

Содержание:

Введение

В наши дни интернет повсеместно проходит в общественную жизнь. Каждый человек желает иметь собственную страничку в интернете. Но не каждый может ее создать. Простейшие страницы в интернете создаются с помощью языком гипертекстовой разметки, которых на сегодняшний день выпущено великое множество. Самым простейшим языком написания интернет-страниц является HTML. HTML является стандартизированным языком разметки документов в Интернете. Большинство web-страниц содержат описание разметки на языке HTML или XHTML.

Актуальностью данной работы является рассмотрение основ языков гипертекстовой разметки, их разновидностей и особенностей каждого языка с целью помощи при выборе наиболее подходящего языка для программирования веб-сайта или другого применения языка гипертекстовой разметки.

Целью данной работы является изучение основных элементов языков гипертекстовой разметки.

Объектом данного исследования является понятие языка разметки и история развития разновидностей языков разметки. Предметом исследования являются разновидности языков гипертекстовой разметки.

Задачами данной работы являются:

- изучение понятия языка разметки;
- обзор конкретных разновидностей языков разметки;
- рассмотрения поэтапной истории развития языков разметки;
- изучение конкретных примеров языков гипертекстовой разметки.

В основу данной работы легли книги таких авторов, как Фримен и Шаффер.

1. Понятия языков разметки

1.1. Язык разметки

Язык разметки в компьютерной терминологии является набором последовательностей или символов, которые вставляются в текст для передачи информации о его строении или выводе. Язык разметки принадлежит классу компьютерных языков. Текстовый документ, который написан с использованием языка разметки, содержит не только сам текст в виде последовательности знаков препинания и слов, но и такую дополнительную информацию о различных его участках, как например, списки, выделения, и указание на заголовки другие подобные элементы. В более сложных случаях язык разметки позволяет вставлять в документ содержание других документов или интерактивные элементы[1].

Следует отметить, что по Тьюрингу язык разметки неполон и обычно не считается языком программирования[2].

Разметки различаются на визуальную и логическую. В случае логической разметки речь идет только о том, какую роль играет данный участок документа в его общей структуре. А в случае визуальной разметки определяется непосредственно отображение элемента. Идея языков разметки состоит в том, что визуальное отображение документа должно автоматически получаться из логической разметки и не зависеть от его непосредственного содержания. Это упрощает автоматическую обработку документа и его отображение в различных условиях. К примеру, один и тот же файл может по-разному отображаться на экране мобильного телефона, компьютера или на печати, так как свойства данных устройств вывода существенно различаются. Но это правило часто нарушается: например, создавая документ в редакторе наподобие MS Word, пользователь может выделять заголовки жирным шрифтом, но нигде не указывать, что эта строка является заголовком[3].

Языки разметки используются везде, где требуется получение форматированного текста на основании одного лишь текста: в типографии. В качестве примеров таких языков можно привести TeX, SGML, RTF и PostScript, Также языки разметки применяются в пользовательских интерфейсах компьютеров, например языки Microsoft Word, troff, OpenOffice, Но чаще всего языки разметки встречаются в Интернете. Среди них распространены HTML, XML, XHTML, VML, XBRL, WML, SVG и PGML[4] [1 6].

1.2. История разновидностей языков разметки

Термин «разметка» подразумевает результат процесса, в английском языке называемого markup. Данный термин произошел от словосочетания «marking up», что в переводе значит размечивание или помечивание. Такое толкование взято из традиционной издательской практики проставления специальных условных пометок в тексте и на полях корректуры или рукописи перед передачей ее в печать. Таким образом разметчики указывали размер, стиль и гарнитуру шрифта для каждой части текста. На сегодняшний день разметкой текста занимаются графические дизайнеры, корректоры, редактор и часто сами авторы[5] [8].

1.2.1. GenCode

Идея использовать языки разметки в компьютерной обработке текстов, вероятнее всего, была впервые обнародована Вильямом Танниклиффом на конференции в 1967 году. Сам он именовал свое предложение «универсальным кодированием». В 1970-е годы Танниклифф руководил разработкой стандарта GenCode для издательской индустрии и позже занял пост руководителя одного из комитетов Международной организации по стандартизации, создавшего SGML, первый описательный язык разметки. Брайан Рэйд в своей диссертации, которую он защитил в 1980 году в университете Карнеги, в развитие предложенной концепции осуществил практическую реализацию описательной разметки[6].

Тем не менее, в настоящее время «отцом» языков разметки обычно называют исследователя IBM Чарльза Голдфарба. Основная концепция родилась у него в 1969 году, при работе над примитивной системой управления документами, предназначенной для адвокатских контор. В том же году он принял участие в создании языка IBM GML, который был впервые представлен в 1973 году[7].

