



+ XML и XHTML: как устроены данные и веб-страницы

Разберём XML для данных и XHTML как строгую HTML-версию: структура, синтаксис и проверка.

Почему XML и XHTML появились в развитии интернет-технологий



XML стандартизировали в 1998 году в W3C как упрощённый поднабор SGML для универсального обмена структурированными данными. XHTML в 2000-м переписали из HTML в строгом XML-синтаксисе, чтобы точнее интерпретировать разметку программами и браузерами.

Что представляет собой XML



XML — расширяемый язык разметки: разработчик создаёт собственные теги под задачу. В нём важна не «красота» файла, а смысл структуры — элементы, атрибуты и вложенность.



Каждый фрагмент XML описывает данные: элементы связывают части документа, атрибуты задают дополнительные сведения, а текстовые значения передают конкретные результаты или параметры.



XML применяют там, где нужно, чтобы документ понимали и человек, и программа: конфигурационные файлы, обмен сообщениями, RSS-ленты, офисные форматы и другие системы, работающие с данными по правилам.

Основные правила построения XML-документа

01.

XML-документ начинается с одного корневого элемента, внутри которого размещаются все остальные части. Такая «точка входа» задаёт структуру и помогает парсеру однозначно понять документ.

02.

Все теги должны быть корректно закрыты, а вложенность — согласованной: открывающий и закрывающий теги образуют чёткую пару. Тогда элементы не «перекрываются» и структура остаётся читаемой.

03.

XML чувствителен к регистру: <Book> и <book> — разные имена элементов. Поэтому даже небольшая ошибка в написании делает документ другим по смыслу.

04.

Если нарушить эти правила, документ будет считаться плохо сформированным (well-formed не выполняется) и не сможет надёжно обрабатываться парсером или инструментами автоматизации.

XML и HTML: сравнение назначения и правил

Признак	XML	HTML
Назначение	Данные и обмен	Отображение страниц
Степень строгости	Высокая формальность	Более гибкий синтаксис
Чувствительность к ошибкам	Документ может быть отклонён	Браузер обычно «переваривает»
Типичное применение	Публикация и хранение данных	Интерфейс для пользователя

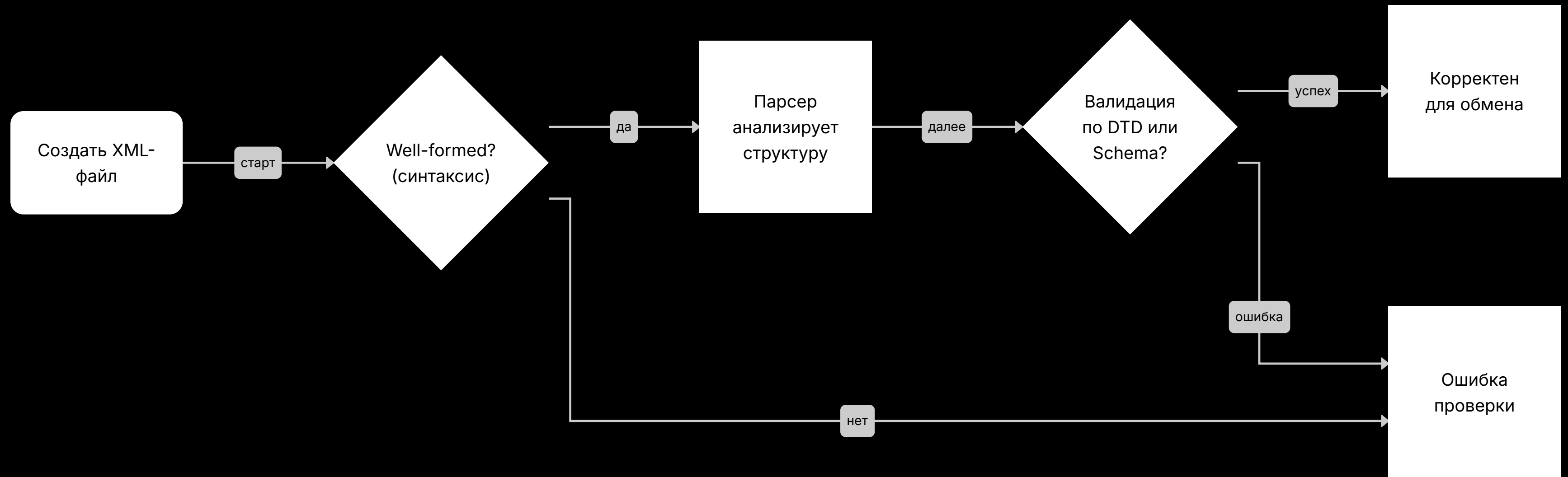
Сопоставление ключевых признаков показывает, что XML строят для данных, а HTML — для отображения. XML ориентирован на структуру и точность, HTML — на отображение и устойчивость к ошибкам.

