Содержание:

Введение

Цель курсовой работы – изучение и разбор основных функция языка, рассмотреть особенности данного языка.

Задача курсовой работы – рассмотреть основные функции языка, оценить разницу между HTML5 and XML.

На данный момент прогресс в развитие html закончилось – все необходимые теги уже есть другие теги уже не нужны, поскольку хватает существующих, к тому же акцент разработки веб-страниц сместился на стили, которые расширяют возможности по оформлению документов.

Каскадная таблица стилей (CSS) не заменяет HTML, но может позволить использовать ограниченный набор тегов, в виде элементов, их расположение и разные параметры задавать через стили.

В HTML ограничениями является и то, что он относится к формальным языкам, в том смысле, что теги и их иерархическая структура жестко описаны в спецификации.

В итоге популярность набирает XML, с помощью него можно создавать свои теги и создавать их структуру. Различие между HTML и XML состоит как в тегах, но еще правилах создания кода. Браузер в работе с HTML закрывает глаза на различные маленькие ошибки и недочеты в структуре или тому, если не верно указан параметр. С XML дела обстоят по-другому, браузер выдаст, что документ сформирован неправильно.

Для того, чтобы разработчики "правильно" мыслили, изменить их стиль написания кода, а также сократить разрыв между HTML и XML, и был разработан XHTML, как промежуточный этап между ними.

XHTML (Расширенный язык разметки гипертекста) предназначен для замены HTML и считается его более строгой версией. Вообще, W3C определяет XHTML как последнюю версию HTML, которая постепенно его вытеснит. Так ли это будет

обстоять на самом деле, покажет только время.

По этим причинам XHTML 1. x является всего лишь подобием HTML, но с более строгим синтаксисом, а не тем перспективным языком разметки, на который обязательно стоит переходить из-за его уникальных возможностей. Об этом языке и пойдет речь в данной курсовой работе.

ГЛАВА 1.

1.1 Основная суть XHTML

XHTML - язык разметки веб-страниц, по возможностям сопоставимый с HTML, однако является подмножеством XML. Как и HTML, XHTML соответствует спецификации SGML. Вариант XHTML 1.1 одобрен в качестве Рекомендации Консорциума Всемирной паутины (W3C) 31 мая 2001 года.

XHTML представляет собой семейство имеющихся на данный момент и могущих появиться в будущем типов документов и модулей, являющихся копиями, подмножествами или расширениями языка HTML 4. Семейство типов документов XHTML базируется на XML и предназначено для работы с пользовательскими агентами на базе. Более подробную информацию об этом семействе и его эволюции можно найти в разделе "Направления развития".

XHTML 1.0 представляет первый тип документов семейства XHTML. В ней три типа документов HTML 4 переформатируются в терминах XML 1.0. Она предназначена для использования в качестве языка содержимого, как соответствующего XML, так и, если соблюдены некоторые простые требования, работающего в конформных пользовательских агентах HTML 4. Разработчики, переносящие свои документы в XHTML 1.0, получат следующие преимущества:

Документы XHTML соответствуют XML. Как таковые они без труда просматриваются, редактируются и проверяются на корректность стандартными средствами XML.

Документы XHTML[1] могут работать лучше, чем они работали в существующих пользовательских агентах, соответствующих HTML 4, а также в новых

пользовательских агентах, соответствующих XHTML 1.0.

Документы XHTML могут использовать прикладные программы, базирующиеся на HTML Document Object Model или XML Document Object Model.

По мере расширения семейства XHTML документы, соответствующие XHTML 1.0, будут с большей вероятностью совместимы с различными средами XHTML.

Семейство XHTML является следующим шагом в эволюции Интернет. Переходя сегодня на XHTML, разработчики содержимого (контента) могут вступить в мир XML со всеми его преимуществами, сохраняя при этом совместимость содержимого с более старыми и более новыми версиями.

Преимущества XHTML.

Для XHTML можно применять множество технологий, разработанных для XML. Например, XSLT и XPath.

Анализ XHTML проще и быстрее, чем HTML. Поскольку синтаксис XML строже, чем SGML, обработка XHTML возможна даже на мобильных телефонах с малыми ресурсами.

Различия между XHTML и HTML.

Все элементы должны быть закрыты. Теги, которые не имеют закрывающего тега (например, <img> или
) должны иметь на конце / (например,
).

Булевы атрибуты записываются в развёрнутой форме. Например, следует писать <option selected="selected"> или <td nowrap="nowrap">.

Все значения атрибутов обязательно должны быть заключены в двойные, либо одинарные кавычки.

