

тема 1.2

- типы адресов сети интернет
 - структура назначение классы ip
 - использование масок при адресации
- Стек протоколов TCP/IP
- Система доменных имён DNS.
 - Ключевые характеристики DNS
 - Домены верхнего уровня.
 - Принципы функционирования доменной системы DNS.
 - Программная поддержка системы DNS.
- Система универсальных идентификаторов ресурсов URI и URL.
 - Принципы системы: расширяемость, полнота, читаемость.
 - Формат URL.
- Структура стеков TCP/IP.
 - Стек TCP/IP и модель OSI.
 - Инкапсуляция протоколов верхнего уровня в протоколы стека TCP/IP.
 - Структура пакетных TCP.

тема 1.3 Сервисы глобальной сети интернет

- типы сервисов глобальной сети
 - программы клиенты
 - браузеры
- веб сервис
 - архитектура www
 - основные компоненты www(htm url http cgi)
 - принципы работы клиент - сервис
 - назначение web технологий и роль www
 - технологии поддерживаемые сервером
 - технологии поддерживаемые клиентом

1.4

- Понятие гипертекста.
- Организационная структура информации Web-узла (иерархическая, линейная, в виде паутины).
- Модели проектирования Web-узла.

Спиральная модель

- Microsoft Solutions Framework (MSF)
- Модель Уолта Диснея
- Содержание этапов проектирования Web-узла: концептуальное, логическое, физическое проектирование.

Концептуальное проектирование

- На этом этапе следует описать следующее:
 - Логическое проектирование
 - На этом этапе следует описать следующее:
 - Физическое проектирование
 - На этом этапе следует описать следующее:
- Хостинг Web-узла.
- Программное обеспечение рабочего места Webразработчика
 - Необходимые программы:
 - Вспомогательные программы:
 - Необходимые программы Html-редактор
 - Macromedia Home Site 5
 - Графический редактор Adobe Photoshop
 - Web-сервер

2.1 разметка языков гипертекста

- ПОНЯТИЕ О СТАНДАРТНОМ ОБОБЩЕННОМ ЯЗЫКЕ РАЗМЕТКИ SGML. КОНСОРЦИУМ W3C.
 - История возникновения
 - Структура консорциума

Особенности внедрения рекомендаций

Рабочий проект (WD — Working Draft)

Кандидат в рекомендации (CR — Candidate Recommendation)

Предложение в рекомендации (PR — Proposed Recommendation)

Рекомендация W3C (REC — W3C Recommendation)

Последующие изменения

ВЕРСИИ ЯЗЫКА ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ HTML. ПОНЯТИЕ О РАСШИРЯЕМОМ ЯЗЫКЕ РАЗМЕТКИ XML.

Типы структур HTML и XHTML

2.2

Основные понятия

Первое правило

Общая структура HTML-файла

Заголовки, абзацы, разрывы строк

Заголовки

Это заголовок второго уровня

Это заголовок третьего уровня

Это заголовок четвертого уровня

Абзацы

Разрывы строк

Предварительно отформатированный текст

Выделенный текст

тема 1.2

типы адресов сети интернет

Типы адресов:

1. Физический (MAC-адрес)
2. Сетевой (IP-адрес)
3. Символьный (DNS-имя)

Компьютер в сети TCP/IP может иметь адреса трех уровней (но не менее двух):

- Локальный адрес компьютера. Для узлов, входящих в локальные сети - это MAC-адрес сетевого адаптера. Эти адреса назначаются производителями оборудования и являются уникальными адресами.
- IP-адрес, состоящий из 4 байт, например, 109.26.17.100. Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и маршрутизаторов.
- Символьный идентификатор-имя (DNS), например,

структура назначение классы ip

IPv4**** - адрес является уникальным 32-битным идентификатором IP-интерфейса в Интернет.

IPv6**** - адрес является уникальным 128-битным идентификатором IP-интерфейса в Интернет, иногда называют **Internet-2**, адресного пространства IPv4 уже стало не хватать, поэтому постепенно вводят новый стандарт.

IP-адреса принято записывать разбивкой всего адреса по октетам (8), каждый октет записывается в виде десятичного числа, числа разделяются точками. Например, адрес

использование масок при адресации

Маска — это число, которое используется в паре с IP-адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Поскольку номер сети является цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность.

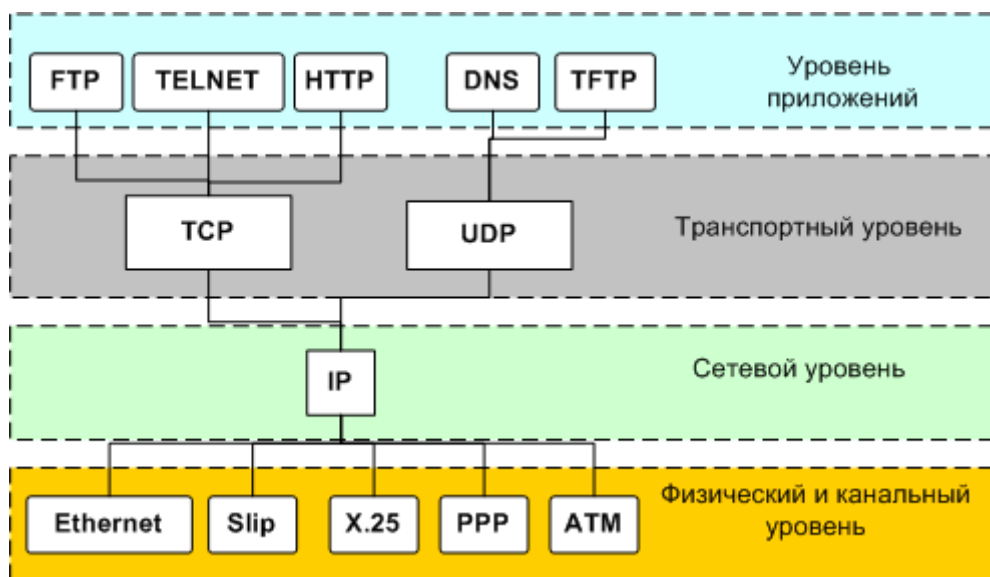
Для стандартных классов сетей маски имеют следующие значения:

- класс А - 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс В - 11111111.11111111. 00000000. 00000000 (255.255.0.0);

Стек протоколов TCP/IP

TCP/IP - собирательное название для набора (стека) сетевых протоколов разных уровней, используемых в Интернет. Особенности TCP/IP:

- Открытые стандарты протоколов, разрабатываемые независимо от программного и аппаратного обеспечения;
- Независимость от физической среды передачи;
- Система уникальной адресации;
- Стандартизованные протоколы высокого уровня для распространенных пользовательских сервисов.



Система доменных имён DNS.

DNS(англ. *Domain Name System* — система доменных имён) — распределённая система преобразования имени хоста (компьютера или другого сетевого устройства) в IP адрес. DNS работает в сетях TCP/IP. Как частный случай, DNS может хранить и обрабатывать и обратные запросы, определения имени хоста по его IP (PTR-записи).

Ключевые характеристики DNS

DNS обладает следующими характеристиками:

- *Распределённость хранения информации.* Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности и (возможно) адреса корневых DNS-серверов.

- *Кеширование информации.* Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.
- *Иерархическая структура,* в которой все узлы объединены в дерево, и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.
- *Резервирование.* За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение работы даже в случае сбоя одного из узлов.