XML выбирают там, где важна строгая структура и корректная интерпретация программами; HTML — там, где важен визуальный результат в браузере.

Спецификации W3C и общие принципы веб-стандартов

Проверка XML: от well-formed до валидации

Последовательность проверки документа на синтаксис и соответствие заданным правилам (DTD или XML Schema)



XML как дерево элементов

- **Иерархия вместо «просто текста»**

XML-документ удобно представлять как дерево: корневой элемент находится сверху, а дочерние узлы располагаются ниже. Внутри каждого узла могут быть ещё вложенные элементы и текстовые значения.



- **Зачем это удобно на практике**

Атрибуты дополняют элементы сведениями, не ломая иерархию. Такое представление помогает быстро находить данные, выбирать нужные фрагменты и выполнять преобразования через XPath и XSLT.

DTD и XML Schema: способы задания структуры



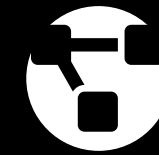
DTD

Задаёт базовые правила для структуры: какие элементы допустимы и в каком порядке они могут встречаться. Его возможности ограничены, особенно когда нужны точные типы значений.



XML Schema

Более выразительна: поддерживает типы данных, включая числовые, строковые и датированные. Часто используется, когда требуется строгий автоматический контроль формата.



Автопроверка

И DTD, и XML Schema служат для валидации: документ сравнивают с заданными правилами. Если структура не совпадает, обмен данными лучше не продолжать.

Где XML используется на практике

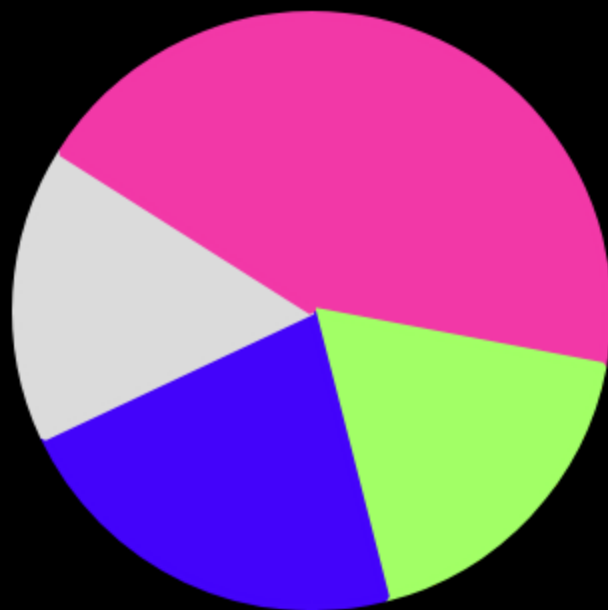
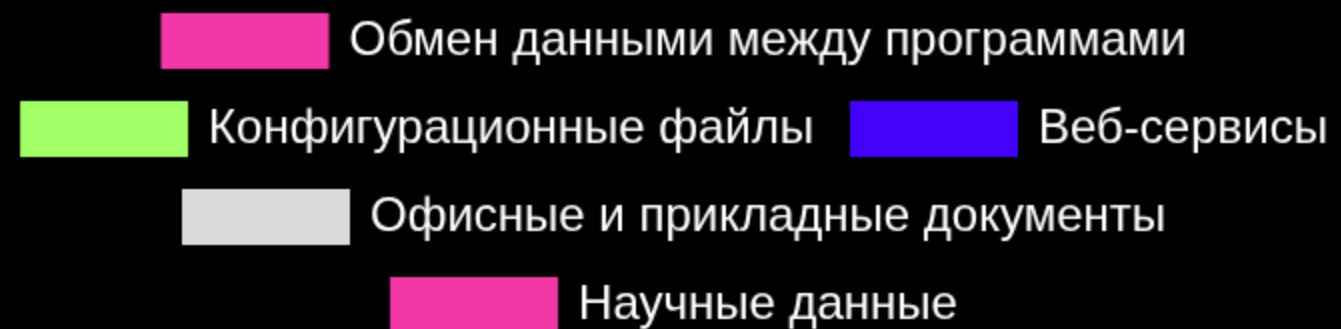


Диаграмма иллюстрирует идею: XML ценят как «контейнер» со строгой структурой, удобной для программной обработки и согласования форматов между системами.