Имена тегов и атрибутов должны быть записаны строчными буквами (например, вместо <IMG ALT="" />).

XHTML гораздо строже относится к ошибкам в коде; < и & везде, даже в URL, должны замещаться < и & соответственно. По рекомендации W3C браузеры, встретив ошибку в XHTML, должны сообщить о ней и не обрабатывать документ.д.ля HTML браузеры должны были попытаться понять, что хотел сказать автор.

Кодировкой по умолчанию является UTF-8 (в отличие от HTML, где кодировкой по умолчанию является ISO 8859-1).

Отличия переходного (англ. transitional) XHTML от HTML незначительны и предназначены лишь для приведения его в соответствие с XML. Остальные версии отличаются лишь набором тегов.

В том случае, если MIME-тип - text/html (а это чаще всего и есть значение по умолчанию), все современные браузеры поддерживают XHTML. Он также совместим и со старыми браузерами, т.к в основе XHTML лежит HTML.

Однако если автор страницы задал MIME-тип как application/xhtml+xml, браузер Internet Explorer 6 не сможет обрабатывать страницу, поскольку у него нет XML-парсера. Это одна из причин, замедляющих процесс перехода от HTML к XHTML.

Впрочем, проблема с MIME-типом легко решается при помощи простого PHP-скрипта, меняющего пресловутый MIME-тип в зависимости от пользовательского агента.

Версии XHTML.

XHTML 1.0 Переходный (Transitional): предназначен для лёгкой миграции из HTML 3.2 и для тех, кто использует инлайн-фрэймы.

XHTML 1.0 Строгий (Strict): полностью отделяет содержание документа от оформления (которое теперь задаётся только через CSS), многие атрибуты (такие как, например, bgcolor и align) более не поддерживаются, их поведение можно задавать только через таблицу стилей.

XHTML 1.0 Фреймовой: используется, если необходимо разделить окно браузера на несколько фреймов.

XHTML 1.1 Модульный: авторы могут импортировать дополнительные свойства в их разметку.

XHTML Основной: специальная облегчённая версия XHTML для устройств, которые не могут использовать полный набор элементов XHTML - в основном используется в миниатюрных устройствах, таких как мобильные телефоны. Подразумевается, что он заменит WML и C-HTML.

XHTML мобильного профиля: основанный на XHTML Basic, добавляет специфические элементы для мобильных телефонов.

XHTML[2] 2.0. Пока в разработке. Синтаксис еще больше приближен к синтаксису XML. Также является модульным языком.

1.2 Валидация XHTML документов

Валидным XHTML-документом считается документ, удовлетворяющий технической спецификации. В идеале, все браузеры должны следовать веб-стандартам и, в соответствии с ними, валидные документы должны отображаться во всех браузерах на всех платформах. Валидация XHTML-документа рекомендована даже несмотря на то, что она не гарантирует кросс-браузерности. Документ должен быть проверен на соответствие спецификации с помощью онлайновой Службы валидации разметки W3C. Валидация обнаружит и разъяснит ошибки в XHTML-разметке.

Валидный документ должен содержать определение типа документа (DTD). DTD должен быть расположен до всех других элементов документа. Вот наиболее распространённые типы DTD для XHTML:

XHTML 1.1

<! DOCTYPE html PUBLIC "- // W3C // DTD XHTML 1.1 // EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11. dtd">

Основными ошибками в XHTML-разметке являются:

Незакрытые элементы (XHTML, в отличие от HTML, требует закрытия всех элементов, в том числе не имеющих закрывающего тега, как, например,
).

Отсутствие альтернативных текстов для изображений (достигающийся применением атрибута alt, который помогает сделать документы доступнее для устройств, которые не в состоянии отображать изображения, или предназначенных для слабовидящих людей).

Присутствие текста непосредственно в теге <body> документа (должен быть объявлен блочный элемент, внутрь которого следует помещать содержимое).

Вложение блочных элементов внутрь инлайновых (внутристрочных) (например, блочные элементы или <p> не могут быть вложены внутрь инлайновых элементов <a>, , и так далее).

Пренебрежение заключением значений атрибутов в кавычки (<a href=http://www.ru/> вместо <a href="http://www.ru/">).

Неправильное вложение элементов (конструкции вида   ).

Написание тегов и/или атрибутов прописными буквами (<DIV STYLE="..."> вместо).

Задание в теге <! DOCTYPE ...> относительного пути к DTD-файлу, скопированное с сайта www.w3.org ("DTD/xhtml11. dtd" вместо "http://www.w3.org/TR/xhtml11/DTD/xhtml11. dtd").

