

XML ([англ. eXtensible Markup Language](#) — расширяемый язык разметки; произносится [экс-эм-эль]) — рекомендованный [Консорциумом Всемирной паутины](#) (W3C) [язык разметки](#). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным [синтаксисом](#), удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете. 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. 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](#).

Содержание

[\[убрать\]](#)

- 1 Язык XML

- 1.1 [Логическая и физическая структура документа](#)
- 1.2 Символы разметки. Решение проблемы неоднозначности разметки
- 1.3 Пролог
 - 1.3.1 Объявление XML
 - 1.3.2 Объявление типа документа
 - 1.3.3 Инструкция обработки
 - 1.3.4 Комментарий
- 1.4 Корневой элемент
 - 1.4.1 Элемент и его разметка
 - 1.4.2 Секция CDATA
- 1.5 Корректный документ
- 1.6 Пространства имён
- 1.7 Пример документа

- 2 Регламентация работы с документами: правила, языки, программные интерфейсы

- 2.1 Кодировка документов
- 2.2 XML-процессор и приложение
- 2.3 Действительный документ. Проверяющие и не проверяющие процессоры
- 2.4 Описание типов: языки схем
- 2.5 Преобразование документа XML
- 2.6 Формат для визуализации документа
- 2.7 Языки запросов
- 2.8 Интерфейс DOM

- 3 Инструменты работы с документами: парсеры, средства создания и визуализации, системы баз данных

- 3.1 Реализации парсеров
- 3.2 Веб-браузеры как инструмент визуализации документа
- 3.3 Редакторы XML
- 3.4 Системы управления базами данных, работающие с данными в формате XML
- 3.5 Поддержка на аппаратном уровне
- 4 Область применения, ограничения, перспективы развития
 - 4.1 Эффективность использования XML
 - 4.2 Скриптовый язык для работы с XML
- 5 См. также
- 6 Примечания
- 7 Литература
- 8 Ссылки

Язык XML [[править](#) | [править исходный текст](#)]

Спецификация XML описывает язык и касается ряда других вопросов, касающихся кодировки и обработки документов. Материал этой секции представляет собой сокращённое изложение описания языка в Спецификации XML, адаптированное для настоящей статьи.

Нормативным считается английский вариант документа, поэтому основные термины приводятся с их английскими оригиналами.

Перевод основных терминов в основном следует доступному в интернете переводу Спецификации на русский язык, исключение составляют термины *tag* и *declaration*.

Для термина *tag* здесь используется перевод *тег* как соответствующий текущим нормам русского языка.

Для термина *declaration* отдано предпочтение распространённому переводу *объявление* (против также распространённой кальки *декларация*).

В литературе и интернете могут встречаться и иные переводы основных терминов.

Логическая и физическая структура документа [[править](#) | [править исходный текст](#)]

С логической точки зрения, документ состоит из *пролога* и **корневого элемента**. Корневой элемент — обязательная часть документа, пролог, вообще говоря, может отсутствовать.

Пролог может включать **объявления, инструкции обработки, комментарии**.

Пролог следует начинать с *объявления XML*, хотя в определённой ситуации допускается отсутствие этого объявления.

Корневой элемент может включать (а может не включать) вложенные в него **элементы** и **символьные данные**, а также комментарии. Вложенные в корневой элемент элементы, в свою очередь, могут включать вложенные в них элементы, символьные данные и комментарии, и так далее.

Элементы документа должны быть *правильно вложены*: любой элемент, начинающийся внутри другого элемента (то есть любой элемент документа, кроме корневого), должен заканчиваться внутри элемента, в котором он начался.

Символьные данные могут встречаться внутри элементов как непосредственно так и в специальных *секциях CDATA*.

Объявления, инструкции обработки и элементы могут иметь связанные с ними *атрибуты*. Атрибуты ([англ. attribute](#)) используются для связывания с логической единицей текста пар имя-значение.

С физической точки зрения, символы, составляющие документ, делятся на **разметку** ([англ. markup](#)) и **символьные данные** ([англ. character data](#)).

