И. Шапошников

bhv
www.bhv.ru
www.bhv.kiev.ua

СПРАВОЧНИК WEB-МАСТЕРА

XML

МАСТЕР

Игорь Шапошников

Справочник Web-мастера

XML

Дюссельдорф ◆ Киев ◆ Москва ◆ Санкт-Петербург

УДК 681.3.06

Справочник по современным технологиям создания и обработки документов, предназначенных для опубликования в сети Интернет, — стандарту XML и его расширениям. Приведены определения структурных элементов языка разметки XML и его синтаксис, вопросы стилевого оформления XML-документов (CSS и XSL), сведения о создании гиперссылок (XLink) и идентификации ресурсов (XPointer), о каналах CDF в Интернете и WAP-ресурсах. Описание сопровождается большим количеством примеров. Дополнительно включены официальные спецификации XML, XML Schema и WML.

Для широкого круга программистов и Web-дизайнеров

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зав. редакцией	*Наталья Таркова*
Редактор	*Евгений Васильев*
Компьютерная верстка	*Натальи Смирновой*
Корректор	*Наталия Першакова*
Дизайн обложки	*Ангелины Лужиной*
Зав. производством	*Николай Тверских*

Шапошников И. В.
Справочник Web-мастера. XML. — СПб.: БХВ-Петербург, 2001. — 304 с.: ил.

ISBN 5-94157-049-X

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 22.01.01.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 24,51.
Тираж 5000 экз. Заказ №
"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар, № 77.99.1.953.П.950.3.99 от 01.03.1999 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с диапозитивов
в Академической типографии "Наука" РАН.
199034, Санкт-Петербург, 9-я линия, 12.

Содержание

Благодарности

Даниленко Ольге

Who wants to live forever without you?

Прежде всего, спасибо вам, что вы держите сейчас книгу в руках и читаете ее. Но поверьте мне, один я не в состоянии был ее сделать. Всегда есть люди, чей вклад в книгу является не менее весомым, чем авторский. Это, прежде всего, редакторский коллектив: главный редактор Екатерина Кондукова, с ее потрясающим знанием специфики компьютерной индустрии и чувством перспективы, заведующая редакцией Наталья Таркова, дирижировавшая всем процессом редактирования, и принимающий редактор Васильев Евгений, который приложил поистине титанические усилия для превращения текста в книгу.

Особо следует отметить корректора, на чью долю выпал поиск ошибок, пропущенных мной и моим спеллчекером. А они, поверьте мне, были.

Честно говоря, упоминать надо весь технический состав издательства "БХВ-Петербург", который принимал участие в работе над этой книгой. Спасибо вам.

Отдельное и просто огромное спасибо моей Ольге, которая постоянно поддерживала меня в работе.

Спасибо всем моим друзьям в Сети. EILA, Bellefi, Платон, Нюта, Анабелька, Vental, Финист, Риоха и Готик — мой ящик всегда открыт для вас.

Шапошников Игорь

shiva1@yandex.ru

Введение

Абсолютное большинство всех документов в WWW написано на языке HTML (HyperText Markup Language). Но, к сожалению, на данный момент возможностей этого языка не хватает Web-мастерам для адекватного воплощения их идей. HTML-документы предназначены для отображения в браузерах, и поэтому не могут служить полноценным интерфейсом между пользователями и базами данных. Содержимое баз данных мы можем публиковать в HTML-документах, но вот обратный перенос является уже нетривиальной задачей.

Косвенным свидетельством того, что HTML исчерпал себя, может служить количество вспомогательных технологий, которые позволяют оживить Web-страницы, придать им интерактивность. Языки сценариев VBScript и JavaScript, CGI-приложения, Java-апплеты, встраиваемые модули. Какие только средства не были созданы для того, чтобы обойти ограничения HTML. Но, несмотря на ряд недостатков, HTML все равно остается сердцевиной всех технологий WWW. Мы можем пытаться обойти его ограничения, но эффективности нашей работе это не прибавит.