Преимущества перехода на XHTML[3] 1.0 описаны выше. Вот несколько основных преимуществ:

Разработчики документов и создатели пользовательских агентов постоянно открывают новые способы выражения своих идей в новой разметке. В XML ввод новых элементов или атрибутов достаточно прост. Семейство XHTML разработано так, чтобы принимать расширения путем модулей и технологий XHTML для разработки новых соответствующих XHTML модулей (описанных в готовящейся спецификации Модуляризации XHTML). Модули позволят комбинировать существующие и новые наборы функций при разработке содержимого и создании новых пользовательских агентов.

Постоянно вводятся альтернативные методы доступа в Интернет. По некоторым оценкам, в 2010 году 95% обращений к документам в Интернет будет выполняться с альтернативных платформ. Семейство XHTML создавалось с учетом общей совместимости пользовательских агентов. С помощью нового механизма профилирования пользовательских агентов и документов серверы, прокси и пользовательские агенты смогут преобразовывать содержимое наилучшим образом. В конечном счете станет возможной разработка соответствующего XHTML [4] содержимого, пригодного для любого соответствующего XHTML пользовательского агента.

1.3 Строго конформные документы

Строго конформный документ XHTML - это документ, которому необходимы только возможности, описанные в настоящей спецификации как обязательные. Такой документ должен соответствовать всем следующим критериям:

Документ обязательно должен пройти проверку на коррекность в связи с одним из трех приложений DTD в приложение A.

Корневой элемент документа обязательно должен являться элемент <html>.

Корневой элемент документа должен назначать пространство имен XHTML с использованием атрибута xmlns. Пространство имен для XHTML определено в http://www.w3.org/1999/xhtml.

В документе до корневого элемента должно иметься объявление DOCTYPE. Открытый идентификатор, включаемый в объявление DOCTYPE, должен ссылаться на одно из трех DTD, приведенных в приложении A, с помощью соответствующего формального открытого идентификатора. Системный идентификатор может изменяться, отражая соглашения, принятые в локальной системе. Вот пример минимального документа XHTML.

<? xml version="1.0" encoding="UTF-8"? >

<! DOCTYPE html

PUBLIC "- // W3C // DTD XHTML 1.0 Strict // EN"

"DTD/xhtml1-strict. dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml: lang="ru" lang="ru">

<head>

<title>Виртуальная библиотека</title>

</head>

<body>

Переехала по адресу vlib.org.

</body>

</html>

Как было отмечено ранее, что бы докумен был валдиным он должен быть верно написан и прост. В данном торговом предприятия необходимо ввести штатную должность маркетолога, который будет постоянно заниматься исследованием рынка, подготовкой различных мероприятий и рекламированием товара.

Если продукция будет хорошо рекламироваться, то предприятие станет наиболее известным и это привлечёт новых клиентов, в результате можно ожидать увеличения объемов продаж и, как следствие, непосредственное увеличение прибыли.

Также стоит привлечь к работе в сети оптик аналитика, который будет заниматься анализом рынка, прогнозировать спрос на продукцию и вырабатывать стратегию предприятия в целом. Необходимо разработать системы написания кода ее структар и проведение различных тестов. Эти методы стимулирования тестирования должны положительно сказаться на результатах деятельности сети, в частности привести правильный работе.

Обратите внимание, что в данном примере включено объявление XML. Такое объявление XML[5] не является обязательным для всех документов XML. Авторам документов XHTML настоятельно рекомендуется использовать объявления XML во всех своих документах. Такое объявление обязательно, если кодировка символов документа отличается от используемых по умолчанию UTF-8 или UTF-16.

ГЛАВА 2.

2.1 Использование XHTML с другими пространствами имен

Пространство имен XHTML может использоваться с другими пространствами XML в соответствии с XMLNAMES, хотя такие документы не являются строго конформными XHTML 1.0 в соответствии с приведенным выше определением. В будущих работах W3C будут определены способы указания конформности документов, в которых используется несколько пространств имен.

В следующем примере показано, как XHTML 1.0 может использоваться с рекомендацией MathML:

<html xmlns="http://www.w3.org/1999/xhtml" xml: lang="ru" lang="ru">

<head>

<title>Пример Math</title>

</head>

<body>

Далее приводится разметка MathML:

<math xmlns="http://www.w3.org/1998/Math/MathML">

<apply> <log/>

<logbase>

<cn> 3 </cn>

</logbase>

<ci> x </ci>

</apply>

</math>

</body>

</html>

Конформный пользовательский агент должен соответствовать всем следующим критериям:

Для соответствия рекомендации XML 1.0 [XML] пользовательский агент должен разбирать документ XHTML и оценивать его правильность. Если пользовательский агент выполняет проверку на правильность, он должен также проверять документы на соответствие с DTD, на которые они ссылаются, в соответствии с [XML].

