

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«Санкт-Петербургский государственный университет
промышленных технологий и дизайна»**

Кафедра информационных технологий

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ИНТЕРНЕТ- ТЕХНОЛОГИИ

Методические указания

для бакалавров заочного обучения

29.03.01 «Технология изделий легкой промышленности»

Составитель

В. Г. Мешкомаев

Санкт-Петербург
2024

Утверждено
на заседании кафедры
24.09.23 г., протокол № 9

Рецензент Е. Н. Якуничева

Методические указания по курсу «Информационные системы и интернет-технологии» разработаны по материалам кафедры информационных технологий в соответствии с требованием ФГОС.

Методические указания включают в себя цели, задачи, тематику и общее содержание курса, задания для исполнения, список литературы.

Предназначены для бакалавров заочного обучения направления 29.03.01 технология изделий легкой промышленности.

Учебное электронное издание сетевого распространения
Издано в авторской редакции

Системные требования:
электронное устройство с программным обеспечением для воспроизведения файлов формата PDF

Режим доступа: http://publish.sutd.ru/tp_get_file.php?id=2018_378, по паролю. – Загл. с экрана.
Дата подписания к использованию .2024 г. Рег. №

ФГБОУВО «СПбГУПТД»
Юридический и почтовый адрес: 191186, Санкт-Петербург, ул. Большая Морская, 18.
<http://sutd.ru>

ОГЛАВЛЕНИЕ

1. ИНФОРМАЦИОННЫЕ СИСТЕМЫ.....	5
1.1. История развития информационных систем.....	5
1.2. Структура информационной системы	6
1.3. Понятие жизненного цикла информационных систем	8
1.4. Стадии жизненного цикла информационных систем	8
1.5. Классификация информационных систем по масштабу	9
1.6. Классификация информационных систем по сфере применения.....	10
1.7. Классификация информационных систем по способу организации	12
1.8. Области применения информационных систем	14
2. ИНТЕРНЕТ-ТЕХНОЛОГИИ	15
2.1. Сервисы интернета	16
3. ПРОЕКТИРОВАНИЕ WEB-СТРАНИЦ	19
3.1. Логическая и физическая структура сайта.....	21
4. ВЕРСТКА WEB-СТРАНИЦ.....	25
5. ОСНОВНЫЕ ПОНЯТИЯ HTML	31
5.1. Базовая структура HTML-документа.....	32
5.2. Создание web-страницы	33
5.3. Элементы форматирования web-страницы	34
5.3.1. Создание заголовков	35
5.3.2. Создание абзацев	35
5.3.3. Создание горизонтальной черты	36
5.3.4. Задание №1	36
5.3.5. Создание списков	37
5.3.6. Задание №2.....	38
5.3.7. Создание таблиц	39
5.4. Шрифтовое оформление web-страницы.....	40
5.4.1. Основные теги форматирования текста	40
5.4.2. Задание №3.....	42
5.5. Цветовое оформление web-страницы	42
5.5.1. Задание №4.....	43
5.6. Гиперссылки	43
5.7. Графическое оформление web-страницы.....	45

5.7.1. Фоновое изображение	45
5.7.3. Задание №5.....	47
6. КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ (CSS)	47
7. ПРАКТИЧЕСКОЕ ЗАДАНИЕ	50
7.1. Основные действия студента по созданию HTML-страниц личного сайта	50
УЧЕБНАЯ ЛИТЕРАТУРА	51

1. ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Под *информационной системой* (ИС) понимается прикладная программная система, ориентированная на сбор, хранение, поиск и обработку текстовой и/или фактографической информации. Подавляющее большинство информационных систем работает в режиме диалога с пользователем. *Автоматизированная информационная система* (АИС) – комплекс технических, программных, других средств и персонала, предназначенный для автоматизации различных процессов.

1.1. История развития информационных систем

Первые информационные системы появились в 50-х гг. XX в. В эти годы они были предназначены для обработки счетов и расчета зарплаты, а реализовывались на электромеханических бухгалтерских счетных машинах. Это приводило к некоторому сокращению затрат и времени на подготовку бумажных документов.

60-е гг. знаменуются изменением отношения к информационным системам. Информация, полученная из них, стала применяться для периодической отчетности по многим параметрам. Для этого организациям требовалось компьютерное оборудование широкого назначения, способное выполнять множество функций, а не только обрабатывать счета и считать зарплату, как было ранее.

В 70-х – начале 80-х гг. информационные системы начинают широко использоваться в качестве средства управленческого контроля, поддерживающего и ускоряющего процесс принятия решений.

К концу 80-х гг. концепция использования информационных систем вновь изменяется. Они становятся стратегическим источником информации и используются на всех уровнях организации любого профиля. Информационные системы этого периода, предоставляя вовремя нужную информацию, помогают организации достичь успеха в своей деятельности, создавать новые товары и услуги, находить новые рынки сбыта, обеспечивать себе достойных партнеров, организовывать выпуск продукции по низкой цене и многое др.

1.2. Структура информационной системы

Структуру информационной системы составляет совокупность отдельных ее частей, называемых подсистемами.

Подсистема – это часть системы, выделенная по какому-либо признаку.

Общую структуру информационной системы можно рассматривать как совокупность независимых подсистем. В этом случае говорят о структурном признаке классификации, а подсистемы называют обеспечивающими. Таким образом, структура любой информационной системы может быть представлена совокупностью обеспечивающих подсистем.

Среди обеспечивающих подсистем обычно выделяют информационное, техническое, математическое, программное, организационное и правовое обеспечение.

Информационное обеспечение – совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных (БД).

Техническое обеспечение – комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Математическое и программное обеспечение – совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.

В состав *программного обеспечения* (ПО) входят общесистемные и специальные программные продукты, а также техническая документация.

К *общесистемному программному обеспечению* относятся комплексы программ, ориентированных на пользователей и предназначенных для решения типовых задач обработки информации. Они служат для расширения функциональных возможностей компьютеров, контроля и управления процессом обработки данных.

Специальное программное обеспечение представляет собой совокупность программ, разработанных при создании конкретной информационной системы. В его состав входят пакеты прикладных программ (ППП), реализующие разработанные модели разной степени адекватности, отражающие функционирование реального объекта.

Техническая документация на разработку программного обеспечения должна содержать описание задач, задание на алгоритмизацию, экономико-математическую модель задачи, контрольные примеры.

Организационное обеспечение – это совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации ИС.

Правовое обеспечение – совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.

1.3. Понятие жизненного цикла информационных систем

Понятие жизненного цикла ИС является одним из базовых понятий. *Жизненный цикл (ЖЦ)* определяется как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

Основным нормативным документом, регламентирующим состав процессов ЖЦ, является международный стандарт ISO. Он определяет структуру ЖЦ, содержащую процессы, действия и задачи, которые должны быть выполнены во время создания ПО.

Полный жизненный цикл информационной системы включает в себя, как правило, стратегическое планирование, анализ, проектирование, реализацию, внедрение и эксплуатацию. В общем случае жизненный цикл можно в свою очередь разбить на ряд стадий. В принципе, это деление на стадии достаточно произвольно.

1.4. Стадии жизненного цикла информационных систем

Согласно международному стандарту ISO, жизненный цикл информационной системы подразделяется на четыре стадии:

- начало;
- уточнение;
- конструирование;
- передача в эксплуатацию.

На *начальной стадии* устанавливается область применения системы и определяются граничные условия. На *стадии уточнения* проводится анализ прикладной области, разрабатывается архитектурная основа информационной

системы. На *стадии конструирования* разрабатывается законченная информационная система, готовая к передаче пользователю. На *стадии передачи в эксплуатацию* разработанное программное обеспечение передается пользователям.

1.5. Классификация информационных систем по масштабу

По масштабу информационные системы подразделяются на следующие виды (*рис. 1.1*):

- одиночные;
- групповые;
- корпоративные.

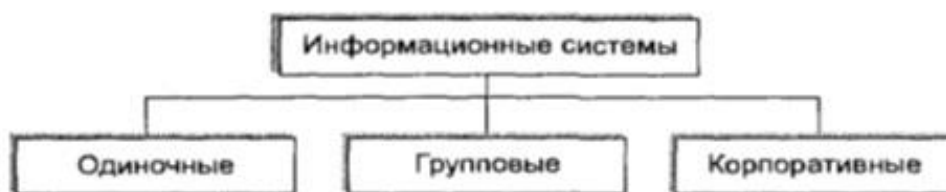


Рис. 1.1. Деление информационных систем по масштабу

Одиночные информационные системы реализуются, как правило, на автономном персональном компьютере. Такая система может содержать несколько простых приложений, связанных общим информационным фондом, и рассчитана на работу одного пользователя или группы пользователей, разделяющих по времени одно рабочее место. Подобные приложения создаются с помощью так называемых *настольных, или локальных, систем управления базами данных (СУБД)*. Среди локальных СУБД наиболее известными являются *Clarion, Clipper, FoxPro, Paradox, dBase* и *QicrosoftAccess*.

