

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ

ЯЗЫК РАЗМЕТКИ ГИПЕРТЕКСТА HTML

*Учебно-методическое пособие
по дисциплине
«ВЕБ-ПРОГРАММИРОВАНИЕ»*

**Набережные Челны
2018**

Галиуллин Л.А. Язык разметки гипертекста HTML: учебно-методическое пособие по дисциплине «Веб-программирование» [Электронный ресурс] / Казанский федеральный университет, Электронный архив, 2018.

Рассматривается язык гипертекстовой разметки документов HTML. Представлены общие принципы создания web-узла, история создания HTML. Приведены контрольные вопросы. Для студентов направлений подготовки «Информатика и вычислительная техника», «Программная инженерия».

Введение

Для создания и размещения в информационном пространстве WWW (**World Wide Web**, *Всемирная паутина*) собственного *Web*-узла надо предпринять шаги, чтобы он был интересен, полезен и, что немаловажно, посещаем. Первый вопрос, на который необходимо дать четкий ответ: с какой целью создается *Web*-узел? От этого зависит многое: *стиль оформления*, необходимые для создания и последующего функционирования *затраты*, формат представления информации для размещения в *Web*, *инструментарий* и требования, предъявляемые к программному обеспечению *Web*-сервера и каналам связи с *Internet*. Здесь возможно несколько вариантов.

Если вы создаете *Web*-узел для компании, реализующей какой-то *товар*, то основной целью может быть распространение информации о фирме и *реклама* продукции, а также организация *Web*-магазина. При этом будут решены следующие задачи:

- изменение имиджа и поднятие престижа компании;
- продвижение торговой марки;
- доступность информации о продукции и ценах для клиентов;
- поддержка дилерской сети, доступность информации о продукции и ценах для дилеров;
- прямая продажа продукции в *Internet*, организация *Web*-магазина;
- доступность внутренней информации для сотрудников, работающих вне офиса.

Другой вариант — создание *Web*-узла научной или общеобразовательной организации, не занимающейся коммерцией в *Internet*, а распространяющей информацию. В этом случае речь пойдет о сборе, переработке и размещении на *Web*-узле больших массивов данных с организацией поиска и доступа к ним.

И последний рассматриваемый вариант — когда вы считаете необходимым разместить в *Internet* свою личную страницу.

Для того, чтобы правильно ответить на поставленные

вопросы, необходимо сформировать категории пользователей, на которые рассчитан *Web*-узел. Исходя из их психологии, должна строиться информационная структура, которая будет привлекать и удерживать клиентов. В дальнейшем все вопросы о целесообразности каких-либо действий, связанных с *Web*-узлом, должны рассматриваться в соответствии с тем, как отреагируют на них посетители и насколько они будут способствовать достижению главной цели.

После того как сформулированы цели и определены категории пользователей, необходимо распределить подготовленную информацию по *Web*-документам, продумать связи между ними и предусмотреть дополнительные навигационные возможности, например, поисковую систему по содержанию *Web*-узла.

Общие принципы создания web-узла

Типичная структура *Web*-узла фирмы обычно представлена так:

Информация о компании. Следует рассказать о целях и деловом облике фирмы, ее истории и т.д. Покажите, какую выгоду получают клиенты от сотрудничества именно с вами, а не с другими компаниями.

Информация о продукции и услугах. Разместите на *Web*-странице фотографии или рисунки своей продукции. Опишите ее свойства и преимущества, приведите примеры использования. Если имеется бумажный каталог продукции, то можно перенести его структуру и содержание в *Web*-узел. Это облегчит создание и дальнейшее обновление электронного варианта каталога. Если планируется прием заказов на продукцию или услуги через *Internet*, то нужно разместить здесь бланк заказа, который будет поступать по электронной почте.

Информационная поддержка. В этом разделе публикуется дополнительная техническая информация, часто задаваемые вопросы, советы по устранению неисправностей и т.п.

Новости. Проинформируйте клиентов о новых товарах и услугах, предоставляемых фирмой, опубликуйте пресс-релизы и т.п.

Обратная связь. Сообщите, как с вами можно связаться, где вы находитесь. Поместите форму для отзыва, гостевую книгу, адреса электронной почты, на которые клиент может отправить *запрос*, и т.п.

При наполнении *Web*-узла всегда нужно помнить два принципа: уникальность и *достоверность* публикуемых материалов.

