специальным серверам и поучаствовать в разговоре со многими людьми в режиме онлайн, узнать необходимую информацию по интересующим его вопросам, а также познакомиться с новыми друзьями, партнерами по бизнесу или просто единомышленниками.

При достаточно высокой скорости передачи данных через Интернет можно смотреть телевидение и слушать радиостанции в режиме онлайн, то есть «живой эфир».

В деловом мире Интернет играет огромную роль: потому что наряду с чисто информативной функцией, он позволяет зарабатывать деньги. С помощью системы кредитных карточек в Интернет можно осуществлять платежи. Во-вторых, сейчас очень распространена система покупок через Интернет, а также заказ услуг, таких как снятие номера в гостинице, покупка билета на самолет либо поезд. В-третьих, деньги можно зарабатывать, пользуясь информацией из Интернет. Пользуясь этой информацией можно быть в курсе торгов на Гонконгской фондовой бирже, находясь в Украине.

Интернет решил проблему передачи информации, дав возможность получать её когда и где угодно, но не решил проблемы ее хранения и упорядочения. Большинство организаций пользуется информацией, знаниями, интеллектом, чтобы создать изделия, услуги, дать образование или развлечение. В прошлом или не было возможности получать надежную информацию, или невозможно было делать это вовремя. Теперь

информация находится непосредственно на рабочем столе безотносительно платформы или программного обеспечения. Благодаря Интернет любой пользователь, на любом уровне, может создавать информацию. Это делает информацию надежной, потому что она исходит прямо из источника. Человек обслуживает информацию, которая может читаться в любом браузере, и помещает ее на сервер. Это редактирование создает рабочий процесс внутри организации. Можно скрывать и предоставлять информацию в зависимости от своих нужд. Интернет может использоваться для различных функций внутри организации. Прикладные программы, которые использовались в течение многих лет, находят применение в Интернет. Интернет сегодня используется для поддержки автоматизированных систем принятия решений; коммерческих инструментальных средств; торговых систем; аналитических систем, работающих в режиме реального времени; финансовых систем торгового зала биржи; текстовых процессоров. Этот список можно продолжить.

Удаленный доступ (telnet). Работа на удаленном компьютере в режиме, когда компьютер пользователя эмулирует терминал удаленного компьютера. Сидя, например, в Швейцарии, можно работать на компьютере в США так, как если бы он стоял рядом.

Передача файлов (FTP – File Transfer Protocol) Протокол Передачи Файлов – протокол, определяющий правила передачи текстовых и двоичных файлов с одного компьютера на другой. Для работы с FTP нужно иметь доступ на ту удаленную машину, с которой вы хотите перекачать себе файлы. FTP также позволяет производить поиск файла на удаленной машине, то есть переходить из одной папки в другую, просматривать содержимое этих папок и файлов. Позволяет пересылать как файлы, так и папки целиком. Пересылка текстовых файлов, поддерживаемая этим протоколом, дает возможность автоматического перекодирования данных при пересылке текста на компьютер с другой кодировкой алфавита, что сохраняет прежний читаемый вид текста. Имеется возможность сжимать данные при пересылке и после их преобразовывать в прежний вид. FTP чаще всего используют для получения файлов из архивных хранилищ файлов. В настоящее время в Интернет появилось огромное количество FTP-узлов – «складов» программного обеспечения, графики, аудио данных, видео, огромных библиотек и просто разнообразных файлов. Для производителей

аппаратного и программного обеспечения стало весьма характерным организовывать FTP-узлы для поддержки и сопровождения клиентов. На этих FTP-узлах хранятся последние версии программ, драйверы поддержки аппаратных средств, которые клиенты могут загрузить бесплатно.

1.4 Структура глобальной сети интернет

Система доменных имён

Система доменных имён (Domain Name System) определяет всю систему адресации в Интернет. Она появилась в результате решения задачи создания архитектуры сети, способной сохранять работоспособность при значительных повреждениях. Особенность доменной структуры состоит в том, что каждый больший её участок повторяет структуру меньшего, как это показано на рисунке 1.8.

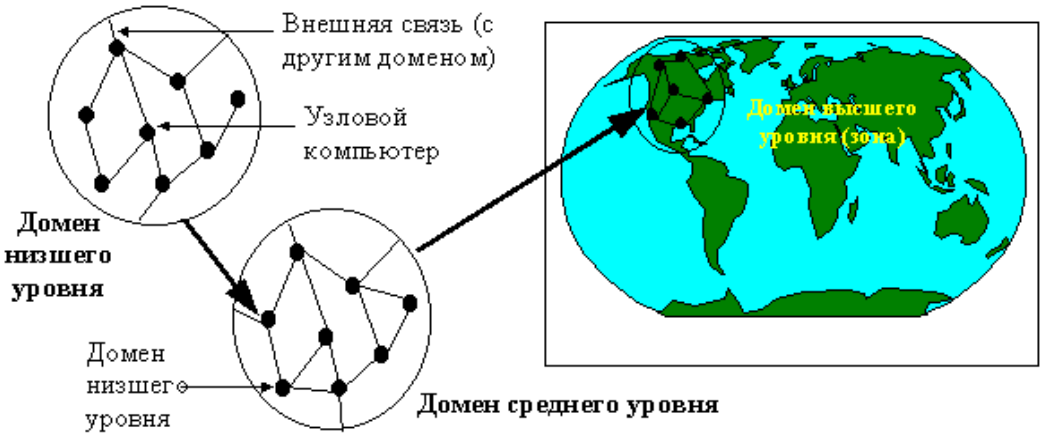

Рисунок 1.8 – Представление системы доменных имен

Домен низшего уровня представляет собой несколько узловых компьютеров (к ним часто подключены другие компьютеры конечных пользователей), соединенных линиями связи. Данные от одного компьютера на другой и дальше в Интернет могут передаваться по любой из линий связи, и, если несколько связей внутри домена оборвутся, всё равно передача данных будет осуществляться по оставшимся. Хорошо видно, что домен низшего уровня имеет несколько внешних, как правило, высокоскоростных, линий связи с другими доменами низкого уровня. Группа таких доменов образует, в свою очередь, домен среднего уровня, а домены среднего уровня образуют домен верхнего уровня, называемый зоной. Под зоной понимается часть земного шара, которую обслуживает данный домен Интернет.

Поскольку внутри домена любого уровня имеется достаточное количество линий связи, то даже при повреждении большого количества линий связи такая система будет работоспособна. Доменная структура Интернет чем-то напоминает паутину – при обрыве нескольких каналов связи она всё равно способна эффективно работать.

Каждый узел и домен Интернет имеет своё имя. Имена доменов высшего уровня – стандартные. Как правило, они состоят из двух латинских букв, определяющих код страны. Например: **ua** – Украина, **ru** – Россия, **ca** – Канада, **uk** – Великобритания, **fr** – Франция, **it** – Италия. Отдельно стоят имена высших доменов в США. Поскольку Интернет родилась и развивалась в США, то изначально все высшие домены (зоны) находились там же. Поэтому они несут трёхбуквенную кодировку. Это такие зоны, как:

- com – зона компаний и коммерческих предприятий;
- net – зона сетевых организаций;
- org – зона прочих организаций;
- mil – зона военных организаций;
- gov – зона правительственных организаций;
- int – зона международных организаций;
- edu – зона учебных организаций.

Домены среднего уровня чаще всего носят имена городов той или иной страны, или крупных компаний, оказывающих услуги в Интернет в рамках страны. Имя домена среднего уровня ставится перед зоной и отделяется от неё точкой, например: **kharkov.ua** – харьковский домен украинской зоны Интернет.

Каждый узловой компьютер, называемый часто просто узлом (от английского слова host), также имеет своё имя, которое ставится перед именем его домена и также отделяется точкой, например: **kname.edu.ua** – узел Интернет ХНУГХ им. А. Н. Бекетова. Таким образом, имя любого узлового компьютера является одновременно и его адресом, явно указывающем на его местонахождение в Интернет и косвенно – в мире. За появлением новых имён следит так называемый DNS-сервер, который хранит информацию о всех компьютерах, имеющихся в настоящий момент в Интернет. При попытке пользователя связаться с каким-либо компьютером в Интернет сервер DNS переводит буквенный адрес в так называемый IP-адрес. IP-адрес представляет собой последовательность

четырех чисел, разделённых точками – своеобразный номер компьютера в Интернет. Каждое число последовательности может быть двух- или трёхзначной. Например, для компьютера online.kharkiv.net IP-адрес выглядит как 192.168.156.18.

Таким образом, система доменных имен представляет собой сетевую инфрастуктуру, состоящую из серверов доменных имен, автоматически преобразующих запрос пользователя в текстовом формате (доменный адрес сайта) в цифровой IP-адрес компьютера, где находится искомый сайт. Термин DNS используется так же для обозначения стандарта на доменные имена и связанные с ними элементы инфраструктуры Интернета (например, DNS-сервер, настройки DNS, DNS-адрес).

Адресация документов в Интернет

Адрес документа в Интернет состоит из 3-х частей:

– Вначале указывается название протокола используемой службы Интернет. Для веб-документов оно имеет вид http://.

– Адрес сетевого компьютера в IP-виде или в форме URL.

– Полный путь к документу на сетевом компьютере.

Например:

– **http://www.microinform.ru/** – http-сервер должен найти и отправить клиенту (браузеру) стартовую страницу сайта компании «Микроинформ»;

– **http://www.microinform.ru/webschool/webschool.htm** – http-сервер должен найти и отправить клиенту (браузеру) файл **webschool.htm**, который находится в папке **webschool**.

Процесс взаимодействия клиент-сервер в этом случае представлен на рисунке 1.9.

Рисунок 1.9 – Представление адреса документа в Интернет

На рисунке 1.9 цифры обозначают следующее:

1 – Метод доступа к ресурсу, т. е. протокол доступа (http-доступ к веб-странице). Это наиболее часто используемый метод доступа к какому-либо HTML-документу в сети.

2 – Сетевой адрес ресурса (имя хост-компьютера).

3 – Полный путь к файлу на сервере.

1.5 История развития HTML

WWW – это глобальная система гипертекста, то есть текста со вставленными в него словами (командами) разметки, ссылающимися на другие фрагменты этого текста, другие документы, изображения.

К 1989 году гипертекст представлял новую, многообещающую технологию, которая имела относительно большое число реализаций с одной стороны, а с другой стороны делались попытки построить формальные модели гипертекстовых систем, которые носили скорее описательный характер и были навеяны успехом реляционного подхода описания данных. Идея Тима Бернерс-Ли заключалась в том, чтобы применить гипертекстовую модель к информационным ресурсам, распределенным в сети, и сделать это максимально простым способом. Он заложил три основные принципа системы из четырех существующих ныне, разработав:

– язык гипертекстовой разметки документов HTML (HyperText Markup Language);

– универсальный способ адресации ресурсов в сети URL (Universal Resource Locator);

– протокол обмена гипертекстовой информацией HTTP (HyperText Transfer Protocol).

Позже команда NCSA добавила к этим трем компонентам четвертый:

– универсальный интерфейс шлюзов CGI (Common Gateway Interface).

Идея HTML – чрезвычайно удачное решение проблемы построения гипертекстовой системы при помощи специального средства управления отображением. На разработку языка гипертекстовой разметки существенное влияние оказали два фактора: исследования в области интерфейсов гипертекстовых систем и желание обеспечить простой и быстрый способ создания гипертекстовой базы данных, распределенной в сети Интернет.

С появлением Интернет стала активно обсуждаться проблема интерфейса гипертекстовых систем, то есть способов отображения гипертекстовой информации и навигации в гипертекстовой сети. Значение

гипертекстовой технологии сравнивали со значением книгопечатания. Исходя из того, что лист бумаги и компьютерные средства отображения и воспроизведения серьезно отличаются друг от друга, то и форма представления информации тоже должна отличаться. Наиболее эффективной формой организации гипертекста были признаны контекстные гипертекстовые ссылки, и кроме того, было признано деление на ссылки, ассоциированные со всем документом в целом и отдельными его частями.

Самым простым способом создания любого документа является его набор в текстовом редакторе. Гипертекстовые системы имеют специальные программные средства построения гипертекстовых связей. Сами гипертекстовые ссылки хранятся в специальных форматах или даже составляют специальные файлы. Такой подход хорош для локальной системы, но не для распределенной на множестве различных компьютерных платформ. В HTML гипертекстовые ссылки встроены в тело документа и хранятся как его часть. Часто в системах применяют специальные форматы хранения данных для повышения эффективности доступа. В WWW документы – это обычные ASCII-файлы, которые можно подготовить в любом текстовом редакторе. Таким образом, проблема создания гипертекстовой базы данных была решена чрезвычайно просто.

В качестве базы для разработки языка гипертекстовой разметки был выбран SGML. Следуя академическим традициям, Бернерс-Ли описал HTML в терминах SGML (как описывают язык программирования в терминах формы Бекуса – Наура). Естественно, что в HTML были реализованы все разметки, связанные с выделением параграфов, шрифтов, стилей, так как реализация для NeXT подразумевала графический интерфейс. Важным компонентом языка стало описание встроенных и ассоциированных гипертекстовых ссылок, встроенной графики и обеспечение возможности поиска по ключевым словам.

Язык HTML предоставлял автору материалов, размещаемых на странице, широкие возможности в отношении того, как эту информацию показать пользователю. Но до 1996 – 97-х годов он обладал весьма скудными возможностями управления представлением информации и внешним видом страницы. Это являлось следствием большого числа нестандартизированных программ просмотра (браузеров) и многоплатформенности Интернет (пользователи входили в него под

разными операционными системами – UNIX, MaxOS, Windows). Каждый браузер в связи с этим, отображал информацию различным образом.

В основу синтаксиса языка HTML положен стандарт ISO 8879:1986 «Information processing. Text and office systems. Standard Generalised Markup Language (SGML)». С момента разработки первой версии языка (HTML 1.0) прошло уже много лет. За это время произошло довольно серьезное развитие языка. Почти вдвое увеличилось число элементов разметки, оформление документов все больше приближается к оформлению качественных печатных изданий, развиваются средства описания не текстовых информационных ресурсов и способы взаимодействия с прикладным программным обеспечением. Совершенствуется механизм разработки типовых стилей. Фактически в настоящее время HTML развивается в сторону создания стандартного языка разработки интерфейсов как локальных, так и распределенных систем.

Такой подход предполагает наличие еще одного компонента технологии — интерпретатора языка. В World Wide Web функции интерпретатора разделены между веб-сервером гипертекстовой базы данных и интерфейсом пользователя. Сервер, кроме доступа к документам и обработки гипертекстовых ссылок, обеспечивает предпроцессорную обработку документов, в то время как интерфейс пользователя осуществляет интерпретацию конструкций языка, связанных с представлением информации.

Если первая версия языка (HTML 1.0) была направлена на представление языка как такового, где описание его возможностей носило скорее рекомендательный характер, то вторая версия языка (HTML 2.0) фиксировала использование его конструкций. Версия представляла новые возможности, расширяя набор тегов HTML в сторону отображения научной информации и таблиц, а также улучшения стиля компоновки изображений и текста.

С появлением в мае 1996 года HTML 3.2 появилась возможность упорядочить все нововведения и согласовать их с существующей практикой. HTML 3.2 позволяет реализовать использование таблиц, выполнение кодов языка Java, обтекание графики текстом, а также отображение верхних и нижних индексов. Кроме возможностей разметки текста, включения мультимедиа и формирования гипертекстовых связей,

уже существовавших в предыдущих версиях HTML, в версию 4.01 включены дополнительные средства работы с мультимедиа, языки программирования, таблицы стилей, упрощенные средства печати изображений и документов. Для управления сценариями просмотра страниц веб-сайта (гипертекстовой базы данных, выполненной в технологии World Wide Web) можно использовать языки программирования этих сценариев, например, JavaScript, Java и VBScript.

Уже с 2013 года браузерами осуществлялась поддержка HTML 5, а разработчиками – использование рабочего стандарта. Цель разработки HTML 5 – улучшение уровня поддержки мультимедиа-технологий с одновременным сохранением обратной совместимости, удобочитаемости кода для человека и простоты анализа. В HTML 5 реализовано множество новых синтаксических особенностей.

Во время чтения гипертекста пользователь видит подсвеченные (выделенные) в тексте слова. Если щелкнуть на них курсором мыши, то высветится то, на что ссылалось это слово, например, другой параграф той же главы текста. Такое представление переходов называется ссылками (URL, Uniform Resource Locator – Унифицированный указатель ресурсов).

В Интернет по ключевым словам можно попасть в совершенно другой текст из другого документа, войти в какую-нибудь программу, произвести какое-либо действие, то есть можно получать доступ к чему угодно. Следовательно, можно ссылаться на данные на других компьютерах в любом месте сети, тогда при активации этой ссылки эти данные автоматически передадутся на компьютер пользователя, и на экране будет показан текст, данные, изображения, мультимедиа, речь.

Первым краеугольным камнем Интернет является то, что эта сеть напоминает паутину, в которой каждый узел или веб-страница представляют собой систему расходящихся связей с другими узлами или страницами, каждая из которых, в свою очередь, связана с еще большим числом страниц. Таким образом, в принципе, зайдя на один сервер можно посетить все серверы Интернет.

Вторым краеугольным камнем WWW стала универсальная форма адресации информационных ресурсов. Universal Resource Identification (URI) представляет собой довольно стройную систему, учитывающую опыт адресации и идентификации e-mail, Gopher, WAIS, telnet, ftp. Без наличия этой спецификации вся мощь HTML оказалась бы бесполезной.

URL используется в гипертекстовых ссылках и обеспечивает доступ к распределенным ресурсам сети. В URL можно адресовать как другие гипертекстовые документы формата HTML, так и ресурсы e-mail, telnet, ftp и многие другие.

Третьим краеугольным камнем является протокол обмена данными в World Wide Web – HyperText Transfer Protocol. Данный протокол предназначен для обмена гипертекстовыми документами и учитывает специфику такого обмена. Так как в процессе взаимодействия клиент может получить новый адрес ресурса в сети, запросить встроенную графику, принять и передать параметры, то управление в HTTP реализовано в виде ASCII-команд. Реально разработчик гипертекстовой базы данных сталкивается с элементами протокола только при использовании внешних расчетных программ или при доступе к внешним относительно WWW информационным ресурсам, например базам данных.

ASCII – это американский стандартный код для обмена информацией. Принятый большинством производителей компьютеров за основу, 7-битный набор символов ASCII может содержать не более 128 символов – служебные символы, знаки препинания, цифры и латинский алфавит. Соответственно, для других алфавитов места там просто нет. Существуют различные 8-битные (256 символов) расширения этого набора символов, однако не существует единого соглашения о том, какой из них должен стать стандартом.

Последняя составляющая технологии WWW это работа группы NCSA – спецификация Common Gateway Interface. CGI была специально разработана для расширения возможностей WWW за счет подключения всевозможного внешнего программного обеспечения. Такой подход логично продолжал принцип публичности и простоты разработки и наращивания возможностей WWW. Если команда CERN предложила простой и быстрый способ разработки баз данных, то NCSA развила этот принцип на разработку программных средств. Необходимо заметить, что в общедоступной библиотеке CERN были модули, позволяющие программистам подключать свои программы к серверу HTTP, но это требовало использования этой библиотеки. Предложенный и описанный в CGI способ подключения не требовал дополнительных библиотек и был довольно простым. Сервер взаимодействовал с программами через стандартные потоки ввода/вывода, чем упрощал программирование. При

реализации CGI чрезвычайно важное место заняли методы доступа, описанные в HTTP.

Ранее документы в Интернет были статическими, в том смысле, что вы можете хоть сотню раз загрузить одну и ту же страницу, но ее содержание от этого не изменится. Однако мы живем в таком мире, где новая информация пользуется повышенным вниманием. Один из методов привлечения внимания пользователей – это введение элемента интерактивности. С появлением CGI обеспечить этот элемент стало значительно проще. Кроме CGI, так же имеется такой инструмент, как форма. Формы являются составляющими той же модели взаимодействия, что и CGI-сценарии, и позволяют вводить данные для своего запроса или в качестве ответа – например, делать заказ на приобретение чего-либо, организовывать поиск необходимого ресурса, регистрироваться в гостевой книге какого-либо сайта. Это – «внешний интерфейс», с которым пользователи взаимодействуют.

CGI-сценарии составляют скрытую от глаз пользователя часть интерактивного взаимодействия. Они принимают информацию, посланную серверу через Web и обрабатывают ее, запрашивая базы данных, выполняя запросы пользователя или просто регистрируя полученные сведения. Все это происходит «за кадром», затем результаты передаются обратно. О такой HTML-странице говорят, что она формируется «на лету» или «динамически генерируется».

CGI – этот доступ к функциональным возможностям, предварительно не запрограммированным в сервере, что позволяет использовать все возможности компьютера, вместо того, что бы использовать только те, которые являются частью программного обеспечения HTTP-сервера.

Усложнение HTML и появление языков программирования для Интернет привело к тому, что разработка веб-узлов стала делом высокопрофессиональным, требующим специализации по направлениям деятельности и постоянного изучения новых веб-технологий. Однако возможности Интернет позволяет пользователям, владеющим основами HTML, создавать и размещать собственные веб-узлы без больших затрат.

Тим Бернерс-Ли – изобретатель Всемирной паутины

Сэр Тимоти Джон Бернерс-Ли (Sir Timothy John Berners-Lee)

(рис. 1.10) – британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины и действующий глава Консорциума Всемирной паутины. Автор концепции семантической паутины. Автор множества других разработок в области информационных технологий.

Рисунок 1.10 – Сэр Тимоти Джон Бернерс-Ли

Тим Бернерс-Ли родился в Лондоне (Англия). Его родители, Конвэй Бернерс-Ли и Мэри Ли Вудс, оба были математиками и трудились над созданием «Manchester Mark I», одного из первых компьютеров. Тим учился в школе Эмануэль в городе Вэндсворте, затем в Королевском колледже в Оксфорде. Там он собрал свой первый компьютер на базе процессора M6800 с телевизором вместо монитора. Один раз Тим и его друг были пойманы при проведении хакерской атаки, за это они были лишены права пользоваться университетскими компьютерами.

После окончания Оксфордского университета в 1976 году Бернерс-Ли поступил на работу в компанию «Plessey Telecommunications Ltd» в графстве Дорсет, где проработал два года, занимаясь в основном системами распределённых транзакций. В 1978 году Бернерс-Ли перешёл в компанию «D.G Nash Ltd», где занимался программами для принтеров и создал подобие многозадачной операционной системы. Затем он полтора года проработал в Европейской лаборатории по ядерным исследованиям ЦЕРН (Женева, Швейцария) консультантом по программному обеспечению. Именно там он для собственных нужд написал программу «Энквайр», которая использовала случайные ассоциации и заложила концептуальную основу для Всемирной паутины. С 1981 по 1984 год Тим Бернерс-Ли работал в компании «Image Computer Systems Ltd» системным архитектором.

С 1991 по 1993 год Тим Бернерс-Ли продолжал работу над Всемирной паутиной. Он собирал отзывы от пользователей и

координировал работу Паутины. Тогда он впервые предложил для широкого обсуждения свои первые спецификации URI, HTTP и HTML.

