Что такое XML

Язык **XML** предназначен для хранения и передачи данных. **HTML** же предназначен для отображения данных.

Прежде чем продолжить, убедитесь, что вы обладаете базовыми знаниями в HTML. Если вы не знаете что такое HTML, то разобраться в этом вам поможет учебник HTML для начинающих. Итак,

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.

Разница между XML и HTML

XML не является заменой HTML. Они предназначены для решения разных задач: XML решает задачу хранения и транспортировки данных, фокусируясь на том, что такое эти самые данные, HTML же решает задачу отображения данных, фокусируясь на том, как эти данные выглядят. Таким образом, HTML заботится об отображении информации, а XML о транспортировке информации.

XML ничего не делает

Возможно вам будет несколько странным это узнать, но XML ничего не делает. Он был создан для структурирования, хранения и передачи информации.

Следующий пример представляет некую заметку от Тови к Джени, сохраненную в формате XML:

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

<note>

<to>Tove</to>

<from>Jani</from>

<heading>Напоминание</heading>

<body>Не забудь обо мне в эти выходные!</body>

</note>

Приведенная запись вполне самоописательна. Здесь есть информация об отправителе и получателе. Также присутствуют данные заголовка и само сообщение. И при всем при этом этот документ XML не делает ничего. Это просто информация, обернутая в теги. Кто-то должен написать программу, которая будет отсылать, получать и отображать эти данные.

В XML вы изобретаете свои собственные теги

Теги в вышеприведенном примере (например, **<to>** и **<from>**) не определяются никакими стандартами XML. Эти теги были "изобретены" автором этого XML документа.

Все потому, что в языке XML нет предопределенных тегов.

Так, в HTML все используемые теги предопределены. HTML документы могут использовать только те теги, которые определяются в стандартах HTML (**<p>**, **<li>** и т. д.).

XML позволяет автору определять свои языковые теги и свою структуру документа.

XML – это не замена HTML

XML – это дополнение HTML.

Важно понять, что XML не является заменой HTML. В большинстве веб-приложениях XML используется для транспортировки данных, а HTML для форматирования и отображения данных.

XML – это программно- и аппаратно-независимый инструмент для транспортировки информации.

XML – везде

В настоящее время XML также важен для сети, как когда-то был важен HTML для рождения современного Интернета. XML – это общий инструмент передачи данных между всеми видами приложений.

Как используется 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

### ML документы формируют древовидную структуру

XML документ должен содержать корневой элемент. Этот элемент является "родительским" для всех других элементов.

Все элементы в XML документе формируют иерархическое дерево. Это дерево начинается с корневого элемента и разветвляется на более низкие уровни элементов.

Все элементы могут иметь подэлементы (дочерние элементы):

<корневой>

<потомок>

<подпотомок>.....</подпотомок>

</потомок>

</корневой>

Для описания взаимоотношений между элементами обычно используют термины "родитель", "потомок" и "брат"/"сестра". У "родительского" элемента есть "потомки" или "дочерние" элементы. "Потомки" на одном и том же уровне называются "братья" или "сестры".

### Все XML элементы должны иметь закрывающий тег

В HTML некоторые элементы могут не иметь закрывающего тега:

<p>Это параграф.

<br>

В XML нельзя опускать закрывающий тег. Абсолютно все элементы должны закрываться:

<p>Это параграф.</p>

<br>

Возможно, вы заметили из предыдущих примеров, что XML декларация не имеет закрывающего тега. Это не ошибка. Дело в том, что декларация не относится к XML документу, поэтому у нее и нет закрывающего тега.

### Теги XML регистрозависимы

Теги XML являются регистрозависимыми. Так, тег **<Letter>** не то же самое, что тег **<letter>**.

Открывающий и закрывающий теги должны определяться в одном регистре:

<Message>Это неправильно</message>

<message>Это правильно</message>

Замечание: "Открывающий и закрывающий теги" иногда еще называют "начальный и конечный теги". Используйте то определение, которое вам более симпатично. По сути это одно и то же.

### XML элементы должны соблюдать корректную вложенность

В HTML иногда можно наблюдать такую картину:

<b><i>Это жирный и курсивный текст</b></i>

и иногда это даже работает должным образом.

В XML все элементы обязаны соблюдать корректную вложенность:

<b><i>Это жирный и курсивный текст</i></b>

Понятие "корректная вложенность" по отношению к приведенным примерам просто означает, что так как элемент **<i>** открывается внутри элемента **<b>**, то и закрываться он должен внутри элемента **<b>**.

### У XML документа должен быть корневой элемент

XML документ должен содержать один элемент, который будет родительским для всех других элементов. Он называется корневым элементом.