Если пользовательский агент поддерживает возможности, определенные в настоящей спецификации или обязательные согласно нормативной ссылке, он должен это делать в соответствии со способами, описанными в определении этой возможности.

Если пользовательский агент обрабатывает документ XHTML как общий документ XML, он должен распознавать только атрибуты типа ID (например, атрибут id большинства элементов XHTML) в качестве идентификаторов фрагментов.

Если пользовательский агент встречает значение атрибута, которое он не распознает, он должен использовать значение атрибута по умолчанию.

Если пользовательский агент встречает ссылку на объект (отличный от заранее определенных объектов), для которой он не обрабатывал объявления (что могло произойти, если объявление расположено во внешнем подмножестве, которое пользовательский агент не прочел), ссылка на объект должна генерироваться в виде символов (начиная с амперсанда и заканчивая точкой с запятой), составляющий ее.

Во время генерации содержимого пользовательские агенты, если они встречают распознаваемые, но не генерируемые символы или ссылки на символьные объекты, должны представлять документ таким образом, чтобы пользователю было понятно, что корректная генерация была невозможна.

Следующие символы определены в XML как пробельные:

пробел ( )

табуляция (	)

возврат каретки (
)

перевод строки (
)

Процессор XML приводит коды конца строки, различные в различных в системах, в одному символу перевода строки, который передается в приложение. Пользовательский агент XHTML, кроме того, должен обрабатывать как пробельные следующие символы:

перевод страницы ()

пробел нулевой ширины ()

В элементах, в которых для атрибута 'xml: space' установлено значение 'preserve', пользовательский агент должен сохранять все пробельные символы (за исключением начальных и конечных пробельных символов, которые должны удаляться). В противном случае пробелы должны обрабатываться по следующим правилам:

Все пробельные символы, окружающие элементы блока, должны удаляться.

Комментарии удаляются полностью и не влияют на обработку пробелов. Один пробельный символ в начале и в конце комментария обрабатывается как два пробела.

Начальные и конечные пробельные символы внутри элемента блока должны быть удалены.

Символы перевода строки в элементе блока должны быть преобразованы в пробел (если для атрибута 'xml: space' не установлено значение 'preserve').

Последовательность пробельных символов должна сокращаться до одного пробела (если для атрибута 'xml: space' не установлено значение 'preserve').

Любой XHTML-файл состоит из трех разделов - тега <! DOCTYPE>, заголовка (<HEAD>) и тела документа (<BODY>). Последние два элемента перекочевали из HTML и ничем не отличаются от своего родоначальника (пример 1).

Пример 1. Простейший XHTML документ.

<! DOCTYPE html PUBLIC "- // W3C // DTD XHTML 1.0 Transitional // EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional. dtd"> <html xmlns="http://www.w3.org/1999/xhtml"> <head> <! - Этот раздел предназначен для заголовка страницы и технической информации. - -> </head> <body> <! - А здесь надо размещать все, что хочется увидеть на странице. - -> </body> </html>

Тег <! DOCTYPE> сообщает браузеру о типе текущего документа и как его интерпретировать. Различают несколько версий и типов XHTML-документов, они приведены в табл.1.

Версия XHTML	Тип документа	Описание

XHTML 1.0	Strict	"Строгое" описание документа, включающее все правила.
XHTML 1.0	Transitional	"Переходный" тип, более лояльно относящийся к коду документа.
XHTML 1.0	Frameset	Устанавливается при использовании на странице фреймов.
XHTML1.1	XHTML 1.1	Эта версия основана на XHTML 1.0 Strict, но понимается браузерами как XML-приложение. В первую очередь предназначено для работы с различными медиа-данными.

Таблица.1. Версии XHTML и допустимые типы документа

Пример 2. Документ со строгой разметкой

<! DOCTYPE html PUBLIC "- // W3C // DTD XHTML 1.0 Strict // EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict. dtd"> <html xmlns="http://www.w3.org/1999/xhtml"> <head>... </head> <body>... </body> </html>

XHTML 1.0 Transitional

Обычно применяется, когда правило разделения оформления и содержания выполняется не в полной мере. В этом случае допускается в коде документа использовать теги физического проектирования (например, тег <TT>) и лишь частично стили. В примере 3 показан вид тега <! DOCTYPE> для подобных документов.

