

## Содержание:

# Введение

Любой документ, который встречается в сети, помимо собственно текста обязательно имеет строгую внутреннюю структуру. Реализация клиентской стороны пользовательского интерфейса (front end), естественно, невозможна без использования языков гипертекстовой разметки.

Это базис которым должен владеть любой опытный разработчик. Именно повсеместная распространённость языков текстовой разметки обуславливает актуальность данного исследования.

Теоретическая значимость исследования связана с необходимостью систематизировать, классифицировать и упорядочить разрозненные данные об истории языков гипертекстовой разметки, о классификации этих языков, их взаимосвязи и функциях.

Объектом работы выступают языки программирования в целом, предметом - языки гипертекстовой разметки в частности, их взаимосвязь, история и иерархическая структура.

Цель работы вытекает из теоретической обоснованности и формулируется следующим образом: в процессе исследования необходимо провести комплексный анализ языков гипертекстовой разметки как предмета изучения информатики, выявить парадигмы их развития и актуальное состояние на сегодняшний день.

Цель работы последовательно решается посредством следующих задач:

1. Проанализировать понятийный аппарат темы;
2. Вывести классификацию и функции языков текстовой разметки;
3. Определить, принадлежат ли языки разметки к классу языков программирования;
4. Проследить развитие языков разметки в исторической динамике;
5. Обозначить основные характеристики наиболее распространённых и важных языков разметки.

Структура работы повторяет выше обозначенные задачи. Работа состоит из двух глав, девяти параграфов и библиографического списка. Первая глава посвящена теоретическим вопросам (понятийный аппарат, историческая справка), вторая глава - практическим аспектам темы.

В процессе проведения исследования автор ссылается на отечественные и зарубежные учебники, книги, статьи научных журналов, интернет-ресурсы и собственный опыт веб-разработчика.

## **Глава I. Языки гипертекстовой разметки: понятийный аппарат, функции, история**

### **1.1. Понятийный аппарат**

Приступая к любому теоретическому исследованию, необходимо первоначально определить понятийный аппарат. Разобрать основные термины и понятия, дать им главную характеристику.

Итак, что же такое язык гипертекстовой разметки? Это определённая совокупность символов, последовательностей, которые интегрированы с текстом и передают устройству информацию о строении текстового массива, об особенностях его вывода.[\[1\]](#)

Языки гипертекстовой разметки формально принадлежал к классу компьютерных языков, вопрос об отнесении языков разметки к языкам программирования достаточно спорный и будет детально разобран в данной главе.

Итак, гипотетический текстовый документ может содержать язык разметки. Кроме самого текста (совокупности знаков препинания и слов) в документе будет содержаться информация об особенностях отображения фрагментов этого текста. Мы передаём устройству информацию о том, что в этой части должен быть заголовок, а этот кусок должен быть выделен, следующей - выведен списком и так далее.[\[2\]](#)

Современные языки разметки позволяют выводить различные интерактивные элементы или даже содержания других электронных документов.

Выделим основные свойства языков разметки, подытожив раздел:

1. Язык разметки имеет свой синтаксис, лексику, семантику;
2. Язык разметки используется не отдельно, а совместно с основным текстом;
3. Язык разметки призван передать устройству информацию о строении текста или об особенностях вывода информации.

Мы выделили три основных критерия, по которым можно ориентироваться, классифицируя компьютерные языки для выделения из них языков разметки.

## 1.2. Языки, разметки как языки программирования

Как было сказано выше, вопрос отнесения языков разметки к языкам программирования является достаточно дискуссионным.[\[3\]](#)

Для начала, по знакомой методологии, выделим основные черты и функции языков программирования.

Итак:

Язык программирования — *формальная знаковая система*, предназначенная для записи компьютерных программ. Язык программирования определяет набор *лексических, синтаксических и семантических* правил, задающих *внешний вид* программы и *действия*, которые выполнит исполнитель (компьютер) под её управлением.[\[4\]](#)

Большинство ресурсов либо полностью употребляет это определение, либо вольно излагает его без потери смысла. Рассмотрим поподробнее составные части определения ЯП:

- Формальный язык — это множество конечных слов (строк, цепочек) над конечным алфавитом.
- Знаковая система — это система однозначно интерпретируемых и трактуемых сообщений/сигналов, которыми можно обмениваться в процессе общения. Иногда знаковые системы помогают структурировать процесс общения с целью придания ему некой адекватности в плане реакций его участников на те или иные «знаки». В качестве примера знаковой системы обычно приводят язык (как в письменной форме так и, в случае естественных