В 1994 году Бернерс-Ли стал главой кафедры Основателей «3Com» в Лаборатории информатики MIT. Он и сейчас является там ведущим исследователем. После слияния Лаборатории информатики с Лабораторией искусственного интеллекта в Массачусетском институте технологий образовалась хорошо известная Лаборатория информатики и искусственного интеллекта (CSAIL). В 1994 году он основал Консорциум Всемирной паутины при Лаборатории информатики. С тех пор и по сей день Тим Бернерс-Ли возглавляет этот консорциум. Консорциум занимается разработкой и внедрением стандартов для Интернета. Консорциум ставит перед собой задачу полностью раскрыть потенциал Всемирной паутины, сочетая стабильность стандартов с их быстрой эволюцией.

Главный литературный труд Бернерса-Ли – это книга «Плетя паутину: истоки и будущее Всемирной паутины» вышла в 1999 году. В этой книге он рассказывает о процессе создания Паутины, её концепции и своём видении развития Интернета.

Ещё одна книга Бернерса-Ли называется «Прядя семантическую паутину: полное раскрытие потенциала Всемирной паутины» вышла в 2005 году. В этой книге он раскрывает концепцию семантической паутины, в которой он видит будущее Интернета.

Семантическая паутина – это надстройка над существующей Всемирной паутиной, которая призвана сделать размещённую в сети информацию более понятной для компьютеров. При этом каждый ресурс на человеческом языке был бы снабжён описанием, понятным компьютеру. Семантическая паутина открывает доступ к чётко структурированной информации для любых приложений, независимо от платформы и независимо от языков программирования. Программы смогут сами находить нужные ресурсы, классифицировать данные, выявлять логические связи, делать выводы и даже принимать решения на основе этих выводов. При широком распространении и грамотном внедрении семантическая паутина может вызвать революцию в Интернете.

В декабре 2004 года Тим Бернерс-Ли стал профессором Саутгемптонского университета. При серьёзной поддержке университета он надеется осуществить проект семантической паутины.

Шестнадцатого июля 2004 года Королева Великобритании Елизавета II произвела Тима Бернерса-Ли в Рыцари. Сейчас сэр Тим живёт в пригороде Бостона с женой и двумя детьми.

1.6 История развития CSS

CSS «каскадные таблицы стилей» – одна из широкого спектра технологий, одобренных консорциумом W3C и получивших общее название «стандарты Web». В 1990-х годах стала ясна необходимость стандартизировать Web, создать какие-то единые правила, по которым программисты и вебдизайнеры проектировали бы сайты. Так появились языки HTML 4.01 и XHTML и стандарт CSS.

В самом начале 1990 года, для того, чтобы отображать веб-страницы, разные браузеры обладали своими собственными стилями. Развитие HTML было очень быстрым и он был способен удовлетворять на тот момент все существовавшие потребности по оформлению информации, именно поэтому тогда и не получил широкого признания CSS. И лишь Хокон Виум Ли в 1994 году предложил для HTML документов использование концепции CSS. В то время браузеры имели ограничение в функционале. А в 1990 году язык HTML, который создал Тим Бернерс-Ли, позволял сделать не визуальное, а структурное отображение документов. Один из основателей Netscape, Марк Андреесен, в 1994 году 13 октября сообщил, что доступна для тестирования от Netscape Navigator первая версия. И за три дня до проведения тестирования, норвежский программист, сейчас он является сотрудником компании Opera Software, Хокон Виум Ли публикует черновой вариант CSS. На сегодняшний день он имеет слишком мало схожего с принятыми современными стандартами, но именно тогда был заложен общий смысл. Самым первым, кто откликнулся на такую идею, был Берт Бос. В ноябре 1994 года в Чикаго на веб-конференции был предоставлен первый черновик CSS. Дебаты различного политического характера и разрешение некоторых технических вопросов продолжались в течение двух лет, но 1996 года 17 декабря W3C официально зарекомендовал CSS1.

Уровень CSS1

В первой версии CSS имелась возможность задавать гарнитуру и размер шрифта, а еще изменять его стиль: обычный, курсив или полужирный. Также определять рамки, фоны, цвета текста и другие

элементы страницы. Можно задавать расстояние между словами, межстрочные отступы и межсимвольный интервал. А также производить выравнивание текста, таблиц, изображений.

Имелись свойства внутренних и внешних отступов и рамок, ширины, высоты и блоков. Входили в данную спецификацию также ограниченные средства по позиционированию различных элементов веб-страницы.

Уровень CSS2

CSS2 основана на CSS1, а также сохранила обратную совместимость с добавлением некоторых функций, а именно:

– Возникло фиксированное, абсолютное и относительное позиционирование. С помощью чего появилась возможность управлять размещением элементов без табличной верстки на веб-странице.

– Для разных устройств отображения веб-страницы появилась возможность устанавливать разные стили.

– Для звуковых устройств появилась возможность определять громкость и голос.

– Возможно устанавливать на нечетных и четных страницах во время печати различные элементы оформления.

– Расширился механизм селекторов.

– Возможность добавлять содержимое, которое не содержится в исходном документе.

Консорциум W3C не поддерживает CSS2 и настоятельно рекомендует применять CSS2.1. CSS2.1 принята 2011 года 7 июня. Она основывается на CSS.2. В этой версии каскадных таблиц стилей исправлены ошибки и удалены некоторые функции.

Уровень CSS3

В отличие от предыдущих версий спецификация разбита на модули, разработка и развитие которых идёт независимо. CSS3 основан на CSS2.1 и дополняет существующие свойства и значения, а также добавляет новые.

Уровень CSS4

CSS4 разрабатывается консорциумом W3C с 29 сентября 2011 года. Модули CSS4 построены на основе CSS3 и дополняют их новыми свойствами и значениями.

Хокон Виум Ли – изобретатель каскадных таблиц стилей CSS

Норвежский учёный (родился 27 июля 1965 года), специалист в области Информационных технологий (рис. 1.11). Известен тем, что впервые сформулировал и предложил идею каскадных таблиц стилей CSS. Это произошло в 1994 году, когда Хокон работал для Консорциума Всемирной паутины (W3C) – организации, разрабатывающей и внедряющей технологические стандарты для Всемирной паутины.

Рисунок 1.11 – Хокон Виум Ли

CSS описывает, как документы будут представлены на экране монитора, на печати, или, может быть, в будущем, когда появятся качественные программы для озвучивания текста, как они произносятся.

Хокона Виум Ли с 1999 года работает главным инженером, норвежской компании Opera Software, которая разрабатывает популярный и очень распространенный браузер Opera. Здесь Хокон как раз и отвечает за разработку стилей и их применение в известном браузере. В 2005 г. предложил тест Acid2, предназначенный для улучшения поддержки стандартов CSS в браузерах. С 2005 г. является членом правления австралийской компании YesLogic, основным продуктом которой является Prince XML – приложение для создания качественных типографских публикаций в формате PDF, в основе работы которого лежит CSS-форматирование.

В 2006 году в Университете Осло Хокон защитил диссертацию по теме «Каскадные таблицы стилей», что стало логическим завершением разработки каскадных таблиц стилей CSS.

1.7 Соответствие документов веб-стандартам

Валидацией называется проверка документа на соответствие веб-стандартам и выявление существующих ошибок. Соответственно,

валидным является такой веб-документ, который прошел подобную процедуру и не имеет замечаний по коду. Код веб-страницы должен подчиняться определенным правилам, которые называются спецификацией, ее разрабатывает W3 Консорциум при поддержке разработчиков браузеров.

На первый взгляд, кажется, что валидация необходима, ведь речь идет о сокращении количества ошибок разработчиков и написании «правильного» кода. На деле все обстоит гораздо сложнее и вокруг валидации до сих пор ведутся горячие споры.

Позитивные стороны валидации

Хотя HTML-код имеет достаточно простую иерархическую структуру, при разрастании объема документа в коде легко запутаться, следовательно, совершить ошибку. Браузеры, несмотря на явно неверный код, в любом случае постараются отобразить веб-страницу. Но поскольку не существует единого регламента о том, как же должен быть показан некорректный документ, каждый браузер пытается сделать это по-своему. А это в свою очередь приводит к тому, что один и тот же документ может выглядеть по-разному в различных браузерах. Исправление явных промахов и систематизация кода приводит, как правило, к стабильному результату.

Негативные стороны валидации

Сайты, конечно же, делают для того, чтобы их посещали люди. Именно посетители выступают мерилом работы сайта, а их интересует информация и способ ее получения. Пользователь желает, чтобы сайт корректно отображался в его любимом браузере, быстро загружался и содержал те материалы, которые ему нужны. Обратите внимание, в этом списке нет ничего про код документа и его валидность, посетителей это просто не интересует. Поэтому совершенно невалидный сайт, но выполненный с душой, наполненный интересными материалами привлечет к себе больше посетителей, чем пустой ресурс, но сделанный по всем «правилам».

Преимущества валидации

Следование стандартам во многом дает множество выгод, которые проявляются в мелочах, в частности, объем кода веб-страницы становится

меньше, компактнее и читабельнее. Соответственно, для пользователей повышается скорость загрузки сайта в целом.

Тенденции валидации

Времена, когда производители браузеров добавляли уникальные возможности в свой продукт вопреки всем стандартам, начинают уходить в прошлое. Каждая новая версия браузера все больше поддерживает спецификации и отображает документы с минимальными ошибками или вообще без них. Разработчики сайтов, также придерживающиеся канонов веб-стандартов, таким образом соответствуют современным тенденциям развития веб-технологий.

Не стоит забывать и об XML (eXtensible Markup Language, расширяемый язык разметки). Этот язык становится стандартом де-факто для хранения данных и обмена информацией между разными приложениями. Синтаксис XML более жесткий, чем HTML и не прощает малейших ошибок. В каком-то смысле XML похож на языки программирования, в которых программа не будет скомпилирована, пока код не отлажен.

Как это не удивительно, но среди веб-разработчиков тоже существует своя мода. Текущая мода – создавать валидные документы и вывешивать специальный значок в виде картинки, что сайт соответствует спецификации HTML. Подобная тенденция затронула даже заказчиков сайтов и при написании технического задания на разработку сайта некоторые из них специально оговаривают, чтобы сайт был выполнен по веб-стандартам.

Проблемы отображения кода веб-страниц в разных браузерах

Разработчики браузеров не всегда следуют спецификации и в некоторых случаях трактуют код не по заданным правилам, а по-своему. В конечном итоге это приводит к тому, что веб-страница, которая правильно (так, как задумывали разработчики) отображается в одном браузере, выводится с ошибками в другом. Следование спецификации в подобных случаях, скорее всего, отпугнет пользователей некоторых браузеров. К примеру, Internet Explorer (IE), который по данным аналитики Net Applications в 2019 году занимает третье место среди браузеров, поддерживает спецификацию HTML и CSS хуже, чем Firefox и Opera. Очевидно, что пользователи Internet Explorer при посещении сайта

выполненного по всем стандартам, но не учитывающего специфику этого браузера, увидят неприглядную картину.

Для заказчиков сайта, а также их разработчиков подобная ситуация неприемлема, поэтому стоя перед выбором: стандарты или браузер, они в большинстве своем выбирают браузер. Получается неутешительная ситуация – тратить время на отладку кода для соответствия спецификации нет особой нужды. Это время лучше посвятить тому, чтобы документ без проблем работал в разных браузерах – так в основном и поступают веб-разработчики.

Так стоит ли проводить валидацию документов и заниматься этим этапом при написании веб-страниц? Безусловно, кому-то она поможет выявить существующие недочеты, кому-то – написать корректный код. Исправлять же ошибки, добиваясь полного соответствия стандартам, или оставить их ради совместимости с разными браузерами – здесь уже каждый разработчик решает сам, какие цели он преследует и что для него важнее.

РАЗДЕЛ 2
ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ ВЕБ-САЙТОВ

В данном разделе рассматривается понятие веб-сайта, его назначение, классификация и основные принципы верстки веб-сайтов с помощью каскадных таблиц стилей (CSS), а также основные рекомендации по их использованию.

2.1 Понятие веб-сайта

Веб-сайт (в переводе с английского) – это место (часть) во всемирной сети Интернет, которое имеет свой уникальный адрес (доменное имя), собственного владельца, и состоит из отдельных веб-страниц, которые пользователь видит как единое целое. веб-сайт состоит из файлов, которые хранятся на удаленном компьютере (сервере).

Отображение сайтов на компьютерах пользователей (читателей) происходит с помощью веб-браузеров – программ, которые с помощью http-запросов к серверу по определенным правилам обрабатывают и формируют страницы сайта. Это называется клиент-серверная технология.

Страницы сайта – это файлы, созданные с помощью языка HTML, содержащие в себе различную информацию: текстовую, фото, видео.

В современном мире сайт для бизнеса, образовательного учреждения является обязательным условием и дает шанс выжить среди множества конкурентов, потому что для потенциального клиента это удобно, и он может быстро выбрать, заказать или ознакомиться с определенной услугой, предоставляемой той или иной организацией.

Сайт позволяет организации, независимо от вида ее деятельности:

- увеличить имидж;
- увеличить доверие;
- увеличить поток клиентов;
- увеличить продажи;
- уменьшить количество рутинных вопросов;
- сэкономить время на документообороте.

Классификация веб-сайтов

По содержанию

Определяются основные два вида веб-сайтов по содержанию:

– статический – представляет собой набор заранее подготовленных файлов;

– динамический – представляет собой программный модуль, который генерирует страницы сайта на основе данных из различных источников.

Интернет-сайт может быть как обычным сайтом, так и Интернет-порталом, который, в свою очередь, бывает горизонтальным (охватывает много тем) и вертикальным (тематический). Сайт может быть интернациональным или региональным по охвату аудитории, на которую он рассчитан. Также по возможности доступа сайты делятся на корпоративные (доступны только для членов корпорации) и публичные (доступны для всех).

По виду

По виду сайты могут быть разделены:

– Сайт-визитка – это небольшой сайт, как правило, состоящий из одной или нескольких веб-страниц и содержащий основную информацию об организации, частном лице, компании, товарах или услугах, прайс-листы, контактные данные.

– Посадочная страница – это страница рекламного характера, содержащая информацию об услуге или товаре, переход на которую осуществляется по ссылке из поискового запроса или через баннер интернет-объявления.

– Блог – это интернет-журнал событий, интернет-дневник, онлайн-дневник. Это веб-сайт, основным содержимым которого являются регулярно добавляемые записи, содержащие текст, изображения или мультимедиа. Для блогов характерны недлинные записи временной значимости, расположенные в обратном хронологическом порядке – последняя запись сверху.

– Интернет-магазин – это сайт, торгующий товарами посредством сети Интернет, позволяет пользователям в своем браузере сформировать заказ на покупку, выбрать способ оплаты и доставки заказа, заплатить за заказ.

Веб-сервис – это сайт, созданный для выполнения каких-либо задач или предоставления услуг в рамках сети Интернет. Под эту категорию попадают социальные сети, поисковые и почтовые системы, форумы, фото и видео хостинги, облачные хранилища данных.

По особенностям тематики

По особенностям тематики сайт может быть:

– официальный (правительственный или корпоративный);

– коммерческий (интернет-магазин или промо-сайт);

– персональный (блог, сайт-визитка);

– информационный (новостной, поисковый, доска объявлений, словарь, энциклопедия или справочник);

– образовательный (каталог учебных заведений, сайт высшего учебного заведения или школы);

– развлекательный (информационно-развлекательный, сайт знакомств, сайт мобильного контента, туристический сайт, сайт ресторана или кафе).

Главным критерием причисления сайта к развлекательной тематике принято считать вид его контента, предложенный пользователю (читателю) – для отдыха, расслабления, получения позитивных эмоций и поднятия настроения.

Четко отделить информационно-развлекательные сайты от остальных сайтов довольно сложно. Создать четкое дерево классификации сайтов невозможно, так как дерево подразумевает независимость групп по разным его ветвям, а в Интернете все сайты каким-то образом связаны (совпадение по типам). Невозможно также собрать статистическую информацию о том, сколько сайтов того или иного типа есть в сети Интернет как в абсолютном, так и в процентом соотношениях.

2.2 Языки для создания веб-страниц

Каскадные таблицы стилей CSS (Cascading Style Sheets) используются создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится

с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

Каскадные таблицы стилей являются следствием дальнейшего развития HTML и дают возможность разработчикам сайтов перейти на следующий уровень представления информации. Таблицы стилей позволяют разделить смысловое содержимое веб-страницы и его оформление.

В первых версиях стандарта HTML не было предусмотрено никаких средств для управления внешним видом отображения информации на веб-странице. Общая концепция гипертекста была направлена на доступность информации для любых устройств, способных воспроизводить текст. Для разметки рекомендовалось использовать только логические теги, определяющие заголовки, подзаголовки, списки, абзацы, цитаты – то есть, те элементы, которые и составляют структуру документа. Интерпретация же внешнего вида зависела полностью от браузера, в котором пользователь просматривал веб-страницу.

С течением времени HTML развивался и стал представлять собой смесь логических и оформительских тегов, несовместимых расширений и полностью перестал отвечать первоначальной концепции – представлять информацию на любом устройстве независимо от его характеристик по выводу информации.

Тогда была предпринята широкомасштабная стандартизация. В результате чего появился стандарт HTML 3.2. Он не был революционным, а лишь расставил по местам все нововведения и выработал общие рекомендации для производителей браузеров. Революционные изменения были введены в новом стандарте – HTML 4.0 или, как его стали называть, Dynamic HTML. В обращение были введены слои, таблицы стилей и универсальная объектная модель браузера. В новом стандарте попытались вернуться к истокам концепции HTML. Четвертая версия, как и первая,

рекомендует создавать странички таким образом, чтобы они могли быть воспроизведены на любом устройстве – будь это монитор 21 дюйм или относительно небольшой экран мобильного телефона.

Каким же образом была решена проблема с представлением внешнего вида информации? В этом и заключается революционность подхода. Все оформление рекомендуется вынести во внешний стилевой файл. Основная же веб-страница будет содержать только информацию и ссылки на необходимые стили. При показе веб-страницы конкретному устройству должна быть задействована соответствующая таблица стилей. Для мобильного телефона и монитора компьютера они должны быть разными. В первом случае мы используем минимальное оформление, которое позволит представить информацию наиболее оптимально и компактно. Во втором же случае в нашем распоряжении имеется все богатство шрифтового и цветового оформления.

Таблицу стилей нужно написать всего один раз при создании сайта для каждого из устройств, на котором планируется вывод информации. К тому же таблица стилей может быть единой для целого сайта. И, следовательно, не нужно будет повторять одни и те же описания стилей на каждой из веб-страниц. Размещение всей стилевой информации в одном внешнем файле открывает и другие полезные возможности – ведь изменив содержимое только одного стилевого файла, можно в считанные секунды сменить весь дизайн сайта. Разумеется, это возможно лишь в том случае, если первоначально сайт был спроектирован верно.

2.3 CSS-вёрстка веб-страниц

До появления CSS оформление веб-страниц осуществлялось исключительно средствами HTML, непосредственно внутри содержимого документа. Однако с появлением CSS стало возможным принципиальное разделение содержания и представления документа. За счёт этого нововведения стало возможным лёгкое применение единого стиля оформления для массы схожих документов, а также быстрое изменение этого оформления.

Преимущества верстки веб-страниц с помощью каскадных таблиц стилей:

1. Несколько дизайнов страницы для разных устройств просмотра. Например, на экране дизайн веб-страницы будет рассчитан на большую

ширину, во время печати меню страницы не будет выводиться, а на мобильном телефоне меню будет следовать за содержимым.

2. Уменьшение времени загрузки страниц сайта за счет переноса правил представления данных в отдельный CSS-файл. В этом случае браузер загружает только структуру документа и данные, хранимые на странице, а представление этих данных загружается браузером только один раз и могут быть помещены в промежуточный буфер для более быстрого доступа.

3. Простота последующего изменения дизайна. Не нужно править каждую страницу, а лишь изменить CSS-файл с общим перечнем правил оформления.

4. Дополнительные возможности оформления. Например, с помощью CSS-вёрстки можно сделать блок текста, который остальной текст будет обтекать (например для меню) или сделать так, чтобы меню было всегда видно при прокрутке страницы.

Однако существуют и **недостатки при верстке веб-страниц с помощью каскадных таблиц стилей:**

1. Различное отображение вёрстки в различных браузерах (особенно устаревших), которые по-разному интерпретируют одни и те же данные CSS.

2. Часто встречающаяся необходимость на практике исправлять не только один CSS-файл, но и теги HTML, которые сложным и ненаглядным способом связаны с селекторами CSS, что иногда сводит на нет простоту применения единых файлов стилей и значительно удлиняет время редактирования и тестирования.

2.4 Основные рекомендации по использованию CSS

Главное правило, которого следует придерживаться при разработке веб-страниц сайта, следующее – **никто никогда не вернется на вашу страницу, чтобы полюбоваться ее дизайном.**

При оформлении сайта с помощью CSS, необходимо придерживаться следующих правил.

Если веб-страница занимает 5 Кб, а стилевой файл 4 Кб, то размер стилевого файла надо уменьшить как минимум в 5 раз. Старайтесь, чтобы ваши стилевые файлы занимали не более 500–1 000 байт. Иначе произойдет такая ситуация: при обращении к вашему сайту, после одной минуты

ожидания (стили загружаются до HTML страницы), пользователь увидит перед собой страницу с оглавлением, состоящем из пяти пунктов. Вряд ли он останется на вашем сайте, скорее всего уйдет с него и попытается найти сайт с аналогичной тематикой, загружающийся значительно быстрее.

Используя внешние таблицы стилей, можно создать единый дизайн для всех страниц вашего сайта. Если вдруг встанет задача смены дизайна всех страниц сайта, это можно будет легко сделать, исправив основной стилевой файл. Если браузер посетителя не поддерживает стили, то он не загрузит стилевой файл, если же были использованны стили, включенные с помощью тега <**style**>, браузер все равно загрузит содержимое этого тега. Если вы хотите изменить стиль конкретного документа, переопределив или доопределив какие-то стили внешнего стилевого файла, используйте стили, определенные с помощью тега <**style**>. Старайтесь не использовать встроенных стилей (определенных с помощью атрибута **style**).

Если для веб-страницы с помощью стилей были определены такие свойства документа, как цвет, размер и начертание шрифтов, цвет фона, фоновое изображение, тип маркеров списка, то браузер, не поддерживающий стили, выведет вашу страницу в приемлемом виде. Конечно, шрифты, цвета, цвет фона будут отображаться не так как было задуманно, но все же страница будет вполне читабельной. В случае использования таких свойств, как **top, bottom, left, right, position** для расположения элементов на веб-странице, при просмотрет в окне браузера, не поддерживающего стили страница вряд ли будет выглядеть привлекательно. Поэтому при использовании CSS необходимо всегда думать о том, как будут отображать веб-страницу браузеры, игнорирующие стили.

Представим себе следующую ситуацию – какому-либо текстовому блоку на веб-странице была задана ширина, равная 650 пикселей. При разрешении экрана 800 × 600 страница будет смотреться нормально. Но при разрешении 1024 × 768 окно браузера будет не целиком заполнено, что будет раздражать посетителей, а при разрешении 640 × 480 пользователь будет вынужден использовать нижнюю полоску прокрутки для просмотра страницы. Вследствие использования самых разных разрешений экранов, желательно избегать использования абсолютных единиц, лучше использовать относительные. Эта рекомендация, не относится к изображениям.