Объявления и инструкции обработки, включая их атрибуты, а также комментарии полностью принадлежат разметке. К разметке также принадлежат теги, обозначающие границы элементов, включая их атрибуты, а также последовательности символов, обрамляющие секции CDATA.

Часть документа, не принадлежащая разметке, составляет символьные данные документа.

Символы разметки. Решение проблемы неоднозначности разметки[\[править\]](#) | [править исходный текст](#)

Разметка всегда начинается символом < и заканчивается символом >. Наряду с символами < и >, специальную роль для разметки играет также символ &. Употребление разметочных символов в символьных данных затрудняет распознавание конструкций разметки и может создать проблему неоднозначности структуры. В XML эта проблема решается следующим образом: три упомянутые символа не могут присутствовать в символьных данных и в значениях атрибутов в их непосредственном виде, для их представления в этих случаях зарезервированы специальные последовательности символов, начинающиеся с & и заканчивающиеся ;

Символ	Замена
<	<
>	>
&	&

Кроме того, для употребления апострофов и кавычек внутри значений атрибутов используются следующие замены:

'	'
"	"

Правило замены символов, используемых в разметке, на зарезервированные последовательности символов не распространяется на символьные данные в секциях CDATA.

Пролог[\[править\]](#) | [править исходный текст](#)

Объявление XML[\[править\]](#) | [править исходный текст](#)

Объявление XML объявляет версию языка, на которой написан документ. Поскольку интерпретация содержимого документа, вообще говоря, зависит от версии языка, то Спецификация предписывает начинать документ с объявления XML. В первой (1.0) версии языка использование объявления не было обязательным, в последующих версиях оно обязательно. Таким образом, версия языка определяется из объявления, и если объявление отсутствует, то принимается версия 1.0.

Кроме версии XML, объявление может также содержать информацию о *кодировке* документа.

Пример:

```
<?xml version="1.1" encoding='UTF-8' ?>
```

или

```
<?xml version="1.0" encoding="windows-1251"?>
```

Объявление типа документа[\[править\]](#) | [править исходный текст](#)

Инструкция обработки[\[править\]](#) | [править исходный текст](#)

Инструкции обработки ([англ. processing instruction, PI](#)), позволяют размещать в документе инструкции для приложений. В следующем примере показана инструкция обработки, передающая xmlstylesheet-приложению (например, браузеру) инструкции в файле my-style.css посредством атрибута href:

```
<?xml-stylesheet href="my-style.css"?>
```

Комментарий[\[править\]](#) | [править исходный текст](#)

Комментарии ([англ. comment](#)) не относятся к символьным данным документа. Комментарий начинается последовательностью «<!--» и заканчивается последовательностью «-->», внутри не может встречаться комбинация символов «--». Символ & не используется внутри комментария в качестве разметки.

Пример:

```
<!-- это комментарий -->
```

Корневой элемент[\[править\]](#) | [править исходный текст](#)

Элемент и его разметка[\[править\]](#) | [править исходный текст](#)

Элемент ([англ. element](#)) является понятием логической структуры документа. Каждый документ содержит один или несколько элементов. Границы элементов представлены *начальным* и *конечным тегами*. Имя элемента в начальном и конечном тегах элемента должно совпадать. Элемент может быть также представлен тегом *пустого*, то есть не включающего в себя другие элементы и символьные данные, *элемента*.

Тег ([англ. tag](#)) — конструкция разметки, которая содержит имя элемента.

Начальный тег: <element1>

Конечный тег: </element1>

Тег пустого элемента: <empty_element1 />

В элементе атрибуты могут использоваться только в начальном теге и теге пустого элемента.