языков, в форме речи).[5]

- Компьютерная программа — последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины.
- Лёксика — совокупность слов того или иного языка, части языка или слов, которые знает тот или иной человек или группа людей.
- Синтаксис — сторона языка программирования, которая описывает структуру программ как наборов символов (обычно говорят — безотносительно к содержанию). Синтаксису языка противопоставляется его семантика. Синтаксис языка описывает «чистый» язык, в то же время семантика приписывает значения (действия) различным синтаксическим конструкциям.[6]
- Семантика в программировании — дисциплина, изучающая формализации значений конструкций языков программирования посредством построения их формальных математических моделей. В качестве инструментов построения таких моделей могут использоваться различные средства, например, математическая логика,  $\lambda$ -исчисление, теория множеств, теория категорий, теория моделей, универсальная алгебра. Формализация семантики языка программирования может использоваться как для описания языка, определения свойств языка, так и для целей формальной верификации программ на этом языке программирования.
- Язык — знаковая система, соотносящая понятийное содержание и типовое звучание (написание)

Более простым языком это может быть изложено так:

Язык программирования — множество заранее определенных, однообразных и понятных исполнителю (читай:

интерпретатору/компилятору/компьютеру/программисту) инструкций, предназначенных для записи последовательно с целью их исполнения неким устройством, являющимся частью вычислительной машины.

Также язык программирования должен обладать рядом особенностей: инструкций должно быть ограниченное число, и все их должны знать; инструкции должны выстраиваться определенным образом для получения определенных результатов, и все должны об этом знать; должны существовать правила написания инструкций и все должны их знать; каждая конструкция языка должна однозначно соотноситься то что написано с тем, что требовалось обозначить.[7]

Рассмотрим классификацию видов различных языков программирования:

- Аспектно-ориентированное программирование (АОП) — парадигма программирования, основанная на идее разделения функциональности для улучшения разбиения программы на модули.
- Структурное программирование — методология разработки программного обеспечения, в основе которой лежит представление программы в виде иерархической структуры блоков. Предложена в 70-х годах XX века Э. Дейкстрой, разработана и дополнена Н. Виртом.[\[8\]](#)
- Процедурное программирование — программирование на императивном языке, при котором последовательно выполняемые операторы можно собрать в подпрограммы, то есть более крупные целостные единицы кода, с помощью механизмов самого языка.[\[9\]](#)
- Логическое программирование — парадигма программирования, основанная на автоматическом доказательстве теорем, а также раздел дискретной математики, изучающий принципы логического вывода информации на основе заданных фактов и правил вывода. Логическое программирование основано на теории и аппарате математической логики с использованием математических принципов резолюций.
- Объектно-ориентированное программирование (ООП) — парадигма программирования, в которой основными концепциями являются понятия объектов и классов. В случае языков с прототипированием вместо классов используются объекты-прототипы.
- Функциональное программирование — раздел дискретной математики и парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних (в отличие от функций как подпрограмм в процедурном программировании).[\[10\]](#)
- Мультипарадигмальный язык программирования — как правило, язык программирования, который был разработан специально как инструмент мультипарадигмального программирования, то есть изобразительные возможности которого изначально предполагалось унаследовать от нескольких, чаще всего неродственных языков.[\[11\]](#)
- Эзотерический язык программирования — язык программирования, разработанный для исследования границ возможностей разработки языков программирования, для доказательства потенциально возможной реализации некоей идеи (так называемое «доказательство концепции», англ. proof of concept), в качестве произведения программного искусства[en], или в качестве шутки (компьютерного юмора).

И, в конце концов, рассмотрим вопрос полноты по Тьюрингу как критерия полноты языка программирования.

Как известно, в теории вычислимости исполнитель (множество вычисляющих элементов) называется тьюринг-полным, если на нём можно реализовать любую вычислимую функцию. Другими словами, для каждой вычислимой функции существует вычисляющий её элемент (например, машина Тьюринга) или программа для исполнителя, а все функции, вычисляемые множеством вычислителей, являются вычислимыми функциями (возможно, при некотором кодировании входных и выходных данных).[\[12\]](#)

Название пошло от Алана Тьюринга, который придумал абстрактный вычислитель — машину Тьюринга и дал определение множества функций, вычисляемых посредством машин Тьюринга.

Другими словами язык является *полным по Тьюрингу*, если любая вычислимая функция, которая Вам нужна, может быть записана на этом языке и решена его исполнителем.