Правильная структура сайта должна быть простой. В ней не должно быть каких-либо подкаталогов и подразделов. Но при этом допускается

неограниченное количество второстепенных страниц. Рекомендуется создавать простые HTML-страницы. Не стоит увлекаться flash и frame, такие сайты будут уступать HTML-сайтам как по индексации, так и по выдаче в поисковых системах.

На каждой веб-странице в теге **title** желательно включать ключевые слова. URL веб-страницы должен быть осмысленным, то есть удобен для поисковых роботов, например – statya-saita.html.

Теги разметки страницы должны быть закрыты. Должны быть исключены все ссылки, ведущие в никуда. Не стоит перегружать сайт Java Script и другой графикой. Не используйте также в меню сайта Java Script, это может привести к проблемам с поисковыми роботами.

Верстку дизайна веб-страницы желательно использовать, разбивая содержимое на блоки с помощью тегов <**div**>, так как табличная верстка довольно тяжелая для поисковиковых роботов. Желательно также создать файл **robots.txt**, куда внести необходимые параметры для поисковых роботов.

По окончании верстки необходимо проверить сайт на наличие ошибок валидатором. В сайте должна быть простота, правильная структура и чистота кода. При наличии этого поисковый робот, скорее всего, проиндексирует ваш сайт.

Следовательно, применение каскадных таблиц стилей позволит перейти на новый уровень создания сайтов и добиться нужных эффектов оформления более простыми и логичными способами.

РАЗДЕЛ 3
ЗАДАНИЯ НА ПРАКТИЧЕСКИЕ ЗАНЯТИЯ ПО СОЗДАНИЮ САЙТА-ВИЗИТКИ

В данном разделе рассматриваются основные шаги по освоению принципов создания веб-страниц с помощью языка гипертекстовой разметки текста и их оформления с помощью каскадных таблиц стилей, а также добавление динамических эффектов к страницам статического сайта-визитки и основные принципы размещения созданного сайта в сети Интернет.

3.1 Основы веб-дизайна. Использование языка гипертекстовой разметки текста HTML для создания веб-документов

Цель: приобрести знания и умения в области создания веб-страниц с использованием языка гипертекстовой разметки текста HTML.

Назначение: выполнив работу, вы научитесь:

– создавать одиночные веб-страницы;

– оформлять веб-страницы по своему усмотрению;

– использовать при оформлении веб-страницы ссылки на рисунки и гиперссылки.

Полученные знания и умения помогут создать свою статическую веб-страницу.

Теоретические сведения

Язык HTML. Назначение языка и основные понятия

Для удобства чтения информация на веб-странице представляется не в виде непрерывного текста, а делится на различные элементы – отдельные строки, абзацы, списки, таблицы, рисунки, гиперссылки и т. п. Чтобы такая веб-страница выглядела одинаковым образом при просмотре пользователями в разных частях Земли с помощью различных браузеров, ее готовят специальным образом. Различные структурные элементы веб-страницы помечают соответствующими операторами, которые называются **тегами.** Когда браузер получает (загружает) веб-страницу с сервера, то

анализирует ее. Браузер находит в ней теги, которые указывают, какого типа элемент нужно отобразить.

В соответствии с этим он выводит фрагмент веб-страницы на экран.

Совокупность тегов и правил их использования называется **языком HTML** (HyperText Markup Language – язык разметки гипертекста).

Файл данных, фрагменты которого размечены тегами языка HTML, называется **HTML-документом.**

Теги. Структура HTML-документа

Тег – это указание браузеру, как отображать фрагмент документа. Тег всегда начинается символом **<** («меньше») и заканчивается символом **>** («больше»). Между этими символами указывается имя тега. В имени строчные и прописные символы не различаются. Например, тег <html> совпадает с тегом <HTML>, а также с тегом <HTml>. Для того чтобы отличить текст фрагмента документа от тега в дальнейшем имена тегов будем писать прописными буквами.

Теги бывают двух видов – парные (контейнер) и одиночные. Одиночный тег ставится перед фрагментом текста, которым он управляет. Например, запись

**
 Тетрадь**

означает, что слово «Тетрадь» будет выведено с новой строки.

Парные теги состоят из двух частей – начальной и завершающей. Начальная часть указывает точку, с которой начинается действие команды, а завершающая – точку ее окончания. Завершающая часть отличается от начальной наличием символа « / » (наклонная) перед именем команды. Например, запись

<B> Синий карандаш **</B>**

означает, что текст «Синий карандаш» будет виден полужирным шрифтом.

Разработка веб-страницы состоит из двух этапов:

– создание макета;
– вставка тегов языка HTML.

При создании макета определяется содержание веб-страницы и ее

внешний вид. Вставка тегов обеспечивает нужный внешний вид при просмотре веб-страницы с помощью браузера.

Признаком того, что текстовый файл является HTML-документом, служит парный тег <HTML> </HTML>. Он начинает и завершает весь документ, т. е. HTML-документ имеет вид:

<HTML>
текст документа
</HTML>

Все HTML-документы состоят из двух частей – **заголовка** и **тела**. Заголовок помечается парным тегом **<HEAD> … </HEAD>,** а тело – парным тегом **<BODY> … </BODY>.** Текст, который выводится в заголовке окна браузера, размещается в парном теге **<TITLE> … </TITLE>.**

В любом месте HTML-документа можно поместить комментарий. Он представляет собой текст, который не отображается на экране и служит для пояснений разработчика веб-страницы. Текст комментария помещается в парный тег **<! – – текст комментария – –>.**

Форматирование абзацев и символов

Чтобы текст страницы не отображался непрерывным текстом, используют следующие теги:

<P> текст абзаца **</P> –** абзац (законченная мысль);

**
** – начало новой строки, следующий текст будет начинаться с новой строки;

<HR> – горизонтальная линия.

При использовании тега **<P>** абзацы друг от друга отделяются пустой строкой, а не красной строкой (отступ первого слова вправо на 5 символов). Далее текст выводится с начала новой строки. Если указан тег **
,** то пустая строка не вставляется, а происходит принудительный перенос текста, следующего за этим тэгом на новую строку в переделах существующего абзаца.

Перед некоторыми фрагментами текста могут размещаться

заголовки. Они выделяются более крупным шрифтом и полужирным начертанием. В HTML имеются теги для определения заголовков шести уровней. Самым крупным шрифтом выделяются заголовки первого уровня, а самым мелким – шестого. Текст заголовка определяется парным тегом **<H_n> … </H_n>,** где n – номер уровня. Например, фрагмент HTML-документа

<H1> Добро пожаловать! **</H1>**

будет отображаться как заголовок первого уровня.

Абзацы и заголовки можно выравнивать по центру, левому или правому краю. Для этого в имени команды в начальной части тега нужно указать параметр ALIGN (выровнять) и после знака = (равно) одно из следующих значений:

CENTER – по центру;
LEFT – по левому краю;
RIGHT – по правому краю.

Например, фрагмент HTML-документа

<P ALIGN = «RIGHT»> Заходите еще! **</P>**

будет отображать текст «Заходите еще!» прижатым к правому краю.

Если в тексте HTML-документа имеются дополнительные пробелы между словами, то при выводе в браузере они будут убираться. Чтобы сохранить пробелы при отображении, нужно использовать парный тег **<PRE> … </PRE>.** Он помечает предварительно отформатированный текст, т. е. текст, отображаемый шрифтом фиксированной ширины (как правило, шрифтом Courier), с сохранением всех пробелов.

В HTML имеются парные теги форматирования символов (табл. 3.1).

Таблица 3.1 – Теги форматирования символов

Тег	Назначение
<B> … </B>	Полужирный
<I> … </I>	Курсив
<U> … </U>	Подчеркнутый

Списки и таблицы

Если требуется перечислять данные, то их выводят в виде списка. Наиболее часто используются списки следующих типов:

– маркированные – отображаются в виде последовательности помеченных элементов, обозначаются тегом **<UL> … </UL>**;

– нумерованные – отображаются в виде списка, элементы которого имеют последовательные номера, обозначаются тегом **<OL> … </OL>**;

Каждый элемент списка (строка) в списках маркированных и нумерованных заключается в парный тег **<LI>...</LI>** .

В языке HTML есть возможность представлять данные в виде таблицы. Для этого используется парный тег **<TABLE> … </TABLE>.** В нем «шапка» таблицы и данные задаются в виде последовательных строк. Каждая строка ограничивается парным тегом **<TR> … </TR>,** название каждого столбца таблицы в «шапке» задается парным тегом **<TH> … </TH>,** а значения в каждой ячейке области данных – парным тегом **<TD> … </TD>**.

Вставка рисунков

Для оживления внешнего вида веб-страницы часто используют рисунки. Чтобы поместить рисунок, необходимо предварительно подготовить файл, в котором он находится. Это осуществляется в специальных программах по работе с графикой. Обработка заключается в подборе такого расширения файла, которое поддерживается Web и количества пикселей по ширине и высоте изображения, позволяющего без потери качества получить минимальный размер в Кб. Затем в HTML-документе надо использовать тег **<IMG SRC = «Имя файла»>.** Например, если требуется поместить свою фотографию, которая содержится в файле «Это я.jpg», то нужно вставить в HTML-документ тег:

<IMG SRC = «Это я.jpg»>

Сам файл рисунка обязательно расположите в той же папке, где хранится html-файл веб-страницы.

Вставка гиперссылок и точек перехода внутри веб-страницы

При просмотре веб-страницы иногда требуется перейти из одной части в другую или даже на другую веб-страницу. Для организации таких переходов используют гиперссылки. На веб-странице они отображаются в

виде подчеркнутого текста оформленного другим цветом. При наведении указателя мыши на нее, указатель отображается в виде кисти руки, а при щелчке – происходит переход в указанное место.

Для организации перехода внутри HTML-документа используется два тега. Первый помечает точку, в которую нужно перейти. Он имеет вид:

<A NAME = «Название_точки»> </A>

Второй тег осуществляет команду перехода в заданную точку. Он имеет вид:

<A HREF = «#Название_точки»> Текст </A>

Текст между начальной и конечной частями тега отображается в браузере в виде гиперссылки. Название точки перехода выбирается произвольно.

Переход на другую веб-страницу осуществляется с помощью тега, который имеет вид **<A HREF = «URL-адрес»> Текст </A>.** В частности, в качестве URL-адреса можно указать имя файла.

Если после просмотра каждой из этих страниц требуется возвратиться обратно на начальную страницу своего веб-сайта, то в каждом из HTML-документов желательно разместить тег:

<A HREF = «URL-адрес главной страницы сайта»>
Переход на главную страницу </A>

Его можно вставить перед конечной частью тега </BODY>.

Изучив этот материал, вы сможете создать свои собственные первые веб-страницы.

Ход работы

Памятка разработчику веб-сайта

1. При разработке веб-сайта первым делом решите для себя, что вы хотите сказать с его помощью, выработайте его концепцию, структуру, содержание. Затем приступайте к выбору дизайнерского решения.

2. Для создания веб-странички необходимо длительное время собирать и систематизировать информацию. Попробуйте проиллюстрировать найденную вами информацию рисунками, схемами.

3. Для того чтобы размещенная (опубликованная) в сети информация имела высокий уровень, необходимо научиться трансформировать информацию, видоизменять ее объем, форму, знаковую систему.

Порядок выполнения работы

Откройте программу, предназначенную для написания HTML кода - **PSPadEditor (либо любую для работы с текстом – например «Блокнот»).** Войдите в меню **Файл**, выберите команду **Новый**. В появившемся окне выберите тип файла – **HTML miltihighlighter** для включения режима подсветки невалидного (не соответствующего стандартам) кода. Сохраните файл (**Файл / Сохранить)** с именем **«index.html»**. Благодаря этому файл будет отображаться в виде значка браузера, который установлен на данном компьютере по умолчанию. Двойной щелчок на значке файла открывает HTML-документ в браузере для просмотра.

Если требуется изменить HTML-документ в редакторе **PSPadEditor**, то следует запустить на выполнение этот редактор, а в нем открыть документ **«index.html»** командой **Файл / Открыть.**

Редактор **PSPad** является бесплатным текстовым редактором для программистов. Это маленький инструмент с простым управлением и мощными возможностями редактора кода. Вы можете пользоваться профессиональным инструментом, не платить денег, и при этом не воровать. Скачать текущую версию редактора вы можете с этого сайта **http://www.pspad.com/ru**

Работу над HTML-документом в редакторе **PSPadEditor** можно организовать следующим образом:

1. Открыть HTML-документ в **PSPad.**
2. После внести изменения в текст HTML-документа и нажать клавишу <**F10**>, чтобы увидеть изменения во встроенном просмотрщике, рассчитанном на **Internet Explorer**.
3. Периодически сохраняйте изменения в **PSPad** с помощью быстрых клавиш <**Ctrl+S**> либо кнопки на панели инструментов  .
4. Повторить пункты 2 и 3 до полной готовности веб-страницы.

!!! 5. Для поддержки кириллицы необходимо внести изменения в тег <meta>, а именно изменить кодировку с 1250 на 1251 – **<meta charset="windows-1251">**

Если вы редактируете HTML-документ в текстовом редакторе (например в «**Блокнот**»), то следует открыть файл **«index.html»** в этом редакторе, открыть браузер для просмотра (например **Firefox**) и в нем открыть тот же файл, расположить два окна слева направо на экране монитора (рис. 3.1). После внесения изменения в файл HTML (рис. 3.2) необходимо его сохранить (сочетание клавиш **Ctrl+S**) а затем обновить окно браузера (рис. 3.3) (клавиша **F5**).

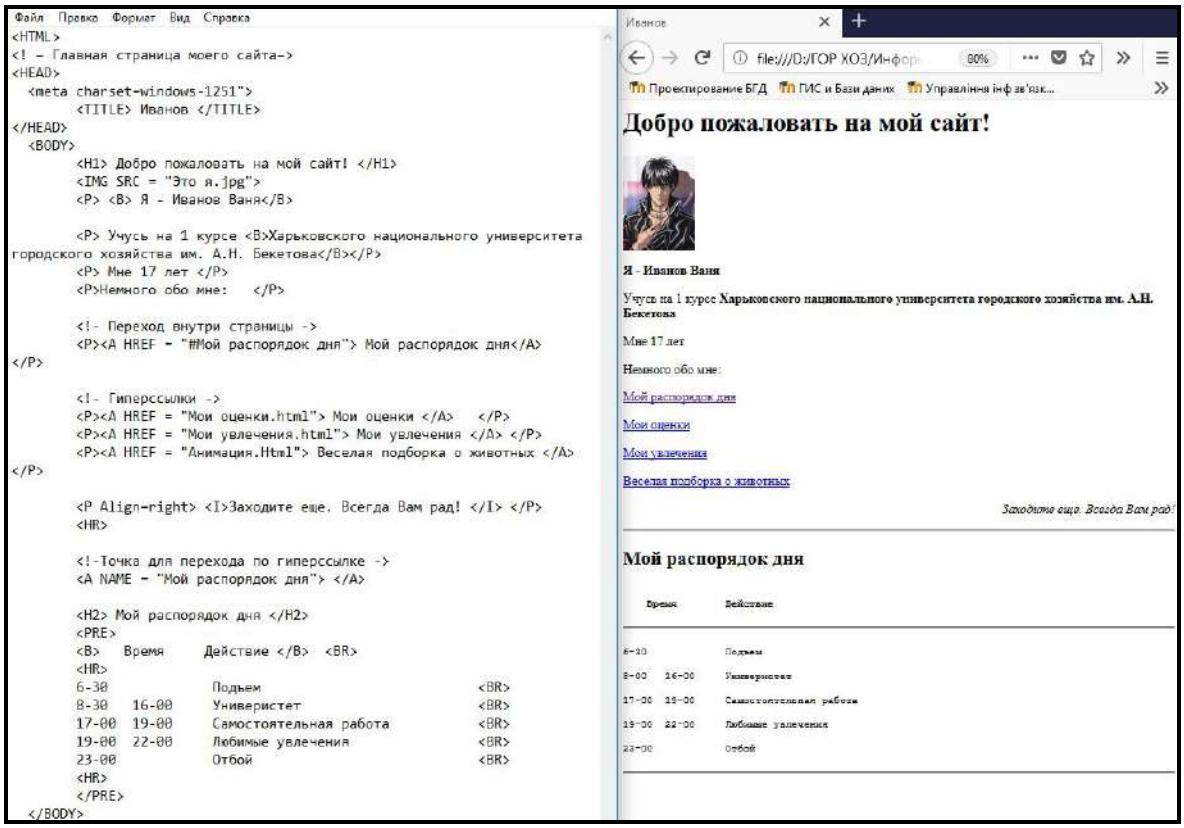

Рисунок 3.1 – Вид окон программы «Блокнот» и браузера, расположенных слева направо

В редакторе **PSPad** очень удобно пользоваться «быстрыми» клавишами для облегчения набора кода и проверки написанного кода:

– **F10** – открыть встроенный просмотрщик для отображения кода в браузере;

– **Ctrl + W** – перенос строк, не поместившихся в заданной на линейке ширины, на новую строку;

– **Ctrl + пробел** – открыть справочник тэгов, где нажав на первую букву тэга можно выбрать необходимый тэг и подставить значения его атрибутов;

– **Ctrl + S** – сохранить файл.

```
<HTML>
<! – Главная страница моего сайта–>
<HEAD>
  <meta charset=windows-1251">
        <TITLE> Иванов </TITLE>
</HEAD>
  <BODY>
        <H1> Добро пожаловать на мой сайт! </H1>
        <IMG SRC = "Это я.jpg">
        <P> <B> Я - Иванов Ваня</B>

        <P> Учусь на 1 курсе <B>Харьковского национального университета
городского хозяйства им. А.Н. Бекетова</B></P>
        <P> Мне 17 лет </P>
        <P>Немного обо мне:   </P>

        <!- Переход внутри страницы ->
        <P><A HREF = "#Мой распорядок дня"> Мой распорядок дня</A>
</P>

        <!- Гиперссылки ->
        <P><A HREF = "Мои оценки.html"> Мои оценки </A>   </P>
        <P><A HREF = "Мои увлечения.html"> Мои увлечения </A> </P>
        <P><A HREF = "Анимация.Html"> Веселая подборка о животных </A>
</P>

        <P Align=right> <I>Заходите еще. Всегда Вам рад! </I> </P>
        <HR>

        <!-Точка для перехода по гиперссылке ->
        <A NAME = "Мой распорядок дня"> </A>

        <H2> Мой распорядок дня </H2>
        <PRE>
        <B>   Время     Действие </B>  <BR>
        <HR>
        6-30             Подъем                              <BR>
        8-30   16-00     Универистет                         <BR>
        17-00  19-00     Самостоятельная работа              <BR>
        19-00  22-00     Любимые увлечения                   <BR>
        23-00            Отбой                               <BR>
        <HR>
        </PRE>
  </BODY>
</HTML>
```

Рисунок 3.2 – Вид HTML-документа в окне редактора кода

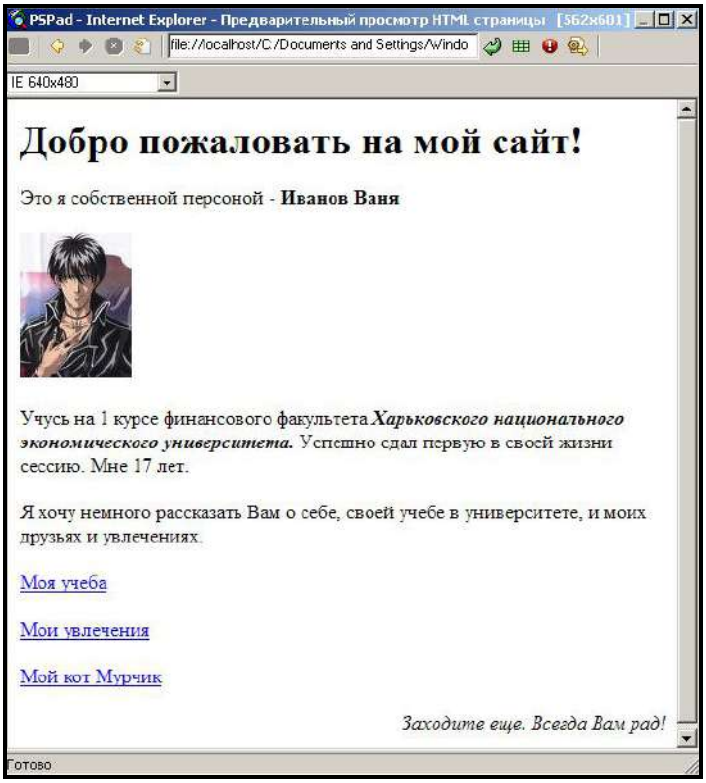

Рисунок 3.3 – Внешний вид веб-страницы

Изучив этот материал, вы сможете создать свои собственные первые веб-страницы. Пока что они будут выглядеть не совсем так, как предполагалось в вашем дизайнерском решении. Но это только первые шаги в разработке сайта.

Задания для самостоятельного выполнения

Создать сайт на следующую тему в соответствии со своим номером по журналу студенческой группы.

1. Как я провел лето.
2. Моя учеба в течении 1 семестра в университете.
3. Кем и где я хочу работать после получения диплома.
4. Мой самый любимый предмет.
5. Мои увлечения.
6. Почему я поступил в ХНУГХ им. А. Н. Бекетова?
7. Чего я хочу достигнуть в будущем.
8. Мое домашнее животное.
9. Мои друзья.
10. Моя семья.
11. Что мне понравилось в студенческой жизни.

12. Мой родной город.
13. Моя родина (страна).
14. Как я собираюсь провести летние каникулы.
15. Любая произвольная тема.

Требования к создаваемому сайту

1. Сайт должен содержать минимум 3 страницы.
2. Текст на страницах должен быть отформатирован (первые 2 требования выполнить в ходе этой лабораторной работы).
3. На страницах должны присутствовать ссылки на рисунки и гиперссылки.
4. Необходимо предусмотреть возможность перехода с главной страницы к другим страницам сайта и обратно.

3.2 Создание контента веб-страницы. Списки, таблицы

Цель: приобрести знания и умения в области создания веб-страниц с использованием языка гипертекстовой разметки текста HTML.

Назначение: выполнив работу, вы научитесь:

– создавать связанные гиперссылками веб-страницы;
– оформлять веб-страницы с использованием списков и таблиц;
– форматировать контент веб-страницы.

Полученные знания и умения помогут объединить отдельные веб-страницы в сайт и наполнить содержимое отдельных страниц табличными данными и списками различной тематики.

Теоретические сведения

Вставка рисунков

Для оживления внешнего вида веб-страницы часто используют рисунки. Чтобы поместить рисунок, необходимо предварительно подготовить файл, в котором он находится. Это осуществляется в специальных программах по работе с графикой. Обработка заключается в подборе такого расширения файла, которое поддерживается Web и количества пикселей по ширине и высоте изображения, позволяющего без потери качества получить минимальный размер в Кб. Затем

в HTML-документе надо использовать тег **<IMG SRC = «Имя файла»>.** Например, если требуется поместить свою фотографию, которая содержится в файле «**My.jpg**», то нужно вставить в HTML-документ тег:

<IMG SRC = «My.jpg»>

Сам файл рисунка обязательно расположите в той же папке, где хранится html-файл веб-страницы.

Для вставки фонового изображения веб-страницы необходимо использовать параметр **background** тега <body>:

<body background=’back.jpg’>

Атрибуты (свойства тэга) IMG

Alt – указывает всплывающий текст подсказки, который будет отображаться, если изображение не найдено или отключено в браузере пользователя.