Пример кулинарного рецепта, размеченного с помощью XML:

```
<recipe name="хлеб" preptime="5" cooktime="180">
  <title>Простой хлеб</title>
  <composition>
    <ingredient amount="3" unit="стакан">Мука</ingredient>
    <ingredient amount="0.25" unit="грамм">Дрожжи</ingredient>
    <ingredient amount="1.5" unit="стакан">Тёплая вода</ingredient>
    <ingredient amount="1" unit="чайная ложка">Соль</ingredient>
  </composition>
  <instructions>
    <step>Смешать все ингредиенты и тщательно замесить.</step>
    <step>Закрыть тканью и оставить на один час в тёплом помещении.</step>
    <!-- <step>Почитать вчерашнюю газету.</step> - это сомнительный шаг... -->
    <step>Замесить ещё раз, положить на противень и поставить в духовку.</step>
  </instructions>
```

</recipe>

Секция CDATA[\[править\]](#) | [править исходный текст](#)

Секция CDATA не является логической единицей текста. Секция может встречаться в любом месте документа, где синтаксис позволяет размещать символьные данные. Секция начинается <![CDATA[и завершается]]>. Между этой разметкой находятся символьные данные, символьные данные при этом включают символы < > & в их непосредственной форме.

Корректный документ[\[править\]](#) | [править исходный текст](#)

Корректный ([англ.](#) *well-formed*) документ соответствует всем общим правилам синтаксиса XML, применимым к любому XML-документу: правильная структура документа, совпадение имен в начальном и конечном теге элемента и т. п. Документ, который неправильно построен, не может считаться документом XML.

Пространства имён[\[править\]](#) | [править исходный текст](#)

Пример документа[\[править\]](#) | [править исходный текст](#)

Регламентация работы с документами: правила, языки, программные интерфейсы[\[править\]](#) | [править исходный текст](#)

Этот раздел содержит изложение некоторых положений рекомендаций W3C, касающихся работы с документами. Соответствующие рекомендации могут относиться как к документам XML, так и к более широкому классу документов. Ссылки, как правило, даются на средства работы с документами, рекомендованные W3C.

Кодировка документов[\[править\]](#) | [править исходный текст](#)

Спецификация требует, чтобы обрабатывающие программы поддерживали по крайней мере две кодировки Юникод: [UTF-8](#) и [UTF-16](#).

XML-процессор и приложение[\[править\]](#) | [править исходный текст](#)

Спецификация XML определяет понятия *XML-процессор* и *приложение*. XML-процессор ([парсер](#)) — программа, анализирующая разметку и передающая информацию о структуре документа другой программе — приложению.

Спецификация XML налагает определённые требования на процессор, не касаясь требований к приложению.

Действительный документ. Проверяющие и не проверяющие процессоры[\[править\]](#) | [править исходный текст](#)

Документ является *действительным*, если с ним связано объявление типа документа и если этот документ отвечает представленным в объявлении типа ограничениям.

XML процессоры делятся на два класса: проверяющие и не проверяющие.

Проверяющие процессоры проверяют действительность документа и должны сообщать (по выбору пользователя) о нарушении ограничений, сформулированных в объявлении типа документа.

Непроверяющие процессоры не проверяют действительность документа, но обязанности по предварительной обработке документа, упомянутые выше, остаются за ними.

Описание типов: языки схем[\[править\]](#) | [править исходный текст](#)

Для описания типов документов используются языки схем ([англ.](#) *schema language*). Поскольку XML является подмножеством языка SGML, то он унаследовал разработанный для SGML язык Document Type Definition ([DTD](#)). Позднее были разработаны и другие языки схем, наиболее известны из которых [XML Schema](#), [RELAX NG](#).

Преобразование документа XML[\[править\]](#) | [править исходный текст](#)

Для решения задачи преобразования документа XML в другую схему или другой формат предназначен язык [XSLT](#).

Формат для визуализации документа[\[править\]](#) | [править исходный текст](#)

Для форматированного документа (документа, подготовленного к визуализации) предназначен формат [XSL-FO](#).

Языки запросов[\[править\]](#) | [править исходный текст](#)

[XPath](#) — синтаксис для адресации содержимого документа, представленного в форме дерева. Выражения XPath используются в языке *XQuery*. Выражения XPath, вообще говоря, могут использоваться в любом контексте, где уместно использовать формальные ссылки на элементы дерева, в частности, в качестве параметров для методов интерфейсов доступа к документу.

[XQuery](#) — язык программирования, ориентированный на работу с документами.

Интерфейс DOM[\[править\]](#) | [править исходный текст](#)

[Document Object Model](#) (DOM) — «объектная модель документа» — программный интерфейс для доступа к содержимому документов. Парсеры XML преобразуют документ XML в объект DOM.