Групповые информационные системы ориентированы на коллективное использование информации членами рабочей группы и чаще всего строятся на базе локальной вычислительной сети. При разработке таких приложений используются серверы баз данных (называемые также SQL-серверами) для рабочих групп. Существует довольно большое количество различных SQL-серверов как коммерческих, так и свободно распространяемых. Среди них наиболее известны такие серверы баз данных, как *Oracle*, *DB2*, *Microsoft SQL Server*, *InterBase*, *Sybase*, *Inforqix*.

Корпоративные информационные системы являются развитием систем для рабочих групп, они ориентированы на крупные компании и могут поддерживать территориально разнесенные узлы или сети. В основном они имеют иерархическую структуру из нескольких уровней. Для таких систем характерна архитектура клиент-сервер со специализацией серверов или же многоуровневая архитектура. При разработке таких систем могут использоваться те же серверы баз данных, что и при разработке групповых информационных систем. Однако в крупных информационных системах наибольшее распространение получили серверы *Oracle*, *DB2* и *Microsoft SQL Server*.

1.6. Классификация информационных систем по сфере применения

По сфере применения информационные системы обычно подразделяются на четыре вида (*рис.1.2*):

- системы обработки транзакций;
- системы поддержки принятия решений;
- информационно-справочные системы;
- офисные информационные системы.

Системы обработки транзакций, в свою очередь, по оперативности обработки данных разделяются на пакетные информационные системы и оперативные информационные системы. В информационных системах организационного управления преобладает режим оперативной обработки транзакций – OLTP для отражения актуального состояния предметной области в любой момент времени, а пакетная обработка занимает весьма ограниченную часть.

Системы поддержки принятия решений – DSS – представляют собой другой тип информационных систем, в которых с помощью довольно сложных запросов производится отбор и анализ данных в различных разрезах: временных, географических и по другим показателям.

Обширный класс *информационно-справочных систем* основан на гипертекстовых документах и мультимедиа. Наибольшее развитие такие информационные системы получили в сети интернет.

Класс *офисных информационных систем* нацелен на перевод бумажных документов в электронный вид, автоматизацию делопроизводства и управление документооборотом.



Рис. 1.2. Деление информационных систем по сфере применения

1.7. Классификация информационных систем по способу организации

По способу организации групповые и корпоративные информационные системы подразделяются на следующие классы (рис. 1.3):

- системы на основе архитектуры файл-сервер;
- системы на основе архитектуры клиент-сервер;
- системы на основе многоуровневой архитектуры;
- системы на основе интернет-технологий.

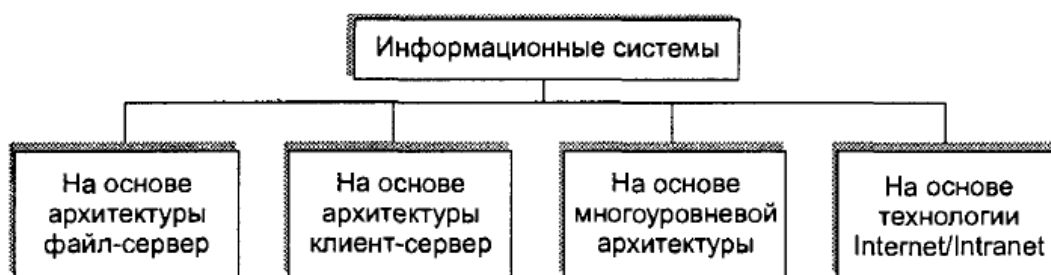


Рис. 1.3. Деление информационных систем по способу организации

Архитектура файл-сервер использует компьютер для функций отображения, что облегчает построение графического интерфейса. Файл-сервер только извлекает данные из файлов, так что дополнительные пользователи и приложения добавляют лишь незначительную нагрузку на центральный процессор. Каждый новый клиент добавляет вычислительную мощность к сети. Разработанное приложение реализуется либо в виде законченного загрузочного модуля, либо в виде специального кода для интерпретации. Недостаток: при выполнении некоторых запросов к базе данных клиенту могут передаваться большие объемы данных, загружая сеть и приводя к непредсказуемости времени реакции.

Архитектура клиент-сервер предназначена для разрешения проблем файл-серверных приложений путем разделения компонентов приложения и

размещения их там, где они будут функционировать наиболее эффективно. Особенностью архитектуры клиент-сервер является использование выделенных серверов баз данных, понимающих запросы на языке структурированных запросов SQL и выполняющих поиск, сортировку и агрегирование информации. Отличительная черта серверов БД – наличие справочника данных, в котором записана структура БД, ограничения целостности данных, форматы и даже серверные процедуры обработки данных по вызову или по событиям в программе.

В настоящее время архитектура клиент-сервер получила признание и широкое распространение как способ организации приложений для рабочих групп и информационных систем корпоративного уровня. Подобная организация работы повышает эффективность выполнения приложений за счет использования возможностей сервера БД, разгрузки сети и обеспечения контроля целостности данных.

Многоуровневая архитектура стала развитием архитектуры клиент-сервер и в своей классической форме состоит из трех уровней:

- нижний уровень представляет собой приложения клиентов, выделенные для выполнения функций и логики представлений PS и PL и имеющие программный интерфейс для вызова приложения на среднем уровне;
- средний уровень представляет собой сервер приложений, на котором выполняется прикладная логика BL и с которого логика обработки данных DL вызывает операции с базой данных DS;
- верхний уровень представляет собой удаленный специализированный сервер базы данных, выделенный для услуг обработки данных DS и файловых операций FS.

Многоуровневая архитектура распределенных приложений позволяет повысить эффективность работы корпоративной информационной системы и оптимизировать распределение ее программно-аппаратных ресурсов. Но пока на российском рынке по-прежнему доминирует архитектура клиент-сервер.

В развитии *технологии интернет/интранет* основной акцент пока что делается на разработке инструментальных программных средств. В то же время наблюдается отсутствие развитых средств разработки приложений, работающих с базами данных. Компромиссным решением для создания удобных и простых в использовании и сопровождении информационных систем, эффективно работающих с базами данных, стало объединение интернет/интранет-технологии с многоуровневой архитектурой. При этом структура информационного приложения приобретает следующий вид: браузер – сервер приложений – сервер баз данных – сервер динамических страниц – web-сервер. Благодаря интеграции интернет/интранет-технологии и архитектуры клиент-сервер процесс внедрения и сопровождения корпоративной информационной системы существенно упрощается при сохранении достаточно высокой эффективности и простоты совместного использования информации.

1.8. Области применения информационных систем

Благодаря стремительному развитию информационных технологий наблюдается расширение области их применения. Если раньше чуть ли не единственной областью, в которой применялись информационные системы, была автоматизация бухгалтерского учета, то сейчас наблюдается внедрение информационных технологий во множество других областей. Эффективное использование корпоративных информационных систем позволяет делать более точные прогнозы и избегать возможных ошибок в управлении. Информационные системы позволяют извлекать максимум пользы из всей имеющейся в компании информации. Именно этим фактом и объясняются жизнеспособность и бурное развитие информационных технологий. Современный бизнес крайне чувствителен к ошибкам в управлении, и для

принятия грамотного управленческого решения в условиях неопределенности и риска необходимо постоянно держать под контролем различные аспекты финансово-хозяйственной деятельности предприятия (независимо от профиля его деятельности). Поэтому можно вполне обоснованно утверждать, что в жесткой конкурентной борьбе большие шансы на победу имеет предприятие, использующее в управлении современные информационные технологии.

Наиболее важными задачами, решаемыми с помощью специальных программных средств, являются:

- бухгалтерский учет;
- управление финансовыми потоками;
- управление складом, ассортиментом, закупками;
- управление производственным процессом;
- управление маркетингом;
- документооборот;
- оперативное управление предприятием;
- предоставление информации о фирме.

2. ИНТЕРНЕТ-ТЕХНОЛОГИИ

В последнее время получают все более широкое развитие глобальные вычислительные и информационные сети – уникальный симбиоз компьютеров и коммуникаций. Идет активное использование всеми странами всемирных сетевых структур. Мировой системой компьютерных коммуникаций ежедневно пользуются более 500 млн чел. Возрастает потребность в средствах структурирования, накопления, хранения, поиска и передачи информации. Глобальные сети – это телекоммуникационные структуры, объединяющие локальные информационные сети, имеющие общий протокол связи, методы подключения и протоколы обмена данными. Крупнейшей глобальной

информационной сетью является Интернет. С точки зрения конечных пользователей Интернет представляет собой единую виртуальную сеть, к которой подсоединены компьютеры, – независимо от их реальных физических соединений. Фундаментальным принципом Интернета является равнозначность всех объединенных с его помощью физических сетей: любая система коммуникаций рассматривается как компонент Интернета независимо от ее физических параметров, размеров передаваемых пакетов данных и географического масштаба. Границы Интернета довольно расплывчаты. Любой компьютер, подключенный к нему, уже можно считать его частью. Web-серверы, на которых располагаются информационные ресурсы, могут находиться в любой части Интернета: у поставщика услуг, в различных локальных сетях и т. д., необходимо лишь соблюдение главного условия – они должны быть подключены к Интернету, чтобы пользователи сети могли получить доступ к их службам. В качестве служб могут выступать электронная почта, FTP, WWW и другие продукты, использующие Интернет как среду передачи информации.