Border – рамка вокруг изображения. Если изображение является ссылкой, то браузер рисует рамку по умолчанию.

Height, width – ширина и высота изображения. Не рекомендуется к использованию. Лучше оговорить эти параметры в графическом редакторе.

Align – отвечает за выравнивание изображения на веб-странице. Имеет следующие значения:

– **bottom** – выравнивание нижней границы изображения по окружающему тексту.

– **left** – выравнивает изображение по левому краю окна.

– **middle** – выравнивание середины изображения по базовой линии текущей строки.

– **right** – выравнивает изображение по правому краю окна.

– **top** – верхняя граница изображения выравнивается по самому высокому элементу текущей строки.

hspace = «единицы измерения» – отступы слева и справа от изображения.

vspace = «единицы измерения» – отступы сверху и снизу от изображения.

Форматы изображения

gif – поддерживает только 256 цветов, но при этом поддерживает прозрачность и анимацию.

jpeg – поддерживает 16 миллионов цветов. Но при этом теряется качество изображения, не поддерживает прозрачность и анимацию.

png – поддерживает 16 миллионов цветов, поддерживает прозрачность, качество изображения обычно лучше, чем у jpg при сопоставимом размере файла. Менее распространен, чем jpg.

bmp – поддерживает 16 миллионов цветов, не поддерживает прозрачность и анимацию. Качество изображения не теряется, но при этом размер файла в разы больше, чем у остальных форматов.

Списки

Если требуется перечислять данные, то их выводят в виде списка. Наиболее часто используются списки следующих типов:

– маркированные – отображаются в виде последовательности помеченных элементов, обозначаются тегом **<UL> … </UL>**;

– нумерованные – отображаются в виде списка, элементы которого имеют последовательные номера, обозначаются тегом **<OL> … </OL>**.

Каждый элемент списка (строка) в списках маркированных и нумерованных заключается в парный тег **<LI>...</LI>**.

Маркированный список

Маркированный список определяется тем, что перед каждым элементом списка добавляется небольшой маркер, обычно в виде закрашенного кружка (табл. 3.2). Сам список формируется с помощью контейнера <ul>, а каждый пункт списка начинается с тега <li>.

В списке непременно должен присутствовать закрывающий тег , закрывающий тег должен четко разделять элементы списка (рис. 3.4).

```
<ul >
 <Li>Стол</li>
 <Li>Стул</Li>
 <Li>Табурет</Li>
</ul>
```

- Стол
- Стул
- Табурет

Рисунок 3.4 – Вид маркированного списка (справа) и его HTML код (слева)

Таблица 3.2 – Стили маркеров списка

Тип списка	Код HTML	Пример
Список с маркерами в виде окружности с заливкой	<ul type="disc"> ... 	• Первый • Второй • Третий
Список с маркерами в виде окружности без заливки	<ul type="circle"> ... 	◦ Первый ◦ Второй ◦ Третий
Список с квадратными маркерами	<ul type="square"> ... 	▪ Первый ▪ Второй ▪ Третий

Вид маркеров может незначительно различаться в разных браузерах, а также при смене шрифта и размера текста.

Нумерованный список

В нумерованном списке добавляются автоматические отступы сверху, снизу и слева от текста списка.

В качестве нумерующих элементов могут выступать следующие значения:

– арабские числа (1, 2, 3, ...);
– прописные латинские буквы (A, B, C, ...);
– строчные латинские буквы (a, b, c, ...);
– прописные римские числа (I, II, III, ...);
– строчные римские числа (i, ii, iii, ...).

Для указания типа нумерованного списка применяется атрибут **type** тега <**ol**>. Его возможные значения приведены в таблице 3.3.

Чтобы начать список с определенного значения, используется атрибут **start** тега <**ol**>. При этом не имеет значения, какой тип списка установлен с помощью **type**, атрибут **start** одинаково работает и с римскими и с арабскими числами.

Таблица 3.3 – Типы нумерованного списка

Тип списка	Код HTML	Пример
Арабские числа	<ol type="1"> ... 	1. Чебурашка 2. Крокодил Гена 3. Шапокляк
Прописные буквы латинского алфавита	<ol type="A"> ... 	A. Чебурашка B. Крокодил Гена C. Шапокляк
Строчные буквы латинского алфавита	<ol type="a"> ... 	a. Чебурашка b. Крокодил Гена c. Шапокляк
Римские числа в верхнем регистре	<ol type="I"> ... 	I. Чебурашка II. Крокодил Гена III. Шапокляк
Римские числа в нижнем регистре	<ol type="i"> ... 	i. Чебурашка ii. Крокодил Гена iii. Шапокляк

```
<ol type="I" start="1000">
 <Li>Чебурашка</li>
 <Li>Крокодил Гена</Li>
 <Li>Шапокляк</Li>
</ol>
```

M. Чебурашка
MI. Крокодил Гена
MII. Шапокляк

Рисунок 3.5 – Вид нумерованного списка (справа) и его HTML код (слева)

В примере показано создание списка с использованием римских цифр в верхнем регистре, начинающихся с 1 000 (рис. 3.5).

Таблицы

В языке HTML есть возможность представлять данные в виде таблицы. Таблица состоит из строк и столбцов ячеек, которые могут содержать текст и рисунки. Обычно таблицы используются для упорядочения и представления табличных данных.

Тэг <**table**> служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов <**tr**> и <**td**>.

Тэг <**td**> предназначен для создания одной ячейки таблицы. Тег <**td**> должен размещаться внутри контейнера <**tr**>, который в свою очередь располагается внутри тега <**table**>.

Тег <**th**> предназначен для создания одной ячейки таблицы, которая обозначается как заголовочная. Текст в такой ячейке отображается браузером обычно жирным шрифтом и выравнивается по центру.

Тег <**tr**> служит контейнером для создания строки таблицы.

Атрибуты тега <table>

align

Задает выравнивание таблицы по краю окна браузера. Допустимые значения: **left** – выравнивание таблицы по левому краю, **center** – по центру и **right** – по правому краю. Когда используются значения **left** и **right**, текст начинает обтекать таблицу сбоку и снизу.

bgcolor

Устанавливает цвет фона таблицы.

border

Устанавливает толщину границ в пикселах. Граница отображается вокруг таблицы и между ячейками.

cellpadding

Определяет расстояние между границей ячейки и ее содержимым. Этот атрибут добавляет пустое пространство к ячейке, увеличивая тем самым ее размеры. Без **cellpadding** текст в таблице «наплывает» на рамку, ухудшая тем самым его восприятие. Добавление же cellpadding позволяет улучшить читабельность текста. При отсутствии границ особого значения этот атрибут не имеет, но может помочь, когда требуется установить пустой промежуток между ячейками.

cellspacing

Задает расстояние между внешними границами ячеек. Если установлен атрибут **border**, толщина границы принимается в расчет и входит в общее значение.

cols

Атрибут **cols** указывает количество столбцов в таблице, помогая браузеру в подготовке к ее отображению. Без этого атрибута таблица будет показана только после того, как все ее содержимое будет загружено в браузер и проанализировано. Использование атрибута **cols** позволяет

несколько ускорить отображение содержимого таблицы.

rules

Сообщает браузеру, где отображать границы между ячейками. По умолчанию рамка рисуется вокруг каждой ячейки, образуя тем самым сетку. Толщина границы указывается с помощью атрибута **border**.

width

Задает ширину таблицы. Если общая ширина содержимого превышает указанную ширину таблицы, то браузер будет пытаться «втиснуться» в заданные размеры за счет форматирования текста. В случае, когда это невозможно, например, в таблице находятся изображения, атрибут **width** будет проигнорирован, и новая ширина таблицы будет вычислена на основе ее содержимого. Как и в случае с высотой, если ширина явно не указана, то она будет вычисляться на основе содержимого таблицы.

Атрибуты тега <td>

Каждая ячейка таблицы, задаваемая через тег <**td**>, в свою очередь тоже имеет свои атрибуты, часть из которых совпадает с атрибутами тега <**table**>.

align

Задает выравнивание содержимого ячейки по горизонтали. Возможные значения: **left** – выравнивание по левому краю, **center** – по центру и **right** – по правому краю ячейки.

bgcolor

Устанавливает цвет фона ячейки. Используя этот атрибут совместно с атрибутом **bgcolor** тега <**table**> можно получить разнообразные цветовые эффекты в таблице.

colspan

Устанавливает число ячеек, которые должны быть объединены по горизонтали. Этот атрибут имеет смысл для таблиц, состоящих из нескольких строк в случае объединения нескольких ячеек в строке – по горизонтали (рис. 3.6).

Рисунок 3.6 – Вид таблицы с объединенными ячейками в первой строке

height

Браузер сам устанавливает высоту таблицы и ее ячеек исходя из их содержимого. Однако при использовании атрибута **height** высота ячеек будет изменена. Здесь возможны два варианта. Если значение **height** меньше, чем содержимое ячейки, то этот атрибут будет проигнорирован. В случае, когда установлена высота ячейки, превышающая ее содержимое, добавляется пустое пространство по вертикали.

rowspan

Устанавливает число ячеек, которые должны быть объединены по вертикали. Этот атрибут имеет смысл для таблиц, состоящих из нескольких строк в случае объединения нескольких ячеек в колонке – по вертикали (рис. 3.7).

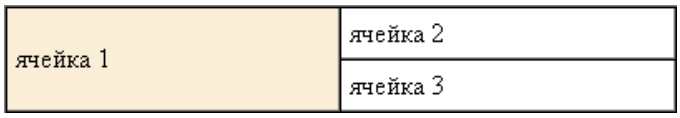

Рисунок 3.7 – Вид таблицы с объединенными ячейками в первой и второй строке

Ниже представлен пример кода для таблицы (рис. 3.8).

```
   <table bordercolor="blue" border="4" bgcolor="red">
<tr>
  <th colspan=3> Прайс-лист</th>
</tr>
<tr>
  <th> № п/п</th>
  <th bgcolor="green"> НАИМЕНОВАНИЕ </th>
  <th> Цена</th>
</tr>
<tr>
  <td> 1</td>
  <td> Болт</td>
  <td> 10 грн.</td>
</tr>
<tr>
  <td> 2</td>
  <td> Гайка</td>
  <td> 20 грн.</td>
</tr>
   </table>
```

Прайс-лист		
№ п/п	НАИМЕНОВАНИЕ	Цена
1	Болт	10 грн.
2	Гайка	20 грн.

Рисунок 3.8 – Вид таблицы (справа) и ее HTML код (слева)

Форматирование шрифта. Тег <font>

Тег <**font**> представляет собой контейнер для изменения

характеристик шрифта, таких как размер, цвет и гарнитура. Хотя этот тег до сих пор поддерживается всеми браузерами, он считается устаревшим и от его использования рекомендуется отказаться в пользу стилей.

<**font**>Текст</**font**> закрывающий тег обязателен.

Атрибуты тега <font>

color

Устанавливает цвет текста.

face

Определяет гарнитуру шрифта.

size

Задает размер шрифта в условных единицах.

Ниже представлен пример кода для текстового абзаца и вид результата в окне браузера (рис. 3.9).

<**p**><**font size**="5" **color**="**red**" **face**="**Arial**">П</**font**>ервая буква этого предложения написана шрифтом Arial, выделена красным цветом и увеличена в размерах.</**p**>

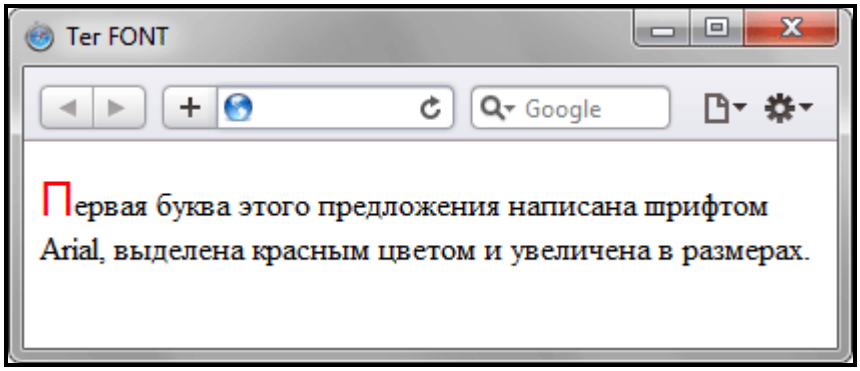

Рисунок 3.9 – Вид оформления первой буквы абзаца текста в окне браузера

Оформление веб-страницы. Тег <body>

Парный тег <**body**> предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера. Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера <**body**>. К такой информации относится текст, изображения, теги, скрипты JavaScript. Тег <**body**> также применяется для определения цветов ссылок и текста на веб-странице.

Атрибуты тега <body>

background

Задает фоновый рисунок на веб-странице.

```
<body background="URL">
...
</body>
```

bgcolor

Задает фон на веб-странице.

bgproperties

Определяет, прокручивать фон совместно с текстом или нет.

```
<body bgproperties="fixed">
... фон веб-страницы (фоновый рисунок) фиксируется
</body>
```

text

Устанавливает цвет текста, используемого на веб-странице по умолчанию. Цвета отдельных элементов можно легко изменять с помощью стилей.

```
<body text="цвет">
...
</body>
```

Представление цветов в HTML-документах

В HTML-документе существует возможность указывать цвет фона документа, фона ячеек таблиц и цвета текстовых фрагментов. Значения цвета можно задавать в одной из трех форм – имя; составляющие красного, зеленого, синего цветов; либо в шестнадцатеричной форме.

Возможно и задание цветов с помощью имен. Шестнадцать стандартных имен цветов (**aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, yellow, white**) поддерживают все браузеры, даже совсем старых версий.

Первая форма использует задание цветов в RGB-палитре (Red-Green-Blue). При этом код цвета указывается как шестиразрядное шестнадцатеричное число, задающее соотношение между красной (первые две цифры), зеленой (следующие две цифры) и голубой (последние две цифры) составляющей.

Интенсивность каждой составляющей в шестнадцатеричной форме изменяется от 00 до FF. Очевидны коды черного (000000) и белого цветов (FFFFFF). Миллионы оттенков могут быть заданы с помощью RGB-триады. При задании цвета в качестве значения атрибута HTML элемента перед шестнадцатеричным числом ставится символ # (табл. 3.4).

Таблица 3.4 – Представление цветов в различных формах кодировки

RGB	HEX	Цвет	Слово
0 255 255	#00FFFF		aqua
127 255 212	#7FFFD4		aquamarine
0 0 0	#000000		black
0 0 255	#0000FF		blue
165 42 42	#A52A2A		brown
255 127 80	#FF7F50		coral
0 255 255	#00FFFF		cyan
255 215 0	#FFD700		gold
128 128 128	#808080		gray
0 128 0	#008000		green
0 255 0	#00FF00		lime
128 128 0	#808000		olive
255 165 0	#FFA500		orange
255 192 203	#FFC0CB		pink
255 0 0	#FF0000		red
255 255 255	#FFFFFF		white
255 255 0	#FFFF00		yellow

Ниже представлен пример кода для трех текстовых абзацев по поводу цветового оформления текста и вид результата в окне браузера (рис. 3.10, 3.11).

```
<body >
      <p><font color="red" >Университет</font></p>
      <p><font color="#FF0000">факультет</font></p>
      <p><font color= "rgb(255,0,0)">Специальность</font></p>
</body>
```

Рисунок 3.10 – HTML-код веб-страницы

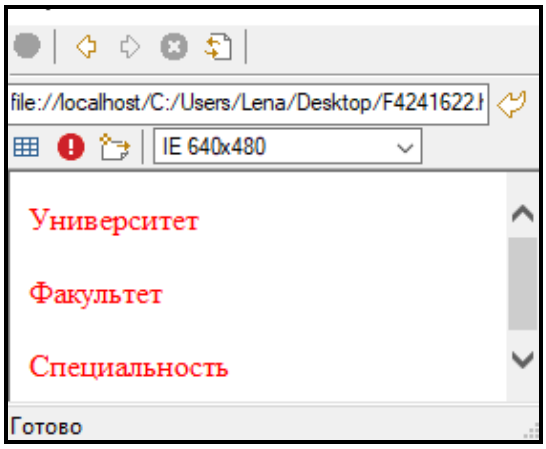

Рисунок 3.11 – Вид оформления трех текстовых абзацев в окне браузера

Имя файла в Web

Значением некоторых свойств может быть URL (Universal Resource Locator, универсальный указатель ресурсов) представляет собой путь к файлу. Для его указания можно использовать как **абсолютный** (перед адресом добавляется упоминание протокола http (http://), так и **относительный** адрес.

Если в начале адреса стоит **слэш (символ /),** это значит, что **отсчет идет от корня сайта**. Подобные ссылки со слэшем впереди работают только на веб-сервере, на локальном компьютере они действовать не будут (рис. 3.12).

Рисунок 3.12 – Вид записи относительного пути для файла

Двоеточие со слэшем (../) в начале адреса говорит о том, что **и файл и веб-страница находятся в разных папках**, которые размещены на одном уровне.

Рисунок 3.13 – Вид записи относительного пути для файлов, находящихся на одном уровне

Имя папки в начале пути, без всяких слэшей и двоеточий, сообщает, что и текущий файл и папка находятся на одном уровне (рис. 3.13).

Ход работы

Доработать созданные на предыдущем занятии HTML-страницы своего сайта.

Требования к создаваемым HTML-страницам своего сайта

1. Должна осуществляться навигация между страницами сайта (с главной страницы сайта возможен переход на любую другую и возврат на главную).
2. На HTML-страницах должен быть осуществлен переход внутри самой страницы для удобства пользователя.
3. Текст на страницах должен быть отформатирован.
4. Главная страница сайта должна называться «**index.html**». В названиях других страниц и файлов рисунков исключить кириллицу (названия только по-английски).
5. Задана кодировка, поддерживающая кириллицу. Должны присутствовать все парные теги, структурирующие веб-страницу (html, meta, title, body).
6. Желательно все графические файлы расположить в отдельной папке, находящейся внутри папки вашего сайта.
7. В контенте страниц должны присутствовать таблицы (содержащие строку с названием самой таблицы и названиями колонок) и нумерованные либо маркированные списки.

3.3 Использование CSS для оформления веб-документов

Цель: приобретение знаний и умений в области создания веб-страниц с использованием каскадных таблиц стилей CSS.

Назначение: выполнив работу, вы научитесь:

– изменять фон (подложку) для веб-страницы;

– разбивать контент веб-страницы на смысловые блоки;

– оформлять эти блоки относительно шрифта, заливки, границ с наименьшими временными затратами.

Полученные знания и умения помогут с помощью каскадных таблиц стилей быстро оформлять веб-страницы своего сайта.

Теоретические сведения

Понятие о каскадных таблицах стилей CSS

CSS (Cascading Style Sheets) – каскадные таблицы стилей. В HTML таблицы стилей управляют внешним видом документа.

Напирмер, необходимо чтобы каждое выделенное жирным слово подсвечивалось красным цветом. Можно добиться нужного эффекта, выделяя каждое слово тегом **<font color="red">**, однако это очень громоздкая конструкция, с помощью стилей этого можно достичь проще. Можно определить стиль (внешний вид) тега **<b>** следующим образом:

b {color: red}

тогда все, что находится внутри тега <**b**>, будет выделено красным цветом. Такой способ намного проще предыдущего, к тому же измениние цвета делается созданием одного стилевого правила для всех фрагментов текста, оформленных тегом <**b**>.

Цвет в CSS

Цветовые значения определяют цвет. Цвет можно указать с помощью **названия** (blue или red) или с помощью RGB кода. Названия цветов CSS совпадают с их названиями в HTML. CSS допускает задание цвета в виде стандартного RGB кода либо как шестнадцатиричный код (#87FED3).

В отличие от HTML, CSS позволяет задавать цвет трехзначным числом, в этом случае каждая цифра числа удваивается. Таким образом #7C5 тоже самое, что и #77CC55. Также можно задавать цвет в десятичных RGB обозначениях. Например: RGB (126, 6, 9) или RGB (30%, 40%, 70%).

Включение стилей в документ

Таблицы стилей на уровне документа

Стили на уровне документа определяются тегом **<style>.** Тег **<style>** может находиться только внутри тега **<head>.** Все, что находиться внутри тега **<style>**, рассматривается браузером как стилевые правила. Стили, определенные в теге **<style>**, действуют на все теги документа (рис. 3.14, 3.15).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
  <meta charset=windows-1251">
  <title></title>
  <style>
    p{color: red;}
  </style>
  </head>
  <body>
<p>В Крым невозможно не влюбиться. </p>
<p >Если вы хотя бы раз побывали на теплом полуострове, то вы обязательно
будете возвращаться в этот чудесный край еще и еще. Потому что Крым многолик.
</p>
  </body>
</html>
```

Рисунок 3.14 – HTML-код веб-страницы со стилевыми правилами

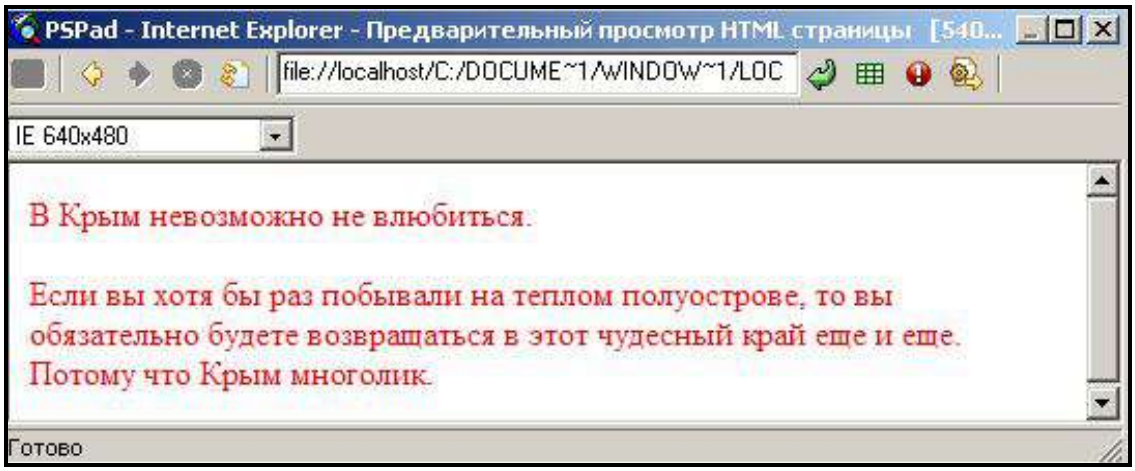

Рисунок 3.15 – Вид веб-страницы в окне браузера. Весь текст отображается красным цветом

В любом месте HTML-документа можно поместить комментарий. В отличие от комментариев в контейнере <**body**>, в контейнере </**style**> формат записи комментария несколько иной. Текст комментария помещается в тег **/* текст комментария */.** Он представляет собой текст,

который не отображается на экране и служит для пояснений разработчику веб-страницы.

Идентификаторы

Описание

Если во время работы веб-страницы требуется изменить стиль некоторых элементов «на лету», с идентификаторами это делается гораздо проще. Идентификатор задает имя элементу (блоку), который позволяет связать этот блок со стилевым оформлением. Имя одного идентификатора в контейнере **<body>** не может использоваться более одного раза.

При использовании в CSS следует учитывать, что идентификаторы обладают более высоким приоритетом по сравнению с классами. При создании правила для определенного идентификатора необходимо в контейнере **<style>** после знака # привести **имя идентификатора**, а затем в **фигурных скобках** записать само правило либо несколько правил через точку с запятой (рис. 3.16, 3.17).