<корневой>

<потомок>

<подпотомок>.....</подпотомок>

</потомок>

</корневой>

### XML пролог

Следующая строка называется XML прологом:

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

XML пролог необязателен. Но если он есть, то это должна быть первая строка XML документа.

В XML документе могут присутствовать международные символы, вроде русских букв, и чтобы не возникало ошибок необходимо указать кодировку, либо сохранить XML файл в формате UTF-8.

UTF-8 — кодировка XML документов по умолчанию.

### Значения XML атрибутов должны заключаться в кавычки

Так же, как и в HTML, у XML элементов могут быть атрибуты в виде пары имя/значение.

В XML значения атрибутов должны заключаться в кавычки.

Посмотрите на следующие два примера XML документа. Первый с ошибкой, второй написан правильно:

<note date=12/11/2007>

<to>Tove</to>

<from>Jani</from>

</note>

<note date="12/11/2007">

<to>Tove</to>

<from>Jani</from>

</note>

Ошибка в первом XML документе заключается в том, что значение атрибута date элемента note не заключено в кавычки.

Что такое XML элемент?

XML элемент — это все от (и включая) начального тега элемента до (и включая) конечного тега элемента.

Элемент может содержать:

* другие элементы
* текст
* атрибуты
* или набор из всего выше названного

### Стили написания имен

Для XML элементов не существует какого-либо определенного стиля написания имен. Тем не менее, вот несколько наиболее часто используемых стилей:

| **Стиль** | **Пример** | **Описание** |
| --- | --- | --- |
| Нижний регистр | <firstname> | Все буквы в нижнем регистре |
| Верхний регистр | <FIRSTNAME> | Все буквы в верхнем регистре |
| С символом подчеркивания | <first\_name> | Слова разделяются символом подчеркивания |
| В стиле Pascal | <FirstName> | Первые буквы всех слов в верхнем регистре |
| "Верблюжий горб" | <firstName> | Первые буквы всех слов за исключением первого в верхнем регистре |

Если вы выбрали какой-либо стиль написания имен, то следует последовательно придерживаться его!

Что такое XPath?

* **XPath** - специальный язык для определения частей XML документа
* **XPath** использует маршрутные выражения для навигации по XML документам
* **XPath** содержит библиотеку стандартных функций
* **XPath** — главный элемент в XSLT
* **XPath** также используется в XQuery, XPointer и XLink

### Маршрутные выражения XPath

**XPath** использует маршрутные выражения для выбора узлов или узловых наборов в XML документе. Эти маршрутные выражения похожи на те выражения, которые можно увидеть при работе с традиционными файловыми системами.

В настоящее время выражения XPath можно использовать в JavaScript, Java, XML схемах, PHP, Python, C и C++, а также во множестве других языках программирования.

# Подводим итоги

XML может использоваться для обмена, совместного использования и хранения данных.

XML документы формируют древовидную структуру, начинающуюся от "корня" и ветвящуюся к "листьям".

XML подчиняется простым синтаксическим правилам. XML с корректным синтаксисом является "синтаксически верным". Валидный XML также проходит проверку по DTD.

XSLT используется для трансформации XML в другие форматы, например, HTML.

Все современные браузеры имеют встроенный XML парсер, который может загружать и манипулировать XML.

Объектная модель документа (DOM) определяет стандартный способ доступа к XML.

Объект XMLHttpRequest предоставляет способ обращаться к серверу после того, как веб-страница уже была загружена.

Пространства имен XML позволяют избежать конфликта имен элементов.

Текст внутри секции CDATA игнорируется парсером.

### Что изучать следующим?

Мы рекомендуем изучить XML DOM и XSLT.

Если вы хотите узнать больше о валидации XML, то обратите внимание на учебники по DTD и XML схеме.

Ниже коротко рассказывается об этих темах.

### Объектная модель документа (XML DOM)

XML DOM определяет стандартный способ доступа к XML документу и манипулирования его элементами.

XML DOM является независимой платформой и может использоваться любыми языками программирования, например, Java, JavaScript и VBScript.

Чтобы подробнее узнать о DOM, смотрите учебник XML DOM.

### Язык преобразования XML документов (XSLT)

XSLT является языком таблиц каскадных стилей для XML файлов.

При помощи XSLT можно трансформировать XML документы в другие форматы, например, XHTML.

### Определение типов документа (XML DTD)

Цель DTD состоит в том, чтобы определить, какие элементы, атрибуты и сущности являются допустимыми в XML документе.

При помощи DTD каждый ваш XML файл может нести описание своего собственного формата.

DTD может использоваться для проверки корректности получаемых данных.

### XML схемы

XML схемы являются альтернативой DTD, в основе которой лежит XML.

В отличие от DTD, XML схемы имеют поддержку типов данных. Также, XML схемы используют синтаксис XML.