Теперь, имея соответствующие теоретическую подготовку, мы в рамках исследования можем сделать вывод относительно принадлежности или, наоборот, не принадлежности языка разметки к классу языков программирования.

Ответим на основные вопросы:

1. Языки разметки - это всё же языки, которые имеют все свойства, присущие языкам.
2. В общем и целом, языки разметки соответствуют конвенционально принятым, формальным определениям, которые дают классы языков программирования. Можно даже конкретизировать и отнести его к декларативным языкам программирования.
3. Языки разметки не удовлетворяют критерию полноты по Тьюрингу. С помощью языков разметки мы можем реализовывать только лишь узкую, ограниченную область действий, которые интерпретируются устройством.
4. В зависимости от контекста и при всех возможных допущениях, мы можем назвать язык разметки языком программирования. Однако следует понимать, что при таких допущениях языками программирования можно с равным успехом назвать с CSS, и SQL, и в том числе язык XML.
5. Вопрос о том, является ли HTML в частности полноценным языком программирования или в целом языком разметки, несмотря на дискутируемость,

решается по простому сопоставлению признаков. Языки разметки не приспособлены для проведения вычислений, они не соответствуют критерию полноты по Тьюрингу и поэтому их нельзя назвать полноценными языками программирования. [\[13\]](#)

## **1.3. История развития языков гипертекстовой разметки**

Идея организовать, представить сложный и разветвлённый текст в интуитивно понятном нелинейном формате занимала лучшие умы нашего человечества на протяжении всей истории, фактически с появления развитой письменности.

Естественно, художественную литературу имеет смысл читать последовательно, переходя от одной главы к другой. Однако когда речь заходит о документах, исследованиях, словарях, монографиях, в общем обо всех сложно организованных публикациях, появляется проблема поиска нужного фрагмента и мгновенного перемещения между несколькими составляющими текста. [\[14\]](#)

Любая современная система навигации построена на гипертекстовой разметке. Мы не можем представить свою жизнь без возможности мгновенно перемещаться по тексту между его частями. На этом построена работа любого браузера и любой программы с графическим интерфейсом.

Ещё 3000 лет назад, в то время, когда человеческая письменность выскабливалась на папирусах и выбивалась на глиняных табличках, древние люди использовали систему перекрёстной навигации: например, на папирусе можно было найти аннотацию или ссылку на другую работу.

Уже в Александрийской библиотеке вопросом подготовки ссылок и аннотаций занимались профессионально. Однако самые первые системы, которые могли бы напомнить современную гипертекстовую разметку, возникли в начале второго тысячелетия. [\[15\]](#)

По сути, это были наборы пометок в печатных изданиях Библии, которые направляли читателя к схожим главам. Занятно, но именно Библия стала первой книгой с идеально организованной и структурированной системой перекрёстных ссылок.

Оставив столь древнюю историю, мы углубимся в более насущные вопросы. Нас интересует система современного гипертекста, которая начинается в июле 1945 года. Доктор В. Буш, научный работник и советник президента Рузвельта, высказал чисто умозрительную идею машины Memex.

В его фантазии это было некое устройство, построенное на логических алгоритмах, в котором на микрофишах хранятся различные научные материалы. При этом разделы любого материала можно коннектировать с любым другим документам.

На практике эта концепция не была реализована, однако В. Буш дал толчок к развитию этой идеи, послужил таким своеобразным пассионарием в области гипертекста.

Спустя 15 лет издатель и ученый С. Райс решил разработать для библиотеки своего магазина удобный гипертекстовый каталог, который содержал бы в себе аннотации и описания различных книг. [\[16\]](#)

Н. Шарф, директор АГК (Ассоциации графических коммуникаций) поддержал его стремление и АГК запустила проект под названием GenCode. Суть проекта заключалась в разработке независимо функционирующих систем, способных представить структуру иерархически организованного массива данных. Однако результаты этой работы появились только спустя десять лет.

Д. Энгелбарт, один из самых ярких пионеров компьютерной инженерии и инженерного программирования, в 1963 году выдвинул теорию интерактивного взаимодействия устройства с человеком. Однако, по стечению обстоятельств, так и не смог воплотить её в жизнь. [\[17\]](#)

Причина была достаточно прозаична - в то время данные можно было хранить только на массивных перфокартах и любая система перекрёстных ссылок сводилось бы к переставлению перфокарт, то есть в конечном итоге к ручному труду.