Уникальность является первоочередным требованием к содержанию. В WWW уже может существовать немало страниц с похожими материалами. Ваш *Web*-узел должен чем-то отличаться от серверов с аналогичной тематикой, хотя бы для того, чтобы привлечь к себе внимание. Наличие уникальных материалов на вашей странице увеличит ее посещаемость. Для того, чтобы создать уникальный информационный *ресурс*, не обязательно изобретать что-то принципиально новое, можно *по-другому* оформить уже существующие ресурсы, но при этом не заставлять клиента тратить много времени на их *поиск*. Проверить же ресурсы на уникальность можно с помощью поисковых серверов. Что касается авторитетности, то все зависит от того, насколько тщательно вы подберете информацию, проверите ее и будете своевременно обновлять.

При создании *Web*-узла необходимо помнить, что составляющие его отдельные документы должны быть объединены общим стилем оформления и средствами навигации. *Единый стиль оформления* — один из показателей, отличающих профессиональный *Web*-узел от любительского. Благодаря единообразно сделанным документам пользователи будут отличать ваш *Web*-узел от других и запомнят его. Это не значит, что документы должны быть похожи друг на друга как две капли воды, но общая идея, единый стиль, должны присутствовать непременно.

То же относится и к средствам навигации *по* страницам. Не стоит рассчитывать, что посетитель знает структуру *Web*-узла так же хорошо, как вы. Он должен без труда понимать, где он находится сейчас и как можно попасть в любое другое *место*. Необходимо предусмотреть возможность перехода к первому документу, программе поиска или к схеме *Web*-узла.

Кроме того, единство стиля позволяет использовать шаблоны

— страницы, содержащие только общие элементы оформления и навигации (без информационного наполнения). С их помощью можно быстро и эффективно создавать новые страницы и распределять работу *по* их созданию между несколькими людьми. При использовании шаблона для получения готовой страницы достаточно лишь внести в него необходимую информацию. Последовательность, логичность, постоянство — вот необходимые качества хорошего *Web*-узла. Значительно упростят работу *по* формированию и изменению стиля вашего *Web*-узла каскадные таблицы стилей, появившиеся в *HTML 4.0*. О некоторых их возможностях будет рассказано ниже, а полностью им посвящена отдельная глава.

После того как определены цели, задана структура и собрана текстовая и графическая *информация*, необходимо разработать внешний вид *Web*-узла. Он также зависит от тех целей, которые необходимо достичь. Спектр возможных решений здесь очень широк: от просмотра уже существующих страниц и создания подобных, до обращения за помощью к профессиональным дизайнерам и художникам. В то же время, необходимо помнить о некоторых уже сложившихся правилах построения *Web*-документов, из которых состоит *Web*-узел.

Структура. На сегодня *представление* о структуре документа достаточно устоялось. *Web*-документ должен содержать в себе следующие *разделы*: заглавие, название компании, навигационную панель, собственно содержание, контактную информацию, дату и время обновления, авторские *права* и статус документа.

Логотип. Создавая *Web*-страницу, необходимо позаботиться о том, чтобы название фирмы всегда присутствовало на экране. Для этого в начале каждого *Web*-документа обычно помещается красочно оформленный *логотип* фирмы. Кроме того, название компании должно присутствовать и в выходных данных ко всем документам.

Навигационная панель. Одним из наиболее важных разделов *Web*-документа является навигационная панель или панель управления. **WWW** завоевала весь мир во многом благодаря тому, что гипертекстовые ссылки обеспечивают полную *связность* публикуемых материалов. Но эти же ссылки

таят в себе опасность погружения в полный хаос, когда, пройдя цепочку из трех-четырёх документов, вы уже не сможете вернуться обратно, запутавшись в обилии ссылок. Ваш *Web*-узел должен обеспечивать пользователю ясные и интуитивно понятные навигационные маршруты.

Многочисленные исследования показали, что посетители *Web*-серверов очень нетерпеливы и дальше, чем на два уровня документов, углубляться в содержание сервера не хотят. Поэтому, создавая *Web*-узел большого объема, следует предусмотреть промежуточные документы, обычно находящиеся на первом-втором уровнях, от которых любая информация находится не далее, чем в двух переходах.