Пример 2.1 "Переходный" документ

<! DOCTYPE html PUBLIC "- // W3C // DTD XHTML 1.0 Transitional // EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional. dtd"> <html xmlns="http://www.w3.org/1999/xhtml"> <head>... </head> <body>... </body> </html>

Тег <! DOCTYPE> хотя и обязателен, но не является непосредственной частью XHTML-документа, поэтому для него закрывающего тега не требуется.

XHTML 1.1

На основе проведённой работы мы можем сделать вывод, что наиболее верным написания кода являются соответствие со стандартом. Для наиболее эффективного анализа стоит применять не только тестирование, но и качественные методы проверка написания.

При анализе структуры HTML документа были использованы узкий круг показателей. Возможно, это привело к некоторым ошибкам ситуации в написание. В качестве рекомендации необходимо проводить такой анализ каждый раз, когда структура страницы меняется, это поможет наиболее эффективно регулировать и управлять index.html документом.

Стоит отметить то, что данный тест не является совершенным и для получения наиболее точных данных об ошибках необходимо проводить оценку тестирования страницы по сравнению с наиболее популярными сайтами, которые могут провести анализ html документа, которые занимаются анализом сайтов. Также необходимо использовать качественный метод тестирования, проводить оценку целевой аудитории пользователей, которые будут вносить какие-то данные, всё это влияет на работу сайта, в частности html документа.

Тег <HTML> является корневым для остальных элементов веб-страницы и располагается сразу после определения типа документа <! DOCTYPE>. Поскольку тип документа может быть любым, а не только тем, что приведен в табл.1, то необходимо дать понять браузеру, что он имеет дело со спецификой XHTML. Для чего в тег <HTML> добавляется параметр xmlns.

2.2 Будущее HTML

Ключевой задачей XHTML 1.0 был переход HTML на присущий XML лексика.

Данная спецификация ввела лимитированы синтаксиса XML в HTML:
регистр подневольность, принудительное заточение в двойные кавычки значений атрибутов и симметричные тэги.

Этим образом XHTML 2.0 пробует решить трудности HTML как языка для разметки web-страниц.

В собственной демонстрации на конференции XTech 2005 в Амстердаме, со-трутник W3C Стивен Пембертон высказал цели плана XHTML 2.0:
Применить XML всюду, где это вполне вероятно:

В случае если функция языка уже есть в XML, она не дублируется и не задумывается повторно;
В начале конструкция, вслед за тем представление: Спасибо таблицам стилей CSS более нет надобности в очевидно презентационных тэгах в HTML;
Устроить HTML легче в написании: Убрать кое-какие бессмысленные идиосинкразии HTML;

Более доступности, аппаратной независимости: Устроить столько догадок, сколько вполне вероятно, о методе, коим документ станет прочитан;

Усовершенствованные формы:

Нужные давным-давно назревшие усовершенствования!

Понизить надобность применения сценариев:

Подключить типичные использования сценариев в синтаксис самого HTML;
Усовершенствованная семантика: Облегчить интеграцию HTML с семантическими web-приложениями.

Разделы и параграфы.

Я испытываю некоторое удивление по поводу текстовых структурных элементов этого языка. Зачем нужны шесть уровней заголовков, и в какой ситуации можно было бы применить их все? Итак, почему бы заголовкам каким-либо образом не включать разделы, которые они озаглавливают? У XHTML[6] 2.0 есть ответ на этот вопрос благодаря новым элементам <section> и <h> (заголовок):

<section>

<h>Level 1 heading</h>

...

<section>

<h>Level 1 heading</h>

...

</section>

</section>

Это более логичное структурирование, чем в XHTML 1.0, и оно будет ближе пользователям многих других словарей. Одно из существенных преимуществ для программистов заключается в том, что они смогут включать разделы материалов прямо в документ, при этом не нужно переделывать номера уровней заголовков.

После этого для заголовков можно использовать стили CSS. Хотя ожидается, что реализация XHTML 2.0 в браузерах по умолчанию будет включать кое-что из описанного здесь как предварительно заданные параметры, но если написать их явным образом, они могут выглядеть примерно так (если абстрагироваться от спецификации XHTML 2.0):

h {font-family: sans-serif; font-weight: bold; font-size: 200%}

section h {font-size: 150%} /* Заголовок второго уровня */

section section h {font-size: 120%} /* Заголовок третьего уровня */

Еще одно логическое несоответствие в XHTML 1.0 заключается в том, что необходимо закрыть абзац, чтобы использовать список. По сути, абзацы следует закрывать для использования любых блочных элементов (блоков цитат, преформатированных разделов, таблиц и т.д.). Поступать так не всегда логично, если материал может вполне правомерно использоваться как часть одного абзаца. XHTML 2.0 устраняет это ограничение. Единственное, чего нельзя делать - это помещать один абзац в другой.