DNS важна для работы Интернета, ибо для соединения с узлом необходима информация о его IP-адресе, а для людей проще запоминать буквенные (обычно осмысленные) адреса, чем последовательность цифр IP-адреса. В некоторых случаях это позволяет использовать виртуальные серверы, например, HTTP-сервера, различая их по имени запроса. Первоначально преобразование между доменными и IP-адресами производилось с использованием специального текстового файла HOSTS, который составлялся централизованно и обновлялся на каждой из машин сети вручную. С ростом Сети возникла необходимость в эффективном, автоматизированном механизме, которым и стала DNS.

Домены верхнего уровня.

Первые домены верхнего уровня были рассчитаны на США:

- gov - государственные организации
- mil - военные учреждения
- edu - образовательные учреждения
- com - коммерческие организации
- net - сетевые организации

Позднее, когда сеть перешагнула национальные границы США появились национальные домены типа:

- uk - Объединенное королевство
- jp - Япония
- au - Австралия
- ch - Чехия
- su - СССР
- ru - Россия
- и т.п.

Принципы функционирования доменной системы DNS.

Служба доменных имен работает как распределенная база, данные которой распределены по DNS-серверам.

Система доменных имен - это сервис прикладного уровня, значит, использует транспорт **TCP** и **UDP**.

Программная поддержка системы DNS.

Сервис DNS строится по схеме "клиент-сервер". В качестве клиентской части выступает процедура разрешения имен - **resolver**, а в качестве сервера DNS-сервер (**BIND ...**)

Например, когда мы хотим обратиться к серверу ipm.kstu.ru, ваш браузер, используя **resolver**, поступает следующим образом:

1. ищет запись ipm.kstu.ru в файле hosts, если не находит, то,

2. посылает запрос на известный DNS-кэширующий сервер (как правило, локальный), если на этом сервере запись не найдена, то,
3. сервер DNS-кэширующий обращается к DNS-ROOT серверу с запросом адреса DNS сервера отвечающего за домен первого уровня ru, если получает адрес, то,
4. сервер DNS-кэширующий обращается к DNS серверу, отвечающего за домен первого уровня ru, с запросом адреса DNS сервера отвечающего за домен второго уровня kstu.ru, если получает адрес, то,
5. сервер DNS-кэширующий посылает запрос на DNS сервер, отвечающий за домен второго уровня kstu.ru, если получает адрес, то,
6. сервер DNS-кэширующий адрес кэширует и передает клиенту
7. клиент обращается по IP адресу - 195.208.44.20

Система универсальных идентификаторов ресурсов URI и URL.

URI (Uniform Resource Identifier, Универсальный идентификатор ресурса) (RFC 2396, August 1998) - компактная строка символов для идентификации абстрактного или физического ресурса. Под ресурсом понимается любой объект, принадлежащий некоторому пространству. Включает и переопределяет определенные ранее URL (RFC 1738/RFC 1808) и URN (RFC 2141, RFC 2611).

URI предназначен для уникальной идентификации любого ресурса.

Некоторые подмножества URI:

URL (Uniform Resource Locator, Универсальный указатель ресурса), - подмножество схем URI, который идентифицирует ресурс по способу доступа к нему (например, его "местонахождению в сети") вместо того, чтобы идентифицировать его по названию или другим атрибутам этого ресурса.

Примеры URL:

- <http://www.ipm.kstu.ru/index.php>
- <ftp://www.ipm.kstu.ru/>

В HTML записывается так:

URN (Uniform Resource Name, Универсальное имя ресурса) - частная URI-схема "urn:" с подмножеством "пространства имен", который должен быть уникальным и неизменным даже в том случае, когда ресурс уже не существует или недоступен.

Предполагается что, например браузер, знает, где искать этот ресурс.

Синтаксис:

urn:namespace: data1.data2,more-data, где namespace (пространство имен) определяет, каким образом используются данные, указанные после второго ":".

Пример URN:

- urn:ISBN: 0-395-36341-6
- ISBN - тематический классификатор для издательств
- 0-395-36341-6 - конкретный номер тематики книги или журнала

При получении URN клиентская программа обращается к ISBN (каталогу "тематический классификатор для издательств" в Интернете). И получает расшифровку номера тематики "0-395-36341-6" (например: "квантовая химия").

URN массово используется в P2P сетях (подобных edonkey).

Пример URN указывающего на образ диска Adobe Photoshop v8.0 в сети edonkey:

urn:ed2k://|file|Adobe Photoshop v8.0.iso|940769280|b34c101c90b6dedb4071094cb1b9f2d3|/

где:

1. ed2k - указывает на сеть
2. file - файл
3. Adobe Photoshop v8.0.iso - название файла
4. 940769280 - размер в байтах
5. b34c101c90b6dedb4071094cb1b9f2d3 - идентификатор файла (вычисляется с помощью хеш-функции)

Принципы системы: расширяемость, полнота, читаемость.

URL - Uniform Resource Locators явно описывает, как добраться до объекта.

Синтаксис:

:

где:

scheme = "http" | "ftp" | "gopher" | "mailto" | "news" | "telnet" | "file" | "man" | "info" | "whatis" | "ldap" | "wais" | ... - имя схемы

scheme-specific-part - зависит от схемы

Имя схемы - последовательность символов [a-z0-9+.-].

В scheme-specific-part можно использовать шестнадцатеричные значения в виде: %5f.

Обязательно должны кодироваться непечатные октеты: 00-1F, 7F, 80-FF. Также всегда кодируются "небезопасные" символы: " ", "<", ">", "\"", "#", "%", "{", "}", "|", ":", "\\", "~", "[", "]", "`".

Некоторые схемы резервируют и другие символы: ";", "/", "?", ":", "@", "=", "&". Их также необходимо кодировать, если хочется "обойти" их специальное трактование. Таким образом остаются [a-z0-9\$-_.+!*'(),] и резервированные символы в их специальном значении для данной схемы.

Формат URL.

URL - Uniform Resource Locators явно описывает, как добраться до объекта.

Синтаксис:

:

где:

scheme = "http" | "ftp" | "gopher" | "mailto" | "news" | "telnet" | "file" | "man" | "info" | "whatis" | "ldap" | "wais" | ... - имя схемы

scheme-specific-part - зависит от схемы

Имя схемы - последовательность символов [a-z0-9+.-].

В scheme-specific-part можно использовать шестнадцатеричные значения в виде: %5f.

Обязательно должны кодироваться непечатные октеты: 00-1F, 7F, 80-FF. Также всегда кодируются "небезопасные" символы: " ", "<", ">", "\"", "#", "%", "{", "}", "|", ":", "^\", "~", "[", "]", "`".

Некоторые схемы резервируют и другие символы: ";", "/", "?", ":", "@", "=", "&". Их также необходимо кодировать, если хочется "обойти" их специальное трактование. Таким образом остаются [a-z0-9\$-_.+!*'(),] и резервированные символы в их специальном значении для данной схемы.

Структура стеков TCP/IP.

Стек TCP/IP и модель OSI.

Инкапсуляция протоколов верхнего уровня в протоколы стека TCP/IP.

Структура пакетных TCP.

тема 1.3 Сервисы глобальной сети интернет

типы сервисов глобальной сети

- World Wide Web;
- сервис имён доменов (DNS – Domain Name System);
- доступ к файловым архивам (FTP);
- электронная почта (e-mail);
- телеконференции (Usenet) и списки рассылки;
- сервисы общения ICQ, IRC и др.;
- сервис Telnet;
- поиск информации в Internet.