Навигационная панель вашего *Web*-узла должна присутствовать в каждом документе. В первую очередь, она должна включать в себя направляющие ссылки типа "Вперед"- "Назад" ("Следующий"- "Предыдущий"), указывающие на соседние документы в структуре *Web*-узла. Далее от панели управления обязательно должны идти ссылки на все крупные *разделы* *Web*-узла — так называемые *разделы* первого уровня. И, наконец, *пользователь* всегда должен иметь возможность мгновенно вернуться на главную страницу *Web*-узла. Помимо ссылок следует указать *путь* к локальной поисковой системе и индексу.

Содержание. Прежде всего, следует отметить, что содержание *Web*-документов должно в полной мере отвечать всем требованиям, предъявляемым к обычным газетным или журнальным публикациям: грамматическая и орфографическая *корректность*, *точность* и *идостоверность* предлагаемых материалов и многое другое. Кроме того, появляется *целый* ряд специфических требований, которым должен удовлетворять *Web*-документ.

Часто возникает вопрос о размерах документа: какое число страниц является оптимальным? Ответ на первый взгляд может показаться странным: одна экранная страница или вообще никаких ограничений. Многочисленные исследования показали, что пользователи не любят работать с полосами прокрутки браузеров. Больше всего им нравятся документы, которые размещаются на одной экранной странице. Так и в WWW — вы

никоим образом не сможете дать пользователю больше информации, чем в концентрированном изложении на одной странице. Если все-таки вы не укладываетесь в эти рамки, создайте еще один документ.

Одна экранная страница оказалась подходящей мерой представления информации. Если размер документа превышает одну страницу, то в большинстве случаев он может быть поделен на несколько логических частей, каждая из которых будет занимать не более одной страницы. Если же логического деления информации произвести не удастся, то необходимо переработать стиль изложения, а может быть, и сами материалы. Сейчас выработалось единое мнение, что *Web-server* необходимо строить на основе одноэкранных документов. Есть только два исключения из этого правила. Оно не распространяется на статьи, публикуемые в WWW, и второе *исключение* — анкетные формы, которые, естественно, нельзя разрывать.

Графика. При разработке *Web*-страницы нужно очень внимательно выбирать оптимальное соотношение графических и текстовых материалов. Одна хорошая картинка может заменить тысячу строк текста, но и загружаться *по* сети она будет в тысячу раз дольше. Поэтому графикой нужно пользоваться осторожно. Можно исходить из того, что графики на странице должно быть чуть меньше, чем хочется *Web*-мастеру. Пользователям может просто не хватить терпения, и они закроют документ еще до того как он полностью загрузится. Задержка отклика системы вызывает у пользователя раздражение. Все понимают, как тяжело сейчас обстоят дела с канальной инфраструктурой в *Internet*. Поэтому *время задержки* возрастает в зависимости от времени суток, *по* разным оценкам до 15-60 секунд. Теперь представьте, что у клиента только *модем* на 19200 бит/с. Тогда за минуту, то есть до того как клиент потеряет терпение, можно передать только около 170 Кбайт данных. Следовательно, размер документа не должен превышать этого значения.

Следует отметить, что обычно панель управления, *логотип* и название фирмы выполняются в виде графических элементов. После создания макета можно приступить к его реализации с

помощью языка *HTML* и иных средств, предлагаемых современными технологиями **WWW**.

Завершив создание *Web*-узла, необходимо разместить его в *Internet*. Здесь возможны два варианта: первый — использовать *компьютер*, который вместе с *Web*-сервером и *Web*-узлом находится в вашем офисе и подключается к *Internet no* выделенной или коммутируемой линии; второй — воспользоваться для размещения *Web*-узла услугами специальных организаций.

Рассмотрим второй вариант. Правильный выбор провайдера, предоставляющего *доступ* к *Web*-странице, позволит вашим клиентам с максимальным удобством получать необходимую информацию. Кроме того, *поддержка Web*-сервером специальных возможностей значительно облегчит разработку *Web*-узла.

На что следует обратить внимание при выборе провайдера, размещающего ваш *Web*-узел на своем сервере?

Пропускная способность каналов. Чтобы вашим посетителям не пришлось слишком долго ждать загрузки страниц, *провайдер* должен обладать надежным высокоскоростным соединением порядка 1-2 Мбит в секунду.