```
  <style>
  /*Правило для идентификатора red*/
  #red {color: red;}
  </style>
  </head>
  <body>
<p id="red">В Крым невозможно не влюбиться. </p>
<p >Если вы хотя бы раз побывали на теплом полуострове, то вы обязательно
будете возвращаться в этот чудесный край еще и еще. Потому что Крым многолик.
</p>
  </body>
```

Рисунок 3.16 – HTML-код веб-страницы со стилевым правилом для идентификатора RED

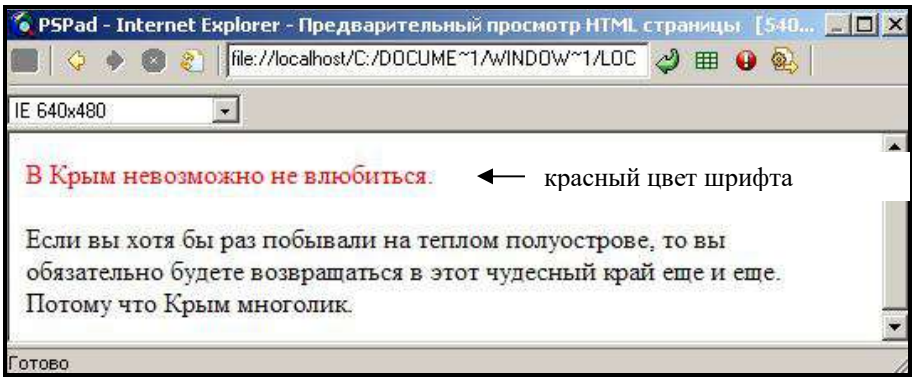

Рисунок 3.17 – Вид веб-страницы в окне браузера

Также при написании стилевых правил для идентификаторов необходимо придерживаться следующего:

– Имена идентификаторов могут содержать в себе латинские буквы (A–Z, a–z), цифры (0–9), символ дефиса (-) и подчеркивания (_). Использование кириллицы (русских букв) в классах недопустимо.

– В коде документа каждый идентификатор уникален и должен быть использован лишь один раз.

– Имя идентификатора чувствительно к регистру.

– Можно получить доступ к элементу по его идентификатору и изменить свойства элемента.

– Стиль для идентификатора (стилевое правило) имеет приоритет выше, чем у класса.

Синтаксис

```
<html>
 <head>
      <style>
       #селектор {свойство1: значение;
                  свойство2: значение;}
```

стилевые правила

```
      </style>
 </head>
 <body>
      <p id="селектор (имя)"> text </p>
 </body>
</html>
```

Классы

Описание

Задает стилевой класс, который позволяет связать определенный тег со стилевым оформлением. Имя класса в контейнере **<body>** может использоваться несколько раз.

Имена классов могут содержать в себе латинские буквы (A–Z, a–z), цифры (0–9), символ дефиса (-) и подчеркивания (_). Использование кирилицы в классах недопустимо.

При создании правила для определенного класса необходимо в контейнере **<style>** после знака « **.** » (точка) привести **имя класса**, а затем в **фигурных скобках** записать само правило либо несколько правил через точку с запятой (рис. 3.18, 3.19).

Синтаксис

```
<html>
 <head>
      <style>
       .селектор {свойство1: значение;
                  свойство2: значение;}
```

стилевые правила

```
      </style>
 </head>
 <body>
      <p class="селектор (имя)"> text </p>
      <p class="селектор (имя)"> text </p>
 </body>
</html>
```

```
  <style>
  /*Правило для класса green*/
   .green {color: green;}
  </style>
  </head>
  <body>
<p class="green">В Крым невозможно не влюбиться. </p>
<p class="green">Если вы хотя бы раз побывали на теплом полуострове, то вы
обязательно будете возвращаться в этот чудесный край еще и еще. Потому что Крым
многолик.</p>
  </body>
```

Рисунок 3.18 – HTML-код веб-страницы со стилевым правилом для класса GREEN

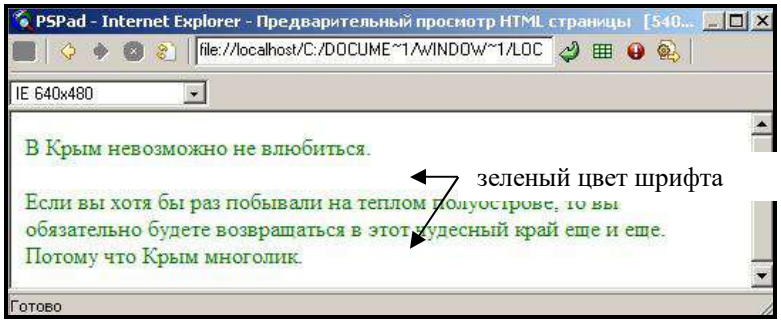

Рисунок 3.19 – Вид веб-страницы в окне браузера

Также при написании стилевых правил для классов необходимо придерживаться следующих правил:

1. Классы могут использоваться в коде неоднократно.
2. Имена классов чувствительны к регистру.
3. Классы можно комбинировать между собой, добавляя несколько классов к одному тегу через запятую.

Приоритеты и наследование в CSS

Каскадность CSS – это механизм, благодаря которому к элементу HTML-документа может применяться более чем одно правило CSS. Правила могут исходить из различных источников: из внешней (когда стилевые правила выделены в отдельный файл) и внутренней таблицы стилей, от механизма наследования, от родительских элементов, от классов и ID, от селектора тега, от атрибута **style**. Поскольку в этих случаях часто происходит конфликт стилей, была создана система приоритетов: в конечном итоге применяется тот стиль, который исходит от источника с более высоким приоритетом.

Если код написан таким образом, что два селектора имеют одинаковый вес, то приоритет отдается тому стилю, который находится ниже в коде. Если для одного элемента задан стиль и во внешней, и во внутренней таблицах стилях, то приоритет отдается стилю в той таблице, которая находится ниже в коде.

Например, если во внутренней таблице стилей задан красный цвет для тегов <**p**>, а во внешней – зеленый цвет для этих же тегов, то в итоге цвет тегов <**p**> будет красным.

Значимость приоритетов для стилевых правил по мере возрастания следующая:

1. Самым низким приоритетом обладают стили, которые прописаны в самом браузере. В файлах каждого интернет-браузера находится информация о стандартных CSS-стилях. В старых версиях браузеров есть возможность загрузить свой CSS-файл или изменить стандартные стили.
2. Следующим по значимости идет авторский стиль – все элементы, которые прописаны в коде страницы – это и подключенный файл и селекторы в атрибуте **style.**
3. Стили, унаследованные элементом от потомков.

4. Далее идут стили, прописанные во внешнем подключенном файле. В данном случае все селекторы, примененные к потомкам, не имеют силы.

5. Более высоким приоритетом обладают стили, которые прописываются в теге <**style**> в начале HTML-документа.

6. В большем приоритете будут стили, прописанные прямо в теге с помощью атрибута **style**.

Во время написания CSS-таблиц стилей могут быть созданны **противоречивые правила**, в которых одно и то же свойство применяется несколько раз. Чтобы избежать этого старайтесь придерживаться следующих правил:

– применяйте только **классы**: используйте **.name** вместо **#name**, даже если этот элемент появляется на вашей веб-странице только один раз;

– не применяйте **несколько классов** к одному элементу HTML: пишите не **<p class="big red important">**, а **<p class="title">**, это является семантически более описательным;

– не используйте **встроенные стили**, такие как ** **.

Тег SPAN

Использование тега SPAN

Этот тег специально зарезервирован в HTML для использования бестеговых стилей. По умолчанию тег **span** ничего не делает с находящимся в нем текстом. Предположим, что некоторые имена на веб-странице вы хотите выделить синим цветом, тогда можно воспользоваться тегом <**span**>.

Тег **** предназначен для определения встроенных элементов документа. В отличие от блочных элементов, таких как <**TABLE**>, <**P**> или <**DIV**>, с помощью тега **** можно выделить часть информации внутри других тегов и установить для нее свой стиль. Например, внутри параграфа (тег <**P**>) можно изменить цвет и размер первой буквы, если добавить начальный и конечный тег <**span**> и определить для него стиль текста. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить параметр **class** или **id** с именем селектора.

Синтаксис

<**span**>...</**span**>

```
  <style>
    span{color: blue;}
  </style>
  </head>
  <body>
<p>В <span>Крым</span>  невозможно не влюбиться. </p>
<p >Если вы хотя бы раз побывали на теплом полуострове, то вы обязательно
будете возвращаться в этот чудесный край еще и еще. Потому что <span>
Крым</span> многолик.
</p>
  </body>
```

Рисунок 3.20 – HTML-код веб-страницы со стилевым правилом для тега <**span**>

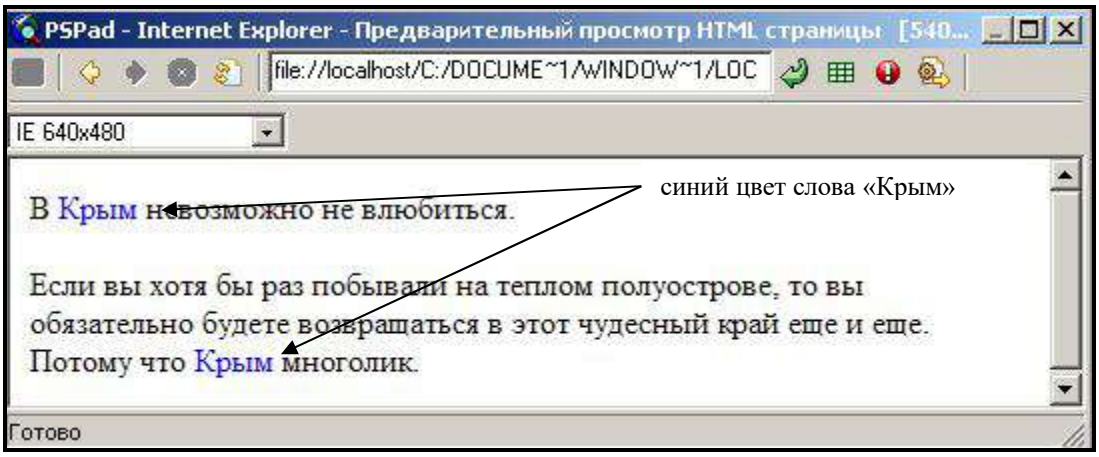

Рисунок 3.21 – Вид веб-страницы в окне браузера

Более наглядно использование тега <**span**> продемонстрировано на рисунках 3.20, 3.21.

Тег DIV

Описание

Элемент <**DIV**> является блочным элементом и предназначен для выделения фрагмента документа (блока) с целью изменения вида содержимого. Как правило, вид блока управляется с помощью стилей. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега HTML добавить параметр **class** или **id** с именем селектора.

Как и при использовании других блочных элементов, содержимое тега <**DIV**> всегда начинается с новой строки. После него также добавляется перенос строки.

```
  <style>
    p{color: blue;}
    div {background: red; }
  </style>
  </head>
  <body>
<div><p>В Крым невозможно не влюбиться. </p>
<p >Если вы хотя бы раз побывали на теплом полуострове, то вы обязательно
будете возвращаться в этот чудесный край еще и еще. Потому что Крым
многолик. </p>  </div>
  </body>
```

Рисунок 3.22 – HTML-код веб-страницы со стилевым правилом для тега <**DIV**>

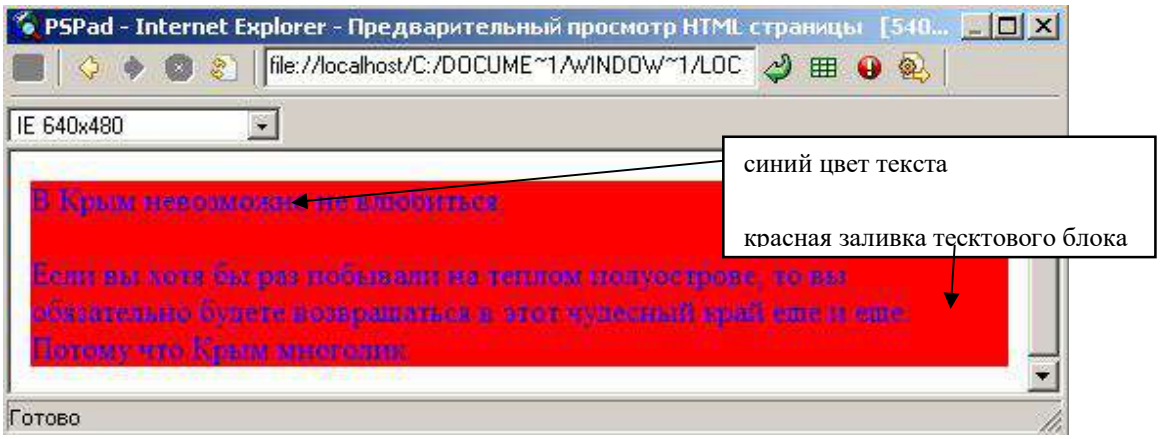

Рисунок 3.23 – Вид веб-страницы в окне браузера

Более наглядно использование тега <DIV> продемонстрировано на рисунках 3.22, 3.23.

Ключевые слова

Значением почти всех свойств в **стандарте CSS** могут быть ключевые слова, такие как **normal, medium, bold**. Ключевые слова не чувствительны к регистру – bold и BOLD – одно и то же.

Размеры элементов. Длины

Этот тип свойств явным образом устанавливает размер какого-либо элемента на веб-странице. Он поддерживает следующие размерности:

– **em** (ширина буквы «**m**» в текущем шрифте) например стилевое правило **p {text-indent: 3em}** задаст красную строку абзаца размером в три буквы «**m**»;

– **ex** (высота буквы «**x**» в текущем шрифте; применяется аналогично **em**);

– **px** (размер в пикселях);

– **in** (дюймы);

– **cm** (сантиметры);

– **mm** (миллиметры);

– **pt** (пункты, 1/72 дюйма);

– **pc** (пики, двенадцать пунктов);

– **процентные значения** (%).

Свойства height и width

Свойства **height** и **width** определяют высоту и ширину элемента соответственно. Значением этих свойств может быть длина или ключевое слово **auto,** которое подразумевает, что элемент обладал изначально какой-то высотой/шириной и отображает его в соответствии с этими значениями. Эти два свойства применяются обычно к таблицам и изображениям. Однако его можно применять ко всем элементам веб-страницы.

```
  <style>
    p{color: blue;}
    div {background: red;
        width: 400 px;
        height: 200 px; }
  </style>
  </head>
  <body>
<div><p>В Крым невозможно не влюбиться. </p>
<p >Если вы хотя бы раз побывали на теплом полуострове, то вы обязательно
будете возвращаться в этот чудесный край еще и еще. Потому что Крым
многолик. </p>  </div>
  </body>
```

Рисунок 3.24 – HTML-код веб-страницы со стилевыми правилами для тега <**DIV**>

Более наглядно использование свойств **height** и **width** для текстового блока продемонстрировано на рисунках 3.24, 3.25.

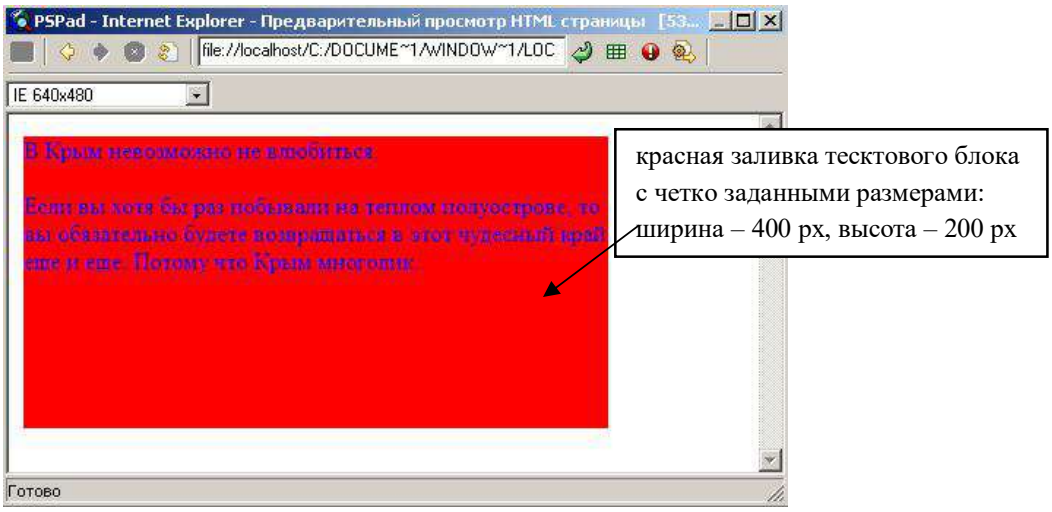

Рисунок 3.25 – Вид веб-страницы в окне браузера

Свойства обрамления

Модель форматирования

Каждый элемент документа занимает прямоугольную область, которая помещена последовательно еще в три области: **подложка (padding**), **рамка (border), поля (margin).** Рисунок 3.26 иллюстрирует эту модель.

Свойство полей, подложки и рамки задается следующим образом:

{Контейнер-Сторона-Свойство: значение;}

где **контейнер** это одно из слов **margin, padding** или **border**;

сторона – это **left, right**, **top** или **bottom**;

свойство – это название определяемого свойства.

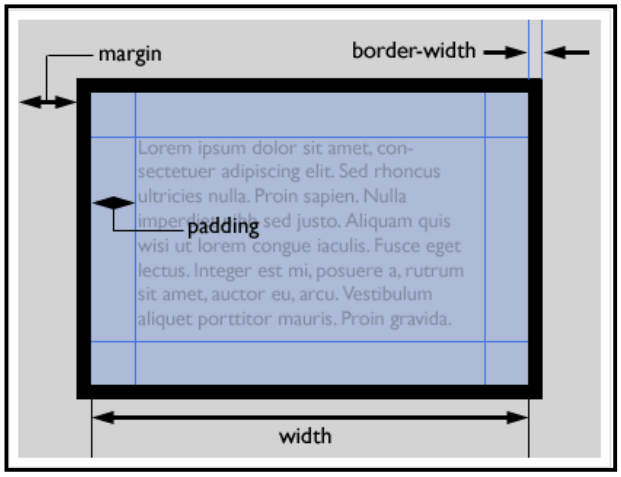

Рисунок 3.26 – Вид отступов для элемента веб-страницы

Название сторон можно пропустить, если вы хотите применить стилевое правило ко всем сторонам прямоугольной области.

Свойство border-color

Это свойство задает цвет рамки. Оно может принимать от одного до четырех значений. Если установлено только одно значение, то все четыре стороны рамки будут одного цвета. Два значения приводят к тому, что верх и низ рамки окрашиваются в первый цвет, а левая и правая стороны рамки – в другой. В случае трех значений первое будет цветом верха рамки, второе цветом низа, а третье задаст цвет левой и правой стороны. Четыре значения определяют цвет каждой из сторон рамки в следующем порядке: верха, правой стороны, левой стороны, низа. Также можно явно задавать цвет определенной стороны рамки.

Свойство border-width

Свойство **border-width** определяет толщину рамки. Как и свойство **border-color** может принимать от одного до четырех значений в том же порядке. Для указания толщины могут использоваться стандартные единицы длины или одно из ключевых слов: **thin** (тонкая), **medium** (средняя). Можно также определить толщину конкретной стороны.

Свойство border-style

Это свойство используется для украшения рамки и может принимать одно из следующих значений: **none** (отсутствует), **dotted** (точечная), **dashed** (штриховая), **double** (двойная), **solid** (сплошная), **groove** (паз), **outset** (приподнятая), **inset** (утопленная), **ridge** (ребро). Значение **none** применяется по умолчанию. Популярные браузеры некорректно поддерживают это свойство, так что лучше его не использовать.

Свойство margin

Это свойство установливает ширину полей. Есть возможность задать ширину всех полей сразу, присвоив свойству **margin** значение, или для каждого поля по отдельности, задав значения свойствам **margin-left, margin-right, margin-top, margin-bottom**. Это свойство может принимать значение **auto**, указывающее браузеру, что надо применить стандартное значение ширины полей.

Свойство padding

Это свойство установливает ширину подложки. Есть возможность задать ширину всех подложек (верхней, нижней, правой и левой) сразу, присвоив свойству **padding** значение, или для каждого поля по отдельности, задав значения **свойствам padding-left, padding-right, padding-top, padding-bottom**. Это свойство может принимать значение

auto, указывающее браузеру, что надо применить стандартное значение ширины полей.