программы клиенты

Некоторые программы-клиенты FTP (File Transfer Protocol) и WWW-браузеры поддерживают полезную функцию FTP-протокола `reget`, позволяющую продолжить загрузку файла, передача которого была прервана по какой-либо причине, а не начинать процесс сначала. Естественно, FTP-сервер тоже должен поддерживать эту функцию.

браузеры

Для доступа к информации в WWW на клиентских компьютерах используется специальное программное средство, называемое браузером (browser). Это приложение позволяет пользователю перемещаться по разнообразным данным Internet, предлагаемым в основном серверами WWW. В настоящее время наиболее популярными браузерами являются Microsoft Internet Explorer, Mozilla Firefox и Opera.

веб сервис

архитектура www

- 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),

основные компоненты www(htm url http sgi)

Позже команда NCSA добавила к этим трем компонентам четвертый: универсальный интерфейс Шлюзов CGI (Common Gateway Interlace)

Технология WWW состоит из четырех основных компонентов:

- 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);
- универсальный интерфейс шлюзов CGI (Common Gateway Interface).

HTML (Hyper Text Markup Language) — язык разметки гипертекста возник на стыке нескольких направлений исследований и разработок. Язык HTML использует команды — теги, вводимые в текстовые документы, которые указывают, каким образом информация должна на выводиться на экран.

HTML-документ представляет собой текстовый файл, который может быть создан с помощью любого текстового редактора, но, в отличие от обычных текстовых файлов, он имеет расширение .htm (или .html).

принципы работы клиент - сервис

От описания основных компонентов перейдем к архитектуре взаимодействия программного обеспечения в системе World Wide Web. WWW построена по хорошо известной схеме «клиент—сервер». Программа-клиент выполняет функции интерфейса пользователя и обеспечивает доступ практически ко всем информационным ресурсам Интернета. В этом смысле она выходит за обычные рамки работы клиента только с сервером определенного протокола.

назначение web технологий и роль www

технологии поддерживаемые сервером

Другую часть программного комплекса WWW составляет сервер протокола HTTP, базы данных документов в формате HTML, управляемые сервером, и программное обеспечение, разработанное в стандарте спецификации CGI. До самого последнего времени реально использовалось два HTTP-сервера: сервер CERN и сервер NCSA. Но в настоящее время число базовых серверов расширилось.

технологии поддерживаемые клиентом

Клиент — это интерпретатор HTML. И, как типичный интерпретатор, клиент в зависимости от команд (разметки) выполняет различные функции. В круг этих функций входит не только размещение текста на экране, но и обмен информацией с сервером по мере анализа полученного HTML-текста, что наиболее наглядно происходит при отображении встроенных в текст графических изображений. При анализе URL-спецификации или по командам сервера клиент запускает дополнительные внешние программы для работы с документами в форматах, отличных от HTML, например GIF, JPEG, MPEG, Postscript и т.п.

1.4

Понятие гипертекста.

гипертекст – это метод баз данных (БД), вводящего новую форму прямого доступа к данным.

Организационная структура информации Web-узла (иерархическая, линейная, в виде паутины).

Модели проектирования Web-узла.

Модель проектирования это метод разработки проекта, основанный на определённом представлении его разработчиков, как необходимо создавать свои работы. Ниже представлены некоторые распространённые модели, которые можно использовать для создания сайта и других проектов. Модель водопада



Рис.1. Модель водопада

Модель водопада предполагает выполнение нескольких, следующих друг за другом этапов, приведенных на (рис. 1) Вначале идет написание технического задания, затем анализ и проектирование сайта; следующие этапы — написание контента, дизайн и программирование. Заканчивается все генеральным тестированием и приёмкой проекта.

***Достоинства*.** Это одна из самых простых и доступных моделей. Все этапы идут последовательно, каждый последующий не начинается, пока не закончится предыдущий.

***Недостатки*.** Эта модель предполагает точное знание того, чего хочется реализовать на сайте. Обычно же, обстоит как раз наоборот, очень трудно сразу сформулировать цели, которые следует выполнить.

Спиральная модель


	

Рис. 2. Спиральная модель

Данная модель использует противоположный подход, нежели модель водопада. Работа начинается с этапа «Планирование и анализ» и по часовой стрелке переходит к этапам выполнения, тестирования полученных результатов и оценки. На следующей итерации всё повторяется по новой, но уже с учётом выявленных недочётов проекта. Таким образом, пройдя несколько итераций и повторив все этапы несколько раз, проект избавляется от недостатков, обрстая дополнительными возможностями и преимуществами.

***Достоинства*.** Возможность разрабатывать проект за несколько итераций позволяет постепенно его улучшать, реализовывая различные идеи.

***Недостатки*.** «Лучшее — враг хорошего». На каком-то этапе всегда следует остановиться, чтобы представить проект. Но в данной модели чёткие критерии его выполнения отсутствуют. Это создает определенные сложности для расчета финансовых затрат на выполнение проекта.

Microsoft Solutions Framework (MSF)


	

Рис.3. Модель Microsoft Solutions Framework

Данная модель сделана компанией Microsoft для своих собственных целей, но приобрела популярность и среди других разработчиков. Все программные продукты Microsoft создаются именно по этой методологии. Модель MSF вобрала в себя лучшее из двух моделей, описанных выше — спиральной и водопада. Состоит из четырех этапов: анализ, планирование, разработка и стабилизация (рис. 3). Каждый этап заканчивается достижением определённого результата, например, после анализа проекта идёт одобрение продуманной концепции. В итоге пишется определённый документ, в котором записывается результат выполнения данного этапа, для того, чтобы каждый из разработчиков чётко понимал своё место в проекте и задачи, которые ему предстоит решить. Модель итерационна и при прохождении всех этапов, проект можно доработать с учётом предыдущей итерации. Но, поскольку, окончание каждого этапа чётко указано, нет, как в спиральной модели, бесконечного повторения одного и того же процесса.

***Достоинства*.** Пожалуй, MSF является одной из самых интересных моделей разработки и создания проектов, взявшей лучшее из других моделей и отказавшись от присущих им недостатков.

***Недостатки*.** Практически их нет. Разве что можно отнести к ним большую, по сравнению с другими моделями, сложность. Поэтому для небольших сайтов рекомендуется использовать другие методы.

Модель Уолта Диснея

Модель Уолта Диснея используется для проектирования сайта и состоит из трех этапов. В терминах модели эти этапы называются фазами мечтателя, реалиста и критика. Однако, мы будем придерживаться других названий, поскольку они более точно отражают суть процесса разработки сайта. Итак, процесс проектирования разбивается на следующие стадии:

- \1. Концептуальное проектирование.
- \2. Логическое проектирование.
- \3. Физическое проектирование.

Этапы следуют последовательно один за другим (рис. 4), но в некоторых случаях возможен переход к следующей стадии без окончания предыдущей. Это может происходить, например, когда разработчиков несколько и каждый работает со своей частью сайта. В любом случае, после окончания этапа физического проектирования следует вернуться к началу и внести соответствующие коррективы.


	

Рис.4. Этапы проектирования

Содержание этапов проектирования Web-узла: концептуальное, логическое, физическое проектирование.

Концептуальное проектирование

Порой бывает сложно оценить эффективность сайта. Есть однако универсальный критерий, который довольно точно характеризует эффективность сайта. Это достижение разработчиками сайта поставленных перед ними целей. В этом случае сайт превращается в действенный инструмент, который выполняет возложенные на него функции.

Концептуальное проектирование служит для указания целей, задач сайта и определения аудитории, на которую он рассчитан.

На этом этапе следует описать следующее:

- \1. Основные и второстепенные цели.
- \2. Действия, которые необходимо предпринять для достижения поставленных целей.
- \3. Состав пользователей.
- \4. Интересы групп пользователей.
- \5. Разделы сайта.
- \6. Критерии достижения цели.