Параллельно с этим учёным схожей проблемой занялся философ Т. Х. Нельсон. Он мечтал разработать программу, которая была бы в состоянии проверять загруженный документ, сравнивать различные массивы и отменять выбранные изменения по желанию пользователя. [\[18\]](#)

## **1.4. Зарождение гипертекста**

Первой полноценной системой, реализующей, в современном понимании, концепцию гипертекста, стала система HES. Т. Нельсон и Э. Ван Дам выдвинул гениальную идею обратного возврата по ссылке (в современном браузере это кнопка “назад”). В 1967 году эти учёные запатентовали это, а год спустя создали первую гипертекстовую систему GRESS для 16-ти разрядного ПК PDP-8. PDP обладал графическим модулем и был скорее даже на персональном компьютере, а простеньким интеллектуальным терминалом.

Год спустя Ван Дам создал первую коммерческую гипертекстовую систему FRESS (File Retrieval and Editing System) для 16-разрядного компьютера PDP-8 с графическим модулем IMLAC, применявшегося в качестве интеллектуального терминала. Она эксплуатировалась компанией Philips на протяжении 20 лет.[\[19\]](#)

К тому времени появились практические результаты деятельности Дугласа Энгельбарта. Он продемонстрировал на конференции Fall Joint Computer Conference систему NLS (oN Line System), способную обрабатывать сотни тысяч документов (она существует и по сей день в виде коммерческого продукта Augment). В процессе ее создания Энгельбарт предложил идеи систем поддержки коллективной работы в масштабе реального времени, телеконференций, электронной почты, онлайн-подсказок и многооконного интерфейса.[\[20\]](#)

В 1969 г. сотрудник IBM Чарльз Голдфарб возглавил проектирование компьютерной системы обслуживания юридических контор. Под его руководством был создан первый язык разметки документов Generalized Markup Language (GML), в котором была реализована концепция типа документа (формально определенного шаблона, описывающего схему внутреннего построения схожих документов) и вложенных друг в друга структур. GML не зависел ни от марки компьютеров, ни от операционной системы, и IBM удалось перевести 90% своей документации в этот формат.

К 1970 г. завершился проект GenCode Ассоциации GCA. Стало ясно, что для разных задач будут требоваться разные способы описания документов (говоря современным языком, не удалось обойтись ограниченным набором тегов разметки текста), поэтому возникла потребность в расширяемой метасхеме такого описания.[\[21\]](#)

Идею гипермедиа в 1974 г. расширил автор термина "гипертекст" Теодор Нельсон. Он предложил понятие гиперграммы (оно, впрочем, не прижилось), с помощью которого можно было организовывать взаимосвязанную сеть спрайтовых картинок

и даже создавать фильмы с меняющимся по требованию пользователя сюжетом. Эту идею воплотила в 1978 г. в системе Aspen Movie Map группа ученых Массачусетского технологического института во главе с Андреем Липпманом. Система предлагала виртуальное путешествие по Аспену (шт. Колорадо).

К 1978 г. комитет по обработке информации Американского национального института стандартов (ANSI) всерьез заинтересовался языками подготовки гипертекстовых данных. Чарльз Голдфарб стал в этом комитете руководителем нового направления, связанного с формированием стандарта на мощный язык разметки документов, который получил название SGML (Standard General Markup Language).

В том же 1983-м увидела свет очередная прикладная гипертекстовая система Symbolics Document Examiner, созданная под руководством Джанет Уолкер. С помощью этой системы удалось перенести в электронный вид 8 тыс. страниц печатной документации к компьютерам Symbolics. Интересно, что в ней впервые была реализована концепция пользовательских закладок.

В 1986 г. Международная организация по стандартизации ISO одобрила стандарт SGML ISO-8879. Основанный на языке GML, он позволил отказаться от конкретных способов представления информации и сосредоточить усилия на продумывании структуры документов с помощью правил определения собственных тегов форматирования, их атрибутов и синтаксиса использования. Для создания конкретных прикладных наборов тегов было введено понятие "SGML-приложение". Так, популярный сегодня язык разметки гипертекста HTML является SGML-приложением.

SGML оказался очень мощным и универсальным. Он требовал точного описания всех нюансов создаваемого синтаксиса документа и подробных правил формирования тегов. В рамках SGML была изобретена концепция DTD (Document Type Definition) – определение типа документа. Она позволила связать конкретные синтаксические правила разбора с заданными способами организации структуры документов, после чего многие компании приступили к активной разработке программ анализа SGML-текстов.[\[22\]](#)

Следующий год вошел в компьютерную историю первой международной конференцией по гипертексту Hypertext'87 и выходом гипертекстовой системы Hypercard для широкого круга пользователей. Ее автором был Билл Аткинсон, сотрудник Apple Computers, разработчик первого графического редактора MacPaint.