Исходя из этих соображений, Консорциум WWW (W3C) разработал более мощную и гибкую технологию XML (eXtensible Markup Language), призванную заменить устаревший HTML.

Данная технология на момент написания книги не является стандартом. Версия, действующая в настоящее время, является только кандидатом на стандарт (Candidate Recommendation). Однако, несмотря на то, что формально статус XML еще не определен, этой технологией уже широко пользуются Web-мастера во всем мире (например, при создании онлайновых магазинов или бюро путешествий). Активно создаются XML-приложения, многие документы преобразуются к стандарту XML. Люди сознательно идут на риск. Ведь если правила не утверждены, то многое может еще измениться.

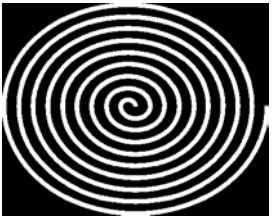
Это, кстати, объясняет ситуацию и с пособиями по XML. Очень часто в разных книгах можно встретить даже различные наборы ключевых слов и предопределенных констант. В этой книге мы не отступим ни на символ от официальной рекомендации W3C. Вы держите в руках справочник, который максимально точно и полно описывает последнюю доступную версию рассматриваемого стандарта.

В первой главе мы рассмотрим непосредственно стандарт XML, точнее — его актуальную версию, которая объявлена кандидатом на стандарт. Разберем примеры, научимся составлять собственные XML-документы. Вторая глава

посвящена обзору обособленной части стандарта — спецификации ссылок XLink. В третьей главе представлены принципы идентификации документов и их фрагментов в XML — язык XPointer. Четвертая глава содержит обзор наиболее популярного приложения XML для push-технологии — специализированного языка разметки CDF (Channel Definition Format, формат определения каналов). В следующей главе мы перейдем к рассмотрению технологии создания Web-страниц, ориентированных на доступ к ним с беспроводных тонких терминалов, то есть с сотовых телефонов. Для этих целей обычно применяется язык WML (Wireless Markup Language), который также является приложением XML. Шестая глава посвящена вопросам правильного отображения XML-документов при помощи стилевых таблиц CSS (Cascading Style Sheets). И, наконец, последняя глава ознакомит вас с преемником CSS, созданным специально для XML. В ней мы рассмотрим язык правил представления (листов стилей) XML-документов для различных устройств и сред — XSL (eXtensible Stylesheet Language).

В каждой главе излагается по возможности предельно подробная информация относительно каждой из перечисленных выше технологий. Вам предлагается описание последней, наиболее свежей версии стандарта. Весь текст в этой книге выверен, а примеры не содержат ошибок. В мир XML мы войдем вместе.

Глава 1



Расширяемый язык разметки XML

Зачем нам это надо?

Как мы уже говорили в предисловии, основой WWW является HTML (HyperText Markup Language). Этот язык представляет собой набор *тэгов* (управляющих дескрипторов), которые позволяют создавать *разметку* документа. То есть помимо указания содержимого документа, мы можем при помощи тэгов HTML управлять отображением этого документа. Технология проста. Браузер получает HTML-документ и анализирует его. Как только в коде документа встречается какой-либо тэг, он распознается, и фрагмент документа, к которому относится тэг, оформляется соответствующим образом.

На данный момент доступна уже четвертая версия языка HTML. Первоначального набора тэгов не хватало для адекватного представления более сложных документов. Поэтому компании Microsoft и Netscape — владельцы двух наиболее популярных браузеров, периодически дополняли наборы тэгов, распознаваемых их браузерами. Вследствие конкуренции этих двух фирм дополнения, естественно, не совпадали. В силу этого разработчики либо не использовали дополнительные средства, либо отказывались от возможности адекватно отображать свой документ в любом браузере. Создавался документ, предназначенный для просмотра в конкретном обозревателе, а часть посетителей сайта, использующих иной браузер, не могла увидеть страницу, содержащую данный HTML-документ (либо видела с искажениями). Проблемой HTML стала его врожденная ограниченность. Даже самый большой список тэгов не в состоянии полностью удовлетворить запросы создателей документов просто в силу того, что этот список ограничен.