Логическое проектирование

Разделы сайта, продуманные на предыдущем этапе, пока не упорядочены и не структурированы, поэтому их нужно привести к удобному и понятному виду. Логическое проектирование включает организацию информации на сайте, построение его структуры и навигации по разделам.

На данном этапе следует задаться вопросом, каким образом будет упорядочена информация. Варианты могут быть самыми разными и зависеть от типа данных и предпочтений создателей сайта: по времени, разделам, в алфавитном порядке, определенным группам или другим критериям. Так, для сайта музыкальной группы, поиск определенной песни можно сделать в виде алфавитного указателя, по названию альбома, первым строчкам песни, году выпуска и по ключевым словам. Одновременное использование различных способов охватывает большую аудиторию и позволяет быстрее найти нужную информацию на сайте.

На этом этапе следует описать следующее:

- \1. Тип структуры сайта (линейная, иерархическая, контекстная, другая).
- \2. Названия разделов.
- \3. Что будет содержать в себе каждый раздел.
- \4. Организация и связь разделов между собой.
- \5. Какая информация будет размещена на определенных страницах сайта.

Конечный результат логического проектирования оформляется в виде блок-схем, структурных диаграмм или другими способами, показывающими взаимосвязь различных частей сайта.

Физическое проектирование

Этап поиска проблем, а не их решений, связанных, по большей части, с технической реализацией сайта.

На этом этапе следует описать следующее:

- \1. Технологии, которые будут применяться на сайте.
- \2. Используемое программное обеспечение.
- \3. Возможные проблемы и способы их устранения.
- \4. Как будет обновляться информация.

После завершения данного этапа следует вернуться к концептуальному проектированию и проверить, не нужно ли внести изменения, в связи с переосмыслением проекта на других стадиях.

Хостинг Web-узла.

WEB-хостинг – это услуга предоставления пространства на дисковых носителях сервера для размещения, хранения и поддержки веб-сайта клиента. Чтобы понять, для чего нужен хостинг, представьте, что Вы создали сайт. Пока он находится на Вашем компьютере, его никто, кроме Вас не увидит.

Большинство сайтов размещаются на удаленных серверах, которые всегда подключены к высокоскоростным Интернет-магистральям и обслуживаются специалистами. Это и есть хостинг. Компания, предоставляющая услуги хостинга называется провайдером хостинга или хостером.

В хостинг входит размещение всех файлов сайта клиента (графики, веб-страниц, скриптов, таблиц стилей и пр.), а также обработка скриптов и баз данных специальным программным обеспечением, находящимся на сервере хостера. Зачастую хостер предоставляет клиенту также адреса электронной почты, базы данных MySQL, PHP, SSI, Perl, Cron, FTP-доступ, доступ к файлу .htaccess, систему управления сайтом (CMS) и прочее. Все это зависит от выбранного вами тарифного плана. В услугу хостинга входит поддержка работоспособности сайта и обеспечение сохранности файлов. Для этого ежедневно осуществляется резервное копирование (back-up). Большой минус хостингу, который не выполняет регулярно резервное копирование данных клиентов.

Хостинг может быть:

§ платным

§ бесплатным

Бесплатный хостинг предоставляется без оплаты услуг, но, как правило, имеет существенные ограничения по предоставляемым функциям (например, отсутствует поддержка PHP или баз данных). Кроме этого, взамен за бесплатное размещение сайтов, хостер может вставлять на страницы клиента свою рекламу. Обычно сайты, пользующиеся услугами бесплатного хостинга – это любительские сайты. Чаще всего они имеют домены третьего уровня, предоставляемые самим хостером.

Платный хостинг надежнее и мощнее бесплатного. От суммы оплаты зависит количество дискового пространства и набор функций. В наиболее дешевые могут не входить поддержка баз данных MySQL, PHP, Perl и другое. Очень часто, при оплате хостинга на долгий срок, бесплатно предоставляется домен второго уровня, что позволяет существенно сократить затраты на содержание сайта. Но стоит обратить внимание на то, чтобы домен был зарегистрирован на ваше имя, а также вы имели доступ к управлению доменом. Это может решить ваши проблемы в будущем при переезде на другой хостинг.

Для того, чтобы разместить свой веб-сайт в сети, необходимо:

§ прежде всего, иметь собственный веб-сайт. Нужно иметь копию сайта на локальном компьютере (в html-файлах), или же готовые материалы + скрипт, который позволял бы создать веб-сайт непосредственно на сервере.

§ доменное имя. Нужно найти и приобрести доменное имя сайта. Желательно, чтобы имя говорило о тематике сайта, и было легко запоминающимся. Доменное имя может быть в любой, на ваш выбор, доменной зоне, например: .com .net .org



.ru и т.д.

§ заказать один из хостинг-планов в компании, осуществляющей услуги хостинга.

Программное обеспечение рабочего места Webразработчика

Необходимые программы:

 Html-редактор


 Графический редактор  Браузер(ы)

 Web-сервер

Вспомогательные программы:

 FTP-клиент

 Программы для просмотра изображений  Программы для работы с CSS

 Notepad

Необходимые программы Html-редактор

Что такое html-редактор? Это программа, помогающая Web-мастеру писать html-коды (html-теги) и создавать html-страницы. Такие программы позволяют генерировать код при нажатии на определенные кнопки или при выборе требуемых опций в меню, освобождая пользователя от огромной рутинной работы по написанию кодов, их многочисленных атрибутов и значений.

Macromedia Home Site 5

Home Site 5 — это профессиональный кодовый редактор, и в процессе создания страниц вы можете и писать код «вручную», и одновременно использовать возможности редактора:

Иными словами, этот редактор создает «чистый» html-код. Он потребует от вас знания языка разметки.

Графический редактор Adobe Photoshop

Лучший на сегодняшний день профессиональный графический редактор растровой графики — Adobe Photoshop

Adobe Photoshop 7 используется не только для создания, редактирования и подготовки изображений для размещения на Web-страницах, но и в полиграфии.

Вместе с Adobe Photoshop 7 поставляется программа Adobe ImageReady 7, в которой можно найти полезные функции для Web-дизайна, например для создания анимированных изображений, карт-ссылок, функции rollover с автоматической генерацией html- и JavaScript кода.

Web-сервер

Для отладки страниц потребуется установить Web-сервер. Динамичная страница предварительно обрабатывается сервером — выполняется встроенный в нее код, и только после этого готовый результат высылается клиенту. В настоящий момент практически любая Web-страница — динамичная, т. е. создается с использованием

серверных технологий. Например, технология SSI (Server Side Include) — включения на стороне сервера — была первой, позволяющей осуществлять предварительную обработку кода страницы на сервере. Более подробно речь об этой технологии пойдет на шестом уроке.

Популярные Web-серверы — Apache и IIS. По статистике около 70% сайтов используют в качестве программного обеспечения Web-серверы Apache и примерно 25% — IIS.

2.1 разметка языков гипертекста

ПОНЯТИЕ О СТАНДАРТНОМ ОБОБЩЕННОМ ЯЗЫКЕ РАЗМЕТКИ SGML. КОНСОРЦИУМ W3C.

Любой документ имеет три составляющих:

- содержание;
- структуру;
- стиль.

Содержание документа на бумаге может быть сугубо текстовым, а также содержать изображения. Если документ представлен в электронном виде, он может содержать и мультимедийные данные, а также ссылки на другие документы.