Инструменты работы с документами: парсеры, средства создания и визуализации, системы баз данных[\[править\]](#) | [править исходный текст](#)

Реализации парсеров[\[править\]](#) | [править исходный текст](#)

XML имеет реализации [парсеров](#) для всех современных языков программирования^[6].

Веб-браузеры как инструмент визуализации документа[\[править\]](#) | [править исходный текст](#)

- Визуализация без использования стилей CSS

Без использования CSS или [XSL](#) XML-документ отображается как простой текст в большинстве веб-браузеров. Некоторые браузеры, такие как [Internet Explorer](#), [Mozilla Firefox](#) и [Opera](#) (встроенный инструмент [Opera Dragonfly](#)) отображают структуру документа в виде дерева, позволяя сворачивать и разворачивать узлы с помощью нажатий клавиши мыши.

- Применение стилей CSS

Процесс аналогичен применению [CSS](#) к [HTML](#)-документу для отображения. Для применения CSS при отображении в браузере, XML-документ должен содержать специальную ссылку на таблицу стилей. Например:

```
<?xml-stylesheet type="text/css" href="myStyleSheet.css"?>
```

Это отличается от подхода HTML, где используется элемент `<link>`.

- Применение преобразований к XS-FO формату

Современные браузеры принадлежат к числу программных средств, способных выполнять преобразования XSLT. В браузере такое преобразование выполняется, как правило, для форматирования документа (преобразования документа в формат XSL-FO). Следующая инструкция в прологе документа XML предписывает браузеру выполнить XSLT-преобразование, описанное в файле transform.xsl:

```
<?xml-stylesheet type="text/xsl" href="transform.xsl"?>
```

Редакторы XML[\[править\]](#) | [править исходный текст](#)

С документом XML можно работать в обычном текстовом редакторе, однако обычные редакторы не поддерживают структуру документа. Существуют специальные [редакторы XML](#), которые делают работу с документом более удобной и эффективной.

Системы управления базами данных, работающие с данными в формате XML[\[править\]](#) | [править исходный текст](#)

[Система управления базами данных DB2](#) позволяет хранить данные в формате XML и предоставляет доступ к таким данным с помощью языка XQuery.

Поддержка на аппаратном уровне[\[править\]](#) | [править исходный текст](#)

XML поддерживается на низком аппаратном, микропрограммном и программном уровнях в современных аппаратных решениях^[7].

Область применения, ограничения, перспективы развития[\[править\]](#) | [править исходный текст](#)

Эффективность использования XML[\[править\]](#) | [править исходный текст](#)

XML — язык разметки, другими словами, средство описания документа. Именно в нише документов, текстов, где доля разнотипных символьных данных велика, а доля разметки мала — XML успешен. С другой стороны, обмен данными в открытых системах не сводится к обмену документами. Избыточность разметки XML (а в целях разработки языка прямо указано, что лаконичность не является приоритетом проекта) сказывается в ситуациях, когда данные не вписываются в традиционную модель документа. Лента новостей, например, оформляемая с использованием синтаксиса XML (форматы [RSS](#), [Atom](#)), представляет собой не документ в традиционном понимании, а поток однотипных мини-документов — многословная и избыточная разметка в этом случае составляет существенную часть передаваемых данных.

W3C озабочен эффективностью применения XML, и соответствующие рабочие группы занимаются этой проблемой (к началу 2013 года нормативные документы не разработаны).

Другая ситуация, когда форматы XML могут оказаться не лучшим решением - работа с данными с простой структурой и небольшим по объёму содержанием полей данных. В этом случае доля разметки в общем объёме велика, а программная обработка XML может оказаться неоправданно затратной, по сравнению с работой с данными более простой структуры. В этой области разработчики рассматривают средства, изначально ориентированные на данные, такие как [INI](#), [YAML](#), [JSON](#).

Скриптовый язык для работы с XML[\[править\]](#) | [править исходный текст](#)

W3C работает над созданием скриптового языка для работы с XML (к началу 2013 года нормативные документы не разработаны).