Еще одной проблемой стала излишняя популярность WWW. На основе этой технологии стали строить даже локальные сети, которые, соответственно, получили название *интрасетей*. Все корпоративные документы переводились в HTML-формат. Общение внутри сети происходило при помощи электронной почты, создавались сайты, не имеющие выхода в Интернет и служащие хранилищем для корпоративных документов и средством совместной деятельности рабочих групп. Но HTML не мог служить интерфейсом между пользователями и данными. HTML-документы ориентированы прежде всего на отображение, а не на автоматическую обработку. Из базы дан-

ных можно передать данные в HTML-документ. Обратная операция намного сложнее.

Исходя из этого, стало ясно, какими свойствами должен обладать преемник HTML. Требовалось, чтобы новый язык был расширяемым, то есть не зависел от конкретного набора тэгов. Требовалась возможность расширять язык автоматически, исходя из нужд пользователя. Также было необходимо создавать файлы, которые могли бы не только отображаться, но и обрабатываться сторонними приложениями. То есть структура документов должна была напрямую привязываться к ним самим. Каждый документ должен был содержать как информацию, так и ее структуру.

Учитывая эти и многие другие соображения, Консорциум World Wide Web (W3C) в 1996 году приступил к разработке спецификаций нового языка, который должен был прийти на смену HTML. Его назвали XML (eXtensible Markup Language). На данный момент вторая редакция спецификации языка версии 1.0 является кандидатом на официальный стандарт. В этой книге мы рассмотрим данную спецификацию полностью, воспользовавшись документацией самого W3C.

Корни XML

Прежде всего, давайте разберемся, откуда появился XML. Очень давно (по меркам компьютерной индустрии, естественно), в 1986 году организацией ISO (International Organization for Standardization) язык SGML (Standard Generalized Markup Language) был принят в качестве официального стандарта. А использоваться он начал еще до этого момента. Язык SGML применяется в качестве стандарта по настоящее время. Он позволяет описывать структурированные данные, организовывать и представлять информацию, содержащуюся в документах. Стандарт SGML позволяет разработчику создавать свои конструкции разметки. Его потрясающие гибкость и универсальность, охватывающие практически все случаи, возникающие в работе над Web-проектами, казалось бы, выдвигали этот язык идеальным кандидатом для принятия его в качестве основного языка WWW, но существовали и другие обстоятельства, которые помешали ему занять лидирующее место.

Большинство документов в WWW предназначены для просмотра специализированными программами — браузерами. Браузеры анализируют код полученного документа, и на основе инструкций разметки, называемых также тэгами, отображают его в окне просмотра соответствующим образом. Но описание спецификации SGML занимает более 500 страниц текста. Отсюда видно, сколько труда потребовалось бы от разработчиков, чтобы заставить браузеры правильно отображать документы, написанные на SGML. Требовалось что-то гораздо более компактное. Естественным образом и был соз-

дан язык HTML, являющийся очень ограниченным и нерасширяемым подмножеством SGML. Так как набор тэгов HTML был невелик, разрабатывать HTML-документы и программы их просмотра было достаточно легко.

Но впоследствии это достоинство HTML превратилось в его недостаток. Посетителям и владельцам сайтов хотелось получать от HTML все больше и больше. Компании — участники "браузерных войн" добавляли все новые и новые тэги в наборы распознаваемых своими браузерами инструкций. Основная проблема состояла в том, что добавленные тэги у различных компаний тоже были разными. Возникли проблемы с совместимостью, которые не решило и доведение стандарта HTML до версии 4.0. Практически всем стало очевидно, что поскольку каждая версия HTML представляет собой ограниченный и нерасширяемый набор тэгов, то рано или поздно она окажется недостаточной.