Поддержка сервером провайдера SSI (*Server Side Includes*, вставки на стороне сервера). Использование *SSI* позволяет *Web*-серверу вставлять небольшие объемы динамических данных непосредственно в пересылаемый пользователю *HTML* - документ. Запрошенная *HTML* -страница "просматривается" в поисках элементов *SSI*. Обнаружив такой элемент, *сервер* вставляет требуемую динамическую информацию. С помощью *SSI* можно включать один *файл* в состав другого, исполнять *CGI*-сценарии и передавать другую информацию. Необходимо уточнить, какие именно функции *SSI* поддерживаются на сервере провайдера.

Поддержка сервером провайдера CGI-сценариев. *CGI* (*Common Gateway Interface*, общий шлюзовой *интерфейс*) — спецификация, позволяющая *Web*-серверу выполнять произвольные прикладные программы. В результате работы таких программ (сценариев, или "скриптов") создаются *HTML* -документы. С помощью *CGI*-сценариев могут

приниматься данные от пользователя, они позволяют организовать диалог на *Web*-страницах, запросы к базам данных и т.д. Создать *CGI-сценарий* можно с помощью любого популярного языка программирования: Perl, *Basic*, C, C++, *Pascal* и т.п.

Поддержка моментальной перекодировки. К сожалению, для русского языка в *Internet* при работе на разных платформах (*Windows*, *Mac*, *Unix* и т.д.) приняты различные кодировки. Чтобы пользователю было легко просматривать страницы, *Web-сервер* провайдера должен уметь автоматически перекодировать документы в зависимости от поступившего запроса. В противном случае либо содержание вашего *Web*-узла для некоторых посетителей будет нечитаемым, либо придется обеспечивать несколько копий *Web*-узла — *no* одной на каждую поддерживаемую кодировку.

Способ обновления страниц. Обычно страницы обновляются *no* протоколу *FTP* (*File Transfer Protocol*, протокол передачи файлов). Некоторые *FTP*-клиенты позволяют работать с файлами на компьютере провайдера так же, как с собственным диском, — копировать, удалять, переименовывать и т.п.

Как правило, возможность размещения *Web*-узла *провайдер* предоставляет своим пользователям за небольшую плату или бесплатно.

Существуют службы, которые предоставляют *место* под *Web*-узлы бесплатно вместе с адресом электронной почты и другими услугами. Как правило, условием такого "бесплатного" размещения является выделение на ваших страницах некоторого места под рекламу, кроме того, накладываются ограничения на размер ваших файлов.

История развития HTML

В 1989 году Тим Бернерс-Ли предложил руководству Европейского Центра ядерных исследований (*CERN*) проект распределенной гипертекстовой системы, которую он назвал *World Wide Web* (*WWW*), *Всемирная паутина*. Первоначально идея системы состояла в том, чтобы при помощи гипертекстовой навигационной системы объединить все

множество информационных ресурсов *CERN* в единую информационную систему. Технология оказалась настолько удачной, что дала толчок к развитию одной из самых популярных в мире глобальных информационных систем. Практически в сознании большинства пользователей глобальной компьютерной сети *Internet* сама эта *сеть* ассоциируется с тремя основными информационными технологиями:

- электронная почта (e-mail);
- файловые архивы FTP;
- *World Wide Web*.

Причем последняя технология постепенно перемещается на первое *место*.

Успех технологии *World Wide Web* определен двумя основными факторами: простотой и использованием протоколов межсетевого обмена семейства *TCP/IP* (*Transmission Control Protocol*, протокол управления передачей/*Internet Protocol*, протокол *Internet*), которые являются основой *Internet*.

Практически все пользователи Сети одновременно получили возможность попробовать себя в качестве создателей и читателей информационных материалов, опубликованных во Всемирной паутине. Но и популярность самого *Internet* во многом вызвана появлением *World Wide Web*, так как это первая *сетевая технология*, которая предоставила пользователю простой современный *интерфейс* для доступа к разнообразным сетевым ресурсам. Простота и удобство применения привели к росту числа пользователей **WWW** и привлекли внимание коммерческих структур. Далее процесс роста числа пользователей стал лавинообразным, и так продолжается до сих пор.

При этом сама технология на начальном этапе была чрезвычайно проста. Дело в том, что при разработке различных компонентов технологии (языка *гипертекстовой разметки HTML* (*HyperText Markup Language*, язык разметки гипертекста), протокола обмена гипертекстовой информацией *HTTP*, спецификации разработки прикладного программного обеспечения *CGI* и др.) предполагалось, что квалификация авторов информационных ресурсов и их

оснащенность средствами вычислительной техники будут минимальными.