Чем выше потребность в совместимости и понятной машине структуре, тем чаще выбирают XML. Это особенно заметно в обмене и сервисах, где ошибки стоят дорого.

Условные доли для учебной наглядности (точные статистические данные не задавались).

Что такое XHTML



- XHTML — это HTML, записанный по правилам XML. Он сохраняет знакомые элементы веб-страницы, но требует строгого XML-синтаксиса, чтобы разметка была однозначной для программ.
- В XHTML обычно обязательно корректно закрывать все элементы, а значения атрибутов указывать в кавычках. Вложение тегов должно быть согласованным, иначе XML-парсер сочтёт страницу некорректной.
- XHTML появился как попытка сделать веб-разметку более формальной: документы можно анализировать инструментами XML и проверять соответствие стандарту, уменьшая неоднозначность чтения разметки браузерами и программами.

Строгие правила синтаксиса XHTML



В XHTML пустые элементы оформляются в самозакрывающейся форме, например `<br />`. Это устраняет неоднозначность для XML-парсера и помогает программе сразу понять границы элемента без «догадок».



Значения атрибутов должны быть в кавычках. Например, `alt` для изображения записывается как `alt="описание"`, а не `alt=описание`. Такой формат снижает риск неверного чтения параметров.



Документ обычно содержит корректное объявление типа и/или использует пространство имён XHTML. Это важно для того, чтобы инструменты проверки и автоматической обработки понимали назначение разметки.



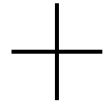
Любое нарушение синтаксиса — незакрытый тег, ошибка в вложенности или некорректные атрибуты — делает страницу некорректной с точки зрения XML-парсера. Проверка завершится ошибкой, а не «частичным» показом.

HTML и XHTML в учебном сравнении

- Сопоставим ключевые параметры: строгость вложенности, обязательность закрывающих тегов, чувствительность к ошибкам и удобство автоматической обработки. Это помогает понять, почему XHTML требует дисциплины при записи, а HTML допускает больше вариативности. Источник: W3C XHTML 1.0. Ключевой вывод: XHTML обеспечивает формальную корректность, но требует более внимательного написания разметки.
- XHTML формально «подталкивает» к корректному коду: меньше вариантов записи — выше вероятность того, что документ будет одинаково прочитан инструментами и браузерами.

W3C XHTML 1.0

Параметр	HTML	XHTML	
Строгость вложенности	Больше гибкости	Строгая	],[



Пространства имён позволяют объединять разметки с одинаковыми именами элементов в одном документе. Это особенно важно, когда используются несколько словарей, например XHTML, SVG и MathML.

3

При использовании пространств имён элементы из разных «словарей» однозначно различаются и не конфликтуют друг с другом.

W3C: материалы по XML Namespaces (общее описание механизма)

Сравнение XML, HTML и XHTML по степени строгости



Высокая формальность не всегда означает «лучше для человека»: она прежде всего удобна для автоматической проверки и преобразований, где важны предсказуемые правила чтения документа.

Диаграмма показывает: XML наиболее строг, HTML — наиболее терпим, а XHTML связывает веб-назначение с дисциплиной XML, делая код более проверяемым.

Сводное сравнение критериев строгого синтаксиса и ошибок по спецификациям W3C (качественная интерпретация)

Как XML преобразуется в другие форматы

01.

XPath позволяет выбирать узлы и фрагменты XML-документа по пути в древовидной структуре. Это даёт точный «адрес» нужных данных внутри файла, например при построении выборок для отчёта.

02.

XSLT преобразует структуру XML в другой формат, а XQuery применяется для запросов к XML-данным. Вместе эти средства обеспечивают публикации, отчёты и объединение данных из разных источников.



Преимущества и ограничения XML



Читаемость

XML читается и человеком, и программой: структура передаётся тегами, а данные отделены от логики. Это упрощает отладку и анализ документов при учебных и прикладных задачах.



Расширяемость

Разработчик может создавать собственные теги под конкретную предметную область. Поэтому XML удобен, когда нужно описывать новые сущности без смены общего механизма разметки.



Независимость и проверяемость

Файлы можно переносить между платформами, а строгие проверки помогают выявлять ошибки в структуре заранее. Это важно для обмена данных между системами и командами.