Изображения.

Тэг <img> в HTML на самом деле имеет довольно ограниченную гибкость. Как отмечает Пембертон, он не предоставляет никакого механизма нейтрализации ошибки, за исключением альтернативного текста alt (что препятствует утверждению новых форматов изображения), текст в тэгах alt нельзя разметить, а атрибут longdesc никогда не войдет в моду из-за своей неуклюжести. (longdesc используется для того, чтобы указать URI более полного описания изображения, чем то, что приводится в атрибуте alt)

В XHTML 2.0 появляется элегантное решение этой проблемы: возможность *любому* элементу иметь атрибут src. Браузер впоследствии заменяет содержимое элемента содержимым, которое находится по указанному URI. В самом простом случае это изображение. Но нигде не сказано, что это не может быть SVG, XHTML или любой другой тип содержимого, которое способен интерпретировать браузер.

Тэг <img> как таковой остается, но теперь он может включать содержимое. Новое действие атрибута src означает, что текст alt теперь представляет собой содержимое элемента, как в этом примере разметки:

<imgsrc="http://example.com/water. png">H₂O</img>

Это особенно приятная новость для таких языков, как японский для которого комментарии Ruby требуют строчной разметки, которая до этого

Расширяемая семантика.

HTML[7] всегда имел некоторые элементы с семантическими ассоциациями, например, <address> и <title>.

Проблема заключается в том, что таких элементов мало, и они не являются расширяемыми. Между тем, предпринимались попытки использовать атрибут class для того, чтобы привнести семантику в элементы HTML.

Это расширение задач атрибута class по сравнению с тем, для чего он был создан; такое применение атрибута затрудняется тем, что он преимущественно используется для применения стилей CSS. (Некоторых людей раздражает такое утверждение задачи атрибута class, но со вторым применением трудно спорить).

Идя еще дальше этих особых методов, XHTML 2.0 вводит метод для описания метаданных типа RDF в документе. Выражения RDF составляются из троек (субъект, свойство, объект).

Например, может существовать тройка: "мой автомобиль", " изображен", "красным цветом".

И, наконец, третье значение в тройке является содержимым элемента, к которому применяются атрибуты about и property, а если эти атрибуты пусты, то значением атрибута content.

Вот пример применения, которое близко к существующему использованию тэга HTML <meta> tag, определяет создателя.

2.3 особенности XHTML 2.0.

Если вам надоело писать <pre><code>... </code></pre>? Теперь можно использовать новый элемент <blockcode>.

Для обеспечения доступности XHTML 2.0 теперь предлагает атрибут role, который можно определить в теле элемента. Например, скудные элементы навигации на странице могут иметь атрибут role="navigation", чтобы механизм речевого воспроизведения текста мог интеллектуально их обработать.

Браузеры в настоящее время поддерживают некоторое перемещение фокуса при помощи клавиши Tab, но это может быть произвольным. Новые атрибуты nextfocus и prevfocus позволяют управлять порядком, в котором фокус будет перемещаться между элементами окна; это может быть важной функцией при создании пользовательских интерфейсов с навигацией.

Подготовка к XHTML 2.0.

Не обращая внимания на размах перемен в будущем, в XHTML 2.0 все ещё возможно признать HTML. Но в нем есть свежие составляющие, почти все в XHTML 2.0 трудится, как прежде. Составляющие - сберегаются в качестве меры обеспечения сопоставимости, как и веществ.

В прочем задача XHTML 2.0 заключается не в сохранении жесткой обратной сопоставимости синтаксиса, в следствие этого интерпретаторы HTML в современных браузерах не сумеют абсолютно преодолеть с выразительными способами документов XHTML 2.0. Что не наименее, основная масса web-браузеров сейчас отлично управляются с случайной интерпретацией XML-плюс-CSS, а почти все из XHTML 2.0 имеет возможность быть интерпретировано данным методикой - в том числе и в случае если вы при данном не получите семантических улучшений.

Кое-какие из различий XHTML 2.0 довольно немаловажны - переход к XForms считается одним из самых видимых, как и абсолютный отказ от не-XML наследства HTML. В следствие этого вы не сможете перевести ваши веб-сайты на управление XHTML 2.0 напрямик в данный момент, но несмотря на все

вышесказанное сможете устроить приготовления на будущее:

В случае если вы до сих времен не создали сего, научитесь трудиться с XHTML 1.0.

Сейчас вполне вероятно управление страничками XHTML 1.0 как простым кодом HTML, в случае если они сделаны в согласовании с советами по совместимости XHTML 1.0 HTML, но при данном имеют все шансы появиться сложности.

