

# XML

## Зачем нужны языки разметки

Идея решения проблемы обмена документами между различными компьютерами и приложениями через Internet основана на языке разметки гипертекста HTML (HyperText Markup Language). Этот язык был создан более 15 лет назад как стандарт оформления документов и был принят подавляющим большинством пользователей Internet, а главное, - всеми производителями программного обеспечения и оборудования для Web. Документы, размеченные согласно HTML, могут читаться на любом компьютере, на котором установлена всего лишь одна программа просмотра таких документов - браузер.

Благодаря языку разметки HTML, клиент Web может на экране своего компьютера просмотреть документ в том виде, в каком его задумал разработчик: с определенными размерами шрифта и разбивкой на абзацы, с определенным расположением рисунков, гиперссылок и проч. ссылок.

- Текстовый документ, составленный на HTML, имеет размер в байтах в несколько раз меньший, чем размер аналогичного документа, подготовленного в текстовом процессоре (например, Word).

- В основу разрабатываемого языка Бернерс-Ли (разработчик) положил язык SGML и приемы работы с гипертекстом, с чем и связано название созданного им языка - HTML. Новый язык использовал основные конструкции SGML для описания документов и гипертекстовых

## Что такое XML?

- **XML** - аббревиатура от англ. *eXtensible Markup Language* (пер. расширяемый язык разметки).

- **XML** – язык разметки, который напоминает HTML.
- **XML** предназначен для передачи данных, а не для их отображения.
- **Теги XML** не предопределены. Вы должны сами определять нужные теги.
- **XML** описан таким образом, чтобы быть самоопределяемым.

Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Расширение XML — это конкретная грамматика, созданная на базе XML и представленная словарём тегов и их атрибутов, а также набором правил, определяющих, какие атрибуты и элементы могут входить в состав других элементов.

Сочетание простого формального синтаксиса, удобства для человека, расширяемости, а также базирование на кодировках Юникод для представления содержания документов привело к широкому использованию как собственно XML, так и множества производных специализированных языков на базе XML в самых разнообразных программных средствах.

XML является подмножеством SGML.

## Почему именно XML?

XML используется во многих аспектах веб-разработки, но основная его задача — облегчение хранения и передачи данных.

## XML отделяет данные от HTML

Если вам в HTML документе необходимо отображать динамические данные, то это будет занимать слишком много времени, если всякий раз, когда эти данные изменились, редактировать сам HTML документ.

С **XML** данные можно хранить в отдельных файлах XML. При этом вы сосредотачиваетесь на использовании HTML/CSS для отображения и шаблонизации и можете быть уверены, что поступающие новые данные не потребуют каких-либо изменений в коде HTML документа.

## XML упрощает распределение данных

В реальном мире компьютерные системы и базы данных используют данные в несовместимых форматах.

XML данные хранятся в простом текстовом формате. Это обеспечивает программную и аппаратную независимость.

Это позволяет легко создавать данные, которые могут использоваться самыми разными приложениями.

## **XML упрощает передачу данных**

Одной из самых время затратных проблем разработчиков всегда была и остается до сих пор проблема обмена данными между несовместимыми между собой системами.

Передача данных в виде XML значительно снижает сложность этой проблемы, так как данные в этом формате могут быть прочитаны разными несовместимыми приложениями.

## **XML упрощает модификацию платформы**

Переход на новые системы (аппаратные или программные платформы) всегда занимает много времени. Множество данных необходимо конвертировать в новые форматы. При этом часто несовместимые данные теряются.

XML данные хранятся в текстовом формате. Это значительно облегчает расширение или модернизацию операционных систем, переход на новые приложения или браузеры без опасности потерять данные.

## **XML делает ваши данные более доступными**

Доступ к вашим данным могут получать не только HTML документы, но и любые другие приложения.

Благодаря XML ваши данные становятся доступными для всех видов "машин чтения" (голосовых машин, новостных каналов и т.д.), что позволяет значительно проще обращаться к ним людям с ограничениями по зрению и другими физическими проблемами.

## **XML используется для создания новых интернет-языков**

С использованием XML было создано множество языков программирования в Интернете.

Вот несколько примеров:

- XHTML
- WSDL для описания доступных веб-сервисов
- WAP и WML как языки разметки для портативных устройств типа КПК
- RSS языки для новостных каналов
- RDF и OWL для описания ресурсов и онтологии

SMIL для описания мультимедиа для сети

## **Какие проблемы решаются использованием XML**

XML решает ряд проблем, которые не решает HTML, например:

- Представление документов любого (не только текстового) типа, например, музыки, математических уравнений и т.д.
- Сортировка, фильтрация и поиск информации.
- Представление информации в структурированном (иерархическом) виде.

## **Что такое DOM и SAX?**

### **DOM**

- Деревяной синтаксический анализатор (Object based) (Дерево узлов).
- DOM загружает файл в память и затем анализирует файл.
- Имеет ограничения памяти, так как он загружает весь XML-файл перед parsingom.
- DOM считывается и записывается (может вставлять или удалять узлы).
- Если XML-контент мал, то предпочитайте парсер DOM.
- Поиск назад и вперед возможен для поиска тегов и оценки информации внутри тегов. Таким образом, это дает легкость навигации.
- Медленнее во время выполнения.