В 1975 году Голдфарб переехал из Кембриджа, Массачусетс в Силиконовую долину, где стал разработчиком исследовательского центра IBM Almaden Research Center. Там он в 1978 году убедил руководство IBM использовать GML в коммерческих целях в составе разработанного компанией «средства формирования документов», после чего GML несколько лет широко использовался в бизнесе[8].

В 1978 году начались первые разработки, которые в конечном итоге привели к созданию стандарта SGML, основанного на GML и GenCode. Сам Голдфарб позднее возглавил комитет по SGML, который был принят организацией ISO в качестве стандартного в октябре 1986 года[9].

Некоторые ранние реализации компьютерных языков разметки можно обнаружить в типографских утилитах UNIX, таких как troff и proff. Они позволяют вставлять команды форматирования в текст документа для его форматирования согласно требованиям редактора.

Доступность издательского софта с функцией WYSIWYG, означающей «что увидишь, то и получишь», вытеснила большинство этих языков среди обычных пользователей, хотя серьезная издательская работа по-прежнему использует разметку для специфических не визуальных структур текста, а WYSIWYG-редакторы сейчас чаще всего сохраняют документы в форматах, основанных на языках разметки[\[10\]](#) [6, 8, 10].

1.2.2. TEX

Другой важный издательский стандарт — это TEX, созданный и впоследствии улучшенный Дональдом Кнутом в 70е-80е года двадцатого столетия. TEX собрал воедино высокие возможности форматирования текста и описания шрифтов, в особенности для математических книг профессионального качества. В настоящее время TEX является стандартом де-факто во многих научных дисциплинах. Помимо TEX существует LaTeX, который представляет собой широко используемую описательную систему разметки на основе TEX[\[11\]](#) [10].

1.2.3. Scribe, GML и SGML

Первым языком с четким и ясным различием между структурой и видом документа был Scribe, был создан и описан докторской диссертацией Брайана Рейда в 1980 году. Scribe был революционным в количестве способов обработки, не в последнюю очередь из-за введенной идеи стилей, отделенных от собственно текста и грамматики и управляющих использованием описательных элементов. Scribe оказал влияние на разработку языка GML и позже SGML, а также он является прямым предком языков HTML и LaTeX[\[12\]](#).

В начале 80х идея, что разметка должна быть сфокусирована на структурных аспектах документа и должна оставить внешнее представление документа интерпретатору, привела к созданию SGML. Язык был разработан комитетом, возглавляемым Голдфарбом. Он объединил идеи из многих источников, включая Тьюнникоффликский проект, GenCode. Шэрон Адлер, Андерс Бергланд и Джеймс А. Марк так же были ключевыми членами комитета SGML[\[13\]](#).

SGML точно определял синтаксис для включения разметки в текст, а также отдельно описывал, какие теги разрешены и где. Это давало возможность авторам создавать и использовать любую разметку, какую они захотят, выбирая теги для использования и давая им имена на нормальном языке. Таким образом, SGML следует считать метаязыком; множественные специальные языки разметок произошли от него. Поздние 80-е были наиболее значимыми в появлении новых языков разметок, основанных на SGML, например, TEI и DocBook[14].

В 1986 году SGML был опубликован как международный стандарт организацией ISO под номером 8879. SGML нашел широкое признание и широко использовался в очень больших проектах. Как бы то ни было, в целом было обнаружено, что он громоздок и труден для изучения, побочным эффектом языка было то, что он пытался сделать слишком много и быть чересчур гибким. Для примера, SGML создавал не всегда нужные закрывающие теги, потому что считал, что эта разметка будет добавлена вручную, штатом поддержки проекта, которые бы оценили экономию нажатий клавиш[15] [4, 8, 10, 12].

1.2.4. HTML

К 1991 году использование SGML было ограничено бизнес-программами и базами данных, а WYSIWYG-инструменты использовались для остальных программ обработки документов. Ситуация изменилась, когда сэр Тим Бернерс-Ли, узнав о SGML от своего коллеги Андерса Бергланда и других сотрудников в ЦЕРНе, использовал синтаксис SGML для создания HTML. Этот язык имел сходство с другими основанными на синтаксисе SGML языками разметки, однако начать его использование было гораздо проще даже тем разработчикам, которые никогда этого не делали. Стивен ДеРоуз убеждал, что HTML, использующий описательную разметку — это основной фактор в развитии Web, потому что в нем была заложена гибкость и расширяемость, а также другие факторы, включая понятие о URL и свободное использование браузерами. В наше время HTML — наиболее привлекательный и самый часто используемый язык разметки в мире[16].