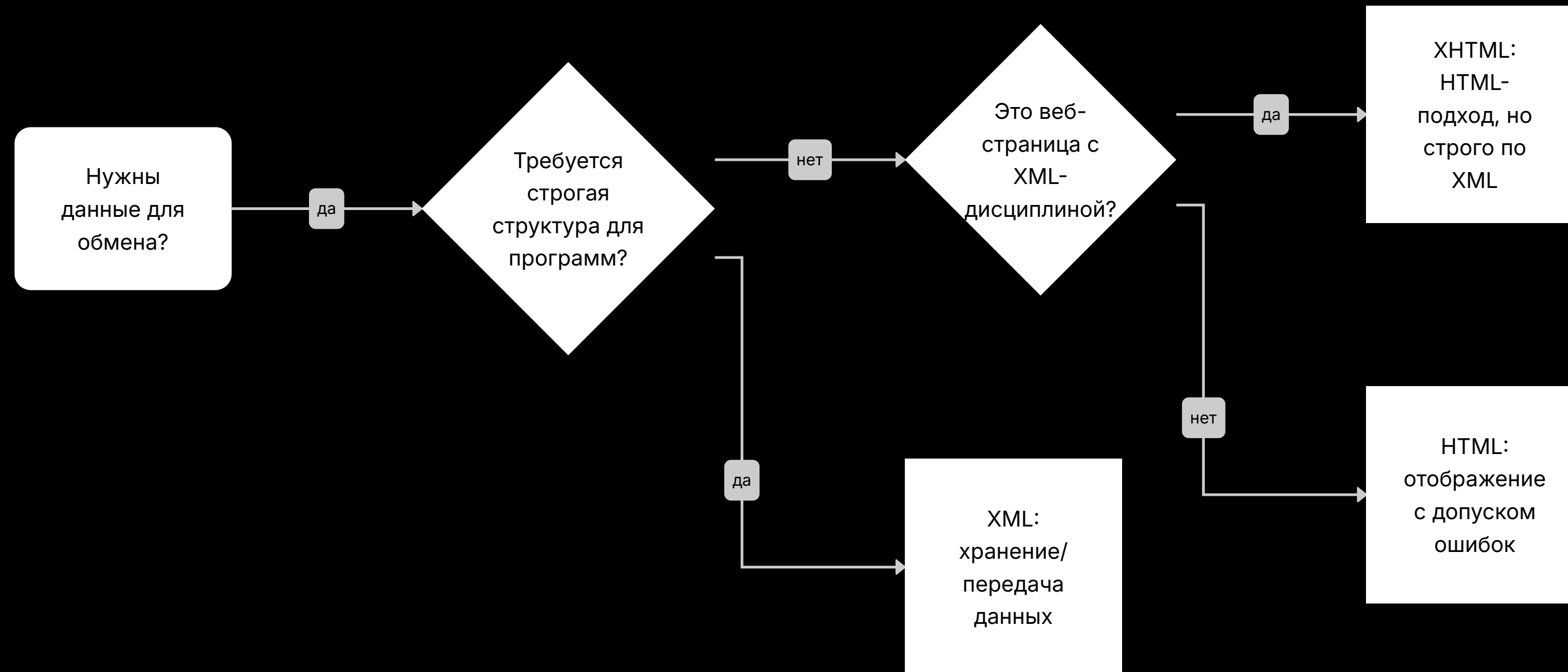


Многословность и объём

Из-за тегов XML часто получается больше по размеру и может работать медленнее, чем компактные форматы вроде JSON. Поэтому его выбирают, когда важнее контроль структуры.

Когда выбирают XML, XHTML или HTML

Выбор делается по назначению документа: данные, строгая веб-разметка или обычное отображение.



Типичные ошибки при работе с XML и XHTML

01.

Незакрытые теги и неправильная вложенность нарушают структуру документа. XML-парсер работает как строгий контролёр: он не принимает «почти правильный» формат и останавливает обработку при первом несоответствии.