Он начал создавать Hypercard для собственных нужд, чтобы отслеживать появление своих статей в прессе. Затем Аткинсона заинтересовали алгоритмы эффективного поиска и упаковки данных.[\[23\]](#)

Эта система быстро завоевала популярность у пользователей Apple и устанавливалась на каждый компьютер Mac.

Судьба же системы Xanadu Теодора Нельсона оказалась менее удачной. В 1988 г. компания Autodesk купила основанную им фирму Xanadu Operating, но потеряла интерес к проекту и закрыла его через пять лет.[\[24\]](#)

В 1992 г. Том Брюс распространил первый браузер Cello для компьютеров класса IBM PC, после чего множество компаний начали выпускать собственные Интернет-навигаторы. Это привело к массовой нестыковке, потому что каждый производитель браузеров старался дополнить тогда еще крайне ограниченный HTML своим набором тегов.[\[25\]](#)

В феврале 1993 г. в Национальном центре суперкомпьютерных приложений США под руководством Марка Андриссена был создан браузер Mosaic для Unix, знаменитый первым графическим интерфейсом среди себе подобных.

Ведущие информационные корпорации, недовольные неразберихой тегов HTML, сформировали в декабре 1994 г. консорциум WWW (W3C), быстро взявший под свой контроль работу практически над всеми стандартами важнейших технологий Сети. Надо отметить, что формально W3C выпускает только рекомендации и некоторые компании их игнорируют, но в целом рекомендации W3C признаются всем рынком в качестве стандартов.

Тем временем была утверждена версия HTML 2.0 – такую ответственность взяла на себя Международная комиссия по стандартам в Интернете (Internet Engineering Task Force), разрабатывавшая сетевые протоколы. Но HTML 2.0 все же оставался SGML-приложением, ориентированным не на более востребованные в практических Web-проектах задачи форматирования текста, а на построение структуры документа.

В феврале 1996 г. Билл Гейтс провозгласил: "HTML стал нашим типом данных". Это было объявлением войны новому конкуренту Netscape.

Между тем Дэйв Рэггетт, специалист W3C, трудился над третьей версией HTML 3.0 (известной также как HTML+). Эта спецификация из-за конкуренции между

производителями Web-приложений, оказывавших давление на W3C, так и не была закончена, а консорциум сосредоточил усилия не на технологической, а на политической стороне вопроса, стараясь примирить враждовавших разработчиков браузеров. В результате была достигнута договоренность о выпуске в январе 1997-го стандарта HTML 3.2, объединившего наборы тегов, поддерживаемые и Netscape Navigator, и Internet Explorer. На скорейшем выходе HTML 3.2 настаивали также IBM, Novell и Sun.[\[26\]](#)

Не выдержав давления софтверной империи Билла Гейтса, Netscape стала допускать новые ошибки. Все еще веря в собственное влияние на рынок, она не отнеслась с должным вниманием к новой технологии Cascading Style Sheets (CSS, аналог таблиц стилей DSSSL, только для HTML), реализовав ее поддержку в виде не очень удобной технологии JASS (JavaScript Accessible Style Sheets), требовавшей знания программирования. Microsoft же наоборот выделила возможность создания сценариев в HTML-документах, предложив пользователям сразу два скрипт-языка – JScript и VBScript. Эта корпорация также начала финансировать подразделение W3C, ответственное за разработку новых рекомендаций HTML, фактически взяв под свой контроль процесс стандартизации браузеров.[\[27\]](#)

В конце 90-х годов быстро набрал популярность новый язык разметки – XML (о нем будет рассказано в следующий раз). Он представляет собой упрощенный вариант SGML и также позволяет создавать пользовательские XML-приложения.

26 января 2000 г. вышла спецификация нового языка разметки XHTML 1.0, в которую консорциум W3C предложил переносить существующие HTML-материалы. Она была названа "переформулировкой HTML 4.0 в виде приложения XML 1.0" (хотя ранее HTML считался приложением SGML). Разработчики дополнительно получили три DTD-документа для описания типов, определяемых требованиями HTML 4, поэтому теперь можно обрабатывать XHTML-файлы с помощью практически любых XML-анализаторов.[\[28\]](#)

Очень важной оказалась возможность создания в XHTML-приложении собственных тегов (ведь XHTML основывается на XML). Да и компании, производящие интеллектуальные телефоны с возможностью выхода в Интернет, пообещали поддержать XHTML в микробраузерах. А W3C назвал XHTML технологией, предназначенной для мягкого перехода с HTML на XML.[\[29\]](#)

# Глава II. Современные языки гипертекстовой разметки

## 2.1. Язык HTML

HTML - это стандартизированный язык разметки в Интернете. Страницы практически всех сайтов построены на HTML или его более строгом родственнике XHTML. Язык разметки интерпретируется браузером, текст автоматически форматируется и в нужном формате отображается на экране устройства.