На смену HTML пришел стандарт (а точнее, рекомендация к стандарту) XML (eXtensible Markup Language). Это — расширяемый язык разметки. Набор тэгов XML много меньше по объему, чем в HTML, но в данном случае это неважно. Изменилась сама парадигма создания документов. Появилась возможность создавать собственные тэги и конструкции из них, наподобие строительных блоков конструктора Lego. Мы теперь можем при помощи тэгов XML создавать свой язык для каждого типа документов, или даже для каждого документа отдельно.

Подобная гибкость языка позволила практически прозрачно стыковать документы с различными источниками данных для них. При помощи XML стало максимально легко интегрировать данные из различных приложений. А ведь именно интеграции различной информации из разнородных приложений подчас не хватает настоящим рабочим, а не презентационным, корпоративным сайтам. XML подоспел вовремя.

XML, как и его предшественник — HTML, является подмножеством SGML. Но XML представляет собой намного более компактный язык. Поэтому реализация браузеров для XML-документов не является излишне сложной задачей.

Любой документ из WWW в конце концов необходимо отобразить на экране компьютера удаленного пользователя. Для этой цели применяются сейчас программы-обозреватели. Браузер Internet Explorer 5-ой версии технически способен распознавать и анализировать XML-файлы, но до полного счастья еще очень далеко. Адекватное отображение содержимого XML-файлов достигается далеко не всегда.

Намного лучше дело обстоит с Netscape Communicator 5. Помимо распознавания и анализа этот браузер может еще и правильно отображать XML-документы. В него встроен полноценный XML-анализатор. Для отладки документов рекомендуется воспользоваться именно браузером Netscape.

Структура XML-документов

В XML-документах можно выделить две основные части. Первая часть XML-документа предназначена для описания его структуры, а во второй находится непосредственно содержание документа. В описании структуры документа мы можем использовать инструкции для XML-процессора, объявление элементов структуры документа, атрибуты для каждого элемента и так называемые *сущности*. Каждую из этих структурных единиц мы рассмотрим отдельно и подробно.

Описание структуры документа называют DTD-блоком (Document Type Definition). В нем мы указываем отношения на иерархии древовидной структуры элементов документа. Наиболее близкой аналогией для этих элементов могут служить объекты. Как и объекты, элементы разметки структуры могут иметь свойства, которые в данном случае называются *атрибутами*.

Как и в любой объектной иерархии, в XML-документе существует некий корневой элемент, от которого наследуются все остальные элементы.

Содержимое XML-документа форматируется при помощи тэгов, которые определяются в описании типа документа. Наименования тэгов полностью совпадают с наименованиями элементов. А параметры тэгов позволяют устанавливать значения атрибутов элементов.

Инструкции XML-процессора

В качестве первой строки каждого XML-документа должна использоваться исполняемая инструкция, предназначенная для XML-процессора (исполняемые инструкции используются для управления процессом разбора документа). В своем минимально применимом виде она выглядит следующим образом:

```
<?xml version="1.0"?>
```

Как видно из примера, исполняемые инструкции обрамляются специальными ограничителями, состоящими из угловой скобки и знака вопроса. Ключевым словом для каждой исполняемой инструкции является сокращение `xml`. Следом за ним указываются параметры инструкции и соответствующие им значения. Иногда блок исполняемых инструкций называют *прологом* XML-документа.

Приведенный нами пример предназначен для указания браузеру, что данный документ написан на XML. Значение параметра `version` указывает на тот факт, что будет использоваться первая версия стандарта XML. (А другой версии у нас пока и нет.)

В этих инструкциях мы можем не только указывать номер версии стандарта. Для XML-документов заготовлено несколько видов кодировок, состоящих из символов набора Unicode. Большинство кодировок предложено международной организацией по стандартизации (ISO).

Для указания конкретной кодировки, которая будет использоваться для отображения содержимого документа, применяется параметр `encoding`, как в следующем далее примере:

```
<?xml encoding="UTF-8"?>
```

В качестве значения параметра здесь задана текстовая строка `"UTF-8"`. Кодировка UTF-8 является одной из наиболее часто используемых кодировок.