```
<style>
 #Heading{ text-align: center;     /*Выравнивание*/
           padding: 5 px;          /*Поля*/
           border: solid 5px blue; /*Границы*/
           }
</style>
 </head>
 <body>
       <!-- Блок заголовка-->
        <div id="Heading"> Мои увлечения <br>
На этой страничке я хочу рассказать об одном из своих увлечений  </div>
   </body>
```

Рисунок 3.27 – HTML-код веб-страницы со стилевыми правилами для текстового блока

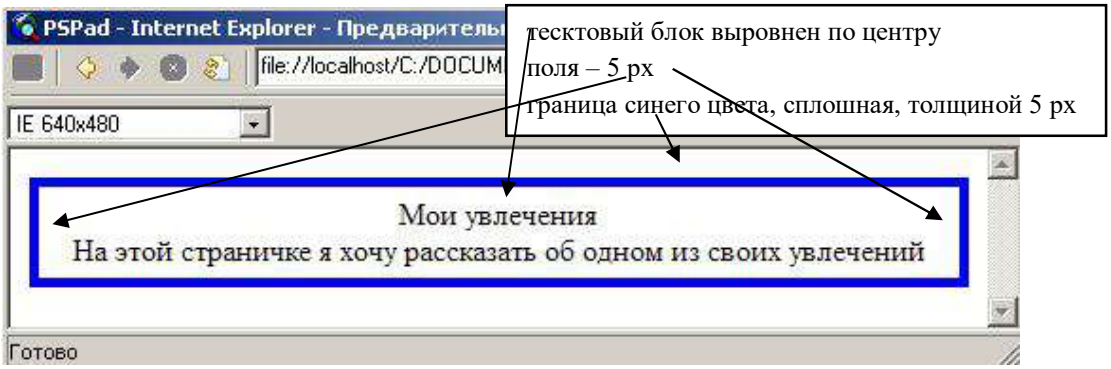

Рисунок 3.28 – Вид веб-страницы в окне браузера

Более наглядно использование свойств обрамления для текстового блока продемонстрировано на рисунках 3.27, 3.28.

Оформление текста. Свойства шрифта

Свойство font-family

Значение этого свойства – список шрифтов, которыми браузер должен отобразить данный объект или одно из семейств шрифтов, определенных стандартом CSS, такие как: **serif** (с засечками), **sans-serif** (без засечек), **fantasy** (с украшениями) и **monospace** (моноширинные). Если указан список шрифтов, то браузер выберет первый в списке, который есть на компьютере пользователя, если указано семейство, то браузер сам решает, какой шрифт из этого семейства выбрать.

Свойство font-size

Это свойство определяет размер шрифта. Может принимать одно из стандартных значений размера или одно из ключевых слов для абсолютного размера. Вот эти ключевые слова по порядку увеличения шрифта: **xx-small, x-small, small, medium, large, x-large, xx-large.** Эти размеры обычно соответствуют семи размерам шрифта, используемым для атрибута **size** тега **<font>.**

Свойство font-stretch

Значение этого свойства указывает, насколько сильно надо растягивать буквы шрифта. Может принимать следующие значения, начиная с самого плотного: **ultra-condenced, extra-condenced, condenced, semi-condenced, semi-expanded, expanded, extra-expanded, ultra-expanded**. Популярные браузеры не поддерживают это свойство.

Свойство font-style

Может принимать три значения: **normal** (нормальное начертание), **italic** (курсивное начертание), **oblique** (наклонное начертание). Курсивное и наклонное начертании обычно не различаются.

Свойство font-variant

Может принимать два значения: **normal, small-caps**. Значение **small-caps** указывает браузеру, что надо заменить строчные буквы на маленькие заглавные.

Свойство font-weight

Используется **для управления жирностью шрифта**. Может принимать значение от **100** (самый тонкий) до **900** (самый жирный). Значение 400 можно заменить ключевым словом **bold**, а значение 700 – ключевым словом **bolder**. Также значением этого свойства могут быть ключевые слова **lighter** и **bolder**, которые показывают, что надо использовать версию шрифта, которая соответственно менее или более жирная, чем принятая для родительского элемента.

Более наглядно использование свойств шрифтов для оформления текста продемонстрировано на рисунках 3.29, 3.30.

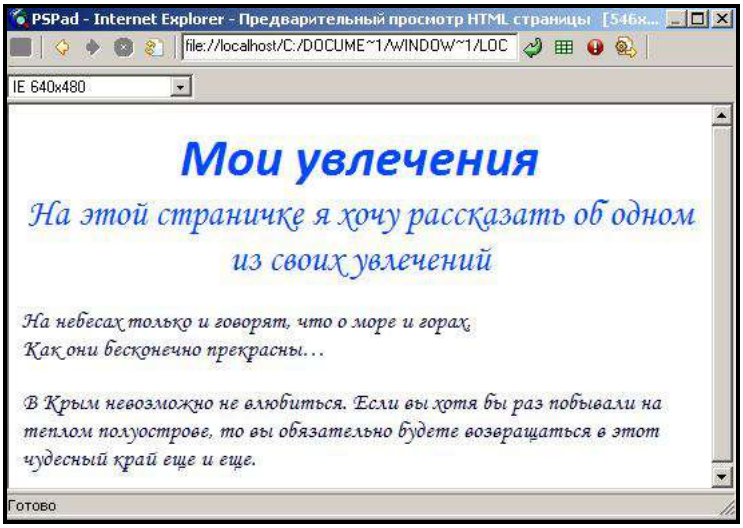

Рисунок 3.29 – Вид веб-страницы в окне браузера

```
 <style>
  /*Фон страницы*/
   /*Форматирование шрифта в абзацах*/
p{font: 18px Monotype Corsiva;
      color: rgb(0,0,64);}
    /*Блок заголовка*/
#Heading{background: white;       /*Фон*/
      text-align: center;     /*Выравнивание*/
      font: 28px Monotype Corsiva;     /*Шрифт*/
      color: rgb(0,72,255);}
   /*Форматирование текста "Мои увлечения"*/
span{ font: bold italic  40px Calibri;}
       /*Форматирование класса "Question"*/

  </style>
  </head>
  <body>
        <!-- Блок заголовка-->
         <div id="Heading"> <span>Мои увлечения</span> <br>
На этой страничке я хочу рассказать об одном из своих увлечений  </div>
       <!-- Эпиграф-->
<p >На небесах только и говорят, что о море и горах, <br>Как они бесконечно
прекрасны…     </p>

<p> В Крым невозможно не влюбиться. Если вы хотя бы раз побывали на теплом
полуострове, то вы обязательно будете возвращаться в этот чудесный край еще и
еще. </p>
  </body>
```

Рисунок 3.30 – HTML-код веб-страницы со стилевыми правилами для оформления текста

Ход работы

Доработать созданные на предыдущих занятиях веб-страницы своего сайта. Оформить контент с помощью стилевых правил CSS по своему усмотрению в соответствии с разработанным дизайнерским решением.

В контейнер **<head> … </head>** вложите контейнер **<style> … </style>** , в который поместите правила, оговаривающие форматирование и расположение отдельных элементов вашей веб-страницы.

Требования к оформлению веб-страниц своего сайта

Добавьте еще одну страницу на свой сайт и организуйте переход на эту страницу. На веб-страницах должны бать созданы «закладки», позволяющие осуществлять навигацию внутри страниц.

С помощью стилевых правил CSS оформите единообразно все страницы своего сайта:

– веб-страницы должны иметь фон (подложку);

– важные смысловые блоки контента страницы должны быть выделены с помощью границ, отступов, заливки, нестандартного шрифта;

– шрифт веб-страницы должен отличаться от стандартного шрифта браузера (но различных гарнитур шрифта на странице должно быть не больше трех).

3.4 Использование внешней таблицы стилей для оформления Веб-документов

Цель: приобретение знаний и умений в области оформления сайта в едином стиле с использованием правил CSS, вынесенных в отдельный файл.

Назначение: выполнив работу, вы научитесь:

– оформлять страницы сайта на основе правил, описанных в одном файле;

– оформлять списки на веб-странице;

– позиционировать (располагать) рисунки на веб-странице и оговаривать их «послойность» (кажущееся приближение к пользователю).

Полученные знания и умения помогут упростить код веб-страниц своего сайта и уменьшить его размер.

Теоретические сведения

Внутренние и внешние таблицы стилей

Каскадные таблицы стилей позволяют разделить описания логической структуры веб-страницы от описания внешнего вида этой веб-страницы (которое теперь производится с помощью CSS). CSS позволяет представлять один и тот же документ в различных методах вывода (стилях), таких как экранное представление, печатное представление.

После того как была создана таблица стилей, можно выбрать, каким образом прикрепить ее к веб-странице. Существует два варианта таблиц – внутренние, которые добавляются прямо на страницу (это было сделано в прошлом практическом занятии), и внешние, которые находятся в отдельном файле с расширением **.css** и подключаются к веб-странице с помощью ссылки.

Внутренние таблицы стилей записываются в HTML-документе между тегами **<style>…</style>**. Недостаток – нужно добавлять стилевые правила на каждую веб-страницу. Если сайт состоит из большого количество страниц, которым необходим одинаковый дизайн, то добавление, а также редактирование стилей становится трудоемким процессом и занимает очень много времени. Поэтому внутренние таблицы стилей считаются неудобными.

Внешние таблицы стилей гораздо более распространены. Они позволяют оформлять все веб-страницы сайта в едином стиле, единожды написав одно правило, а не повторяя его на каждой странице. Для этого нужно подключить файл **.css** ко всем необходимым веб-страницам.

Файл **.css** – это обычный текстовый файл, в котором написаны различные стили для элементов веб-страниц и комментарии к ним. И больше ничего лишнего.

Таблицу стилей, которая находится в отдельном файле (.css), можно подключить к веб-документу используя тег **<link>** с атрибутом **rel** (определяет отношение между страницей и подключаемым файлом) и значением **stylesheet**, которое означает, что в подключаемом файле содержится таблица стилей. Еще один атрибут **href** – это путь (URL) к вашему файлу **.css**. Тег **<link>** в свою очередь должен находиться между тегами **<head>** и **</head>**. Код для подключения файла «style.css» со стилевыми правилами представлен ниже.

```
<html>
 <head>
 /*Подключение внешнего стиля*/
 <link rel="stylesheet" href="style.css" />
 /*Внутренние стили*/
 <style>
 ……
 </style>
 </ head >
 <body>
 …….
 </ body >
</ html >
```

В результате редактируя только один файл можно изменять стиль для всего сайта сразу, вне зависимости от того, сколько в нем страниц. Это очень удобно, особенно для крупных ресурсов.

Комментарии

Комментарии в коде нужны, чтобы легче было ориентироваться разработчику, оставляя свои заметки о том, какую задачу решает тот или иной фрагмент. На разных языках программирования они реализуются по разному. Текст, прописанный в комментарии, браузером на веб-странице не отображается.

Формат записи комментария следующий:

/* комментарий в css */

Приоритеты стилей

Существуют внутренние стили и внешние стили. Если в коде для веб-страницы прописаны правила оформления, например цвета текста в контейнере **<style> …. </style>** и они отличаются от цвета, описанного в файле внешней таблицы стилей – **style.css**, то сначала будут выполняться правила, описанные в контейнере **<style> …. </style>** (во внутренних таблицах).

Если к одному элементу применены одинаковые стили с одним приоритетом, то сработает тот стиль, который назначен последним.

Группировка селекторов

Селекторы в стилевых правилах можно указывать через запятую, тогда указанные свойства будут применены ко всем указанным тегам, классам, идентификаторам.

h1, h2, h3 {свойства}

В результате свойства будут применены для тегов h1, h2 и h3.

Свойства позиционирования

Система координат

Как и в языках программирования, система координат в модели CSS отличается от декартовой. Начало координат находится в левом верхнем углу окна браузера.

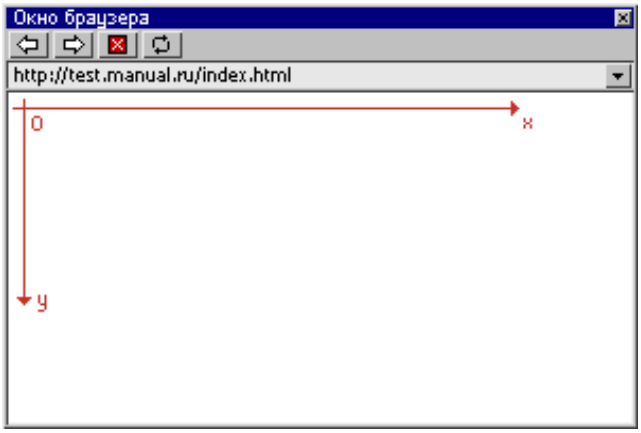

Рисунок 3.31 – Вид системы координат в окне браузера

Координата **X** возрастает слева на право, а координата **Y** – сверху вниз (рис. 3.31).

Свойство position

Установить способ позиционирования элемента относительно окна браузера или других объектов на веб-странице поможет свойство **POSITION.**

Это свойство указывает браузеру относительно какого элемента располагать содержимое тега в окне. Может принимать три значения: **absolute** (координаты задаются относительно верхнего левого угла самого элемента), **static** (координаты задаются относительно верхнего левого угла страницы), **relative** (координаты задаются относительно предыдущего элемента).

Синтаксис

position: absolute | fixed | relative | static

(абсолютное | фиксированное | относительное | статичное)

Положение блока в документе зависит от заданных значений стилевых свойств left, top, right и bottom, они устанавливают соответственно позицию слоя слева, сверху, справа и снизу.

Свойство left

Это свойство определяет **X-координату** верхнего правого угла элемента. Значение может задаваться либо длиной, либо в процентах от ширины окна браузера. Может также принимать значение **auto**, указывающие браузеру, что надо выбрать X-координату элемента автоматически.

Свойство top

Свойство **top** задает **Y-координату** верхнего правого угла элемента. Значение может задаваться либо длиной, либо в процентах от высоты окна броузера. Как и **left** может принимать значение **auto**.

Свойство z-index

Свойство **z-index** задает порядок перекрытия элементов друг другом. Если элементы перекрываются, то **сверху будет отображаться элемент, имеющий большее значение z-index**. Этот параметр может задаваться либо числом, либо ключевым словом **auto**, которое меньше любого значения **z-index**. Если этому свойству для двух элементов присвоено значение **auto** и они перекрываются, то отображаться сверху будет элемент, определенный позже в исходном коде. Значение **auto** принято по умолчанию.

```
  <style>
  /*Фон страницы*/
body { background: url(heading.png);
        repeat-x;
        background-attachment: fixed; }
   /*Расположение и послойность картинок*/
#a{  position:absolute;       /*Для картинки A*/
     z-index=1}
#b{   position:relative;
      top:20px;left:180px;    /*Смещение относит "A"*/
      z-index=2}
#c{   position:relative;
      top:-50px;left:150px;  /*Смещение относит "b"*/
      z-index=3}
  </style>
```

```
  <body>
     <p class="Question">Вам еще напонятно зачем идти в поход?</p>
     <!-- Вставка рисунков-->
    <div> <img id="a" src="Crimea1.JPG">
     <img id="b" src="Crimea2.JPG">
    <img id="c" src="Crimea3.JPG">  </div>

<p>Крым - неповторимый уголок на земле, с уникальной природой, историей и
культурой. Крым прекрасен. И особенно прекрасен Крым весной.  </p>
  </body>
```

Рисунок 3.32 – HTML-код веб-страницы со стилевыми правилами для оформления рисунков

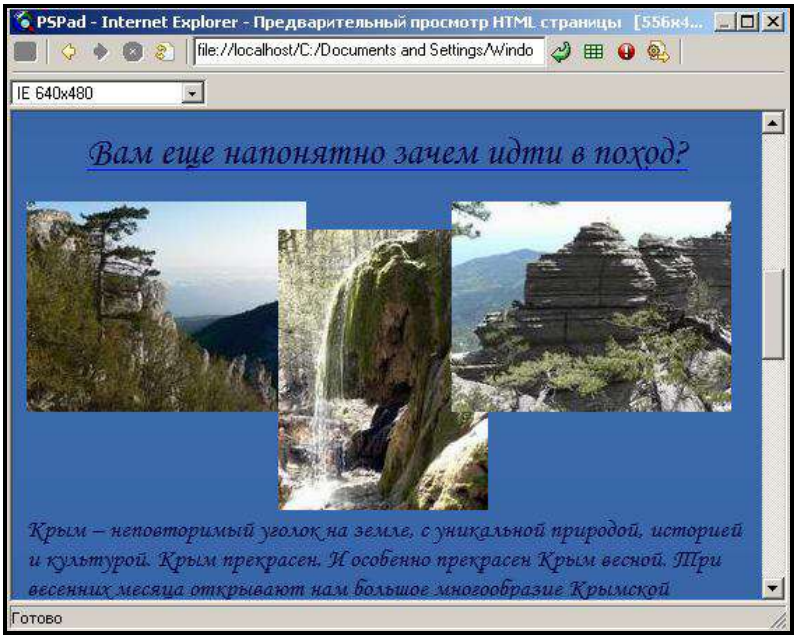

Рисунок 3.33 – Вид веб-страницы в окне браузера

Стилевое свойство background-image

Свойство background-image определяет картинку фонового изображения элемента. Принимает значение URL или ключевое слово **none**. Ключевое слово **none** означает, что у элемента нет фонового изображения.

body {background-image: url(bg3.gif)}

Это стилевое правило задает цвет (рис. 3.34) фона документа.

```
  <style>
     div {background-image: url(Crimea1.JPG);}
  </style>
  </head>
  <body>
<div><p>В Крым невозможно не влюбиться. </p>
<p >Если вы хотя бы раз побывали на теплом полуострове, то вы обязательно
будете возвращаться в этот чудесный край еще и еще. Потому что Крым
многолик. </p>  </div>
  </body>
```

Рисунок 3.34 – HTML-код веб-страницы со стилевым правилом для задания фона страницы

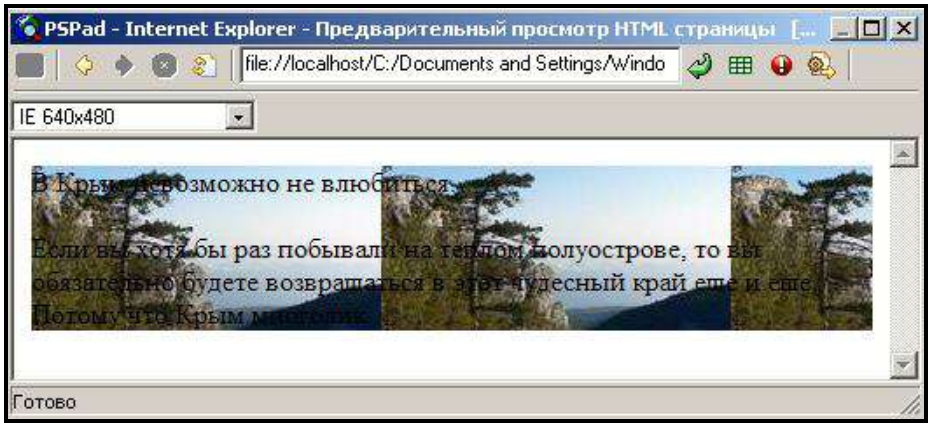

Рисунок 3.35 – Вид веб-страницы в окне браузера

Более наглядно это свойство продемонстрировано на рисунках 3.34, 3.35.

Стилевое свойство background-repeat

Это свойство управляет способом размножения фонового изображения. Может принимать следующие значения: **repeat-x** (размножает изображение только по вертикали), **repeat-y** (размножает изображение только по горизонтали), **no-repeat** (не размножает фоновое

изображение). Если это свойство не было указано, то изображение размножается стандартным образом.

Стилевое свойство background-position

Обычно броузеры выводят фоновое изображение, размножая его вправо и вниз, начиная с левого верхнего угла браузера. Однако с помощью свойства **background-position** можно сместить точку начала размножения фонового изображения. Этому свойству можно приписать одно или два значения, разделенные пробелом. Если вы используете одиночное значение, оно применяется к положению по горизонтали и по вертикали, если двойное, то первое значение устанавливает горизонтальное смещение, а второе вертикальное. Смещение изображения можно задавать как стандартными единицами длины и процентами, так и ключевыми словами **left, right, center** в случае горизонтального смещения и **top, center, bottom** в случае вертикального.

```
  <style>
     div {background-color: rgb(102,255,255);
          background-image: url(Crimea1.JPG);
          background-repeat: no-repeat;
          background-position: center center;}
  </style>
  </head>
  <body>
<div><p>В Крым невозможно не влюбиться. </p>
<p >Если вы хотя бы раз побывали на теплом полуострове, то вы обязательно
будете возвращаться в этот чудесный край еще и еще. Потому что Крым
многолик. </p>  </div>
  </body>
```

Рисунок 3.36 – HTML-код веб-страницы со стилевыми правилами для задания фона блока

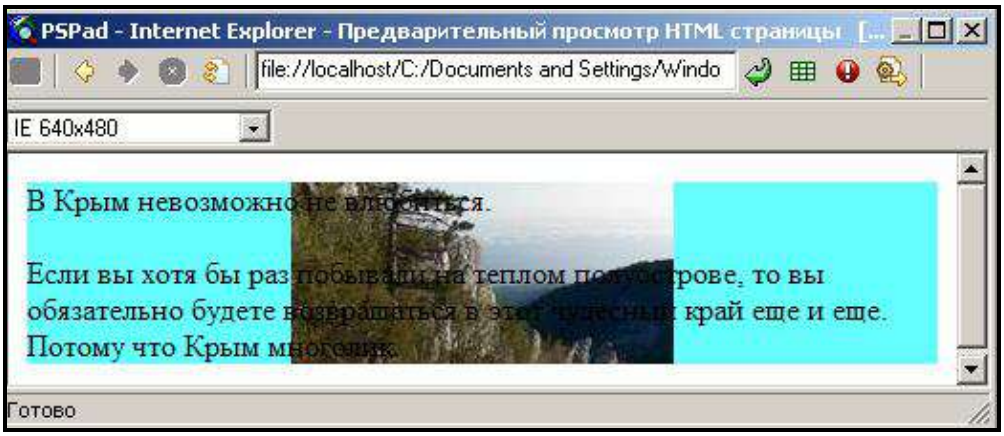

Рисунок 3.37 – Вид веб-страницы в окне браузера

Более наглядно эти свойства для оформления фона блока продемонстрированы на рисунках 3.36, 3.37.

Стилевое свойство background-attachment

Это свойство управляет способом прикрепления изображения к окну броузера. При принятом по умолчанию, значении **scroll** браузер перемещает фоновое изображение вместе с текстом. Если же этот параметр принимает значение **fixed**, то изображение не двигается относительно окна браузера во время прокрутки документа.

Стилевое свойство float

Свойство **float** описывает область отображения тега в качестве плавающего элемента. Это свойство аналогично атрибуту **align** тегов **<img>** и **<table>,** но оно может применяться к любым HTML-элементам. Это стилевое свойство позволяет верстать веб-страницы, состоящие из нескольких колонок. При этом колонки будут не жестко привязаны к определенным координатам страницы, а будут менять свое местоположение в случае просмотра веб-страницы на устройствах с разными диагоналями и пропорциями экрана. Что является особо актуальным в настоящее время, когда пользователь просматривает страницу сайта на устройствах с разными диагоналями и пропорциями.

Свойство **float** принимает одно из трех значений: **left, right, none**. Значение **none** принято по умолчанию, оно выключает свойство **float**. Значения **left** и **right** указывают браузеру поместить содержимое элемента слева и справа от потока соответственно и позволить другому содержимому обтекать его.

Стилевое свойство clear

Это свойство говорит браузеру, помещать ли содержимое тега рядом с «плавающим» элементом или на первой строке перед ним. Значением свойства **clear** может быть **none**, **left**, **right** или **both**. Значение **none** принято по умолчанию и подразумевает, что браузер размещает содержимое тега рядом с плавающими элементами по обе стороны, если там есть место. Значение **left** запрещает располагать содержимое слева от плавающего элемента, **right** – справа. **Both** не допускает размещение содержимого тега рядом с плавающим элементом вообще.