## **Что такое DOM и SAX?**

### **SAX**

- Парсер, основанный на событиях (последовательность событий).
- SAX анализирует файл по мере его чтения, то есть разбирает узел по узлу.
- Нет ограничений памяти, так как он не хранит содержимое XML в памяти.
- SAX доступен только для чтения, т. Е. Не может вставить или удалить узел.
- Используйте синтаксический анализатор SAX, когда объем содержимого большой.
- SAX читает XML-файл сверху вниз, а обратная навигация невозможна.
- Быстрее во время выполнения.

## **Как выбрать между SAX и DOM**

Использовать вам DOM или SAX, зависит от нескольких факторов:

- **Назначение приложения:** Если вы собираетесь делать изменения в данных и выводить их как XML, то в большинстве случаев способом для этого является DOM. Нельзя сказать, что вы не можете делать изменения при использовании SAX, но этот процесс значительно более сложен, так как вы должны делать изменения в копии данных, а не в самих данных.

- **Объем данных:** Для больших файлов SAX является лучшим выбором.

- **Как будут использованы данные:** Если на самом деле будет использована только небольшая часть данных, вам, возможно, лучше применить SAX, чтобы выделить ее в ваше приложение. С другой стороны, если вы знаете, что вам придется ссылаться назад в большом объеме информации, которая уже была обработана, SAX, возможно, не является правильным выбором.

- **Требования к быстродействию:** реализации SAX обычно быстрее, чем реализации DOM.

Важно помнить, что SAX и DOM не являются взаимоисключающими. Вы можете использовать DOM для создания потока событий SAX, и вы можете использовать SAX для создания дерева DOM. Фактически, большинство парсеров, применяемых для создания дерева DOM, используют SAX, чтобы сделать это!

## **XSD — умный XML**

**XSD** — это язык описания структуры XML документа. Его также называют XML Schema. При использовании XML Schema XML парсер может проверить не только правильность синтаксиса XML документа, но также его структуру, модель содержания и типы данных.

Такой подход позволяет объектно-ориентированным языкам программирования легко создавать объекты в памяти, что, несомненно, удобнее, чем разбирать XML как обычный текстовый файл.

Кроме того, XSD расширяем, и позволяет подключать уже готовые словари для описания типовых задач, например веб-сервисов, таких как SOAP.

Стоит также упомянуть о том, что в XSD есть встроенные средства документирования, что позволяет создавать самодостаточные документы, не требующие дополнительного описания.

## **XSLT**

**XSLT** (eXtensible Stylesheet Language Transformations) — язык преобразования XML-документов. Спецификация XSLT входит в состав XSL и является рекомендацией W3C.

При применении таблицы стилей XSLT, состоящей из набора шаблонов, к XML-документу (исходное дерево) образуется конечное дерево, которое может быть сериализовано в виде XML-документа, XHTML-документа (только для XSLT 2.0), HTML-документа или простого текстового файла. Правила выбора (и, отчасти, преобразования) данных из исходного дерева пишутся на языке запросов XPath.

XSLT имеет множество различных применений, в основном в области веб-программирования и генерации отчетов. Одной из задач, решаемых языком XSLT, является отделение данных от их представления, как часть общей парадигмы MVC (англ. Model-view-controller). Другой стандартной задачей является преобразование XML-документов из одной XML-схемы в другую.

## **Процесс выполнения XSLT-преобразования**

В процессе выполнения XSLT-преобразования задействованы:

- один или несколько входных XML-документов;
- одна или несколько таблиц стилей XSLT;
- XSLT-процессор;
- один или несколько выходных документов.

В простейшем случае XSLT-процессор получает на входе два документа — входной XML-документ и таблицу стилей XSLT — и создаёт на их основе выходной документ.

## **XPath**

**XPath** (XML Path Language) — язык запросов к элементам XML-документа. Разработан для организации доступа к частям документа XML в файлах трансформации XSLT и является стандартом консорциума W3C. XPath призван реализовать навигацию по DOM XML. В XPath используется компактный синтаксис, отличный от принятого в XML.

На данный момент самой популярной версией является XPath 1.0. Это связано с отсутствием поддержки XPath 2.0 со стороны открытых библиотек. В частности, речь идёт

о LibXML, от которой зависит поддержка языка в браузерах, с одной стороны, и поддержка со стороны серверного интерпретатора, с другой.

XML имеет древовидную структуру. В самостоятельном XML-документе всегда имеется один корневой элемент (инструкция `<?xml version="1.0"?>` к дереву элементов отношения не имеет), в котором допустим ряд вложенных элементов, некоторые из которых тоже могут содержать вложенные элементы. Так же могут встречаться текстовые узлы, комментарии и инструкции. Можно считать, что XML-элемент содержит массив вложенных в него элементов и массив атрибутов.

У элементов дерева бывают элементы-предки и элементы-потомки (у корневого элемента предков нет, а у тупиковых элементов (листьев дерева) нет потомков). Каждый элемент дерева находится на определённом уровне вложенности (далее — «уровень»). Элементы упорядочены в порядке расположения в тексте XML, и поэтому можно говорить об их предыдущих и следующих элементах. Это очень похоже на организацию каталогов в файловой системе.

Строка XPath описывает способ выбора нужных элементов из массива элементов, которые могут содержать вложенные элементы. Начинается отбор с переданного множества элементов, на каждом шаге пути отбираются элементы, соответствующие выражению шага, и в результате оказывается отобрано подмножество элементов, соответствующих данному пути.

### **XSLT и XPath**

XSLT использует язык XPath для доступа к отдельным частям входного XML-документа и для организации вычислений.

XSLT 1.0 использует XPath 1.0, а XSLT 2.0 использует XPath 2.0.