Обычно содержание документа представляется не в произвольном порядке, а имеет определенную **структуру**. Структура – это состав и порядок следования частей (блоков) документа.

Стиль документа определяет форму вывода его содержания на то или иное устройство (например, принтер или дисплей). В понятие стиль входят характеристики шрифта (наименование, размер, цвет) всего выводимого документа или отдельных его блоков, порядок разбивки на страницы, расположение блоков на страницах и другие параметры.

Языки разметки документов являются искусственными языками, предназначенными для описания структуры документа и отношений между различными объектами структуры. Данные разметки называются также **метаданными**.

Первым языком разметки является **язык GML**. Его непосредственным наследником стал **язык SGML** – стандартный обобщенный язык разметки, определяющий правила записи элементов разметки документа.

SGML был разработан для совместной разработки машинных документов в больших правительственных и аэрокосмических проектах. Он широко применялся в печатном деле и издательской сфере, но его сложность затруднила повседневное использование. Основные наследники SGML — форматы HTML и XML.

требования к языку разметки документов:

- \1. Язык должен быть доступен для чтения человеком.
- \2. Размеченные файлы документов должны быть текстовыми и кодироваться с помощью символов кода ASCII
- \3. В языке могут использоваться ссылки как на внутренние ресурсы (в том же документе), так и на внешние ресурсы (в других документах).

В языке SGML и подобных ему языках используются специальные инструменты разметки документа:

- структуры документа;
- дескрипторы или элементы и сопутствующие им атрибуты;
- сущности (entities);
- комментарии.

Документы SGML имеют древовидную структуру.

Дескрипторы в SGML размещаются в начале (открывающий дескриптор) и в конце (закрывающий дескриптор) каждого *элемента* (item).

Атрибуты – это простые символьные конструкции (items), которые добавляются к элементам для придания им уточнения действия дескрипторов.

Языки обобщенной разметки, подобные SGML, допускают использование атрибутов, с которыми могут быть соотнесены до 15 различных типов значений, в том числе:

- Ссылки на любые ресурсы, находящиеся вне документа, на которые обычно ссылаются как на сущности (entities).
- Уникальный идентификатор (ID) элемента в документе.
- Указатели идентификаторов (ID Pointers), имеющие перекрестные ссылки для тех элементов, которые имеют ID, упомянутые в документе.
- Обозначения или атрибуты элементов, которые определяют обозначения в содержании элемента.
- Символьные данные (character data), или CDATA, представляющие собой любые допустимые символы, которые не могут выступать в качестве значений атрибутов.

Комментарии позволяют добавить информацию, которая не будет видна после обработки документа. Комментарии не влияют на скорость обработки документа, не рассматриваются и не обрабатываются как часть содержания SGML-документа. Они просто включаются в исходный текст.

Для проверки соответствия документа разметке заданного типа используются специальные программы – **анализаторы** (parsers). Анализаторы являются либо отдельными программами, либо частью программы обработки документа SGML. Чтобы анализатор мог выполнить проверку документа, создается специальный документ, называемый **определением типа документа**

Язык **HTML** является приложением языка SGML для использования в Internet с фиксированной структурой, фиксированным набором элементов (дескрипторов) и их атрибутов, а также фиксированным набором сущностей. расширенный язык разметки **XML** (Extensible Markup Language).

HTML соответствует международному стандарту ISO 8879. Текущий стандарт HTML 4.01 существует с 1999 г. В стоящее время опубликован проект пятого стандарта языка. Новая версия HTML обещает дополнить язык многочисленными расширениями и обеспечить более простую, логичную и удобную систему правил.

Dynamic HTML или DHTML — это способ создания интерактивного Web- сайта. DHTML возник как набор методов динамического создания и изменения Web- страниц путем вызова из HTML- документа сценариев. Однако развитие этих методов

привело к полному пересмотру концепции Web-документа и формированию понятия объектной модели документа DOM (Document Object Model).

DOM представляет собой платформенно — независимый программный интерфейс, позволяющий программам и скриптам управлять содержимым документов HTML и XML, а также изменять их структуру и оформление.

Модель DOM не накладывает ограничений на структуру документа. Любой документ известной структуры с помощью DOM может быть представлен в виде дерева узлов, каждый из которых содержит некоторый объект. Узлы связаны между собой отношениями «родитель-потомок».

Изначально многие браузеры имели собственную модель DOM, не совместимую с остальными. Для того чтобы обеспечить совместимость, специалисты международного консорциума W3C классифицировали эту модель по уровням, для каждого из которых была создана своя спецификация. Все эти спецификации объединены в общую группу, носящую название W3C DOM.

Язык XML является подмножеством языка SGML, полностью совместимым с ним.

Язык XML обеспечивает широкий спектр функциональных возможностей, которые отсутствуют в HTML

Что касается Интернета в целом, нет такого человека или группы людей, которые бы управляли Всемирной паутиной. Однако, Консорциум Всемирной паутины (W3C, World Wide Web Consortium) играет проактивную роль в разработке рекомендаций и прототипов технологий, относящихся к сети. Темы, которых касается W3C – веб-архитектура, стандарты веб-дизайна и обеспечение доступности сайтов для инвалидов. В попытке стандартизировать веб-технологии W3C создает спецификации, которые называются рекомендациями. В их создании принимают участие множества крупных корпораций по разработке веб-технологий. Следование рекомендациям W3C – первый шаг на пути осуществления доступности вашего сайта

История возникновения

Консорциум был создан в 1994 году как консультативный орган для лидеров компьютерной индустрии. Крупнейшие мировые компании и корпорации договаривались в W3C об обеспечении совместимости своих продуктов и внедрении новых

технологических стандартов. Первым крупным успехом консорциума стала стандартизация языка гипертекстовой разметки HTML (англ. HyperText Markup Language) в 1996 году.

Дело в том, что в середине 1990-х годов ряд крупнейших производителей программного обеспечения планировал выпустить каждый свою версию языка HTML со своими названиями тегов. Разумеется, это привело бы к хаосу в Интернете, и в результате веб- страница одной компании была бы размечена совершенно не так, как страница другой компании. Из-за этого веб-браузер одной компании не мог бы отображать страницы, созданные по правилам другой компании.

Именно W3C принадлежит заслуга в том, что HTML был выпущен с единым базовым набором тегов и атрибутов и веб-страницы стали такими, какими мы их знаем сейчас. Хотя полного совпадения тегов и атрибутов HTML достичь, к сожалению, не удалось до сих пор. Это, конечно, усложняет жизнь веб-разработчикам

Структура консорциума

Общую администрацию консорциума Всемирной паутины осуществляют 3 организации:

- Массачусетский технологический институт (англ. Massachusetts Institute of Technology, MIT) в США;

- Европейский консорциум по исследованиям в области информатики и математики (англ. European Research Consortium for Informatics and Mathematics, ERCIM) во Франции;

- Университет Кейо (англ. Keio University) в Японии.

Членом консорциума может стать юридическое или частное лицо, занимающееся веб-технологиями и заинтересованное в развитии Интернета. Основным источником финансирования консорциума являются членские взносы. Членами консорциума уже являются более 350 организаций.

Международную координацию осуществляют так называемые «офисы W3C» (англ. W3C Offices), которые созданы уже в 14 странах мира. Время от времени консорциум Всемирной паутины также устраивает международные конференции.

Процесс выработки рекомендаций проходит в специальных группах. Рабочие и исследовательские группы консорциума включают штатный персонал, представителей организаций-членов и приглашённых экспертов. С предложениями в адрес рабочих групп

может выступить любое лицо, даже не члены W3C. 16 февраля 2012 W3C совместно с НИУ ВШЭ открыли представительство консорциума в России.