Однако статус HTML как языка разметки оспаривается некоторыми компьютерными учеными. Их основным аргументом является то, что HTML ограничивает размещение тегов, требуя, чтобы оба тега были вложены в другие теги или в главные теги документа. Вследствие чего, эти ученые считают HTML контейнерным языком, следующим иерархической модели[17] [2, 8].

1.2.5. XML

XML обозначает расширяемый язык разметки и представляет собой мета-язык разметки, широко используемый в настоящее время. XML разработан консорциумом World Wide Web в комитете, возглавляемом Джоном Босаком. Основное предназначение XML — быть более простым, чем SGML и сфокусироваться на специфичной проблеме — документах в интернете. XML — мета-язык как SGML, пользователям разрешается создавать любые теги, какие необходимы. Становлению XML помогли, так как каждый XML-документ мог быть написан таким же способом, как и SGML-документ, а программы и пользователи, использующие SGML, могли перейти на XML достаточно легко[18].

Тем не менее, XML лишился многих ориентированных на людей особенностей языка SGML, упрощавших его использование. Другие улучшения исправляли некоторые проблемы SGML на международном уровне и делали возможным разбор документа иерархически, даже если не был доступен DTD.

XML был спроектирован, в основном, для не полностью структурированной среды, например, для документов и публикаций. Тем не менее, это привело к золотой середине между гибкостью и простотой, и он был быстро принят многими пользователями. В настоящее время XML широко используется для передачи данных между программами. Как HTML, он может быть охарактеризован как «контейнерный» язык[19] [2, 6].

1.2.6. XHTML

Начиная с января 2000 года, все рекомендации для W3C были основаны на XML в большей степени, чем на SGML, была предложена аббревиатура XHTML, расшифровывающаяся как Расширяемый Гипертекстовый Язык Разметки. Спецификации языка требовали, чтобы XHTML-документы были оформлены как XML-документы, это позволяет использовать XHTML для более четких и точных документов, используя теги от обычного языка HTML[20].

Но, бесспорно, одним из самых достойных внимания различий между языками HTML и XHTML является правило, которое гласит, что все теги должны быть закрытыми: пустые теги, такие как `
`, должны быть оба закрыты стандартным закрывающим тегом или специальной записью: `
`. Пробел перед `</>` в закрывающем теге является опциональным, но часто используется, поскольку его используют некоторые пре-XML браузеры, также SGML-парсеры. Другие атрибуты в тегах должны быть в кавычках. Наконец, все теги и имена атрибутов должны быть написаны в нижнем регистре, чтобы восприниматься правильно; HTML

невосприимчив к регистру[21] [1, 2].

По итогам данной главы можно сделать вывод, что языки разметки начали свое развитие в 70-х годах 20 века и до сих пор развиваются, привнося новые элементы и создавая новые языки, более удобные для пользователя.

2. Конкретные реализации языков гипертекстовой разметки

2.1. Особенности языков разметки

Общая особенность всех языков разметки в том, что они перемешивают текст документа с инструкциями разметки в потоке данных или файле. Это не необходимость, возможно изолировать разметку от текста, используя указатели, метки, идентификаторы или другие методы координации. Такая «отделенная разметка» характерна для внутреннего представления программ, работающих с размеченными документами. Тем не менее, внедренная или «междустрочная» разметка более принята где бы то ни было. Вот, для примера, небольшая часть текста, размеченного при помощи HTML:

Код разметочных инструкций обведен угловыми скобками <как здесь>. Текст, находящийся между этими инструкциями, является текстом документа. Коды h1, p и em — примеры структурной разметки, они описывают позицию, назначение или значение текста, включенного в них[22].

Более точно, h1 означает «это заголовок первого уровня», p значит «это параграф», а em означает «это подчеркнутое слово или фраза».

Интерпретирующая программа может применять эти правила или стили для показа различных частей текста, используя различные гарнитуры, размеры шрифта, отступы, цвет или другие стили, если это необходимо. Такой тег, как h1, может быть, например, представлен большим жирным типографским шрифтом, или в документе с моноширинным текстом может быть подчеркнутым, или может вообще не менять внешний вид[23].

Для контраста, тег i в HTML — пример визуальной разметки; обычно он используется для определения специфичных особенностей текста без объяснения

причин.

TEI опубликовала всесторонние руководящие документы, указывающие кодировать текст в интересах человечества и научных обществ. Эти руководства использовались для кодирования исторических документов, специфичных работах ученых, периодике и так далее[24] [1, 2, 6].