2.1. Сервисы интернета

В настоящее время одной из самых распространенных стала технология компьютерного способа пересылки и обработки информационных сообщений, позволяющая поддерживать оперативную связь между всеми желающими. Такая технология получила название электронной почты.

Электронная почта. Электронная почта – специальный пакет программ для хранения и пересылки сообщений между пользователями Интернета. Посредством электронной почты реализуется служба безбумажных почтовых отношений. Она является системой сбора, регистрации, обработки и передачи любой информации (текстовых документов, изображений, цифровых данных,

звукозаписи и т. д.) по компьютерным сетям и выполняет такие функции, как редактирование документов перед передачей, их хранение в специальной базе, пересылка корреспонденции, проверка и исправление ошибок, возникающих при передаче, получение и хранение информации в собственном «почтовом ящике», просмотр полученной корреспонденции. При этом все компьютеры получателей подключены к ближайшему почтовому серверу, получающему, хранящему и пересылающему дальше по сети почтовые отправления, пока они не дойдут до адресата.

Телеконференции. Телеконференции – вторая по распространенности служба Интернета, предоставляющая информационные услуги.

Служба телеконференций состоит из множества тематических телеконференций – групп новостей, поддерживаемых серверами новостей. Сервер новостей – это компьютер, который может содержать тысячи групп новостей самых разнообразных тематик. Каждый сервер новостей, получивший новое сообщение, передает его всем узлам, с которыми он обменивается новостями. Доступ к группам новостей осуществляется через процедуру подписки, которая состоит в указании координат сервера новостей и выбора интересующих пользователя групп новостей. В обсуждении темы телеконференции может участвовать множество людей, независимо от того, где они находятся физически.

Наряду с описанной формой служб телеконференций широкое распространение получили WWW-телеконференции, также называемые форумами. Отличие состоит в том, что они работают через web-интерфейс и размещаются не централизованно на серверах новостей, а на web-сайтах.

Кроме служб распространения информации, востребованы службы Интернета, предоставляющие возможность общения, например чат.

Чат позволяет проводить текстовые дискуссии в режиме реального времени. От традиционной формы разговора их отличает то, что они ведутся в текстовом виде – путем набора текста на клавиатуре. Самым популярным открытым стандартом, лежащим в основе чатов, является IRC (*Internet Relay*

Chat). IRC – это многопользовательская, предназначенная для чата многоканальная сеть, с помощью которой пользователи могут беседовать в режиме реального времени независимо от своего месторасположения.

В свое время чаты, в основе которых лежал стандарт IRC, получили достаточно широкое распространение. Однако сегодня все более популярными становятся чаты, создаваемые на отдельных web-сайтах.

Служба FTP. Следующая популярная в прошлом служба Интернета – служба FTP. FTP (*File Transfer Protocol*) – протокол передачи файлов, но при рассмотрении FTP как службы Интернета имеется в виду не просто протокол, а именно служба доступа к файлам в файловых архивах. Одна из причин достаточно высокой ее популярности в прошлые годы – это огромное количество информации, накопленной в FTP-архивах за десятилетия эксплуатации компьютерных систем

Информационная система – WWW. Самая популярная служба Интернета – это служба прямого доступа к Интернету, позволяющая интерактивно взаимодействовать с представленной на web-сайтах информацией, – WWW (*World Wide Web*). Это самая современная и удобная служба Интернета. Ее идея основана на принципе гипертекстовой разметки, то есть способе совмещения текстовой информации с элементами ее разметки (тегами). Эта служба также может использовать всевозможные мультимедийные ресурсы (видео, аудио, графику, текст и т. д.) для представления информации. Контакт с сетью осуществляется по принципу клиент-сервер с использованием протокола передачи гипертекста (*Hyper Text Transfer Protocol* – HTTP). С помощью протокола HTTP служба WWW позволяет обмениваться документами в формате языка разметки гипертекста – HTML, который обеспечивает надлежащее отображение содержимого документов в браузерах пользователей. В HTML-документе каждый его элемент может являться ссылкой на другой документ или его часть, при этом любой документ может ссылаться как на документы на этом же сервере, так и на других серверах Интернета. Ссылки WWW могут указывать не только на

документы, свойственные службе WWW, но и на прочие службы и информационные ресурсы Интернета. Необходимо подчеркнуть, что Интернет и WWW – это не тождественные понятия. Узкое определение Интернета представляет его как взаимосвязь компьютерных сетей на базе семейства протоколов TCP/IP (*Transmission Control Protocol/Internet Protocol*), в пространстве которых становится возможным функционирование протоколов более высокого уровня, в том числе протокола передачи гипертекста (HTTP) – протокола *World Wide Web*, гипертекстового сервиса доступа к удаленной информации. Таким образом, WWW – это одна из служб Интернета, которая предлагает простой в использовании интерфейс и дает возможность пользователям, даже не слишком хорошо знающим компьютер, получать доступ к web-службам в любой части Интернета. Обычно любой текст представляется как одна длинная строка символов, которая читается в одном направлении. Гипертекстовая технология заключается в том, что текст представляется как многомерный, т.е. с иерархической структурой типа сети. Материал текста делится на фрагменты. Каждый видимый на экране компьютера фрагмент, дополненный многочисленными связями с другими фрагментами, позволяет уточнить информацию об изучаемом объекте и двигаться в одном или нескольких направлениях по выбранной связи.

3. ПРОЕКТИРОВАНИЕ WEB-СТРАНИЦ

На Web-страницах присутствуют не только текст и статические изображения, но и анимация, видео- и аудиоклипы. Внешний вид просматриваемых Web-страниц динамично изменяется при выполнении пользователем некоторых действий. Это делает Web-страницу сложным произведением искусства, создаваемым на стыке литературы, журналистики, изобразительного искусства, кино, телевидения, радио, фотоискусства.

Проектирование страниц или сайтов осложняется еще и тем обстоятельством, что одна и та же страница отображается по-разному разными браузерами и в окнах разного размера. Во многих случаях браузер самостоятельно «решает», как будет выглядеть загружаемая страница.

При проектировании сайта дизайнеру приходится обращать внимание на правильный выбор цвета шрифта и цвета фона. Специалистами художественной фотографии установлено, что глаз зрителя обращается прежде всего к светлым тонам снимка и к тем местам, где имеется наивысший тональный контраст. Целесообразно не перегружать страницу большим числом деталей. Это связано с тем, что человек за короткое время способен воспринять и запомнить не более 7–8 объектов.

Считается, что чем меньше размер объекта, тем больше должна быть его контрастность. Замечено, что четко выраженный фоновый рисунок на Web-странице повышает утомляемость глаз и снижает эффективность восприятия текста. Для исключения пестроты страницу рекомендуется создавать с использованием не более четырех основных цветов.

На всех страницах одного сайта рекомендуется использовать одинаковые панели навигации, размещенные в одинаковых местах. Каждая Web-страница сайта должна иметь ссылку на главную страницу сайта.

Контентом сайта принято называть его содержимое, информационное наполнение страниц сайта. Под контентом подразумевают тексты, но сюда также входят изображения, видео- и аудиофайлы, размещенные на сайте. Информация на сайте не менее важна, чем его дизайн, а иногда даже более. Большинство пользователей Интернета ищут в сети нужную им информацию, а не красивые сайты ни о чем.

Любой текст, любой элемент сайта, каждая его страница должна иметь цель. Бессмысленный или присутствующий на сайте по непонятной причине контент бесполезен, а иногда и вреден. Если страница создана без цели, а ее содержание неинтересно и не полезно посетителям сайта, она с высокой

степенью вероятности станет последней страницей, которую посетит пользователь на этом сайте (станет точкой выхода).

Итак, каким должен быть контент сайта (и прежде всего текст), чтобы он работал на своего владельца?

Правило 1. Текст должен быть написан интересно, привлекать внимание и не отпускать до самого конца.

Правило 2. Текст должен быть уникальным для поисковых систем.

Правило 3. Текст должен быть «читабельным» для посетителей сайта.

Правило 4. Текст должен «продавать», когда нужно продавать, информировать, когда нужно информировать, убеждать, когда нужно убеждать. В общем, должен достигать заданной цели, ни больше ни меньше.

Правило 5. Текст необходимо удобно размещать на странице, чтобы посетитель не путался.

Отличные тексты – одна из важнейших вещей, которая непременно проявится и в таком деле, как раскрутка интернет-ресурса. Именно такие тексты помогут увеличить конверсию, а следовательно, и эффективность сайта.