Одним из компонентов технологии создания распределенной гипертекстовой системы *World Wide Web* стал язык гипертекстовой разметки *HTML*, разработанный Тимом Бернерсом-Ли на основе стандарта языка разметки печатных документов — *SGML (Standard Generalised Markup Language, стандартный обобщенный язык разметки)*. Дэниел В. Конноли написал для него *Document Type Definition*— формальное описание синтаксиса *HTML* в терминах *SGML*.

Разработчики *HTML* смогли решить две задачи:

- предоставить дизайнерам гипертекстовых баз данных простое средство создания документов;
- сделать это средство достаточно мощным, чтобы отразить имевшиеся на тот момент представления об интерфейсе пользователя гипертекстовых баз данных.

Первая задача была решена за счет выбора *теговой* модели описания документа. Такая модель широко применяется в системах подготовки документов для печати. Примером такой системы может служить хорошо известный язык разметки научных документов *TeX*, который был создан Дональдом Кнутом и предложен Американским математическим обществом, и программы его интерпретации.

Язык HTML позволяет размечать *электронный документ*, который отображается на экране с полиграфическим уровнем оформления; результирующий документ может содержать самые разнообразные метки, иллюстрации, аудио- и видеофрагменты и так далее. В состав языка вошли развитые средства для создания различных уровней заголовков, шрифтовых выделений, различные списки, таблицы и многое другое.

Вторым важным моментом, повлиявшим на судьбу *HTML*, стало то, что в качестве основы был выбран обычный текстовый *файл*. Выбор был сделан под влиянием следующих факторов:

- такой файл можно создать в любом текстовом редакторе на любой аппаратной платформе в среде какой угодно операционной системы;
- к моменту разработки *HTML* существовал американский

стандарт для разработки сетевых информационных систем — *Z39.50*, в котором в качестве единицы хранения указывался простой текстовый файл в кодировке *LATIN1*, что соответствует *US ASCII*.

Таким образом, гипертекстовая база данных в концепции **WWW** — это набор текстовых файлов, размеченных на языке *HTML*, который определяет форму представления информации (разметка) и структуру связей между этими файлами и другими информационными ресурсами (гипертекстовые ссылки). Гипертекстовые ссылки, устанавливающие связи между текстовыми документами, постепенно стали объединять самые различные информационные ресурсы, в том числе звук и видео; в результате возникло новое понятие — *гипермедиа*.

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

Первая версия языка (*HTML 1.0*) была направлена на представление языка как такового, где описание его возможностей носило скорее рекомендательный характер. Вторая версия языка (*HTML 2.0*) фиксировала практику использования его конструкций. Версия ++ (*HTML++*) представляла новые возможности, расширяя набор *тегов HTML* в сторону отображения научной информации и таблиц, а также улучшения стиля компоновки изображений и текста. Версия 3.2 смогла упорядочить все нововведения и согласовать их с существующей практикой. *HTML 3.2* позволяет реализовать использование таблиц, выполнение кодов языка *Java*, обтекание графики текстом, а также *отображение* верхних и нижних индексов.

Контрольные вопросы

1. Что Вы знаете о HTML?
2. Что Вы знаете об общих принципах создания web-узла?
3. Что Вы знаете о способах обновления страниц?
4. Что Вы знаете о поддержке моментальной перекодировки?
5. Что Вы знаете о поддержке CGI-сценариев?
6. Что Вы знаете о поддержке SSI?
7. Что Вы знаете о пропускной способности каналов?
8. Что Вы знаете о навигационной панели?
9. Что Вы знаете о HTTP?
10. Что Вы знаете о хостинге?

Рекомендуемые источники

1. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2017. - 416 с. [Электронный ресурс]. <http://znanium.com/bookread.php?book=336649>.
2. Гагарина Л.Г. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2015. - 400 с. [Электронный ресурс]. <http://znanium.com/bookread.php?book=389963>.
3. Голицына О. Л. Программирование на языках высокого уровня: Учебное пособие / О.Л. Голицына, И.И. Попов. - М.: Форум, 2016. - 496 с. [Электронный ресурс]. <http://znanium.com/bookread.php?book=139428>.