2.2. Язык HTML

HTML расшифровывается как HyperText Markup Language и обозначает язык гипертекстовой разметки. HTML является стандартизированным языком разметки документов в Интернете. Большинство web-страниц содержат описание разметки на языке HTML или XHTML. Язык HTML интерпретируется браузерами, после чего текст, полученный в результате интерпретации форматированный, отображается на экране монитора компьютера или мобильного устройства[25].

Язык HTML является приложением стандартного обобщенного языка разметки и соответствует международному стандарту ISO 8879.

Язык XHTML является более строгим вариантом HTML, он следует всем ограничениям XML и, фактически, XHTML можно воспринимать как приложение языка XML к области разметки гипертекста[26].

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования[27].

Язык HTML был разработан британским ученым Тимом Бернерсом-Ли приблизительно в 1986—1991 годах в стенах Европейского центра ядерных исследований в Женеве в Швейцарии. HTML создавался как язык для обмена технической и научной документацией, пригодный для использования людьми, не являющимися специалистами в области верстки. HTML успешно справлялся с проблемой сложности SGML путем определения небольшого набора семантических и структурных элементов — дескрипторов. Дескрипторы также часто называют «тегами». С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже[28].

Изначально язык HTML задумывался и создавался как средство форматирования и структурирования документов без их привязки к средствам отображения. В идеале, текст с разметкой HTML должен был без структурных и стилистических искажений воспроизводиться на оборудовании с различной технической оснащённостью, таком как цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов. Но современное применение HTML очень далеко от его изначальной задачи. Например, тег `<table>` предназначен для создания в документах таблиц, но часто используется и для оформления размещения элементов на странице. С течением времени основная идея платформонезависимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом оформлении[29].

Текстовые документы, содержащие разметку на языке HTML, традиционно имеют расширение `htm` или `html` и обрабатываются специальными приложениями, отображающими документ в его форматированном виде. Такие приложения, называемые «интернет-обозревателями» или «браузерами», обычно предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра и вывода на иные внешние устройства и, при необходимости, отправки введенных пользователем данных на сервер. Наиболее популярными на сегодняшний день браузерами являются Mozilla Firefox, Google Chrome, Safari, Internet Explorer и Opera.

В настоящее время Консорциум Всемирной паутины разработал HTML версии 5. Черновой вариант спецификации языка появился в Интернете 20 ноября 2007 года [30].

Сообщество WHATWG, расшифровывающимся как Web Hypertext Application Technology Working Group, начиная с 2004 года, разрабатывается спецификация Web Applications 1.0, часто неофициально называемая «HTML 5», которая расширяет HTML, имея также совместимый с XHTML 1.0 XML-синтаксис, с целью лучшего представления семантики различных типичных страниц, например, сайтов аукционов, форумов, онлайн-магазинов, поисковых систем и прочих ресурсов, которые не очень удачно вписываются в модель XHTML 2.0[31] [3, 5, 7].

2.3. Язык HTML5

HTML5 является языком HTML 5 версии, и представляет собой язык для представления и структурирования содержимого всемирной паутины. Несмотря на

то, что стандарт был завершен только в 2014 году, еще с 2013 года браузерами оперативно осуществлялась поддержка, а разработчиками — использование рабочего стандарта под названием HTML Living Standard. Целью разработки HTML5 является улучшение уровня поддержки мультимедиа-технологий с одновременным сохранением обратной совместимости, удобочитаемости кода для человека и простоты анализа для парсеров[32].

Во всемирной паутине долгое время использовались стандарты HTML 4.01, XHTML 1.0 и XHTML 1.1. Веб-страницы на практике оказывались сверстанными с использованием смеси особенностей, представленных различными спецификациями, включая спецификации программных продуктов, таких как веб-браузеры, а также сложившихся общеупотребительных приемов. HTML5 был создан как единый язык разметки, который мог бы сочетать синтаксические нормы HTML и XHTML. Он рационализирует, улучшает и расширяет разметку документов, а также добавляет единый API для сложных веб-приложений[33].