3.1. Логическая и физическая структура сайта

Каждый ресурс Интернета, от любительской домашней странички до большого информационного портала, содержит несколько тематических рубрик, соединенных между собой гиперсвязями. Как правило, ссылки на все разделы сайта с краткими анонсами их содержимого приводятся на первой, так называемой стартовой, странице, которой присваивается имя index.htm (index.html). Если тематические рубрики содержат собственные подразделы, каждая из них также имеет свою стартовую страницу, называющуюся index.html.

Подобный набор тематических рубрик с распределенными по соответствующим разделам документами и заранее спроектированными гиперсвязями между всеми страницами ресурса и называется *логической структурой* сайта. *Физическая структура*, напротив, подразумевает алгоритм размещения физических файлов по поддиректориям папки, в которой хранится опубликованный сайт. Пример сравнения логической и физической структур одного и того же ресурса Интернета показан на *рис. 3.1*.

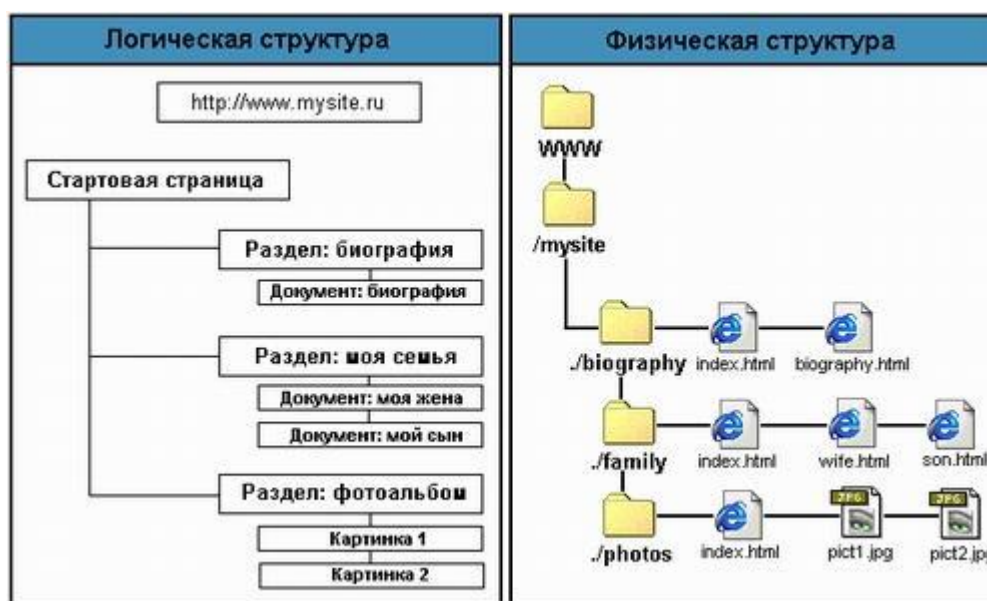


Рис. 3.1. Сравнение логической и физической структуры сайта

Очевидно, что логическая и физическая структуры могут не совпадать, поскольку в общем случае физическая структура ресурса разрабатывается, исходя из удобства размещения файлов.

Физическая структура сайта скрыта от посетителей ресурса: они могут наблюдать только логическую структуру, причем именно так, как она представлена при помощи элементов навигации. Отсюда следует вполне логичный вывод: строение системы навигации должно если не полностью повторять, то хотя бы максимально соответствовать разработанной вами логической структуре сайта.

Доступность информации на сайте в большей степени определяется его структурой. Чем легче получить на сайте полезную информацию, тем чаще на сайт будут заходить, тем больше рекомендаций получит сайт.

Структуры сайтов бывают 3-х видов:

- линейная;
- иерархическая;
- сетевая.

Линейная структура. При такой структуре (рис. 3.2) веб-страницы просматриваются по порядку, одна за другой. С текущей страницы можно перейти либо на следующую, либо на предыдущую. В чистом виде линейная структура не применяется для организации сайтов из-за неудобства пользования.

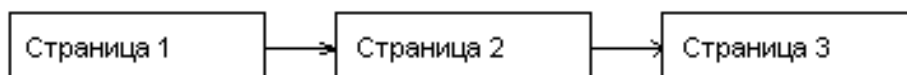


Рис. 3.2. Пример линейной структуры сайта

Иерархическая структура. Верхняя страница в иерархии (рис. 3.3) является обычно «домашней» страницей, предоставляющей обзор сайта со ссылками на основные тематические подразделения. Каждое из этих подразделений в свою очередь соединяется со страницами, содержащими дополнительные подробности по теме. Навигация направлена вниз по иерархии со ссылками возврата вверх по иерархии, которые проходят по тем же самым направленным вниз маршрутам, только в обратном направлении. Могут существовать редкие перекрестные ссылки между темами на одном уровне дерева. Такая структура задает логическую организацию сайта и позволяет посетителям найти информацию рациональным, направленным от общего к частному образом.

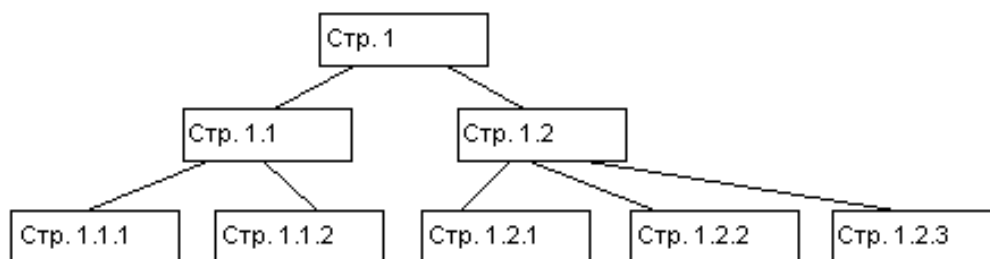


Рис. 3.3. Пример иерархической структуры сайта

Сетевая структура. Это наиболее распространенная в Интернете структура (рис.3.4). При такой организации сайта можно перейти с любой страницы на любую логически с ней взаимосвязанную, но совершенно необязательно связывать все страницы со всеми. Связи должны быть максимально естественными.

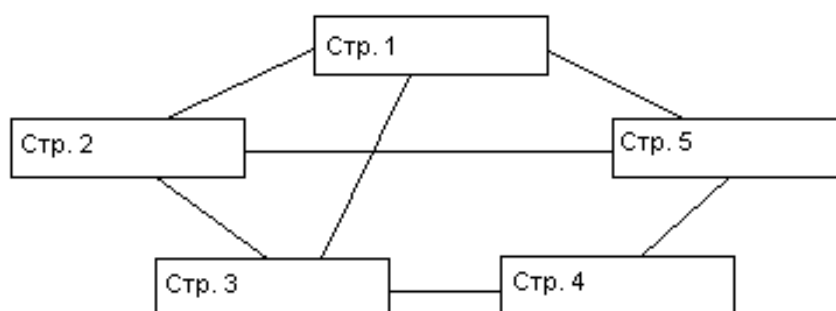


Рис. 3.4. Пример сетевой структуры сайта

Структура сайта реализуется с помощью:

- навигационных меню;
- гиперссылок с одних страниц на другие.

На сайте принято располагать несколько навигационных меню, как правило, основное и дополнительное (контекстное). В основном меню перечисляются крупные и самые важные разделы сайта. В дополнительных

меню размещаются ссылки на страницы со смежными темами и вопросами (отступления от основного содержания сайта).

Гиперссылки устанавливают возможность перехода с одной страницы сайта на другую, таким образом позволяя посетителю путешествовать в соответствии со своими целями и интересами не только по страницам своего сайта, но и переходить на другие сайты в информационном пространстве WWW.

Заглавная страница представляет собой html-документ, который не включает в себя какую-либо содержательную информацию и элементы навигации. Файлу заглавной страницы присваивается имя index.html, при этом стартовая страница называется иначе и вызывается посредством организации гиперссылки с заглавной страницы, загружающейся при обращении к сайту первой. Заглавная страница содержит, как правило, логотип компании – владельца данного ресурса, счетчик посещений и выбор между английской и русской версиями сайта.

При обращении к сайтам, не оснащенным заглавной страницей, первым отображается стартовый документ, включающий какое-либо информационное наполнение, элементы навигации и иногда анонсы составляющих данный ресурс тематических рубрик.

4. ВЕРСТКА WEB-СТРАНИЦ

Веб-страница запускается на компьютере под управлением клиентской программы, называемой браузером. Понятно, что операционная система, ее настройки и собственно сам браузер отличаются от компьютера к компьютеру. Из чего можно сделать вывод, что один и тот же документ сайта по-разному отображается у каждого пользователя.

Поэтому задача верстки веб-страниц формулируется так: сформировать документ, который бы корректно отображался с небольшими различиями на основных платформах и в браузерах.