Поэкспериментируйте с браузером, который приглашает помощь XHTML 2.0 в одном ряду с ведущими способами SVG, XForms, и SMIL 2.0;
В случае если вы делаете свежие клиентские системы на основе XHTML-подобных функций, всерьез задумайтесь об применении XHTML 2.0 в качестве начальной точки.

В конце концов, обратите забота на то, собственно, что XHTML 2.0 - это все ещё не бесповоротная спецификация. На момент написания нашей заметки она все ещё располагается в W3C в стадии рабочего черновика, собственно, что значит, собственно, что ей ещё надо пройти кое-какой дорога, до этого чем она добьется стадии Recommendation (Рекомендуемая спецификация).

Принципиально, собственно, что имеет возможность быть ещё и стадия Candidate Recommendation (кандидат в рекомендуемые спецификации), которая применяется для скопления навыка реализации.

Наверное, XHTML 2.0 не будет подходящей спецификацией W3C (Recommendation) до 2007 года, в согласовании с трудящимся намерением рабочей группы W3C по HTML. Это значит, собственно, что 2006 год будет годом получения актуального навыка размещения.

Сравнение W3C XHTML 2.0 с WHATWG HTML 5.

В этой курсовой работе я рассказала о точках излома спецификаций HTML 5 от WHATWG и XHTML 2.0 от W3C. Эти две инициативы в корне различаются: Организованная пользователями WHATWG стремится к поэтапному усовершенствованию HTML 4 и XHTML 1.0, тогда как финансируемая консорциумом спецификация XHTML 2.0 представляет собой комплексную реорганизацию языка HTML.

Хотя эти две спецификации очень разные, они не являются несовместимыми. Некоторые из наиболее понятных разработок спецификации WHATWG уже получили реализацию в браузерах, а часть разработок WHATWG фактически

представляет собой расширения HTML.

Самые значительные из них, например, XMLHttpRequest, найдут свое выражение в спецификациях группы W3C Rich Client Activity. WHATWG также действует, как полезный катализатор в мире web-стандартов.

Если посмотреть еще дальше, подход XHTML 2.0 предлагает очищенный словарь для web, в котором модульная обработка XML, CSS и ECMAScript быстро станет нормой.

Встроенным устройствам, например, телефонам и цифровым телевизорам, нет необходимости поддерживать традиционные средства беспорядочного HTML, они свободно могут воспользоваться преимуществами XHTML 2.0 как чистого XML-словаря. Кроме того, новые функции, способствующие доступности и интернационализации, делают XHTML 2.0 первым словарем документа XML, который можно обоснованно назвать универсальным, а, следовательно, прочной и экономичной основой для многих попыток, основанных на использовании разметки.

Как и прошлое, будущее HTML будет разным - некоторые могут назвать его беспорядочным - но я думаю, что XHTML 2.0 в конце концов получит широкое распространение и применение. Если бы это был единственный словарь XML в Web, возможно, возникли бы некоторые вопросы, но, поскольку браузеры готовы работать с SVG, XForms и другими технологиями, XHTML 2.0 начинает выглядеть точно так же, как любой из этих основанных на XML словарей.

Чтобы устранить разрыв между этими двумя языками разметки и был разработан XHTML.

По существу, это обычный HTML, в который добавили синтаксические правила XML для создания well-formed документов. Так что веб-страницы станут XML-совместимыми, а веб-разработчики познакомятся с синтаксисом XML.

На практике, в HTML надо добавить четыре правила, чтобы получился XHTML:

Все теги должны быть записаны в нижнем регистре, то есть нельзя писать <BODY>, а надо писать <body>

Все теги должны быть закрыты 2a. В случае если элемент не имеет закрывающего тега (например, <IMG> или
), надо добавлять слэш в конце тега <img /> и

Вложенность тегов должна быть корректной. Например, нельзя писать <B><P>текст</B></P>, а надо писать текст

Итак, зачем использовать XHTML вместо старого доброго HTML? Консорциум W3C выделяет следующие причины:

В будущем улучшения XHTML[8] будут позволять разработчикам использовать новейшие, пока не написанные, модули для расширения XHTML, чтобы включать новые, пока не определенные, вещи в свои веб-страницы.

В добавок ко всему, W3C ожидает, что в будущие браузеры будут использовать XHTML вместо HTML.

Вторая причина пока тоже не важна. В настоящее время нет чистых XHTML-конформных браузеров, которым необходим XHTML. Да и вообще неизвестно, появятся ли они когда-либо.