В HTML5 реализовано множество новых синтаксических особенностей. Например, элементы `<canvas>`, `<audio>` и `<video >`, а также возможность использования SVG и математических формул. Эти новшества разработаны для упрощения управления и создания мультимедийных и графических объектов в сети без необходимости использования сторонних API и плагинов. Другие новые элементы, такие как `<article>`, `<section>`, `< nav>` и `<header>`, разработаны для того, чтобы обогащать семантическое содержимое документа. Новые атрибуты были введены с той же целью, хотя ряд элементов и атрибутов был удален. Некоторые элементы, такие как `<cite>`, `<menu>` и `<a>`, были изменены, переопределены или стандартизированы. DOM и API стали основными частями спецификации HTML5. HTML5 также определяет некоторые особенности обработки ошибок верстки, поэтому синтаксические ошибки должны рассматриваться одинаково всеми совместимыми браузерами[34].

WHATWG начал работу над новым стандартом в 2004 году, когда консорциум World Wide Web Consortium, обозначаемый как W3C сосредоточился на будущих разработках XHTML 2.0, а HTML 4.01 не изменялся с 2000 года. В 2009 году W3C признал, что срок работы у рабочей группы XHTML 2.0 истек, и решил не возобновлять его. Впоследствии W3C и WHATWG совместно разрабатывали HTML5 [35].

Даже несмотря на то, что HTML5 был хорошо известен среди веб-разработчиков в течение нескольких лет, он стал основной темой средств массовой информации

только в апреле 2010 года. После этого Стив Джобс, будучи главой компании Apple Inc., написал публичное письмо, заголовок которого гласил: «мысли по поводу Flash», где он заключил, что с разработкой HTML5 нет больше необходимости смотреть видеоролики или использовать другие виды приложений с помощью Adobe Flash. По этому поводу вспыхивали дебаты в кругу веб-разработчиков, причем некоторые намекали, что, хотя HTML5 и обеспечивает расширенную функциональность, разработчики должны принимать во внимание различия браузеров и необходимость поддержки различных частей стандартов, равно как и функциональные различия между HTML5 и Flash[\[36\]](#) [3, 5, 9, 11].

2.4. Язык XHTML

XHTML расшифровывается как extensible hypertext markup language, что значит расширяемый язык гипертекстовой разметки. XHTML является семейством языков разметки веб-страниц на основе XML, повторяющих и расширяющих возможности HTML 4. Спецификации XHTML 1.0 и XHTML 1.1 являются рекомендациями консорциума Всемирной паутины. Развитие XHTML остановлено; новые версии XHTML не выпускаются; рекомендуется использовать HTML[\[37\]](#).

Главное отличие XHTML от HTML заключается в обработке документа. Документы XHTML обрабатываются своим модулем аналогично документам XML. В процессе этой обработки не исправляются допущенные разработчиками ошибки.

XHTML соответствует спецификации SGML, поскольку XML является ее подмножеством. HTML обладает множеством особенностей в процессе обработки и фактически перестал относиться к семейству SGML, что и закреплено в черновике спецификации HTML 5[\[38\]](#).

Браузер выбирает парсер для обработки документа на основании заголовка content-type, полученного от сервера:

- для HTML — text/html;
- для XHTML — application/xhtml+xml;
- для локального просмотра на клиенте выбор основывается на расширении файла;
- в Internet Explorer вплоть до 8-й версии парсер обработки XHTML-документов отсутствует[\[39\]](#).

Декларация DOCTYPE может не оказывать никакого влияния на определение парсера для обработки в зависимости от используемого браузера.

Согласно синтаксису XHTML:

- все элементы должны быть закрыты. Теги, которые не имеют закрывающего тега (например, `` или `
`), должны иметь на конце `/` (например, `
`);
- логические атрибуты записываются в развернутой форме. Например, следует писать `<option selected="selected">` или `<td nowrap="nowrap">`;
- имена тегов и атрибутов должны быть записаны строчными буквами (например, `` вместо ``);
- XHTML гораздо строже относится к ошибкам в коде; `<` и `&` везде, даже в URL, должны замещаться `<` и `&` соответственно. По рекомендации W3C браузеры, встретив ошибку в XHTML, должны сообщить о ней и не обрабатывать документ. Для HTML браузеры должны были попытаться понять, что хотел сказать автор;
- кодировкой по умолчанию является UTF-8[40].

Для XHTML-страниц рекомендуется задавать MIME-тип — `application/xhtml+xml`, но это не является обязательным, более того — браузер Internet Explorer 8 и младшие версии не смогут обрабатывать страницу, поэтому с XHTML 1.0 традиционно используется MIME-тип для HTML — `text/html`[41].

Существует три типа документов XHTML: `strict`, `transitional` и `frameset`. Наиболее употребительной и универсальной из версий XHTML является переходная, поскольку она позволяет использовать `iframe` и атрибут `target` у ссылок для указания, например, того, что ссылке необходимо открываться в новом окне. Фреймовая версия представляет собой расширенный вариант `transitional` и добавляет к нему, как следует из названия, возможность установки `frameset` вместо `body`. Определение типа документа строгой версии XHTML не содержит многих тегов и атрибутов, описанных в DTD `transitional` и признанных устаревшими [42] [1, 2, 4, 10].