Изначально разработчику сайта ширина окна браузера пользователя неизвестна, поскольку она может меняться в самых широких пределах. Ширина зависит от разрешения монитора, длины его диагонали, размера окна и еще некоторых варьируемых данных. Иными словами, предугадать ее заранее простыми средствами не представляется возможным. С учетом этой особенности утвердилось два способа верстки: фиксированный и «резиновый».

Фиксированный макет. В данном случае устанавливаем общую ширину макета жестко заданной и равной определенной величине.

Недостаток, который инкриминируют этому виду верстки, фактически один – недостаточно эффективное использование свободной площади.

«Резиновый» макет. Этот вид макета основывается на том, что в качестве одной из единиц измерения выступают проценты. Общая рабочая ширина окна браузера – 100 %, и колонки макета в сумме не должны ее превышать, поэтому для удобства, как правило, везде применяют процентную запись. При изменении размеров окна происходит переформатирование данных страницы, чтобы они вписались в новую ширину.

Этот вид верстки имеет большую популярность, и практически все известные сайты выбрали именно его в силу того, что эффективно задействуется вся площадь веб-страницы. Однако «резиновой» верстке присущи некоторые особенности и недостатки.

Хотя веб-страница и подстраивается под ширину окна браузера, при достижении некоторой величины читать текст становится неудобно – строки слишком длинные.

Верстать «резиновый» макет сложнее, чем аналогичный фиксированной ширины.

Любой макет имеет некоторую минимальную ширину, при достижении которой веб-страница «рассыпается» или появляется горизонтальная полоса прокрутки.

Исторически сложилось, что пролистывание большого документа на компьютере происходит сверху вниз. Для удобства листания предназначены вертикальные полосы прокрутки, клавиатурные комбинации, колесо прокрутки мыши. А вот перемещение по горизонтали происходит не так удобно, поэтому горизонтальной полосы прокрутки быть не должно.

Рисунки на веб-страницах применяются для иллюстрации текста и выполняют на сайте много различных ролей, например используются для создания привлекательного дизайна, служат распоркой между ячейками таблицы, создают градиентные заливки, фоновые изображения и т. д.

Фоновый рисунок удобен тем, что он может заполнять всю отведенную ширину или высоту под блоком. Это позволяет создавать линии или другие декоративные элементы, которые привязываются к ширине или высоте текста и не зависят от размеров окна. К тому же поверх фона можно накладывать текст, что также расширяет возможности дизайна веб-страниц.

Современные видеокарты поддерживают несколько произвольно устанавливаемых видеорежимов, характеризующихся экранном разрешением и количеством цветов, используемых при отображении информации на экране компьютера. При открытии в одном и том же браузере какой-либо веб-страницы она может отображаться неодинаковым образом в зависимости от используемого посетителем сайта экранного разрешения.

Для того чтобы избежать «съезжания» элементов html-документа друг относительно друга и, как следствие, деформации веб-страницы в целом при изменении параметров экрана, применяется простой и действенный прием: все компоненты веб-страницы заключаются в соответствующие ячейки невидимой таблицы, при этом каждому объекту назначается одно, строго определенное положение. Таким образом, появляется критерий, по которому можно разделить все существующие веб-сайты на две условные категории. Данной

таблице можно назначить строго определенную ширину в пикселях, после чего жестко позиционировать ее по центру экрана или "прижать" к левому его краю.

Для построения сложных таблиц можно применять вложение одной таблицы в другую. Такой подход позволяет дробить различные участки web-страницы независимо друг от друга. Такой вариант компоновки сайта называют *статическим*, поскольку ширина таблицы не меняется в зависимости от экранного разрешения. Здесь при изменении параметров экрана не происходит ни малейшего смещения элементов дизайна страницы.

Иной подход – когда ширину невидимой таблицы, содержащей фрагменты web-страницы, задают в процентах от текущей ширины экрана. При увеличении экранного разрешения таблица "растягивается" по горизонтали, и все размещенные в ее ячейках элементы, позиционированные либо по центру, либо по краям столбцов, смещаются согласно установленному алгоритму. В силу того, что параметры таблицы изменяются в зависимости от настроек экрана, такой принцип компоновки html-документа называют *динамическим*. И тот и другой подход обладает как достоинствами, так и недостатками.

Статическая компоновка страницы. *Достоинства.* Простота алгоритма верстки документа. Оптимизировав страницу для отображения с определенным экранным разрешением, вы можете быть уверены, что при изменении пользовательских экранных настроек элементы дизайна не "поплывут". Кроме того, данный вариант компоновки сайта в большинстве случаев (при соблюдении ряда дополнительных условий) позволяет добиться идентичности отображения html-документа в различных браузерах.

Недостатки. При отображении документа на компьютере с высоким экранным разрешением по краям экрана или с правой его стороны остается заметное пустое поле.

Динамическая компоновка страницы. *Достоинства.* Документ растягивается по всей ширине экрана, не остается неиспользованных пустых полей.

Недостатки. Сложность верстки и отладки страницы, весьма часто проявляется неадекватность отображения.

Любая web-страница содержит определенный набор стандартных элементов, являющихся обязательными компонентами каждого ресурса Интернета. Безусловно, ассортимент и количество подобных объектов могут варьироваться в зависимости от тематической направленности сайта, объема опубликованных на нем материалов, а также от целей и задач, которые ставит перед собой создатель данного ресурса. Компоновка таких элементов, проектирование их взаимного расположения и составляет одну из главных задач web-мастера.

Первым элементом web-страницы является ее *заголовок*. Он может быть выполнен как в текстовом, так и в графическом варианте, однако и в том и в другом случае он должен располагаться в верхней части документа. Иногда с заголовком совмещают кнопку для перехода с русскоязычной на англоязычную версию сайта, если данный web-ресурс представлен на двух языках. Непосредственно под заголовком документа, как правило, располагается пространство, отведенное для размещения рекламного баннера.

Основную часть документа занимает так называемое *текстовое поле* – участок, где и размещается смысловое наполнение страницы: текст и иллюстрации. Перечисленные элементы еще называют «*контент*» (от англ. *content* – содержание). Расположение текстового поля зависит в первую очередь от того, каким образом web-дизайнер разместит остальные элементы документа.

Следующей обязательной составляющей частью web-страницы являются элементы навигации – *гиперссылки*, связывающие данный документ с другими разделами сайта. Элементы навигации могут быть выполнены в виде текстовых строк, графических объектов, то есть кнопок, либо активных компонентов, например Java-апплетов. Последние представляют собой те же кнопки, которые, в отличие от своих «традиционных» сестер, умеют реагировать на движения мыши, выполняя при наведении на них курсора какие-либо

несложные действия (включение подсветки, создание эффекта «нажатия», изменение формы и т. д.). Располагать элементы навигации следует таким образом, чтобы они всегда были «на виду», «под рукой», то есть так, чтобы пользователю не приходилось «отматывать» страницу назад, если текстовое поле занимает по высоте несколько физических экранов, после чего подолгу искать ссылки на другие разделы.

В нижней части документа принято публиковать информацию о разработчиках сайта и адрес электронной почты, по которому посетители ресурса могут направить владельцам странички свои отклики, предложения и пожелания.

Элементы навигации можно разместить не только вблизи правой или левой границ страницы, но и в верхней части документа. Такой вариант компоновки наиболее подходит при создании домашних страничек: в этом случае все объекты страницы гармонично «вписываются» в заданную ширину невидимой таблицы, при этом подготовка самой таблицы значительно упрощается. Единственным недостатком подобного подхода является необходимость продублировать элементы навигации в нижней части документа, поскольку при вертикальной прокрутке страницы они исчезают за верхней границей экрана, и, чтобы добраться до них, пользователю придется «отматывать» экран назад, но окончательное решение всегда остается за web-мастером. В конечном счете, какой бы дизайн вы не заложили в основу своего будущего проекта, результат ваших трудов все равно будет правильным: в Интернете нет ни цензуры, ни каких-либо регламентов, загоняющих создателя сайта в те или иные жесткие рамки. Важно лишь, чтобы внешний вид сайта не вызывал нареканий у посетителей. В конце концов, только вы как разработчик вправе проявить все свои способности и таланты и скомпоновать страницу по собственному вкусу.

5. ОСНОВНЫЕ ПОНЯТИЯ HTML

HTML (*Hyper Text Markup Language*) – язык разметки гипертекста был разработан Тимом Бернерсом-Ли в рамках реализации проекта создания распределенной гипертекстовой системы, которую он назвал *World Wide Web* (WWW), или Всемирная паутина, и предназначен для написания гипертекстовых документов, публикуемых в *World Wide Web*.

Документ на языке HTML может включать следующие компоненты:

- стилизованный и форматированный текст;
- команды включения графических, звуковых и видео- файлов;
- гиперсвязи с различными ресурсами Интернета;
- скрипты на языках *JavaScript* и *VBScript*;
- различные объекты, например Flash-анимацию.

Документы HTML являются обычными текстовыми файлами, содержащими специальные *теги* (управляющие элементы) разметки. Теги разметки указывают web-браузеру (программе для отображения web-страниц), как надо вывести страницу на экран.