Особенности внедрения рекомендаций

Рекомендации консорциума Всемирной паутины открыты, то есть не защищены патентами и могут внедряться любым человеком без всяких финансовых отчислений консорциуму. В отличие от других организаций, занимающихся разработкой стандартов для Интернета, консорциум Всемирной паутины не имеет программ сертификации (на соответствие рекомендациям консорциума) и не планирует их вводить, поэтому рекомендации W3C получили гораздо большее распространение, нежели стандарты любых других организаций.

В то же время, из-за отсутствия сертификации многие производители следуют рекомендациям лишь частично. Рекомендации консорциума построены таким образом, что частичное внедрение не нарушает общих стандартов. Некоторые популярные рекомендации имеют несколько степеней внедрения — кому как удобнее. Степени внедрения — это новое слово в сетевых

стандартах, которое принесло консорциуму Всемирной паутины и его рекомендациям заслуженную популярность и признание Процесс стандартизации

Рекомендации W3C – это заключительный этап ратификации процесса рассмотрения вопросов стандартов рабочей группой Консорциума Всемирной паутины (W3C). Эта ратификация означает, что документ был подвергнут общественному рассмотрению и обзору организациями членами W3C. Рекомендации W3C направлены на стандартизацию веб-технологий.

В соответствии с описанием процесса разработки рекомендации W3C документ проходит через несколько этапов:

Рабочий проект (WD — Working Draft)

На рабочем уровне проекта, стандарт публикуется для рассмотрения сообщества. WD документ является первой формой стандарта, которая выкладывается в публичный доступ. Обычно принимаются практически все комментарии, хотя это и не является обязательным, и нет гарантий что комментарий будет учтён при дальнейшей работе над документом.

На данном этапе описание стандарта, скорее всего, может иметь существенные отличия от его окончательной формы. Таким образом, любой, кто разрабатывает WD документ должен быть готов к значительному изменению их реализации стандарта.

Кандидат в рекомендации (CR — Candidate Recommendation)

Кандидат в рекомендации версии стандарта (CR) это уже более стабильная версия документа чем WD. На этот момент группа разработки отвечает на вопросы тех, кто считает что нужно внести изменения в суть реализации стандарта. удовлетворен тем, что стандарт не то, что нужно его. Целью CR является получение помощи от сообщества разработчиков, которым предстоит реализовывать этот стандарт. Документ стандарта может меняться и дальше, но начиная с этого момента, возможность внесения существенных изменений с стандарт как правило закрыта. Дизайн небольших деталей все еще может измениться из-за обратной связи с реализаторами стандарта.

Предложение в рекомендации (PR — Proposed Recommendation)

Вариантом стандарта на уровне «предложения в рекомендации» (PR) документ становится после прохождения предыдущих двух этапов. Пользователи уже внесли свои предложения в стандарт и реализаторы стандарта тоже внесли свои изменения. На данном этапе документ представляется на рассмотрение Консультативному Совету W3C для окончательного утверждения.

Хотя этот шаг является очень важным, он редко приводит к внесению каких-либо существенных изменений в стандарт.

Рекомендация W3C (REC — W3C Recommendation)

Это заключительный этап развития стандарта. На данный момент стандарт уже прошёл широкий обзор и тестирование, как в теоретических, так и практических условиях. На этом этапе документ уже одобрен W3C в качестве стандарта для широкого развертывания в соответствующей области.

Последующие изменения

Рекомендации могут обновляться позднее отдельно опубликованными исправлениями, которые аккумулируют достаточно существенные изменения, сделанные со времени последней публикации. W3C также публикует различные виды информационных заметок, которые не предназначены для рассмотрения в качестве стандарта.

Рекомендации могут время от времени обновляться. К рекомендациям публикуются сообщения о выявившихся ошибках и неточностях (англ. errata). Когда накапливается достаточный запас выявленных ошибок, выходит новая, исправленная и доработанная редакция (англ. edition) рекомендации (например, «редакция 1.1»). В исключительных случаях вся рекомендация может быть отозвана консорциумом для переработки.

Для удобства пользователей консорциумом созданы специальные программы-[валидаторы](#) (англ. Online Validation Service), которые доступны по Сети и могут за несколько секунд

проверить документы на соответствие популярным рекомендациям W3C. Консорциумом также созданы многие другие утилиты для облегчения работы веб-мастеров и программистов. Большинство утилит — это свободные программы, все они бесплатные. В последнее время, следуя мировым тенденциям, консорциум в целом гораздо больше внимания уделяет проектам с открытым исходным кодом.

ВЕРСИИ ЯЗЫКА ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ HTML. ПОНЯТИЕ О РАСШИРЯЕМОМ ЯЗЫКЕ РАЗМЕТКИ XML.

Первая версия **языка гипертекстовой разметки** – **HTML** (HyperText Markup Language), так же, как и сама технология Web, была разработана Тимом Бернерсом Ли в 1991 г. Язык HTML является реализацией правил языка SGML для типа документов, которые были названы *документами* HTML. Язык задает фиксированную структуру, фиксированный набор тегов и их атрибутов, а также фиксированный набор сущностей. Программы обработки документов HTML называются *Web-браузерами*. Результатом обработки документа является *Web-страница*, выводимая на экран дисплея.

В 1994 г. группа поддержки Интернет – IETF (Internet Engineering Task Force) разработала спецификацию HTML 2.0, с которой началось широкое распространение языка HTML в сети Internet. В том же году был создан консорциум W3C (World Wide Web Corporation), объединивший 165 коммерческих и академических организаций, разработчиков и пользователей (с момента создания и по настоящее время эту организацию возглавляет Т.Б. Ли). Последняя версия спецификации HTML – HTML 4.01 была принята консорциумом в декабре 1999 г.

- Язык XML обеспечивает широкий спектр функциональных возможностей, которые отсутствуют в HTML

Последняя версия спецификации языка XML – XML 1.1 была принята в апреле 2004 г.

На основе языка XML консорциум W3C разработал дальнейшее развитие языка HTML

– **язык XHTML** (Extended HTML – расширенный HTML). Первая версия этого языка – XHTML 1.0 была принята в январе 2000 г. Эта версия фактически представляет собой переформулирование HTML 4 как приложения XML 1.0. Предполагается, что дальнейшее развитие языка HTML будет осуществляться в соответствии со спецификациями XHTML.

Новая версия XHTML – XHTML 1.1 была принята консорциумом W3C в мае 2001 г. Эта рекомендация определяет новый тип документа – XHTML на основе модулей. Каждый модуль XHTML 1.1 содержит один или несколько элементов и/или атрибутов языка HTML.

В соответствии со спецификацией, документы XHTML 1.1 состоят из следующих групп модулей XHTML:

Модули ядра – это модули, наличие которых необходимо в любом типе документа, соответствующего спецификации XHTML (в эту группу входят модули Structure, Text, Hypertext и List).

Модуль Applet, содержащий единственный элемент (этот элемент признан устаревшим и вместо него рекомендуется использовать элемент).

Модули текстовых расширений, в которых определены различные дополнительные модули текстовой разметки (в эту группу входят модули Presentation, Edit и Bi-directional Text).

Модули форм (в эту группу входят модули Basic Forms и Forms). Модули таблиц (в эту группу входят модули Basic Tables и Tables).

Модуль Image, предоставляющий базовые возможности внедрения изображений (этот модуль также может независимо использоваться в некоторых реализациях клиентскими картами-изображениями).