2.5. Язык SGML

SGML расшифровывается как `Standard Generalized Markup Language`, что значит стандартный обобщенный язык разметки. SGML является метаязыком, на котором

можно определять язык разметки для документов. SGML — наследник разработанного в 1969 году в IBM языка Generalized Markup Language[43].

Изначально SGML был разработан для совместного использования машинно-читаемых документов в больших правительственных и аэрокосмических проектах. Он широко использовался в печатной и издательской сфере, но его сложность затруднила его широкое распространение для повседневного использования[44].

Основными частями документа SGML:

- SGML-декларация, которая определяет, какие ограничители и символы могут появляться в приложении;
- Document Type Definition, который определяет синтаксис конструкций разметки. DTD может включать дополнительные определения, такие, как символьные ссылки-мнемоники;
- спецификация семантики, которая относится к разметке и также дает ограничения синтаксиса, которые не могут быть выражены внутри DTD;
- содержимое SGML-документа, в котором, по крайней мере, должен быть корневой элемент[45].

Язык SGML предоставляет множество вариантов синтаксической разметки для использования различными приложениями. Изменяя SGML-декларацию, можно даже отказаться от использования угловых скобок, хотя этот синтаксис считается стандартным, так называемым concrete reference syntax.

Языки HTML и XML произошли от SGML. HTML является приложением SGML, а XML — подмножеством SGML, которое разработано для упрощения процесса машинного разбора документа. Другими приложениями SGML являются SGML Docbook и «Z Format»[46] [10, 12].

2.6. Язык XML

XML расшифровывается как eXtensible Markup Language, что значит расширяемый язык разметки. XML рекомендован Консорциумом Всемирной паутины.

Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров. XML-процессорами называют программы, которые читают XML-документы и обеспечивают 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 подчеркиванием 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. Расширение XML — это конкретная грамматика, созданная на базе XML и представленная словарем тегов и их атрибутов, а также набором правил, определяющих какие атрибуты и элементы могут входить в состав других элементов. Если бы HTML появился позднее, чем XML, то он, возможно, также бы являлся его расширением. 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[\[47\]](#) [10].

2.7. Язык JSON

JSON расшифровывается как JavaScript Object Notation и является текстовым форматом обмена данными, основанным на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми. Формат JSON был разработан Дугласом Крокфордом.

Несмотря на происхождение от JavaScript, формат считается независимым от языка и может использоваться практически с любым языком программирования. Для многих языков существует готовый код для создания и обработки данных в формате JSON[\[48\]](#).

За счет своей лаконичности по сравнению с XML, формат JSON может быть более подходящим для сериализации сложных структур. Если говорить о веб-приложениях, в таком ключе он уместен в задачах обмена данными как между браузером и сервером, так и между самими серверами.

Поскольку формат JSON является подмножеством синтаксиса языка JavaScript, то он может быть быстро десериализован встроенной функцией `eval()`. Кроме того, возможна вставка вполне работоспособных JavaScript-функций. В языке PHP, начиная с версии 5.2.0, поддержка JSON включена в ядро в виде функций

`json_decode()` и `json_encode()`, которые сами преобразуют типы данных JSON в соответствующие типы PHP и наоборот[\[49\]](#).

JSON-текст в закодированном виде представляет собой одну из двух структур:

- набор пар ключ: значение. В различных языках это реализовано как объект, запись, структура, словарь, хэш-таблица, список с ключом или ассоциативный массив. Ключом может быть только регистрозависимая строка, где имена с буквами в разных регистрах считаются разными, а значением — любая форма.
- упорядоченный набор значений. Во многих языках это реализовано как вектор, массив, последовательность или список[\[50\]](#).

Это универсальные структуры данных: как правило, любой современный язык программирования поддерживает их в той или иной форме. Они легли в основу JSON, так как он используется для обмена данными между различными языками программирования.