```
   <style>
.Question {text-align: center;  /*Выравниевание по центру*/
          font-size: 30 px;}     /*Размер шрифта*/
#left {float: left;       /*Обтекание*/
      width: 250 px;     /*Ширина колонки*/
      border-right: 2px solid blue;     /*Правая граница колонки*/
      padding: 5 px;      /*Поля*/
      margin: 10px;}       /*Отступы*/

#right {float: left;       /*Обтекание*/
        padding: 5 px;      /*Поля*/
        margin: 10px;}      /*Отступы*/
.clear {clear: both;}       /*Запрет обтекания*/

  </style>
  </head>
  <body>
 <div id="left"><p> В Крым невозможно не влюбиться. Если вы хотя бы раз побывали
на теплом полуострове, то вы обязательно будете возвращаться в этот чудесный
край еще и еще. Потому что Крым многолик. Это шумные набережные, тихие пляжи и
лазурные морские закаты. Туризм в Крыму для меня - это новые открытия, новые
встречи, новые впечатления, дающие заряд энергии на долгое время, и в то же
время оставляющие легкую грусть, заставляющие скучать по друзьям, зовущие
обратно. Здесь в Крыму я вроде бы как и не отдыхаю, я здесь, по-настоящему живу,
здесь душа по-настоящему работает. 20-30 километров в день, тысячи фотоснимков,
общение с такими же как и я сам... Идет  энергетическая подзарядка всех
подсистем огранизма а мозг это все интенсивно обрабатывает,  чтобы сделать
информационные запасы на весь предстоящий год..."  </p>   </div>

<div id="right"><p>Крым - неповторимый уголок на земле, с уникальной природой,
историей и культурой. Крым прекрасен. И особенно прекрасен Крым весной. Три
весенних месяца открывают нам большое многообразие Крымской природы: холодный и
непредсказуемый март раскроет перед нами приход весны, апрель поразит цветением
весеннего леса, а теплый май даст возможность насладиться буйством зелени и
красок первозданной природы. Весной, Крым представляет собой сказочную картину:
холмы ярко зелёные, в степи от ветра колышутся волны ковыля, миллионы птиц вьют
гнёзда, снуют в своих заботах перед глазами путешественника. С наступлением
вечера всюду поют соловьи, а ведь их песню можно услышать только весной. Весной
реки Крыма, многие из которых пересыхают летом, наполняются талой водой. Только
весной любители пешего туризма могут увидеть полноводные крымские каньоны,
горные озера, водопады, перекаты, ванны во всей своей красе.  </p>  </div>
 <div class="clear"></div>
<p class="Question">Так зачем же идти в поход?  <p>
 </body>
```

Рисунок 3.38 – HTML-код веб-страницы со стилевыми правилами для многоколоночной верстки

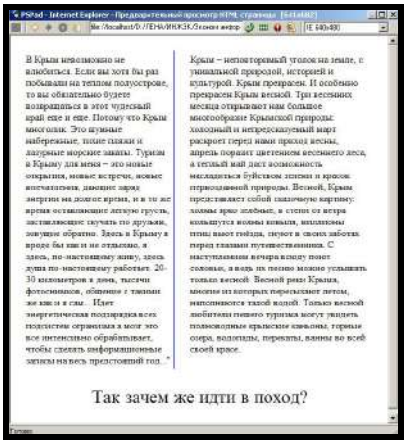

Рисунок 3.39 – Вид веб-страницы в окне браузера

Более наглядно эти свойства для оформления многоколоночной веб-страницы продемонстрированы на рисунках 3.38, 3.39.

Стилевые свойства списков

Свойство list-style-image

Это свойство определяет изображение, используемое вместо маркера. Укажите в значении свойства URL изображения, и вместо обычного маркера браузер выведет нестандартный рисунок.

Свойство list-style-position

Свойство **list-style-position** управляет способом вывода маркера. Может принимать два значения: **inside** (маркер расположен снаружи) и **outside** (маркер включен в текстовый блок).

```
   <style>
    /*Форматирование маркированного списка*/
 li {list-style-image: url(smile.jpg);
     list-style-position: outside; }
  </style>
  </head>
  <body>
<p class="Question">Так зачем же идти в поход?   <p>
<ul>
    <li>Это здорово, весело и интересно :)  </li>
    <li>Это ночи в палатке, звездное небо, песни у костра и романтика :)</li>
    <li>Это тяжелый рюкзак и высокие горы... </li>
    <li>Это невероятные пейзажи и способ испытать себя на прочность. </li>
    <li>Это всегда веселая компания и интересные люди. </li>
    <li>Это масса впечатлений, новых друзей и приключений!</li>
</ul>
  </body>
```

Рисунок 3.40 – HTML-код веб-страницы со стилевыми правилами для оформления маркированного списка

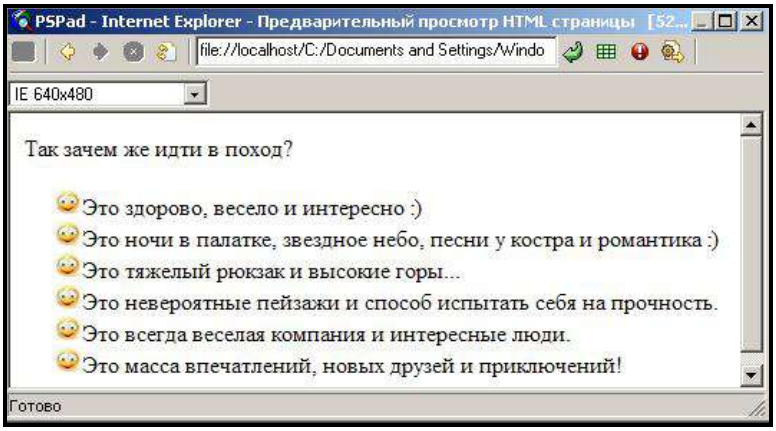

Рисунок 3.41 – Вид веб-страницы в окне браузера

Более наглядно эти свойства для оформления многоколоночной веб-страницы продемонстрированы на рисунках 3.40, 3.41.

Ход работы

Доработать созданные на предыдущих занятии HTML-страницы своего сайта.

Порядок выполнения работы

Уже созданные и связанные гиперссылками веб-страницы своего сайта оформите с помощью внешней таблицы стилей – пропишите в ней те правила, которые одинаковы для всех страниц. Оформите свой сайт по своему усмотрению в соответствии с разработанным дизайнерским решением.

Требования к оформлению WEB-страниц:

1. Все веб-страницы сайта должны быть оформлены в едином стиле – фон (подложка), заголовки на страницах. Эти правила должны быть описаны с помощью внешних стилей.
2. Списки на веб-страницах должны иметь нестандартное форматирование, касаемо маркеров.
3. На одной из веб-страниц должна быть размещена фотогалерея с графическими объектами, наплывающими один на другой. То есть необходимо спозиционировать (расположить) рисунки и оговоривть их «послойность» (кажущееся приближение к пользователю).
4. Текст на веб-страницах должен быть сверстан в несколько колонок, способных «перестраиваться» в случае изменения ширины окна браузера (для этого создайте хотя бы одну веб-страницу с достаточно большим количеством текста).
5. Ваш сайт-визитка должен иметь презентабельнй вид и выглядеть как законченный продукт.

3.5 Технология создания динамических веб-страниц средствами языка JavaScript

Цель: приобрести знания и умения в области создания программ на языке **JavaScript**.

Назначение: выполнив работу, вы научитесь:

– «оживлять» свои WEB-страницы;

– изменять изображения во времени;

– создавать календарь, отображающий текущую дату, а также количество дней до определенного события.

Полученные знания и умения помогут «оживить» вашу статическую WEB-страницу по своему усмотрению.

Теоретические сведения

Программы можно располагать в любой месте **HTML** страницы, но нельзя перемешивать **HTML**-теги и операторы **JavaScript**. Месторасположения программы зависит от её предназначения. Если сценарии используются для динамического создания кода **HTML**, то предназначенные для генерации программных или системных параметров контейнеры **<script>** будут расположенны по всему тексту документа **HTML**. Однако если требуется определить функции для решения различных задач, то самым лучшим местом для них будет отдельный контейнер **<script>**, расположенный в начале документа.

Синтаксис тега <script>

<**script language="JavaScript**">
//операторы JavaScript ...
</**script**>

Код должен обязательно выглядеть так, как показано выше, с точным соблюдением регистра для всех программ на языке **JavaScript**. Несмотря на то, что язык указывать необязательно, лучше всё-таки указывать его. Весь текст за пределами тегов **<script>** и **</script>** должен быть обязательно в формате **HTML.**

На HTML страницу можно добавить несколько программ на языке JavaScript. Они будут выполняться в порядке их расположения в коде (сверху вниз). Браузер всегда анализирует программы JavaScript после загрузки всей HTML-страницы. Функции загружаются в память и выполняются браузером, когда генерируется соответствующее событие или при явном вызове функции JavaScript (либо других функций, определенным пользователем. Под функцией здесь понимается группа операторов языка JavaScript).

HTML дает возможность создавать статические веб-документы. Тем не менее они получат более захватывающий вид, если к ним прибавить динамизацию: анимацию и изменяемое содержимое. Динамические страницы сайтов более привлекательны. Такая технология может быть осуществлена с помощью специальных анимационных тегов и небольших программ-сценариев, которые выполняют, как правило, простые задачи обработки данных или украшают страницы документа.

Динамические эффекты могут применяться к текстам, изображениям, видео, музыке. Наличие в документе бегущей строки или эффекта выскакивания букв в строке текста динамизирует документ и вызывает дополнительный интерес к документу. Интерес возрастает, если в документ включены гиперссылки на сайты с музыкой (файлы *.mp3 или *.wma) или с видео-клипами.

Большинство анимационных операций имеют логико-математическую основу, а потому нуждаются в программировании. Оно не должно вызвать трудностей, если известны основы алгоритмизации.

Сценарии пишутся на любом языке программирования – JavaScript, VBScript, PHP.

JavaScript – это клиентский скриптовый язык сценариев с элементами объектно-ориентированной модели. Сценарии JavaScript выполняются непосредственно на компьютере пользователя клиент-программой (браузером). Это – язык-интерпретатор. За синтаксическую основу в нем принята упрощенная версия языка Java, которая во многом совпадает с языками C и C++. JavaScript имеет ряд операторов и возможность написания пользовательских функций, которые могут быть связаны с методами, свойствами и событиями.

Операторы JavaScript

Программы JavaScript состоят из операторов (инструкций), каждый из которых записывается в одной строке. В конце оператора проставляется (но не обязательно) символ точки с запятой (;). Как и в любом языке программирования высокого уровня, основными операторами JavaScript являются:

- операторы ввода данных;
- операторы комментариев;
- операторы присваивания;

– операторы разветвления;
– операторы циклов;
– операторы вывода результатов.

Типы данных

В языке программирования JavaScript используются следующие типы данных:

– текстовый;
– логический;
– целые числа (в десятичной и шестнадцатеричной системах счисления);
– действительные числа (с фиксированной и плавающей точками);
– бинарные числа;
– массивы.

Данные представляются в виде переменных и констант. Переменные имеют имена. При написании имен нужно учитывать регистр. Переменные создаются с помощью оператора **var** или путем непосредственного присвоения значений операцией (=).

Синтаксис оператора var

var varyablename[=value | expression]

Оператор var создает новую переменную с именем **varyablename**. В JavaScript переменные могут быть локальными или глобальными. Локальные переменные обычно применяются при написании функций и их нужно объявлять с помощью ключевого слова **var**. Глобальные переменные не объявляются – их тип определяется по умолчанию, но при вычислениях нужно контролировать правильность результатов.

В JavaScript применяются выражения (expression) – объединение переменных, констант, функций знаками операций и скобками, операторами и методами. Например: f=10*Sin(x*t).

Операции **i++** и **i--** известные как инкремент и декремент.

Операции сравнения:

– ==равняется;
– != не равняется;
– ! логическое возражение;
– >=больше или равно;

– <=меньше или равно;
– > больше;
– < меньше.

Логические операции:
– (&&) логическое **И**;
– (||) логическое **ИЛИ**.

Операции выполняются с учетом приоритетов в следующем порядке: ++, --, ! , *, /, +,-,<, <=, >=, ==, !=, & , I .

В операторах применяются следующие операции (табл. 3.5).

Таблица 3.5 – Перечень операций присвоения

Операция	**Содержание операции**	**Выражение**	**Эквивалент**
=	непосредственное присвоение значения левому операнду	i=j	i=j
+=	составляет печали левого и правого операндов и присваивает результат левому операнду	i+=j	i=i+j
+	составляет печали левого и правого операндов и присваивает результат левому операнду	i+j	i=i+j
++	увеличивает значение левого операнда	i++	i=i+1
-=	отнимает значение правого операнда от левого и присваивает результат левому операнду	i-=j	i= i-j
-	отнимает значение правого операнда от левого и присваивает результат левому операнду	i-j	i= i-j
--	уменьшает значение левого операнда	i--	i= i-1
*	умножает значение левого и правого операндов и присваивает результат левому операнду	i*j	i=i*j
*=	умножает значение левого и правого операндов и присваивает результат левому операнду	i*=j	i=i*j
/	делит значение левого операнда на правый и присваивает результат левому операнду	i/j	i=i/j
/=	делит значение левого операнда на правый и присваивает результат левому операнду	i/=j	i=i/j

Встроенные функции

Для расчетов в JavaScript часто применяются следующие встроенные математические функции: **abs**(), **acos**(), **asin**(), **atan**(), **cos**(), **exp**(), **log**(), **max**(arg1, arg2,...), **min**(arg1, arg2,...), **pow**(основание, степень), **random**(1), **round**(), **sin**(), **sqrt**(), **tan**().

Обращение к функциям такое:

var functionname=Math.функция

Например, функция **random**() задается в интервале (0,1). Чтобы изменить диапазон, нужно функцию представить в таком виде:

```
function rand(min,max){
return Math.random()*( max-min)+min}
```

Структуры операторов

В JavaScript применяются **простые** и **сложные** операторы. К простым относятся: операторы присваивания и операторы вызова функций. Сложными считаются операторы, которые включают в себя простые операторы. Задаются они в виде блоков операторов, которые замыкаются в фигурные скобки { }. Например:

```
if (a==1)
{
a++;
// другие действия
}
```

Комментарии

В JavaScript применяются два вида комментариев:

1. **однострочный** комментарий со следующим синтаксисом:

// -текст комментария;

2. **многострочный** комментарий со следующим синтаксисом:

```
/*
Текст комментария
*/.
```

Операторы присваивания

Это наиболее распространенные операторы JavaScript. Их назначение – присваивать значение переменным. Синтаксис оператора:

Переменная = Выражение.

Например:

– **x=3**; // Переменной x присваивается значение 3;

– **y=x*x;** // Переменной y присваивается значение значения выражения x*x.

Операторы ввода

Ввод данных в JavaScript осуществляется с помощью операторов присваивания или специальных функций **alert**(), **confirm**() и **prompt**(). Эти функции удобны при организации интерактивного взаимодействия с пользователем. При их применении формируются диалоговые окна сообщений с кнопками **ОК** (функция **alert**); **ОК** и **Отменить** (функции **confirm** и **prompt**). В окнах функции **alert**() можно передавать числа, массивы, логические значения, объекты и функции. Данную функцию использует браузер для передачи текстов со сценария в HTML-документ. Синтаксис функции следующий:

alert(текст сообщения)

Для ввода текста пользователем удобно применять функцию **prompt**. Синтаксис ее следующий:

prompt(текст сообщения, текст по умолчанию)

Например:

```
<script language= JavaScript,
<!- -
pass= prompt('Введите пароль');
If pass==='1111'{
alert('Вход разрешен')
}else{
alert('Пароль неверный'), top.location href=URL ссылки}
//-->
}
</script>
```

Операторы выбора

В JavaScript применяются три вида операторов выбора:

1. оператор **?** – оператор вопросительный знак;
2. **if** – условный оператор;
3. **switch** – оператор выбора варианта.

Оператор ?.

Формат оператора следующий:

Условие ? выражение 1: выражение 2.

Здесь **Условие** – логическое выражение, которое принимает значение **True** или **False**.

Например:

(x>3)?y=x*x:y=x* x-1.

Согласно этому оператору вычисляется y: если x>3, то y=x*x, в противном случае y= x* x-1.

Оператор if

Довольно распространенный условный оператор. Формат его следующий:

If (условие) оператор 1
else оператор 2.

Оператор 1 и оператор 2 это – простые или сложные операторы. Сложные операторы замыкаются в фигурные скобки { }.

Например:

```
If (hour>=12)
type_time=”PM”;
else
type_time=”AM”;
```

Оператор switch

Оператор, обеспечивающий выбор варианта (case) и соответствующие ему действия по значению «выражения». Синтаксис оператора следующий:

```
switch (выражение){ // определение варианта case
case 1:
операторы, которые выполняются по варианту 1
```

case 2:
операторы, которые выполняются по варианту 2
…
defoult
операторы, которые выполняются по умолчанию
}

Например:

```
switch (var){
case 1: // операторы, которые выполняются при var==1, например: y=x;
break;
case 2: // операторы, которые выполняются при var==2, например: y=a*x;
break;
case 3: // операторы, которые выполняются при var==3, например: y=a*x*x;
break;
defoult;
}
```

Операторы циклов

JavaScript поддерживает три вида циклов:

1. for;
2. while;
3. do-while.

Оператор for

Представляется как итерационный цикл. Синтаксис этого оператора следующий:

for(выражение1; условие; виражение2){операторы},

где for – ключевое слово,

выражение1 – устанавливает начальное значение итерации**,**

условие – задает конечное значение числа итераций**,**

выражение2 – задает операцию изменения итераций**.**

Большей частью оператор **for** связан со счетчиком итераций, например:

var i // i объявляется как переменная (параметр цикла)

```
for (i=0;i<10;i++){ // работает счетчик
// Действия в цикле
}
```

Приведенный оператор удобно применять при обработке массивов данных.

Оператор while

Это оператор цикла с предусловием. Имеет следующий синтаксис:

while (условие) операторы.

Например:

```
var i=0
while (i<10){
// операторы действий
I++; // переход к следующей итерации
}
```

Оператор do-while

Это оператор цикла с постусловием. Формат оператора следующий:

do операторы **while** условие**.**

Например:

```
var i=0;
do{ // действия, которые должны выполняться
i++; //переход к следующей итерации
}while (i<10);
```

Операторы break и continue

В операторах **switch** и операторах циклов могут применяться операторы **break** с целью прерывания в нужных местах программы выборки вариантов или итерации циклов.

Например:

```
var I;
for (i=0;i<10;i++){
// Действия в цикле
If (i==8) break;
}
```

На итерации i==8 выполнение оператора for перерывается и следующие операции не выполняются. Если в теле цикла встречается оператор **continue**, то следующие операторы игнорируются, и

осуществляется переход к проверке условия цикла. Например:

```
var i, sum=0;
for(i=0;i<10;i++){
// Действия в цикле
If (i==5 II i==8) continue;
Sum+=i;
}
```

В приведенном фрагменте игнорируются итерации 5 и 8, а будут выполняться итерации 0 – 4, 6, 7, 10.

Использование функций в скриптах

Удобным способом написания программ сценариев в JavaScript является использование функций. Если функция должна выполняться при загрузке HTML-документа, то в нужном месте сценария должен быть указан оператор вызова функции.

Если функция должна выполняться, как реакция на событие, то в HTML-документе необходимо применить HTML-элемент с нужным атрибутом обработчика события, например **onclick**.

Операторы вывода

В качестве оператора для вывода информации в JavaScript используется **document.write().** В скобках записывается информация, которую необходимо вывести. Например:

```
document.write(«Полученный результат:»+Y+«<br>»)
```

После выполнения оператора браузер выведет на экран сообщение «Полученный результат:» и значение переменной Y.

Использование в сценариях объектной модели документа

Скрипт сценария может быть введен в HTML-документ двумя способами:

1. как обработчик результата события;
2. как утверждение или функция, которые используют тег <**SCRIPT**>.

Веб-приложения для обработчиков событий

С помощью веб-приложений в браузере в основном осуществляется управление событиями. Имеет место определенный набор событий

(**events**), которые браузер распознает и подключает соответствующий обработчик результата. Обработчик результата события записывается в документ как атрибут тега HTML, к которому дописывается код JavaScript для выполнения. Например, если созданная функция JavaScript с названием **compute**, то можно принудить браузер выполнять эту функцию, когда пользователь нажмет на кнопку, которой методом **onClick** выполняется обработка события, например:

<input type=«button» value=«calculate» onClick=«compute(this.form)»>

Обработчики событий это – методы (процедуры и функции) реакции на разные события. События делятся на три группы:

– связанные с мышью;
– связанные с клавиатурой;
– системные (программные).

Имена событий принято начинать с префикса **on**: **onKeyDown**, **onKeyPress**, **onKeyUp**, **onMauseDown**, **onMauseMove**, **onMauseUp**, **onClick**, **onClose**, **onCreate**, **onActivate** и т.д.

Объектная модель документа

JavaScript является объектно-ориентированным языком, а потому программирование веб-приложений основывается на использовании объектной модели документа (DOM). Объекты с точки зрения программиста – это сложные типы данных, которые поддерживают совокупность значений переменных (свойств) и функций для их обработки (методов). Эта сложность качественно другая, чем у массивов, файлов или записей. Массив по обыкновению определяется как совокупность элементов одного и того же базового типа. Запись – совокупность разнотипных данных. Объект – это совокупность не только данных (чаще всего разных типов), а и действий. При объектном программировании разработчик оперирует при создании кода объектами.

Компоненты-данные объектов называют свойствами объектов, а функции – методами. Объединение в одной структуре данных и выполняемых над ними действий называют **инкапсуляцией**. Это – первая из особенностей объекта. Вторая особенность – **наследование**. Суть ее в том, что, объявляя новый объект, имеется возможность включить к его составу все данные и методы, которые относятся к данному объекту. Объект-наследник называют рожденным типом, а предшественник –

родительским. За счет наследования можно создавать иерархию объектов.

Объектно-ориентированное программирование связано со средой визуального программирования. **Класс** в объектном программировании определяется как особый вид записи, которая может состоять из полей, методов и свойств.

Объекты одного типа (они имеют одинаковые наборы свойств и методов) представляют собой экземпляры одного класса. **Свойства** объектов используются в программе как переменные, т. е. могут получать некоторые значения. **Методы** объектов – это функции, которые должны выполняться над этими свойствами. Объекты, свойства и методы имеют имена. Чтобы узнать, к какому объекту относится свойство или метод, в программе имя отделяют от свойства или метода **точкой**.

Прежде чем применять объект, его сначала нужно создать. Это реализуется с помощью оператора **new**. Например, оператор **var today new Date()** создает объект **today** как новый объект в классе объектов **Date**. В данном случае это – текущие дата и время.

В JavaScript используются встроенные классы объектов. Наиболее употребляемыми среди них есть: **Date**, **Number**, **String**, **Math**.

Примеры программ JavaScript-сценариев

Ниже представлены примеры, которые имеют целью показать методы динамизации HTML-документов, а также закрепить навыки программирования веб-приложений средствами языка JavaScript.

Представленные в примерах программы – это HTML-документы, в виде самостоятельных веб-приложений (скрипты), или в виде скриптов веб-приложений в составе HTML-документов, или как интерактивные веб-приложения в составе HTML-документов, доступ к которым выполняется применением форм в HTML-документах. Во всех случаях браузер воспринимает и отображает их содержание.

В первом способе имеется возможность отрабатывать веб-приложения автономно. Во вторым – подготовленные веб-приложения включаются непосредственно в HTML-документы. В третьем – обеспечивается интерактивная связь веб-документа с веб-приложением путем применения формы для ввода данных пользователем и вывода результатов.

Все примеры сопровождаются методическими рекомендациями и комментариями в листингах программ.

Пример 1. Файл динамического HTML-документа «animate.html» со сценарием изменения демонстрации изображений во времени.

Изображение можно показывать в динамике – чтобы они появлялись периодически с заданным интервалом. Листинг программы, которая реализует такой процесс, следующий:

```
<html>
<head>
<title> анимации </title>
</head>

<!-Задаем цвет фоновой страницы->
<body bgcolor = lightcyan >
<!-Задаем форматирование шрифта заголовка страницы->
<p><font color = "lightscyblue">
<h1> <b><center> А еще я люблю животных </center></b></h1></font></P>

<p><font color = "lightscyblue">
<h3> <i><center> Посмотрите на подборку картинок с анимацией – мне они очень нравятся </center></i></h3></font></P>

<!-Программа на javascript->
<SCRIPT type="text/javascript">
// Назначаем функцию, которая действует по таймеру, и изменяет изображение от рисунка1.gif до рис. 9.gif с шагом 1
        window.setInterval(new_frame,2500);
        var inc=1;
        var StartFrame=1;
        var maxFrame=9;
// Функция изменения кадров
        function new_frame(){
```