Следующие параметры исполняемых инструкций XML-процессора напрямую связаны с обработкой блоков описания структуры документа (DTD-блоков). Забегая немного вперед, необходимо сказать, что подобные блоки могут внедряться в сам XML-документ, или находиться во внешнем файле. Для того чтобы XML-процессор смог правильно их обработать, применяется параметр `standalone`. При помощи этого параметра можно указать, где именно находится описание структуры для данного XML-документа. В следующем примере мы используем объявление XML-документа, в котором указывается, что DTD-блок для него оформлен в виде отдельного файла.

```
<?xml standalone='no'?>
```

У параметра `standalone` есть два предопределенных значения: `yes` и `no`. Значение `no`, как мы уже видели, извещает XML-процессор, что для данного документа DTD-блок выделен в отдельный файл. Значение `yes` указывает на то, что DTD-блок размещен в теле XML-файла.

Мы рассмотрели возможные варианты исполняемых инструкций, помещаемых в начале документа. Практически эти инструкции могут находиться не только в начале документа, но и в любом другом его месте. Но использование исполняемой инструкции в качестве первой строки XML-документа является обязательным условием.

Любое рассмотрение языка на примерах, без четких определений, будет всего лишь обзором. Чтобы избежать этой участи, для каждой из рассматриваемых нами структурных единиц XML-документа мы будем приводить их определение из спецификации XML. Для исполняемых инструкций, помещаемых в пролог документа, оно выглядит следующим образом:

```
[23]  XMLDecl ::=  '<?xml' VersionInfo EncodingDecl? SDDecl? S? '?>'
[24]  VersionInfo ::=  S 'version' Eq (' VersionNum ' | " VersionNum ")
[25]  Eq ::=  S? '=' S?
[26]  VersionNum ::=  ([a-zA-Z0-9_.:] | '-')+
```

На первый взгляд смотрится достаточно устрашающе. Тем не менее, все спецификации XML выглядят подобным образом. В примере приведена запись конструкций XML в форме Бэкуса-Наура (EBNF, Extended Backus-Naur Form). Разберемся с правилами чтения деклараций в нотации EBNF. В левой части каждой декларации указывается имя конструкции, затем следует оператор эквивалентности (::=), а в правой части следует расшифровка имени, которая содержит его формат и правила оформления. Перед каждой конструкцией в квадратных скобках указывается номер строки спецификации.

Итак, из приведенного фрагмента мы видим, что определение исполняемой инструкции находится в 23-й строке спецификации языка XML. При изучении спецификации необходимо помнить несколько правил записи определений в форме EBNF.

- Если в определение включено несколько терминов, разделенных вертикальной чертой (|), то следует выбрать только один из них. Подобным образом определяется перечислимое множество компонентов.
- Для указания диапазона символов, которые могут использоваться в каком-либо месте определения термина, этот диапазон символов помещается в квадратные скобки.
- Круглые скобки ограничивают так называемые *регулярные выражения*. Проще всего их воспринимать как обычное средство группировки элементов. В случае, когда согласно синтаксису вместо одиночного литерала нужно указать несколько, используются круглые скобки, группирующие их.
- Знак звездочки (*) примыкает справа к символьному выражению, если требуется, чтобы это выражение в результирующей строке могло быть повторено несколько раз или вовсе там не использоваться.
- Знак плюса (+) применяется в качестве модификатора символьного выражения подобно знаку звездочки. Но есть некоторое отличие. Символьное выражение, после которого присутствует этот модификатор, должно встречаться в результирующей строке хотя бы один раз.
- Вопросительный знак (?) используется для указания, что тот или иной элемент могут не найтись в результирующей строке. То есть при помощи этого модификатора указываются опциональные элементы.
- Если в выражении не должны встречаться некоторые символы, то к нему добавляется последовательность [^, после которой указываются исключаемые символы. Например, правило [^&] исключает символ амперсанда.
- Если внутри какого-либо литерала необходимо вставить пробел, то этот пробел обозначается при помощи символа S.
- Любая последовательность символов, заключенная в двойные или одиночные кавычки, является литералом, и в результирующей строке должна появляться именно в том виде, в каком она указана в декларации.