Тэги – это зарезервированные слова. Все тэги HTML начинаются с символа < (левой угловой скобки) и заканчиваются символом > (правой угловой скобки).

Как правило, существует стартовый тэг и завершающий тэг. Завершающий тэг выглядит так же, как стартовый, он отличается от него прямым слешем перед текстом внутри угловых скобок. Некоторые тэги не требуют завершающего тэга. HTML не реагирует на регистр символов, но теги рекомендуется писать маленькими буквами.

Файлы HTML обычно имеют расширения *htm* или *html*. Их можно создавать при помощи любого специализированного текстового редактора.

Основное преимущество HTML заключается в том, что документ может быть просмотрен на web-браузерах различных типов и на различных платформах.

Для представления текста в браузере необходимо создать файл с расширением .HTML, в который будет записан этот текст.

Каждый HTML-документ содержит три основных обязательных раздела: HTML, HEAD и BODY. Эти разделы задаются парами тегов: <HTML> и </HTML> , <HEAD> и </HEAD>, <BODY> и </BODY>.

5.1. Базовая структура HTML-документа

<HTML>

<HEAD>

<TITLE>Название документа </TITLE>

</HEAD>

<BODY>

Тело документа – все, что мы видим на странице.

</BODY>

</HTML>

Раздел HTML является внешним, между его начальным и конечным тегами должна находиться вся web-страница.

HEAD – это раздел, в котором помещается название web-страницы.

BODY – один из самых важных разделов любого HTML-документа, в нем располагается содержательная часть, которая выводится браузером на экран монитора пользователя. Внутри раздела BODY можно использовать все элементы, предназначенные для дизайна web-страницы.

Слова, заключенные в угловые скобки, являются служебными словами языка HTML. Их принято называть *тегами (tags)*.

Теги никогда не отображаются на экране монитора пользователя – они служат для управления оформлением страницы.

Тег `<HTML>` в первой строке означает, что текст действительно написан на языке HTML.

Большинство тегов языка HTML – парные. Поэтому в первой строке стоит открывающий тег `<HTML>`, а в последней – закрывающий тег `</HTML>`. Все, что расположено между ними, считается HTML-документом.

В языке HTML совершенно не имеет значения, строчными или прописными буквами записаны теги. Можно написать `<HTML>` или `<html>` – для браузера это одно и то же.

Между парными тегами `<HEAD>` и `</HEAD>` расположено название web-страницы. Оно оформлено парными тегами `<TITLE>` и `</TITLE>`, между которыми расположена служебная текстовая информация.

Далее находится парный тег `<BODY>`. Все, что расположено между ним и его закрывающим тегом `</BODY>`, считается телом документа и отображается на экране. Теги BODY имеют ряд атрибутов, которые условно можно разделить на четыре основные группы: фон, границы документа, текст и гиперссылки. Если эти атрибуты не указывать, то соответствующие параметры страницы – отступы, цвета и др. – будут такими, какие заданы по умолчанию.

В теле находится само содержимое документа: оригинальный текст и встроенные теги – инструкции о структуре и внешнем виде содержимого.

5.2. Создание web-страницы

1. Откройте программу Блокнот (Пуск > Программы > Стандартные > Блокнот).

2. Напишите:

`<HTML>`

<HEAD>

<TITLE> Страница №1 </TITLE>

</HEAD>

<BODY>

Учебная страница студента _____ (ФИО студента).

</BODY>

</HTML>

3. Сохраните этот файл с расширением имени .html. (Меню> Файл>Сохранить, в окне «Тип файла» в выпадающем списке выбрать строку «Все файлы», в окне «Имя файла» вписать «ПЕРВАЯ СТРАНИЦА.html»).

4. Вы создали основу для первой странички сайта. Для того чтобы ее просмотреть, просто откройте файл. Страница будет загружена в браузере, установленном по умолчанию на вашем компьютере.

5.3. Элементы форматирования web-страницы

Структурное форматирование текста, расположенного между тегами <BODY> и </BODY>, подразумевает разбиение текстовых фрагментов электронного документа на логические блоки, которым соответствует определенный формат: абзац, текстовый блок, центрирование, отступы и перенос строки, горизонтальный разделитель, предварительно отформатированный текст и комментарии.

Рассмотрим каждый из названных элементов форматирования.

5.3.1. Создание заголовков

Для того чтобы выделять заголовки, в HTML существуют так называемые теги заголовков. Это теги <H1>, <H2>, <H3>, <H4>, <H5> и <H6> (и их закрывающие теги). Для самых крупных заголовков можно использовать тег <H1>, для заголовков помельче – <H2> и так далее. Например, так:

<H2>ИВАН-ЦАРЕВИЧ И СЕРЫЙ ВОЛК - сказка</H2>

5.3.2. Создание абзацев

Для обозначения абзаца используется специальный тег <P>, который разделяет фрагменты текста отступом.

Тег <P> может содержать атрибут ALIGN, отвечающий за тип горизонтального выравнивания текста в окне браузера, который может принимать ряд значений:

- ALIGN="left" – текст выровнен по левому краю;
- ALIGN="center" – текст располагается посередине окна браузера;
- ALIGN="right" – выравнивание текста по правому краю;
- ALIGN="justify" – выравнивание по ширине окна браузера.

Например:

<P ALIGN="CENTER">Текст абзаца </P>.

Переход на новую строку осуществляет тег
. В том месте, где он стоит, происходит принудительный переход на новую строку (причем в исходной записи можно продолжать текст на этой же строке – это не имеет значения.) Тег
 не имеет закрывающего парного тега.

5.3.3. Создание горизонтальной черты

Для создания горизонтальной черты достаточно в нужном месте поставить тег <HR> (он не имеет закрывающего тега). Однако в этом случае горизонтальная черта займет всю ширину страницы. Определение ширины горизонтальной черты можно задать атрибутом WIDTH, например так:

```
<HR WIDTH="75%">.
```

В этом случае горизонтальная линия займет 75% от полной ширины экранной страницы. Можно также определять ширину линии и в пикселях. Например, запись <HR WIDTH="275"> определит ширину линии в 275 пикселей. Задать толщину линии в пикселях можно, добавив атрибут SIZE.

```
<HR WIDTH="75%" SIZE="50">.
```

5.3.4. Задание №1

Разработать код HTML-документа, который выводит на экран следующий контент:

Учебная страница студента _____ (ФИО студента)

Задание №1

Вывести заголовок с произвольным содержанием текста, используя тег <H1>.

Вывести заголовок с произвольным содержанием текста, используя тег <H5>.

Создать горизонтальную черту толщиной 50 пикселей, занимающую 75% ширины экрана и расположенную по центру экрана.

5.3.5. Создание списков

Списки бывают маркированные и нумерованные.

Для организации маркированного списка применяется тег `<UL>`. Все, что находится между ним и его закрывающим тегом `</UL>`, считается маркированным списком. Каждый элемент списка должен быть при этом обозначен тегом `<LI>`. Этот тег можно употреблять без закрывающего, хотя можно и с ним.

Например:

```
<UL> Услуги
<li>Вкручивание электрических лампочек
<li>Подметание пола
<li>Дефрагментация жестких дисков
</UL>
```

Каждый элемент маркированного списка будет при просмотре отмечаться закрашенным кружком.

Для организации нумерованного списка применяется тег `<OL>`, а элементы списка также обозначаются тегом `<LI>`.

И нумерованные, и маркированные списки в большинстве браузеров выделяются небольшим отступом. Благодаря тому, что списки отображаются с отступом, легко можно организовывать вложенные списки с помощью тех же тегов. Для этого надо начать новый список внутри уже начатого. Схема расположения тегов списка при этом получится примерно такая:

```
<UL>
<OL>
</OL>
</UL>
```

При вложении нескольких маркированных списков разумно каждый из них обозначить своим типом маркера. Для задания типа маркера в теге `<UL>`

следует установить атрибут TYPE. У него могут быть три значения: "disc", "square" и "circle", что означает, соответственно, закрашенный кружок, квадратик и незакрашенный кружок.

В теге нумерованного списка можно установить атрибут TYPE для определения типа нумерации. Если не указано ничего или установлено значение TYPE="1", то нумерация происходит обычными цифрами. Если установить TYPE="I" или "i", то получится нумерация римскими цифрами (соответственно большими или маленькими). Для буквенных обозначений элементов списка устанавливают атрибут TYPE="A" или "a" (соответственно с использованием прописных или строчных букв). Кроме того, иногда может потребоваться начать нумерацию не с единицы, а с какого-нибудь иного числа. Для этого существует атрибут START. Например, запись <OL START="43"> вызовет нумерацию элементов списка, начиная с числа 43. Далее пойдут элементы с номерами 44, 45 и т. д.

5.3.6. Задание №2

Разработать код HTML-документа, который выводит на экран следующий контент:

Учебная страница студента _____ (ФИО студента).