```
// Показ текущего кадра из папки Gif
        animate.src="Gif/" + StartFrame + ".gif";
        StartFrame+= inc;
        if (StartFrame>maxFrame){
// Начинаем просмотр в обратном порядке
        StartFrame=maxFrame;
        inc=-1;
        }
else
        if (StartFrame==0)
        {
// Начинаем просмотр в прямом порядке
        StartFrame=1;
        inc=1;
        }
        }
        </SCRIPT>

<!-Форматирование рисунков из папки Gif по поводу ширины, высоты и рамки->
        <p align="center">
        <img id="animate" src="Gif/.gif" width="400" height="450" border="10">
        </p>
        </body>
        </html>
```

Количество кадров изображений задается переменной **maxFrame**. Предполагается, что все кадры помещены в папку Gif. Фрагмент одного из рисунков, воспроизводимых в окне браузера, показан на рисунке 3.42.

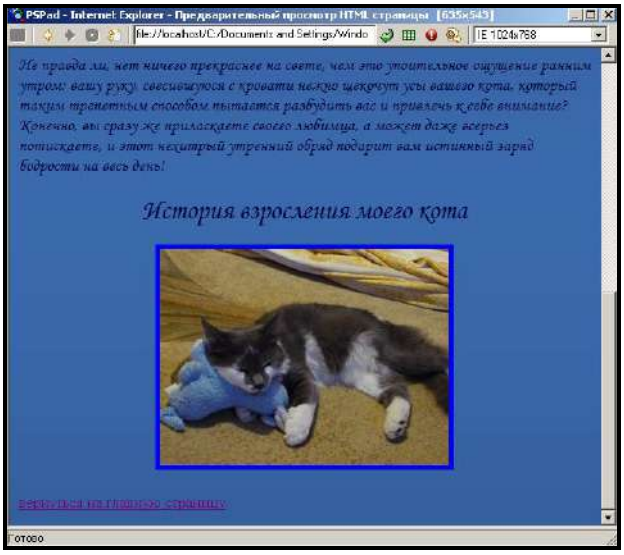

Рисунок 3.42 – Вид веб-страницы в окне браузера

Используя соответствующие теги, можно динамично изменять фон страницы. В приведенном примере используется объект **window** для установления интервала показа изображений.

Пример 2. HTML-документ «Календарь.html», обеспечивающий формирование календаря (рис. 3.43).

Пример знакомит со средствами применения свойств и методов объекта **Date()**, а также с конструкцией оператора **if**.

Рисунок 3.43 – Вид календаря, полученного с помощью скрипта

```
<html>
<head>
<meta name=robots content="noindex">
</head><body bgcolor=white>
<title>http://programinet.narod.ru/ - База JavaScript (bjs)</title>
<SCRIPT language=Javascript>
calendar = new Date();
day = calendar.getDay();
```

```
document.write("<font face=arial><center><table width=100 border=1><td><center><font size=2>")
if (day == 0) {document.write("Воскресенье")}
if (day == 1) {document.write("Понедельник")}
…
if (day == 6) {document.write("Суббота")}
document.write("</font></center></td><tr><td><center><font size=2>")
month = calendar.getMonth();
if (month == 0) {document.write("Январь")}
if (month == 1) {document.write("Февраль")}
…
if (month == 11) {document.write("Декабрь")}

document.write("</font></center></td><tr><td><center><font size=7>")
date = calendar.getDate();
document.write(date)
document.write("</font></center></td><tr><td><center><font size=2>")
year = calendar.getYear();
if (year < 100) {
document.write("19" + year + "")
}
else if (year > 1999) {
document.write(year)
}
document.write("</font></center></td></table>")
</SCRIPT>
</HTML>
```

Ход работы

Требования к создаваемому сайту

В уже созданные и связанные гиперссылками веб-страницы своего сайта поместите какую-либо программу **JavaScript** для придания динамических эффектов контенту.

Для размещения кода программы на языке **JavaScript,** создайте контейнер следующего вида:

<script language = «JavaScript»>
//операторы JavaScript ...
</script>

в необходимом месте своей веб-страницы для решения какой-либо задачи, позволяющей «оживить» вашу статическую страницу. Можете воспользоваться примерами, рассмотренными в теоретической части данной лабораторной работы.

3.6 Создание на сайте информеров. Размещение сайта в сети Интернет

Цель: приобрести знания и умения в области публикации (расположения) созданного сайта из нескольких веб-страниц в сети Интернет а также размещения различных информеров на своих веб-страницах.

Назначение: выполнив работу, вы научитесь:

– размещать различные информеры на страницах своего сайта, позволяющие «оживить» статичную веб-страницу;

– выбирать в зависимости от специфики сайта и своих предпочтений хостинг;

– располагать сайт на удаленном сервере;

– вносить изменения в уже опубликованный сайт в случае необходимости.

Полученные знания и умения помогут опубликовать свой сайт в сети Интернет.

Теоретические сведения

Для вставки видео- либо аудиозаписи можно в любом месте HTML-документа поместить тег **Embed** со следующими атрибутами:

<Embed src = «имя файла с расширением»>

Атрибуты этого тега имеют следующие значения:

– **Width** = «число» – размер аудиозаписи по горизонтали;

– **Height** = «число» – размер аудиозаписи по вертикали;

– **Autostart** = «значение» – автозапуск или запуск по нажатию кнопки (True – сразу после запуска, False – запуск после нажатия кнопки);

– **Loop** = «True» – зацикливание ролика;

– **Controller** = «начение» – есть/нет клавиши управления воспроизведением (True – есть, False – нет).

Более наглядно результат работы тега Embed продемонстрированы на рисунках 3.44, 3.45.

```
<!--Вставка видео о коте-->
 <Embed src="2011_02.AVI" Width="300px" Height="225px" Autostart=False Loop=
False Controller=True>
```

Рисунок 3.44 – HTML-код веб-страницы с тегом Embed для вставки видео

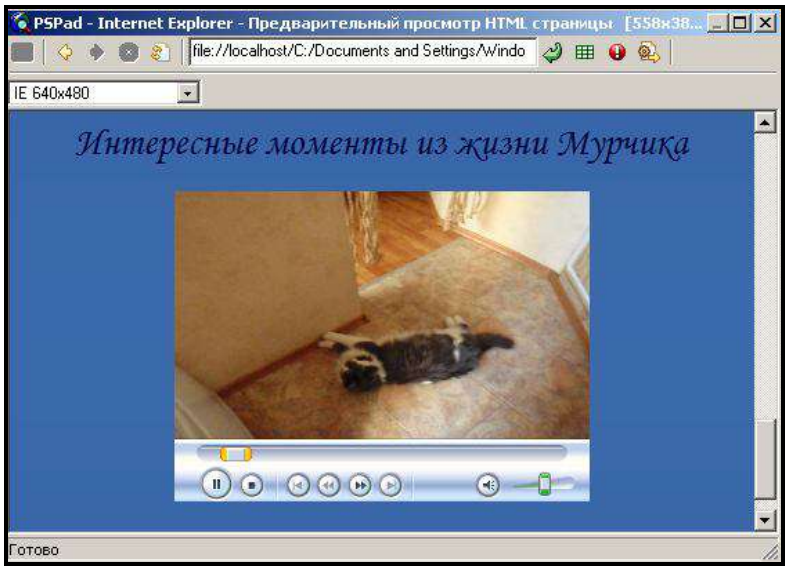

Рисунок 3.45 – Вид веб-страницы в окне браузера

Также для вставки видео- либо аудиозаписи на свою HTML-страницу можно использовать сервис YouTube. Для этого необходимо выполнить следующие действия:

1. Прежде всего необходимо зарегистрироваться.
2. После регистрации, перейти по ссылке **Добавить Видео.**

3. В появившейся форме, используя кнопку **Обзор** загрузить видео со своего компьютера, размер которого не должен превышать 1 Гб.

4. После успешной загрузки, необходимо подождать некоторое время, чтобы видео перецифровалось в нужный формат.

5. Выполнить щелчок левой кнопкой мыши по своему логину, под которым была осуществлена регистрация на сервисе и выбрать «**Мое Видео**», затем выполнить щелчок по уменьшенному изображению видео, и на открывшейся странице в правом верхнем углу скопировать HTML код плеера, который затем можно вставить на свою веб-страницу.

6. Размеры плеера задаются с помощью атрибутов **height** и **width.**

Для вставки на свою веб-страницу прогноз погоды необходимо вставить так называемый «информер». Информеры прогноза погоды позволят показывать погоду не только в вашем регионе, но параллельно и в любой другой части света. Это может понадобиться например для веб-страницы туристического агентства – для рекламы того места отдыха, куда предлагаются путевки. Конечно, информер погоды должен вписываться в дизайн сайта, не должен быть чужеродным элементом на вашей странице. Разнообразие предлагаемых на сайтах прогнозов погоды информеров велико, есть даже такие, которые позволяют полностью настроить все элементы по их дизайну. Информеры погоды можно взять на следующих сайтах:

– http://meteoservice.ru/content/informers.html;
– http://weather.yandex.ru/informer.xml;
– http://gismeteo.ua;
– http://prognoz-pogoda.com (полная настройка всех параметров дизайна).

Для вставки на свою веб-страницу информации о текущем времени, либо времени того региона, о котором идет речь на вашей веб-странице также необходимо, как и в случае с погодой, воспользоваться разнообразными предлагаемыми на сайтах видами часов как по дизайну, так и по функциональности. Адрес одного из таких сайтов: http://clocklink.com.

Размещение сайта в сети Интернет

Для размещения сайта в сети нужно лишь определиться с провайдером на сервере, которого вы бы хотели разместить свой сайт. Вопрос выбора хостинга полностью лежит на вебмастере, а выбор провайдера хостинга полностью лежит на вас.

Хостинг – это определенного вида услуга, которая дает возможность создателям сайтов, тоесть вебмастерам, размещать свои сайты на серверах хостинговых компаний, выполняющих обслуживание хостинговых серверов и обеспечивающих доступ к сайту через Интернет. Другими словами, хостинг – это предоставление во временную аренду некоторого дискового пространства сервера для размещения на нем сайтов.

Платный хостинг – при выборе хостинга для размещения сайта в сети, который предоставляется в пользование вебмастеру за деньги, компания провайдер несет обязанности по обслуживанию и сервису предоставленного вида услуги. Заказчик обязан вовремя производить оплату услуг по предоставлению хостинга и несет ответственность за соблюдение правил, установленных провайдером. Цена на предоставление хостинга в пользование для размещения ресурса колеблется и зависит от тарифного плана, который будет выбран.

Каждый тарифный план включает в себя те или иные дополнительные функции по управлению сайтом и имеет определенный предоставляемый размер дискового пространства. При выборе платного хостинга предоставляется постоянная техническая поддержка, возможность использовать при создании сайта несколько языков програмирования и стабильность работы хостингового сервера с быстрой загрузкой страниц размещаемого сайта, а также доменное имя второго уровня.

Размещение сайта в сети на коммерческих условиях дает возможность выбрать доменное имя. Немного о доменном имени: **ru, biz, com, ua** – это все доменные имена первого уровня. **Rambler, narod** – доменные имена второго уровня. **Имя сайта.narod.ru** – в этом случае имя сайта будет иметь доменное имя третьего уровня или называемое еще поддомен. Значимость доменного имени в раскрутке сайта очень важна. По возможности адрес сайта должен быть коротким, запоминающимся и интуитивно понятным. URL такого домена наверняка быстрее отложится в памяти пользователя, нежели длинный и сложный адрес. Немаловажным

следствием регистрации своего доменного имени является то, что вы не будете зависеть от организации, поддерживающей сервер, на котором размещается ваш сайт. Если появится необходимость сменить хостинг, потребуется всего лишь изменить данные об этом на сервере, предоставившем вам домен. Отсюда следует, что не придётся, перерегистрировать свой ресурс в поисковых системах, писать письма администраторам разных каталогов, чтобы они изменили ссылки на ваш сайт.

Также сайты, расположенные на платных хостингах, вызывают больше доверия. И если у вас, например, интернет магазин или вы что-либо продаёте со своего сайта, то покупатели будут вам больше доверять, если разместите сайт на платном хостинге.

Бесплатный хостинг – такой хостинг предоставляется некоторыми провайдерами. При размещении сайта на бесплатном хостинге вам предоставляется дисковое пространство небольшого объема – это первый минус. Невозможность использовать некоторые языки програмирования и скрипты. Провайдеры бесплатного хостинга будут размещать на сайте свою рекламу, что не всегда подходит под дизайн сайта. Еще один минус бесплатного хостинга заключается в том, что сайту будет выдано доменное имя третьего уровня. Скорость загрузки страниц в браузере будет не такой быстрой, как на платном хостинге.

Есть и другие недостатки серверов, предоставляющих бесплатный хостинг: большинство бесплатных серверов не поддерживают такие языки программирования, как: PHP, Perl, CGI, MySQL. Бесплатные сервера могут закрыться и перестать предоставлять бесплатное место под сайт. Бесплатный хостинг по сравнением с платным имеет единственное достоинство – отсутствие необходимости тратить деньги.

Бесплатный хостинг подойдет вам, если вы только начинающий разработчик сайтов, создаете свою личную страничку.

www.narod.ru – довольно популярный бесплатный хостинг в рунете. Неограниченный трафик и дисковое пространство, 100 готовых шаблонов для вашего сайта, гостевая книга, форум, чат, статистика посещений. Из недостатков хочется отметить низкую скорость, отсутствие поддержки PHP, CGI, возможны проблемы с закачкой файлов по FTP. Реклама: небольшой баннер в правом верхнем углу.

Как разместить сайт в сети Интернет и получить доменное имя

Если ваш сайт готов и необходимо получить для него доменное имя, то есть имя, по которому люди будут заходить на сайт, и разместить его в сети Интернет. Для размещения сайта в Интернете и получения доменного имени существует несколько решений.

Первый способ

Размещение созданного сайта на сервере интернет-провайдера вашего города или иного населенного пункта. Вы приходите в центр технической поддержки и узнаете, как разместить на их сервере имеющийся у вас сайт. Персонал центра на месте расскажет вам, что для этого необходимо и сколько это будет стоить.

Второй способ

Обратиться к фирме, предоставляющей услуги хостинга через Интернет. В данном варианте вы ознакамливаетесь с прайсом предоставляемых услуг непосредственно на сайте хостера. Вся работа и общение будет происходить удаленно.

Третий способ

Наиболее сложный – разместить сайт у себя на компьютере. Однако тут есть много но:

– Ваш компьютер должен быть всегда включен.

– Должен быть надежный, скоростной канал связи с провайдером: выделенная линия или что-то подобное.

– На вашем компьютере должна быть установлена и правильно сконфигурирована серверная операционная система – Windows server, UNIX.

– У провайдера нужно получить выделенный IP адрес, зарегистрировать домен и договориться о поддержке вашего доменного имени на DNS-сервере провайдера.

Четвертый способ

Этот способ наименее трудоемкий и капиталоемкий. Он заключается в размещении сайта на бесплатном хостинге. Как самостоятельно разместить сайт на бесплатном хостинге? Необходимо зайти на сайт, предоставляющий бесплатный хостинг, заполнить форму, получить письмо о регистрации сайта на ваш электронный адрес.

Самые важные показатели для размещения сайта в сети Интернет

Uptime

Этот показатель характеризует время бесперебойной работы сервера в течение установленного промежутка времени. Измеряется он в процентах, и должен он быть близким к 100 процентам.

Какой показатель Uptime считается достаточным? Разместить Интернет-проект на хостинге с Uptime равным 100 % просто невозможно. Любой хостинг должен останавливаться на профилактику и обслуживание.

Если на странице сайта хостинга размещен информер с реальным Uptime – это показатель высокой его надежности.

Скорость загрузки страниц

Разместить сайт в Интернете недостаточно, нужно еще чтобы его страницы достаточно быстро загружались при просмотре их посетителями. Если загрузка идет долго, посетителю это просто надоест, и он уйдет, так и не дождавшись, пока страница откроется.

Скорость загрузки страниц измеряется в килобайтах в секунду (Кб/с). Прежде чем разместить сайт на хостинге, этот показатель нужно проверить. В отличие от Uptime, скорость загрузки страниц проверить можно просто зайдя на любой сайт, расположенный у этого провайдера.

Время ответа сервера

Суть этого показателя ясна. Это время, которое проходит с момента, когда браузеру дана команда начать загрузку страницы (нажатием клавиши или кнопки) до момента начала загрузки.

Какой показатель для размещения своего сайта важнее? Из двух показателей, которые реально можно измерить, конечно же, важнее скорость загрузки страниц.

Для любого посетителя любого сайта пауза перед началом загрузки страницы довольно естественна. Психологически оценивать сайт посетитель начинает одновременно с началом загрузки страницы. Раздражает посетителя задержка начала загрузки страницы, превышающая 1,5–2 секунды. Меньшее значение задержки большинство посетителей сайта просто не замечают.

Другое дело скорость загрузки страницы. Если загрузка страницы уже началась, но идет медленно, это может привести к тому, что посетитель покинет ваш сайт.

Размещение сайта на сервере бесплатной службы

В качестве примера размещения сайта на бесплатном хостинге рассмотрим сервер бесплатной службы http://www.narod.ru/. Эта служба предоставляет неограниченный объем дискового пространства, а также доменное имя третьего уровня. Для размещения сайта на сервере этого провайдера необходимо пройти процедуру регистрации перейдя по ссылке на сайт провайдера, набрав в адресной строке браузера **http://www.narod.ru/.** После этого в левом верхнем углу страницы увидите ссылку **Регистрация,** перейдя по которой, откроется новая форма первого этапа регистрации. После заполнения формы необходимо будет нажать кнопку **Дальше** внизу страницы и перейти ко второму этапу регистрации. В формах обычно предлагают указать фамилию, имя, отчество, домашний адрес, e-mail.

Заранее необходимо придумать имя для вашего сайта. Имя которое вы выберете для размещения сайта в сети, может оказаться уже занятым, так как сайтов огромное множество и вероятность совпадения имен очень велика. Если так случится, то появится предупреждение о том, что такое имя уже занято, необходимо будет изменить имя сайта, пока система не зарегистрирует его как свободное. После заполнения всех форм служба выдаст сообщение об успешной регистрации и предложит ввести дополнительные данные.

После полного завершения регистрации и заполнения всех данных на адрес электронной почты, который был указан при регистрации на сервере, придет письмо со ссылкой для подтверждения регистрации.

Сайт зарегистрирован, но это фактически лишь его название и некоторое место на сервере, а для того чтобы он начал функционировать, необходимо загрузить созданные вами веб-странички.

Публикация своего сайта. Инструкция по размещению

1. Для перемещения вашего сайта с жёсткого диска на сервер лучше всего воспользоваться файловым менеджером. Например **Total Commander.** Запустите файловый менеджер и щелкните по кнопке **Соединиться с ftp-сервером.**

2. После этого на экране монитора появится форма с аналогичным названием. Поскольку вы еще не устанавливали параметры соединения с сервером, то список возможных соединений будет пуст. Для того чтобы

добавить в него новое место соединения, нужно щелкнуть по клавише **Добавить**, после чего откроется форма **Настройка ftp-соединения.**

3. В строке **Заголовок** можно написать что угодно, например «**Мой первый сайт**», при написании заголовка можно использовать кириллицу.

4. В строке **Адрес (порт)** необходимо указать доменное имя сервера файлового архива – **ftp.narod.ru.**

5. В строке **Учетная запись** необходимо указать логин вашего сайта, например адрес сайта **my_site.narod.ru**, соответственно логин – **my_site.**

6. В строке **Пароль** необходимо указать пароль для доступа к сайту.

7. В строке **Локальный каталог** необходимо указать путь к локальной версии вашего сайта. Это необходимо для того, чтобы после соединения с сервером файловых архивов, содержание документов, из которых состоит ваша локальная версия сайта, загружалось на сервер провайдера.

8. Нажмите на кнопку **ОК.**

9. После закрытия окна настройки ftp-соединения на форме **Соединиться с ftp-сервером** появится название нового соединения. Теперь достаточно выделить его и щелкнуть по кнопке **Соединиться** и в одну из половин файлового менеджера будет загружено содержимое локальной версии сайта, а вторая окажется пустой, т. к. пока на сервер не загружен ни один файл.

10. После того как будет установлено соединение, вы можете выделить все файлы в локальной папке и скопировать их в другую половину файлового менеджера, тем самым опубликовав (загрузив) ваш сайт в сети Интернет.

Загрузка сайта браузером с сервера Интернет-провайдера

Обычно просмотр сайта начинается с главной страницы. При обилии в корне сайта различных файлов как веб-сервер узнает, какой файл ему загружать автоматически? Конечно, если путь к файлу указан напрямую, никаких вопросов не возникает. Но в большинстве случаев, адрес сайта указывается коротко, без лишних файлов на конце. Вот тогда читаются настройки сервера и определяется файл, который следует показать, а также, есть ли указанный файл в наличии. Как правило, такой файл имеет имена **index.html, index.htm, default.htm.** Имя файла, загружаемого первым (главную страницу сайта), можно установить самостоятельно,

используя для этого файл **.htaccess.** Данный файл является конфигурационным для популярного веб-сервера Apache и представляет собой обычный текстовый документ. В нем следует прописать такую строку.

DirectoryIndex index.html index.htm

В этом файле необходимо через пробел указать имена файлов, которые браузер будет первыми запускать при обращении к вашему сайту.

Порядок выполнения работы

Уже созданные и связанные гиперссылками веб-странички дополните различными информерами в соотвтствии с темой и дизайном вашего сайта. После этого полностью завершенный и отредактированный веб-сайт расположите в сети Интернет на сайте бесплатного провайдера и покажите преподавателю.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Основные

1. Дакетт Д. HTML и CSS. Разработка и дизайн Web-сайтов / Д. Дакетт. – М. : Эксмо, 2017. – 480 с.

2. Минник К. HTML5 и CSS3 для чайников / К. Минник, Э. Титтел. – М. : Диалектика, 2016. – 400 с.

Дополнительные

3. Хрусталев А. HTMLS + CSS3. Основы современного WEB-дизайна / А. Хрусталев, А. Кириченко. – М. : Наука и Техника, 2018 – 352 с.

4. Веру Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – Питер, 2017. – 336 с.

Ресурсы сети Интернет

5. Сайт Влада Мержевича [Электронный ресурс]. – Режим доступа : http://htmlbook.ru – Загол. с экрана.

6. Сайт «Создаем и раскручиваем сайт самостоятельно». [Электронный ресурс]. – Режим доступа: https://site-do.ru/css/ – Загол. с экрана.

Навчальне видання

ПОМОРЦЕВА Олена Євгенівна

МЕРЕЖЕВІ ТЕХНОЛОГІЇ. ОСНОВИ ВЕБ-ДИЗАЙНУ

НАВЧАЛЬНИЙ ПОСІБНИК

(Рос. мовою)

Відповідальний за випуск *О. Є. Поморцева*

Редактор *О. В. Михаленко*

Комп'ютерне верстання *І. В. Волосожарова*

Дизайн обкладинки: *К. П. Растріполко,*
Т. А. Лазуренко

Підп. до друку 19.03.2020. Формат 60 × 84/16.
Друк на ризографі. Ум. друк. арк. 7,7.
Тираж 60 пр. Зам. №

Видавець і виготовлювач:
Харківський національний університет
міського господарства імені О. М. Бекетова,
вул. Маршала Бажанова, 17, Харків, 61002.
Електронна адреса: rectorat@kname.edu.ua
Свідоцтво суб'єкта видавничої справи:
ДК № 5328 от 11.04.2017.