В качестве значений в JSON могут быть использованы:

- объект являющийся неупорядоченным множеством пар ключ:значение, заключенным в фигурные скобки «{ }». Ключ описывается строкой, между ним и значением стоит символ «:». Пары ключ-значение отделяются друг от друга запятыми;
- одномерный массив являющийся упорядоченным множеством значений. Массив заключается в квадратные скобки «[]». Значения разделяются запятыми;
- число;
- литералы `null`, `true` и `false`.
- строка являющаяся упорядоченным множеством из нуля или более символов юникода, заключенным в двойные кавычки. Символы могут быть указаны с использованием escape-последовательностей, начинающихся с обратной косой черты «\» (поддерживаются варианты `"`, `\\`, `\/`, `\/t`, `\/n`, `\/r`, `\/f` и `\/b`), или записаны шестнадцатеричным кодом в кодировке UTF-8 в виде `uFFFF`[\[51\]](#) [10, 11, 12].

По итогам данной главы можно сделать вывод, что язык существует большое количество разнообразных языков разметки, каждый из которых удобен при применении в определенной ситуации.

Заключение

В рамках данной работы было рассмотрено понятие языков разметки в целом. Язык разметки в компьютерной терминологии является набором последовательностей или символов, которые вставляются в текст для передачи информации о его строении или выводе. Также была рассмотрена история отдельных разновидностей языков разметки, где были выделены GenCode, TEX, Scribe, GML, SGML, HTML, XML и XHTML.

Во второй части работы были рассмотрены основы языка HTML и его разновидности HTML5. HTML расшифровывается как HyperText Markup Language и обозначает язык гипертекстовой разметки. HTML является стандартизированным языком разметки документов в Интернете. Большинство web-страниц содержат описание разметки на языке HTML или XHTML. HTML5 является языком HTML 5 версии, и представляет собой язык для представления и структурирования содержимого всемирной паутины. Стандарт был завершен в 2014 году. Также были рассмотрены и другие языки, такие как XHTML, SGML, XML и JSON.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Веру Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – СПб.: Питер, 2016. – 336 с.
2. Лабберс П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б. Олберс, Ф. Салим. – М.: Вильямс, 2011. – 272 с.
3. Макфарланд Д. Новая большая книга CSS / Д. Макфарланд. – СПб.: Питер, 2016. – 720 с.
4. Мирный А. HTML5 против Flash-видео / А. Мирный // UP Special : журнал. – 2010. – № 5. – С. 42 – 45.
5. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – СПб.: Питер, 2015. – 688 с.
6. Прохоренок П. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / П. Прохоренок, В. Дронов. – СПб.: БХВ-Петербург, 2015. – 766 с.
7. Роббинс Д. HTML5. Карманный справочник / Д. Роббинс. – М.: Вильямс, 2015. – 192 с.
8. Титтел Э., HTML, XHTML и CSS для чайников / Э. Титтел, Д. Ноубл. – М.: Диалектика, 2011. – 400 с.

9. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен. – СПб.: Питер, 2012. – 656 с.
10. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – СПб.: Питер, 2014. – 737 с.
11. Хольцнер С. HTML5 за 10 минут / С. Хольцнер. – М.: Вильямс, 2011. – 384 с.
12. Шафер С. HTML, XHTML и CSS. Библия пользователя / С. Шафер. – М.: Диалектика, 2010. – 656 с.

1. Веру Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – СПб.: Питер, 2016. – С. 135. [↑](#)
2. Прохоренок П. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / П. Прохоренок, В. Дронов. – СПб.: БХВ-Петербург, 2015. – С. 346. [↑](#)
3. Прохоренок П. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / П. Прохоренок, В. Дронов. – СПб.: БХВ-Петербург, 2015. – С. 261. [↑](#)
4. Веру Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – СПб.: Питер, 2016. – С. 127. [↑](#)
5. Титтел Э., HTML, XHTML и CSS для чайников / Э. Титтел, Д. Ноубл. – М.: Диалектика, 2011. – С. 326. [↑](#)
6. Титтел Э., HTML, XHTML и CSS для чайников / Э. Титтел, Д. Ноубл. – М.: Диалектика, 2011. – С. 362. [↑](#)
7. Прохоренок П. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / П. Прохоренок, В. Дронов. – СПб.: БХВ-Петербург, 2015. – С. 126. [↑](#)
8. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – СПб.: Питер, 2014. – С. 437. [↑](#)
9. Титтел Э., HTML, XHTML и CSS для чайников / Э. Титтел, Д. Ноубл. – М.: Диалектика, 2011. – С. 336. [↑](#)