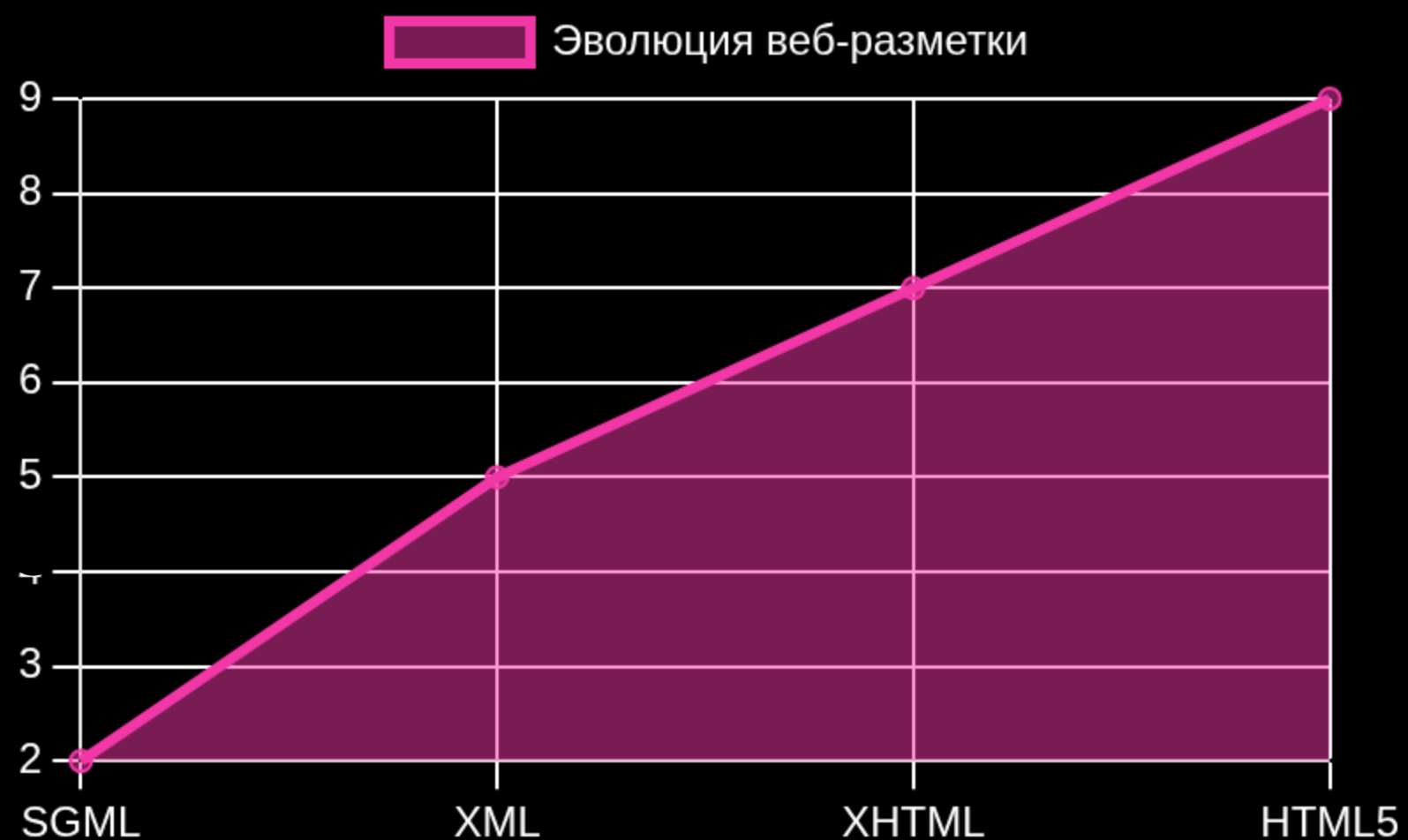
02.

Частая проблема — смешение строчных и прописных букв в именах тегов. В XML `<Tag>` и `<tag>` рассматриваются как разные сущности, поэтому ошибка в регистре ведёт к неверным связям элементов.

03.

Для XHTML критично и корректное объявление кодировки UTF-8. Если документ записан не так, как ожидают инструменты, результаты могут отличаться: от ошибок проверки до некорректного отображения текста.

Эволюция веб-разметки: от SGML к HTML5



Развитие шло к практичности и совместимости: строгие модели помогали в проверке, а затем усилился акцент на удобство написания и устойчивость веб-страниц в реальной среде.

XML сохраняет свою роль «контейнера» для данных, а XHTML выступает мостом между веб-страницами и дисциплиной XML. HTML5 расширяет свободу, но идея структурности не исчезает.

Обобщение линии развития стандартов веб-разметки по материалам W3C и историческим обзорам спецификаций (без количественных измерений)

Итоги: почему XML и XHTML важны для информатики



XML — универсальный язык описания структуры данных, а XHTML показывает, как строгие правила XML применяются к веб-страницам. Формальная структура повышает надёжность, совместимость и возможность автоматической обработки информации.