В конце концов, если вы создадите браузер, который отображает только XHTML, он не будет корректно отображать HTML-страницы. Производители браузеров этого совсем не хотят.

Так что, если новый браузер выйдет, разработчики все равно позаботятся о поддержке старого доброго HTML. Новые браузеры на каких-то новых платформах возможно и будут требовать XHTML (хотя я так не думаю), но Netscape и Explorer никогда, потому что они должны быть консервативными в выборе языка.

Новые браузеры на новых платформах могут требовать XHTML. Но тогда они столкнутся с той же проблемой, что и старые браузеры на старых платформах: они не смогут корректно отображать существующие HTML-страницы, а это означает крайнее недовольство конечных пользователей. Во избежании этого, новые браузеры должны поддерживать HTML.

Конечно, XHTML[9] может стать стандартом для новых областей Интернет, как WML стал стандартным языком для WAP. Это одна из причин, по которой W3C разрабатывал XHTML. Но, откровенно говоря, я в это не верю. Новые области Интернет требуют действительно новых языков, потому что они отличаются от WWW, тогда как XHTML хорошо подходит только для традиционных WWW-страниц способом, чтобы пользователи не отвернулись от их продуктов. Только в этом случае за ними потянется остальная часть веба.

Заключение

Этим образом мы можем устроить вывод, собственно, что создание Web по праву возможно считать одним из огромнейших научно - технических достижений последнего десятилетия 21 века.

Спасибо реализации сего плана появляется весь ряд свежихинформационныхтехнологий

Однимиз более популярных классов систем обработки данных считаются информационны системы.

В реальное время увеличивается желание глобализации ИС.

Современные информационные Web-технологии проворно изменяют наш вселенная и именно воздействуют на становление Web-технологий.

Данная технологическая революция крепко повлияла на все сферы людской работы.

Внутренняя сложность и максимальная простота использовании применения передовых информационные Web-технологии готовит их дешевыми любому, кто каждый день встречается с использование

Ключевое превосходство Web-технологий в передовых критериях заключается в их простоте и как последствие в увеличении производительности их использования.

Список литературы

1. Брайан Пфаффенбергер, Стивен Шафер, Чак Уайт, Билл Kapoy.html, XHTML и CSS. Библия пользователя 3-е изд., 2006 г., 752 с.

2. Дидре Хейз. Освой самостоятельно HTML и XHTML. 3-е издание. 2002 г., 224 с.

3. Муссиано, Кеннеди.html и XHTML. Подробное руководство, 2002 г., 752 с.

4. Галактионов В.В. Расширяемый язык разметки XML (Extensible Mark-up Language, Р10-2000-44, Дубна, 2015.

5. Справочник по спецификации XHTML 2.0 (http://www.w3.org/TR/xhtml2)

6. Официальное письмо W3C XHTML Media Types (Медиа-типы XHTML) (www.w3.org/TR/xhtml-media-types)

7. www.xhtml.ru

8. Основы Web - технологий: учеб. пособие / П.Б. Храмцов [и др.]. - М.: Изд-во Интуит.ру “Интернет-Университет Информационных Технологий”, 2003.-422с.

9. Пауэл Томас, А. Справочник программиста / Томас А Пауэл, Д. Уитворт. - М.: АСТ, Мн.: Харвест, 2005. - 360 с.

10. Петров, В.Н. Информационные системы: учеб. пособие / В.Н. Петров. - СПб.: Питер, 2002. - 588 с.

11. Когаловский М.Р. XML: возможности и перспективы

12. Когаловский М.Р. XML:сферы применений

1. Дидре Хейз. Освой самостоятельно HTML и XHTML. 3-е издание. 2002 г., 224 с. ↑
2. Справочник по спецификации XHTML 2.0 (http://www.w3.org/TR/xhtml2) ↑
3. Дидре Хейз. Освой самостоятельно HTML и XHTML. 3-е издание. 2002 г., 224 с. ↑
4. Справочник по спецификации XHTML 2.0 (http://www.w3.org/TR/xhtml2) ↑
5. Справочник по спецификации XHTML 2.0 (http://www.w3.org/TR/xhtml2) ↑
6. Галактионов В.В. Расширяемый язык разметки XML (Extensible Mark-up Language, P10-2000-44, Дубна, 2015 ↑
7. Дидре Хейз. Освой самостоятельно HTML и XHTML. 3-е издание. 2002 г., 224 с. ↑

8. Когаловский М.Р. XML:сферы применений ↑

9. Галактионов В.В. Расширяемый язык разметки XML (Extensible Mark-up Language, P10-2000-44, Дубна, 2015 ↑