Модуль Client-side Image Map, предоставляющий элементы для клиентских карт-изображений (для функционирования этого модуля необходимо включение модуля Image).

Модуль Object, предоставляющий поддержку включения объектов общего назначения.

Модуль Frames, предоставляющий элементы, относящиеся к фреймам.

URL (с помощью этого элемента вычисляются относительные URL документа).

Модуль Name Identification, используемый для идентификации определённых элементов в документах HTML.

Модуль Legacy, определяющий элементы и атрибуты, которые уже не рекомендовались в предыдущих версиях HTML и XHTML и не рекомендуются в дальнейшем.

Типы структур HTML и XHTML

Согласно спецификации HTML 4.01 для документов HTML определены **три структуры**, описываемые тремя DTD. Разработчики Web-страниц должны включать в свои документы одно из трех объявлений типов. Разница между DTD заключается в поддерживаемых ими элементах. Объявление DTD должно размещаться в самом начале документа.

HTML 4.01 Strict DTD (строгое определение) включает все элементы и атрибуты, не являющиеся отмененными (deprecated) и не использующиеся в документах с фреймами.

Определение **HTML 4.0 Transitional DTD** (переходное определение) включает все элементы, включенные в строгое DTD, а также отмененные элементы и атрибуты.

Определение **HTML 4.0 Frameset DTD** (определение для фреймов) включает, помимо элементов переходного DTD, фреймы.

Первая строка документа HTML, определенного в соответствии со спецификацией XHTML

Эта строка определяет используемую версию XML и кодировку символов документа. При кодировании символов в XML используется двухбайтовый код Unicode. В качестве значений параметра **encoding** наиболее часто используются кодировки **UTF-8**, в котором значения первых 128 символов представляются в однобайтовой кодировке, символы наиболее распространенных языков (в том числе русского и украинского) – двумя байтами, а остальные символы тремя байтами. В кодировке **UTF-16** все символы представляются двумя байтами (эти кодировку рекомендуется использовать для русских и украинских документов HTML).

2.2

Основные понятия

HTML (*HyperText Markup Language*) - язык разметки гипертекста - предназначен для создания Web-страниц.

Под *гипертекстом* в этом случае понимается текст, связанный с другими текстами указателями-ссылками.

HTML представляет собой достаточно простой набор кодов, которые описывают структуру документа. HTML позволяет выделить в тексте отдельные логические части (заголовки, абзацы, списки и т.д.), поместить на Web-страницу подготовленную фотографию или картинку, организовать на странице ссылки для связи с другими документами.

HTML не задает конкретные и точные атрибуты форматирования документа. Конкретный вид документа окончательно определяет только *программа-браузер* на компьютере пользователя Интернета.

HTML также не является языком программирования, но web-страницы могут включать в себя встроенные программы-скрипты на языках *Javascript* и *Visual Basic Script* и программы-апплеты на языке *Java*.

Даже, если вы не собираетесь в дальнейшем редактировать "вручную" текст HTML (предполагая использовать графические редакторы), знание языка HTML даст вам возможность как лучше использовать эти средства, так и увеличит ваши шансы сделать HTML-документ более доступным и "читаемым" при просмотре браузерами разных фирм.

Основными компонентами HTML являются:

- **Тег (tag).** Тег HTML это компонент, который командует Web- браузеру выполнить определенную задачу типа создания абзаца или вставки изображения.
- **Атрибут (или аргумент).** Атрибут HTML изменяет тег. Например, можно выравнивать абзац или изображение внутри тега.
- **Значение.** Значения присваиваются атрибутам и определяют вносимые изменения. Например, если для тега используется атрибут выравнивания, то можно указать значение этого атрибута. Значения могут быть текстовыми, типа *left* или *right*, а также числовыми, как например ширина и высота изображения, где значения определяют размер изображения в пикселях.

Теги представляют собой зарезервированные последовательности символов, начинающиеся с < (знака меньше) и заканчивающиеся > (знаком больше).

Закрытие тега отличается от открытия только наличием символа *'/'*.

Предположим, у нас есть гипотетический атрибут форматирования текста, управляемый кодом , и мы хотим применить его к словам "Это мой текст".

HTML-последовательность кодов и собственно текста будет выглядеть так:

Это мой текст

Теги могут вкладываться друг в друга иерархически, но без пересечений, то есть допустимо вложение вида `<div><p>...</p></div>`, но не `<div><p>...<div>...</div>...</p></div>`.

Действие вложенных тегов объединяется. Например, если внутри тега, создающего жирное начертание шрифта, вложен тег курсива, то в результате получится жирный курсив.

Первое правило

Первое правило HTML: закрывайте все, что вы открыли!

НО! Из этого правила, как и из всех остальных, существуют исключения.

HTML- программа должна начинаться тегом `<html>` и заканчиваться тегом `</html>`

..... (здесь будут другие теги программы)

HTML- программы состоят из двух основных частей: заголовка и тела. Заголовок ограничивается парой тегов `<head>` и `</head>`, а тело - парой тегов `<body>` и `</body>`

В результате HTML- программа выглядит следующим образом:

... (здесь будет заголовок)

.... (здесь будут другие теги тела программы)

Кроме того, каждая HTML- программа имеет заголовок, который помещается в заголовок окна броузера. Заголовок окна броузера создается при помощи двух тегов и содержится между тегами и. Тогда программа принимает следующий вид:

;

.... (здесь будут другие теги тела программы)

Некоторые авторы, пишущие об языке HTML, советуют записывать теги прописными буквами, другие - используют строчные. Редактор HTML - Allaire HomeSite 4.5.1,

например использует по умолчанию нижний регистр для записи тегов. При создании моих страниц использовались оба варианта написания тегов. Как видите, допустимо и то и другое. Современные браузеры допускают запись тегов в любом регистре.

Уже позднее я узнала, что нельзя делать категоричные заявления по этому вопросу. Существуют теги и атрибуты "чувствительные" к написанию прописными или строчными буквами. Это регламентируется стандартами языка HTML, определенными [Консорциумом W3C](#).

Обращайтесь к [первоисточнику](#)!

Хорошее знание технического английского обязательно!

	<hr/>

При написании HTML-программ возникает необходимость вставки *комментариев* - поясняющих текстов, которые невидны при загрузке документа в браузер. Для этой цели служит тег `<!--`. Все, что заключено между символами `<!--` и `-->` считается комментарием и не отображается в браузере.

	<hr/>

Еще один тег, который очень важен в HTML-программе, но так же не предназначается для отображения какого-либо объекта в браузере - тег `<!--`. Этот тег служит специальным целям, а именно - указания языка, на котором написан документ, его кодовой страницы, ключевых слов, используемых поисковыми системами для классификации этого документа и т.п. Теги обычно вставляются в HTML- программу на заключительном этапе создания Web-страницы - *публикации*.

	<hr/>

Для вставки в HTML-программу фрагмента программ, написанных на языке JavaScript или Visual Basic Script *сценариев* используют теги `<script>` и `</script>`.

	<hr/>

Общая структура HTML-файла

Суммируя вышесказанное приведем общую структуру HTML-файл :

<Мета-теги>

<Функции скриптов>

Основная часть документа

Заголовки, абзацы, разрывы строк

Заголовки

Каждый пользователь компьютера, работающий в текстовом редакторе Microsoft Word знаком с понятием *стиля заголовка*. В HTML тоже применяется это понятие для структурирования документа и выделения важности заголовка. Всего существуют 6 стилей заголовка. Каждый из них обозначается в HTML-документе парными тегами

и *H1* Здесь *i* обозначает важность стиля. *H1* обозначает самый важный стиль заголовка,

H2 - стиль заголовка второго уровня, а *H6* - стиль заголовка самого нижнего уровня.