Задание №2

Вывести на экран следующий список, используя атрибут TYPE со значением «1»:

ФРУКТЫ:

1. Яблоки
2. Груши
3. Сливы

Вывести на экран следующий список, используя атрибут TYPE со значением «I»:

ФРУКТЫ:

I.Яблоки

II.Груши

III.Сливы

5.3.7. Создание таблиц

Любая таблица определяется в HTML с помощью тега <TABLE>. Все, что расположено между тегами <TABLE> и </TABLE>, считается таблицей. Строки таблицы создаются с помощью тега <TR> и его закрывающего тега, а ячейки – с помощью тега <TD> и его закрывающего тега.

По умолчанию таблицы обычно отображаются без рамки. Чтобы таблица имела рамку, необходимо задать ее ширину, установив атрибут BORDER. С помощью атрибута WIDTH можно устанавливать любую желаемую ширину таблицы.

Например, если мы напишем <TABLE WIDTH="50" BORDER="3">, то браузер отобразит таблицу шириной в 50 пикселей с рамкой толщиной в 3 пикселя. А если установить WIDTH="50%", то таблица растянется на половину ширины окна браузера.

Для создания цветной рамки таблицы служит атрибут BORDERCOLOR, которому можно дать в качестве значения любой доступный цвет. Цвет фона таблицы можно задать с помощью атрибута BGCOLOR.

5.4. Шрифтовое оформление web-страницы

5.4.1. Основные теги форматирования текста

Теги форматирования текста web-документа имеют следующий вид:

`<B> </B>` – отображение полужирным шрифтом;

`<I> </I>` – отображение курсивом;

`<U> </U>` – отображение подчеркнутым шрифтом.

`<BIG></BIG>` – отображение текста шрифтом большего, чем непомеченная часть текста, размера;

`<SMALL></SMALL>` – отображение текста шрифтом меньшего размера;

`<SUB></SUB>` – сдвиг текста ниже уровня строки и вывод его (если возможно) шрифтом меньшего размера;

`<SUP></SUP>` – сдвиг текста выше уровня строки и вывод его (если возможно) шрифтом меньшего размера.

`<FONT></FONT>` – один из основных тегов физического форматирования текста, отображающий свойства шрифтов. Для него могут использоваться следующие атрибуты:

1. FACE – служит для указания типа шрифта, которым программа просмотра пользователя будет выводить текст. Значение параметра – название шрифта, которое должно в точности совпадать с названием шрифта, имеющегося у пользователя. Если такого шрифта не будет найдено, то данное указание будет проигнорировано и будет использован шрифт, установленный по умолчанию. Например, `<FONT FACE="Verdana", "Arial", "Helvetica">`.

2. SIZE – служит для указания размеров шрифта в условных единицах от 1 до 7. Размер шрифта указывается абсолютной величиной, например `<FONT SIZE="2">`. Если написать `<FONT SIZE="+1">` текст `</FONT>`, то весь текст, оказавшийся между этими тегами, будет отображен шрифтом на один пункт

крупнее обычного. Как правило, базовый шрифт имеет размер 3 (по умолчанию), если не определено иное с помощью тега <BASEFONT> (например, так: <BASEFONT SIZE="6">).

3. COLOR – служит для указания цвета шрифта. Существует две формы задания цвета: символьная (указывается название одного из предопределенных цветов) и цифровая (комбинация RGB: #RRGGBB).

Символьные идентификаторы основных цветов и их RGB-комбинации приведены ниже:

- BLACK (#000000) – черный;
- SILVER (#C0C0C0) – серебряный;
- GRAY (#808080) – серый;
- WHITE (#FFFFFF) – белый;
- MAROON (#800000) – темно-красный;
- RED (#FF0000) – красный;
- PURPLE (#800080) – темно-сиреневый;
- FUCHSIA (#FF00FF) – сиреневый;
- GREEN (#008000) – зеленый;
- LIME (00FF00) – ярко-зеленый;
- OLIVE (#808000) – оливковый;
- YELLOW (#FFFF00) – желтый;
- NAVY (#000080) – темно-синий;
- BLUE (#0000FF) – голубой;
- TEAL (#008080) – сине-зеленый;
- AQUA (#00FFFF) – ярко-голубой.

Например, Текст будет написан красным цветом .

5.4.2. Задание №3

Разработать код HTML-документа, который выводит на экран следующий контент:

Учебная страница студента _____ (ФИО студента)

Задание №3

Вывести заголовок со следующим содержанием текста: Шрифтовое оформление web-страницы.

Вывести на экран произвольный текст из 10 строк с размером 7 и шрифтом *Arial*. Цвет символов – зеленый.

5.5. Цветовое оформление web-страницы

Для этого проще всего установить соответствующие атрибуты тега <BODY>. Атрибут TEXT определяет цвет текста на всей странице, а атрибут BGCOLOR – цвет фона. Названия цветов можно задавать либо символьно, либо цифровой комбинацией RGB.

Например, <BODY BGCOLOR="#BABAAO" TEXT="#1D1D18">.

Иногда при создании web-страниц необходимо упорядочить представленную на них информацию. В языке HTML для этого применяют списки и таблицы.

5.5.1. Задание №4.

Разработать код HTML-документа, который выводит на экран следующий контент:

Учебная страница студента _____ (ФИО студента)

Задание №4

Вывести заголовок со следующим содержанием текста: Цветовое оформление web-страницы.

Используя атрибут BGCOLOR, сформировать цвет фона страницы (например желтый).

Вывести на экран следующий список, используя атрибут TYPE со значением «disc»:

ФРУКТЫ:

- Яблоки
- Груши
- Сливы

Вывести на экран произвольный текст из 5 строк с размером 5 и шрифтом *Times New Roman*. Цвет символов – красный.

5.6. Гиперссылки

Гипертекстовые связи (ссылки) являются наиболее важным элементом web-страниц. С их помощью делают документ связанным и структурированным, что позволяет пользователю получать необходимую ему информацию максимально быстро и удобно. Они являются основой построения сайтов в виде наборов связанных между собой страниц в *World Wide Web*.

Ссылка состоит из двух частей. Первая из них – это то, что видно на web-странице; она называется указателем ссылки (*anchor*). Вторая часть, дающая инструкцию браузеру, называется адресной частью ссылки (URL-адрес). Когда происходит щелчок мышью по указателю ссылки, браузер загружает документ, адрес которого дается URL-адресом.

Ссылка реализуется тегом <A>. Тег <A> имеет единственный параметр HREF, значением которого является URL-адрес. Этот тег является парным, поэтому необходимо поставить закрывающий тег .

Текстовый указатель ссылки

В текст web-страницы можно вставлять гиперссылки трех типов:

- ссылки в другие места той же страницы (внутренние);
- ссылки на другие страницы этого же сайта или сервера;
- ссылки на страницы, расположенные на любом другом сервере в

Интернете.

Ссылки на разные части текущего документа. Большой документ читается лучше, если он имеет оглавление со ссылками на соответствующие разделы. Для построения внутренней ссылки сначала нужно создать указатель, определяющий место назначения. Например, если необходимо сделать ссылку на текст определенной главы документа, нужно разместить там указатель и дать ему имя при помощи параметра NAME тэга <A>. При этом параметр HREF не используется.

Пример:

Необходимо обратить внимание, что в приведенном примере отсутствует содержимое тэга <A>. Обычно именно так и делают, поскольку здесь нет необходимости как-то выделять текст, а требуется лишь указать местоположение.

После того как место назначения определено, можно приступить к созданию ссылки на него. Для этого вместо указания в параметре HREF адреса

документа, как это делалось ранее, поместим туда имя ссылки с префиксом #, говорящим о том, что это внутренняя ссылка.

Пример:

```
<A HREF="#glava_5">Глава 5</A>
```

Теперь, если пользователь щелкнет кнопкой мыши на словах «Глава 5», браузер выведет соответствующую часть документа в окне просмотра.

Гиперссылки в пределах сайта. Обычно бывает целесообразно разместить разные логические фрагменты текста в разных файлах (особенно если они большие). Чтобы установить гиперссылку на другой файл, необходимо в качестве значения параметра HREF указать имя этого файла:

```
<A HREF="rasskaz.html"> Рассказ можно прочитать здесь</A>
```

Такие гиперссылки можно устанавливать только на файлы, содержащиеся в каталоге сайта на сервере.

Внешние гиперссылки. Однако гиперссылки можно устанавливать не только на файлы, содержащиеся в каталоге сайта на сервере, но и на любые другие Интернет-ресурсы. В этом случае в качестве значения параметра HREF необходимо указать полный URL-адрес ресурса.

Например:

```
Поисковая машина <A HREF = "http: // www.yandex.ru " > "Yandex" </A>.
```

5.7. Графическое оформление web-страницы

5.7.1. Фоновое изображение

Чтобы поместить рисунок на web-страницу в качестве фона, нужно установить в теге <BODY> атрибут BACKGROUND со значением, содержащим имя файла рисунка.