Протоколы передачи HTML также стандартизированы - это HTTP или HTTPS. Кроме того, HTML может передаваться в виде обычного текста или с использованием шифрования.

HTML состоит из элементов и сущностей. Элементы HTML - это основной кирпичик, квант веб-страницы. Начало и конец каждого элемента обозначаются тегами. Элементы могут иметь атрибуты, которые указываются в открывающемся теге.

Вторая составная часть языка гораздо интереснее и менее тривиальна - это сущности. **HTML-сущности** — это части текста ("строки"), которые начинаются с символа амперсанда (&) и заканчиваются точкой с запятой (;). Сущности чаще всего используются для представления специальных символов (которые могут быть восприняты как часть HTML-кода) или невидимых символов (таких как неразрывный пробел). Также вы можете использовать их вместо символов, печать которых с обычной клавиатуры труднодоступна.[\[30\]](#)

HTML крайне прост в изучении, его структура понятна и ясна. Он не столь строг, как его родственник XHTML, и гораздо функциональнее языка XML.

## 2.2. SGML

SGML (англ. Standard Generalized Markup Language) — метаязык, на котором можно определять язык разметки для документов.

Метаязык — язык, предназначенный для описания другого языка, называемого объектным языком. Метаязык — язык лингвистики. Лингвистическая лексика, на

основе которой формируются словари. Язык построения теории, слов, фраз в сфере грамматической лингвистики.[\[31\]](#)

По сути, это метаданные (дополнительные данные, которые используются для описания уже имеющихся).

Изначально SGML был разработан для совместного использования машинно-читаемых документов в больших правительственных и аэрокосмических проектах.

Основные части документа SGML:

SGML-декларация — определяет, какие символы и ограничители могут появляться в приложении;

Document Type Definition — определяет синтаксис конструкций разметки. DTD может включать дополнительные определения, такие, как символьные ссылки-мнемоники;

Спецификация семантики, относится к разметке — также даёт ограничения синтаксиса, которые не могут быть выражены внутри DTD;

Содержимое SGML-документа — по крайней мере, должен быть корневой элемент.

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*.

Пример синтаксиса SGML:

```
<QUOTE TYPE="example">
```

```
typically something like <ITALICS>this</ITALICS>
```

```
</QUOTE>
```

SGML стандартизован ISO: “ISO 8879:1986 Information processing—Text and office systems—Standard Generalized Markup Language (SGML)”

HTML и XML произошли от SGML. HTML — это приложение SGML, а XML — это подмножество SGML, разработанное для упрощения процесса машинного разбора документа. Другими приложениями SGML являются SGML Docbook

(документирование) и “Z Format” (типография и документирование).[\[32\]](#)

## 2.3. XML

В чём заключается отличие между HTML и XML? Язык XML предназначен для хранения и передачи данных. HTML же предназначен для отображения данных. 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, по сути, не заменитель HTML, а лишь его дополнение. 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.[\[33\]](#)

XML позволяет представлять сложные, иерархические объекты в текстовом формате. Если язык разметки гипертекста HTML определяет, как элементы будут расположены на Web-сайте, спецификации HTML позволяют лишь форматировать текст, то XML определяет, что эти элементы будут содержать, XML обеспечивает создание собственных дескрипторов, помогающих идентифицировать объекты.

Т.е. XML от HTML отличается тем, что позволяет определять (создавать) собственные элементы, адаптированные под специфические нужды компании или какой-либо отрасли. С помощью XML можно описать данные практически любой предметной области.

Язык XML используется для создания документов, управления, хранения и передачи информации, в том числе и в сети Интернет. XML-документ не зависит от операционной системы и может создаваться при помощи различных языков программирования. Данные из прикладной системы любого типа можно выгрузить в виде XML-документов, используя встроенный в нее язык программирования, даже если он не располагает специальной библиотекой поддержки XML.

Технология XML предоставляет возможность универсального доступа к данным и используется для обмена информацией в бизнесе:

- обмена данными между различными бизнес-приложениями;
- обмена данными с удаленными филиалами предприятия/банка;
- обмена данными между разными организациями;
- обмена данными между БД и Интернет-приложением.