В подавляющем большинстве случаев для заголовков Web-страниц используют три первых уровня заголовков

И

**. Объясняется
это тем, что
размеры
шрифтов
оставшихся
заголовков
(теги**



**) меньше
размера
обычного
шрифта
Web-
страницы.**

**Вот как в
документ
можно
добавить
очень
важный
заголовок.**

**An
impo
rtant
head**

ing

А ВОТ

результат.

**An
important
heading**

**Посмотрим
другие
примеры:**

**Это
заголовок
второго
уровня**

Это заголовок третьего уровня

Это обычный текст

Это заголовок четвертого
уровня

Это заголовок пятого уровня

Это заголовок шестого уровня

Абзацы

Понятие абзаца в HTML-документе также аналогично понятию абзаца в Microsoft Word. Абзац обозначается в документе парными тегами

и

. Впрочем, применение закрывающего тега не является строго обязательным.

НО! Специфика тега

заключается в том, что после текста, который находится в его пределах, пустая строка добавляется *автоматически*.

Следует помнить и о другой особенности текстовых абзацев: когда текст достигает правой границы окна Web-браузера, переход на новую строку осуществляется автоматически, независимо от расположения тега

.

Для отдельного абзаца можно указать тип, размер и цвет шрифта отличным от стиля остального документа.

Например:

My

greetings to you!

А вот и результат -

My greetings to you!

Разрывы строк

Если в середине строки появилась необходимость ее разорвать - используйте одиночный тег переноса строки

.(Это соответствует нажатию клавишной комбинации [Shift]-[Enter] в текстовых процессорах Word). Код

не означает конца логического абзаца, и за строкой с этим кодом дополнительная пустая строка не появится.

Вот и нарушено первое правило!

Кроме тега

не требует закрытия тег добавления изображения .

Использование закрывающего тега

абзаца

также не является строго обязательным.

**HTML довольно "демократичен":
неправильный тег или
неправильное вложение тегов
обычно не приводят к
"зависаниям" броузера, а только
вызывает сообщение об ошибке в
строке состояния окна броузера
вашего Интернет-читателя.
Разумеется ошибки могут
привести к неправильному
форматированию HTML-
документа.**

**Примером использования тега
может служить написание
почтового адреса или
стихотворения.**

**Броузеры показывают текст в
своем окне, автоматически
осуществляя перенос слов.
Поэтому, если вы считаете
необходимым запретить разрыв
блока текста с пробелами между
словами - воспользуйтесь
специализированным символом * * -
символом незазрывного
пространства (non-breaking space).
Например,**

***Освежающий и бодрящий напиток
Coca Cola, приобрел широкую
популярность в нашей стране.***

А вот и результат:

***Освежающий и бодрящий напиток
Coca Cola, приобрел широкую
популярность в нашей стране.***

	<hr/>

Не следует использовать цепочку символов * * для выравнивания текста в окне броузера. Для этой цели рекомендуется использовать таблицы стилей.

	<hr/>

Предварительно отформатированн ый текст

Достоинством броузеров является их способность самостоятельно распределять текст в окне броузера. Но иногда вы не нуждаетесь в этой услуге, хотите самостоятельно определить представление вашего текста в окне броузера. Например, вы хотите представить код программы в наиболее удобочитаемом виде. Такую возможность вам предоставляет тег

. Посмотрим пример:

```
void Node::Remove()
```

```
{
```

```
if (prev)
```

```
prev->next = next; else if  
(parent)
```

```
parent->SetContent(null);
```

```
if (next)
```

```
next->prev = prev;
```

```
parent = null;
```

}

Это выполнится так

void Node::Remove()

{

if (prev)

prev->next = next; else if (parent)

parent->SetContent(null);

if (next)

next->prev = prev;

```
parent = null;
```

```
}
```

Красиво, правда? Привлекает внимание, удобно читать программный код. Это сделано при помощи таблицы стилей.

Обратите внимание на то, что в случае использования тега

*, текст отображается
броузером точно в таком виде,
как он был создан в HTML-
документе. Сохраняются все
пробелы, табуляции и переводы
строк. Исключением является
только новая строка,
следующая сразу же за тегом*

*. Таким образом, эти два
примера кода HTML на экране
дисплея будут показаны
одинаково:*

*предварительно
отформатированный текст*

*предварительно
отформатированный текст*

А именно:

*предварительно
отформатированный текст
предварительно
отформатированный текст*

*У этого тега существует
необязательный атрибут,
указывающий желаемый размер
строки в символах, а именно:*

Попробуем?

*Предмет истории есть жизнь
народов и человечества.*

*Непосредственно уловить и
обнять словом – описать жизнь
не только человечества, но
одного народа, представляется
невозможным.*

*Обычно, для отображения
предварительно
отформатированного текста
используются моноширинные
шрифты, все символы которого
имеют одинаковую ширину. Для
того, чтобы браузер "не*

*забыл" об этом следует
применить таблицы стилей.*

*Например, это можно сделать
так:*

```
pre { color: green;  
background: white; font-  
family: monospace; }
```

*Результат такого определения
стиля для тега*

вы можете видеть на этой странице.

	<hr/>

Если вы устанавливаете цвет текста, желательно также установить цвет фона. Это поможет вам избежать ситуации, когда буквы будут практически неразличимы на близком к ним по цвету фоне.

	<hr/>

***Совет № 2. Вместо того, чтобы
устанавливать цвет фона для
элемента***

, установите цвет фона для элемента . Например, это можно сделать так:

```
body { color: black;  
background: white; }
```

```
pre { color: green; font-  
family: monospace; }
```

Выделенный текст

В тех случаях, когда необходимо обратить особое внимание пользователя на тот или иной фрагмент текста,

*следует использовать
стандартные средства
форматирования.*

*Стандартные средства
форматирования представлены
специальными тегами, которые
обязательно являются парными
(т.е. имеются открывающий и
закрывающий теги). Чтобы
увеличить размер шрифта на
один пункт, используйте тег .*

*Чтобы выделить текст
полужирным шрифтом,
воспользуйтесь тегом
или тегом*

.

Чтобы выделить текст курсивом, воспользуйтесь тегом `<i>` или тегом ``.

Вы заметили, что в данном случае выделение текста тегами `<i>` и `` различаются? Объяснение простое – в HTML-коде этой страницы используется внедренная таблица стилей, в которой для тега `i` указан

*шрифт: полужирный
курсив.*

*Шрифт пишущей машинки
можно имитировать с
помощью тега `<small>`.*



*С помощью тега `<small>` размер шрифта
можно уменьшить на один пункт.*

*Существуют и другие теги, которые
также предназначены для выделения
текста.*

Кроме того, в HTML включена поддержка математических символов и научных обозначений. Для построения простейших равенств и выражений вам могут пригодиться два тега (нижний индекс) и (верхний индекс).

Например:

$A_2+B_2=C_2$

CO_2 =углекислый газ

В HTML-коде это записано так:



$$A^2+B^2=C^2$$

CO₂=углекислый газ



*Если вы намерены придерживаться
хорошего стиля программирования
web-страниц (с помощью HTML и
XHTML), то вместо тегов
форматирования используйте
каскадные таблицы стилей.*



	<hr/>



*На сегодняшний день многие
старые теги HTML, которые
предназначались специально для
выделения текста, вытеснены
новыми возможностями
форматирования с помощью
каскадных таблиц стилей.*