Например: <BODY BACKGROUND="img1.jpg">.

5.7.2. Графические иллюстрации

Их можно вставить в текст с помощью тега . Например: . На месте этой записи браузер отобразит нашу картинку.

Чтобы браузер мог корректно загружать текст параллельно с картинкой, ему нужно указать размеры будущей картинки с помощью атрибутов WIDTH и HEIGHT определяющих ширину и высоту картинки на экране в пикселах:

.

Вокруг рисунка можно отобразить рамку. Толщина этой рамки задается значением атрибута BORDER:

Тег может иметь собственный атрибут ALIGN, который управляет расположением рисунка относительно текста. Если рисунок встроен в текст, то есть несколько вариантов его расположения:

- ALIGN="left" – текст обтекает рисунок, который располагается слева;
- ALIGN="right" – текст обтекает рисунок, который располагается справа;
- ALIGN="top" – рисунок располагается внутри текстовой строки, и его верхняя граница совпадает с верхней границей текста;
- ALIGN="middle" – рисунок располагается внутри текстовой строки, которая вертикально выравнивается по центру;
- ALIGN="bottom" – рисунок располагается внутри текстовой строки, и его нижняя граница совпадает с нижней границей текста.

Кроме того, чтобы рисунок не слишком «прижимался» к тексту, можно оставить между ними немного «воздуха». Для этого в теге используют

атрибуты HSPACE и VSPACE, значения которых задают соответственно горизонтальное и вертикальное поля отступа рисунка от текста.

5.7.3. Задание №5

Разработать код HTML-документа, который выводит на экран следующий контент:

Учебная страница студента _____ (ФИО студента)

Задание №5

Вывести заголовок со следующим содержанием текста: Графическое оформление web-страницы.

Вывести на экран произвольный текст из 15 строк с шрифтом *Times New Roman*. Цвет символов – черный.

Вставить в тексте в произвольном месте изображение размером 200 x 200 пикселей с рамкой толщиной 4 пикселя.

6. КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ (CSS)

Дизайн web-узлов – это точное размещение компонентов HTML-страниц относительно друг друга в рабочей области окна браузера.

К компонентам страницы относятся: блоки текста, графика и встроенные приложения. Размер и границы каждого из этих компонентов в рамках HTML-разметки задаются с разной степенью точности. Размер графики и приложений можно задать с точностью до пикселя. Размеры текстовых блоков в HTML задать нельзя: они вычисляются браузером исходя из относительного размера

шрифта по умолчанию. Автор страницы не может заранее определить настройки браузера пользователя, что существенно ограничивает число вариантов представления информации на странице.

CSS (*Cascading Style Sheets*) предоставляют средства оформления, которые позволяют более точно следовать правилам форматирования и отделять структуру страницы от ее визуального представления. Они позволяют назначить собственный стиль визуального представления любому тегу HTML. Другой способ управления настройками браузера – программирование на *JavaScript*.

Каскадные таблицы стилей призваны разрешить противоречие между точностью определения размеров картинок и приложений, с одной стороны, и точностью определения размеров блоков текста и его начертания – с другой.

Таблицы стилей также позволяют определить цвет и начертание текстового фрагмента, изменять эти параметры внутри текстового блока, выполнять выравнивание текстового блока относительно других блоков и компонентов страницы.

Наличие подобных возможностей позволяет говорить о CSS как о средстве разделения логической структуры документа и формы его представления. Логическая структура документа определяется элементами HTML-разметки, в то время как форма представления каждого из этих элементов задается CSS-описанием элемента.

CSS позволяет полностью переопределить форму представления элемента разметки по умолчанию.

Различают четыре способа применения стилей:

- переопределение стиля в элементе разметки;
- размещение описания стиля в заголовке документа в элементе `STYLE`;
- размещение ссылки на внешнее описание через элемент `LINK`;
- импорт описания стиля в документ.

Рассматривая реализацию первого способа применения стилей, можно добавить в тег `<body>` атрибут `style`, и он сформирует стиль для всех элементов (абзацев, заголовков и т. д.), помещенных внутри этого тега-контейнера, в случае отсутствия собственных стилей для этих элементов. Таким образом, нам уже не нужно прописывать теги `<font>` и атрибуты `color`, `size` и т. п. для каждого абзаца на странице – достаточно задать некий стиль для тега `<body>`, и все абзацы на странице будут отображены в соответствии с этим стилем.

Например, чтобы все абзацы отображались шрифтом *Times New Roman* размером 12 пунктов темно-зеленого цвета, следует прописать атрибут `style` тега `<body>`, присвоив ему соответствующее значение:

```
<body style="font-family: Times New Roman; font-size: 12pt; color: darkgreen;">
```

Если необходимо, чтобы все абзацы на странице выглядели, как в предыдущем примере, а все заголовки первого уровня отображались шрифтом *Arial* зеленого цвета полужирного начертания размером 16 пунктов, то нам понадобится создать в заголовке страницы (между тегами `<head>` и `</head>`) внедренную таблицу стилей, в которой мы пропишем несколько правил сразу. Для этого создаем тег-контейнер таблицы стилей, начинающийся открывающим тегом `<style>` и заканчивающийся закрывающим тегом `</style>`. Внутри этого тега-контейнера можно задать любое количество правил CSS, состоящих из селектора (названия тега HTML, к которому будет применяться правило) и его определения (непосредственно набора средств форматирования), заключенного в фигурные скобки. Синтаксис для приведенного выше примера такой:

```
<head>
...
<style>
body { font-family: Times New Roman; font-size: 12pt; color: darkgreen;}
h1 {font-family: Arial; font-size: 16pt; color: green; font-weight: bold;}
</style>
```

</head>

Как видно из приведенного примера, использование CSS дает возможность существенно сократить объем кода.

7. ПРАКТИЧЕСКОЕ ЗАДАНИЕ

1. Выполнить задания 1-5.
2. Разработать эскизы web-страниц личного сайта студента. Общее количество страниц три и более. На главной странице расположить текстовую информацию и графическую информацию об авторе сайта и главное меню сайта. На следующих страницах отразить информацию о будущей профессии студента, спортивные достижения, увлечения и т. д .
3. Связать все разработанные страницы сайта гиперссылками.
4. Проверить работоспособность сайта в различных браузерах.

7.1. Основные действия студента по созданию HTML-страниц личного сайта

1. Создать HTML-документ с использованием какого-либо обычного (Блокнот) или специализированного текстового редактора. В соответствии с правилами оформления HTML-страниц наполнить страницу соответствующей информацией. Сохранить первый документ (главную страницу сайта) под названием «index.html» в отдельную папку с названием «SITE». Файлы последующих страниц сайта, присваивая им соответствующие тематические имена, и изображения для сайта сохранять также в папку «SITE».

2. Задать заголовок, отображаемый в строке заголовка окна браузера. Текст строки заголовка — фамилия студента. Использовать при этом тэг <TITLE>.

3. Написать текст подходящим шрифтом. В данном случае следует использовать тэги-контейнеры, отвечающие за размер шрифта. Произвести форматирование текста. Осуществить выравнивание различных абзацев (отцентрировать, прижать вправо, влево, выравнивать по ширине). Добавить тематические изображения.

4. Установить гиперссылки для связи с другими страницами сайта.

5. Проверить функционирование сайта активацией файла «index.html» в папке «SITE».

УЧЕБНАЯ ЛИТЕРАТУРА

1. Золотов, С. Ю. Проектирование информационных систем [Электронный ресурс]: учеб. пособие / С. Ю. Золотов. – Электрон. текстовые данные. – Томск: Томский гос. ун. систем управления и радиоэлектроники, Эль Контент, 2015. – 88 с. – Режим доступа: <http://www.iprbookshop.ru/13965>. – ЭБС «IPRbooks», по паролю.

2. Балдин, К. В. Информационные системы в экономике [Электронный ресурс]: учебник / К. В. Балдин, В. Б. Уткин. – Электрон. текстовые данные. – М.: Дашков и К, 2015. – 395 с. – Режим доступа: <http://www.iprbookshop.ru/24785>. – ЭБС «IPRbooks», по паролю.

3. Мешкомаев, В. Г. Разработка и стандартизация программных средств и информационных технологий [Электронный ресурс] / В. Г. Мешкомаев. – Электрон. текстовые данные. – СПб.: ФГБОУ ВПО «СПГУТД», 2015. – 26 с. – Режим доступа: <http://publish.sutd.ru>

4. Муссиано, Ч. HTML и XHTML. Подробное руководство / Ч. Муссиано, Б. Кеннеди. – 6-е изд. – СПб.: Символ-Плюс, 2012. – 752 с.

5. Гаевский, А. Ю. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А. Ю. Гаевский. – М.: Триумф, 2008. – 464 с.

6. Соловьёв, Д. Как создать интернет-магазин без правил / Д. Соловьёв, А. Писарев. – СПб: Издательский дом «Питер», 2013. – 220 с., 63 ил.