10. Прохоренок П. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / П. Прохоренок, В. Дронов. – СПб.: БХВ-Петербург, 2015. – С. 124. [↑](#)
11. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – СПб.: Питер, 2014. – С. 553. [↑](#)
12. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – СПб.: Питер, 2014. – С. 666. [↑](#)
13. Титтел Э.,HTML, XHTML и CSS для чайников / Э. Титтел, Д. Ноубл. – М.: Диалектика, 2011. – С. 336. [↑](#)
14. Шафер С. HTML, XHTML и CSS. Библия пользователя / С. Шафер. – М.: Диалектика, 2010. – С. 366. [↑](#)
15. Лабберс П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б. Олберс, Ф. Салим. – М.: Вильямс, 2011. – С. 166. [↑](#)
16. Мирный А. HTML5 против Flash-видео / А. Мирный // UP Special : журнал. – 2010. – № 5. – С. 42. [↑](#)
17. Титтел Э.,HTML, XHTML и CSS для чайников / Э. Титтел, Д. Ноубл. – М.: Диалектика, 2011. – С. 225. [↑](#)
18. Лабберс П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б. Олберс, Ф. Салим. – М.: Вильямс, 2011. – С. 51. [↑](#)
19. Прохоренок П. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / П. Прохоренок, В. Дронов. – СПб.: БХВ-Петербург, 2015. – С. 116. [↑](#)
20. Веру Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – СПб.: Питер, 2016. – С. 124. [↑](#)

21. Лабберс П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б. Олберс, Ф. Салим. – М.: Вильямс, 2011. – С. 177. [↑](#)
22. Лабберс П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б. Олберс, Ф. Салим. – М.: Вильямс, 2011. – С. 133. [↑](#)
23. Веру Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – СПб.: Питер, 2016. – С. 138. [↑](#)
24. Прохоренок П. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / П. Прохоренок, В. Дронов. – СПб.: БХВ-Петербург, 2015. – С. 346. [↑](#)
25. Роббинс Д. HTML5. Карманный справочник / Д. Роббинс. – М.: Вильямс, 2015. – С. 46. [↑](#)
26. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – СПб.: Питер, 2015. – С. 72. [↑](#)
27. Макфарланд Д. Новая большая книга CSS / Д. Макфарланд. – СПб.: Питер, 2016. – С. 362. [↑](#)
28. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – СПб.: Питер, 2015. – С. 462. [↑](#)
29. Роббинс Д. HTML5. Карманный справочник / Д. Роббинс. – М.: Вильямс, 2015. – С. 26. [↑](#)
30. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – СПб.: Питер, 2015. – С. 362. [↑](#)
31. Макфарланд Д. Новая большая книга CSS / Д. Макфарланд. – СПб.: Питер, 2016. – С. 337. [↑](#)

32. Хольцнер С. HTML5 за 10 минут / С. Хольцнер. – М.: Вильямс, 2011. – С. 274. [↑](#)
33. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен. – СПб.: Питер, 2012. – С. 274. [↑](#)
34. Хольцнер С. HTML5 за 10 минут / С. Хольцнер. – М.: Вильямс, 2011. – С. 16. [↑](#)
35. Макфарланд Д. Новая большая книга CSS / Д. Макфарланд. – СПб.: Питер, 2016. – С. 537. [↑](#)
36. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – СПб.: Питер, 2015. – С. 62. [↑](#)
37. Лабберс П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б. Олберс, Ф. Салим. – М.: Вильямс, 2011. – С. 325. [↑](#)
38. Веру Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – СПб.: Питер, 2016. – С. 136. [↑](#)
39. Мирный А. HTML5 против Flash-видео / А. Мирный // UP Special : журнал. – 2010. – № 5. – С. 44. [↑](#)
40. Веру Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – СПб.: Питер, 2016. – С. 267. [↑](#)
41. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – СПб.: Питер, 2014. – С. 477. [↑](#)
42. Веру Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – СПб.: Питер, 2016. – С. 277. [↑](#)

43. Шафер С. HTML, XHTML и CSS. Библия пользователя / С. Шафер. – М.: Диалектика, 2010. – С. 247. [↑](#)
44. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – СПб.: Питер, 2014. – С. 247. [↑](#)
45. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – СПб.: Питер, 2014. – С. 364. [↑](#)
46. Шафер С. HTML, XHTML и CSS. Библия пользователя / С. Шафер. – М.: Диалектика, 2010. – С. 466. [↑](#)
47. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – СПб.: Питер, 2014. – С. 472. [↑](#)
48. Шафер С. HTML, XHTML и CSS. Библия пользователя / С. Шафер. – М.: Диалектика, 2010. – С. 236. [↑](#)
49. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – СПб.: Питер, 2014. – С. 362. [↑](#)
50. Хольцнер С. HTML5 за 10 минут / С. Хольцнер. – М.: Вильямс, 2011. – С. 17. [↑](#)
51. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – СПб.: Питер, 2014. – С. 257. [↑](#)