## 2.4. XHTML

XHTML (англ. *extensible **h**ypertext **m**arkup **l**anguage — расширяемый язык гипертекстовой разметки)* — семейство 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.[\[34\]](#)

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

XHTML соответствует спецификации SGML, поскольку XML является её подмножеством. HTML обладает множеством особенностей в процессе обработки и фактически перестал относиться к семейству SGML, что и закреплено в черновике спецификации HTML 5.

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

- HTML — text/html
- 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.

Декларация DOCTYPE может не оказывать никакого влияния на определение парсера для обработки (зависит от используемого браузера).

Различия между HTML и XHTML сводятся к следующему:

Согласно синтаксису XHTML:

- все элементы должны быть закрыты. Теги, которые не имеют закрывающего тега (например, `<img>` или `<br>`), должны иметь на конце `/` (например, `<br />`);
- логические атрибуты записываются в развёрнутой форме. Например, следует писать `<option selected="selected">` или `<td nowrap="nowrap">`;
- имена тегов и атрибутов должны быть записаны строчными буквами (например, `<img alt="" />` вместо `<IMG ALT="" />`);[\[35\]](#)
- XHTML гораздо строже относится к ошибкам в коде; `<` и `&` везде, даже в URL, должны замещаться `&lt;` и `&amp;` 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 (в отличие от HTML, где кодировкой по умолчанию является ISO 8859-1).[\[36\]](#)

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`.

Существует три типа документов XHTML: `strict`, `transitional` и `frameset`. Наиболее употребительной и универсальной из версий XHTML является переходная, поскольку она позволяет использовать `iframe` (включение содержимого одной веб-страницы в другую) и атрибут `target` у ссылок (для указания того, например, 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`. DTD определение типа строгой версии XHTML не содержит многих тегов и атрибутов, описанных в DTD `transitional` и признанных устаревшими.

Самыми распространёнными ошибками в XHTML-разметке являются:

- незакрытые элементы (XHTML, в отличие от HTML, требует закрытия всех элементов, в том числе не имеющих закрывающего тега, как, например, `<br />`);
- отсутствие альтернативных текстов для изображений (достигающихся применением атрибута `alt`, который помогает сделать документы доступнее для устройств, которые не в состоянии отображать изображения, или предназначенных для слабовидящих людей);

- присутствие текста непосредственно в теге `<body>` документа (должен быть объявлен блочный элемент, внутрь которого следует помещать содержимое);
- вложение блочных элементов внутрь внутрискриптовых (`inline`) (например, блочные элементы `<div>` или `<p>` не могут быть вложены внутрь инлайновых элементов `<a>`, `<span>`, `<em>` и так далее);
- пренебрежение заключением значений атрибутов в кавычки (`<a href=http://example.com/>` вместо `<a href="http://example.com/">`);
- неправильное вложение элементов (конструкции вида `<strong><em></strong></em>`);
- неправильное использование ссылок-мнемоник (например, `&` вместо `&amp;`);
- написание тегов и/или атрибутов прописными буквами вместо строчных (`<DIV STYLE="...">` вместо `<div style="...">`);
- задание в теге `<!DOCTYPE ...>` относительного пути к DTD-файлу ("`DTD/xhtml11.dtd`" вместо "`http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd`").

Этот список не полный и содержит самые частодопускаемые ошибки при составлении XHTML-документов.

## 2.5. Облегчённые языки разметки

Языки, предназначенные для простого и быстрого написания текста в простом текстовом редакторе, называются *облегчёнными* (en:Lightweight markup language).

Особенности таких языков:

- Минимум функций.
- Небольшой набор поддерживаемых тегов.
- Легки в освоении.
- Исходный текст на таком языке читается с такой же лёгкостью, как и готовый документ.[\[37\]](#)

Применяются они там, где человеку приходится подготавливать текст в обычном текстовом редакторе (блоги, форумы, вики), либо там, где важно, чтобы пользователь с обычным текстовым редактором также мог прочитать текст. Вот несколько широко распространённых облегчённых языков разметки:

- BBCode
- Markdown
- reStructuredText

- Textile
- Различные системы автодокументирования (например, Javadoc).

BBCode (аббр. от англ. bulletin board code) — язык разметки, используемый для форматирования сообщений на многих электронных досках объявлений (BBS) и форумах. Для форматирования текста используются теги, подобные тегам HTML, но, в отличие от них, заключённые в квадратные скобки («[» и «]»). Перед отображением страницы движок форума производит разбор текста и преобразование его в HTML-код.

BBCode создан в 1998 году для движка досок объявлений Ultimate Bulletin Board (UBB), написанном на языке Perl. В 2000 году поддержка BBCode была добавлена в движок для форумов PhpBB, написанный на языке PHP.[\[38\]](#)

BBCode был придуман с целью предоставить более простой, безопасный и ограниченный по сравнению с HTML способ форматирования сообщений. Первоначально многие BBS позволяли пользователям использовать HTML-разметку, что могло приводить к побочным эффектам, таким как искажение разметки на странице, запуск JavaScript-сценариев (иногда в целях вандализма). Некоторые реализации BBCode, использующие упрощённые методы трансляции в HTML, также могут испытывать проблемы с безопасностью.

Хотя основные теги BBCode мало отличаются в большинстве интернет-форумов, существует большое количество вариаций. Некоторые реализации чувствительны к регистру символов внутри тегов, другие позволяют подсвечивать текст HTML, PHP, SQL и других языков разметки и программирования при помощи специальных тегов.

**eStructuredText** (сокращение: **ReST**, расширение файла: **.rst**) — облегчённый язык разметки. Хорошо применим для создания простых веб-страниц и других документов, а также в качестве вспомогательного языка при написании комментариев в программном коде. reStructuredText можно считать потомком легковесных систем разметки StructuredText и Setext. Парсер reStructuredText поддерживает достаточное количество конечных форматов, в том числе PDF, HTML, ODT, LaTeX и формат презентаций S5.

Система документирования Sphinx, написанная для нужд документирования языка программирования Python на сайте docs.python.org и построенная на базе Docutils, применяется уже в нескольких десятках проектов. ReST является одним из языков разметки, используемых на github, в частности, для README-файлов.

## **Заключение**

Языки разметки - важнейший класс компьютерных языков. Организация текста, установление перекрёстных ссылок, графическое оформление - реализация всех этих функций была бы слишком сложной, если не сказать невозможной, без удобных и простых в использовании языков гипертекста.

Цель работы достигнута, поставленные задачи последовательно выполнены и раскрыты. Был проведён комплексный анализ языков гипертекстовой разметки как предмета изучения информатики, были выявлены парадигмы их развития развития и актуальное состояние на сегодняшний день.

Проведён подробный анализ понятийного аппарата, предложен перечень общих признаков, объединяющих все языки гипертекстовой разметки.

Рассмотрен вопрос об отнесении языков разметки к языкам программирования. Проанализировано развитие языков разметки в исторической динамике.

Во второй главе обозначены основные характеристики наиболее популярных и важных языков разметки, рассмотрены такие языки как XML, XHTML, SGML, а также облегчённые языки разметки.

Можно сделать вывод, что языки гипертекстовой разметки - это достаточно ригидная и самодостаточная система, которая не нуждается в постоянном обновлении и совершенствовании. Разработанных стандартов хватает для обеспечения жизнеспособности большей части веб-страниц, едва ли в ближайшем времени будет придумана кардинально новая система разметки.

## **Библиографический список**

1. Алернов И. П. Алгоритмы упаковки данных в метапрограммировании. - 2 изд. - М.: Астра, 2013. - 304 с.
2. Гавриленко С. Ю., Шитьков П. А. Использование языка xml для промежуточного представления программы // Вестник НТУ ХПИ. 2008. №24.
3. Грищук Д. А. Объектно-ориентированное программирование. - М.: Научный Мир, 2012. - 298 с.

4. Дуглас Энгельбрат и его вклад в развитие программной инженерии // Всё обо Всём URL: <https://vsobs.com/ru/post/y332/> (дата обращения: 10.05.2019).
  5. Иванцова Е. Л., Иванцов Д. И. История HTML и перспективы развития. - М.: Флинта, 2015. - 189 с.
  6. История гипертекста: краткая и понятная // habr URL: <https://habr.com/ru/post/2341897/> (дата обращения: 10.05.2019).
  7. Лавров Святослав Сергеевич Функциональное программирование // КИО. 2002. №3-4. URL: <https://cyberleninka.ru/article/n/funktsionalnoe-programmirovanie-1> (дата обращения: 16.05.2019).
  8. Мальцев Д. И. Толковый словарь программиста. - М.: Проспект, 2016. - 471 с.
  9. Орлов С. А. Программная инженерия: учебник для вузов. - СПб.: Питер, 2015. - 501 с.
  10. Пахунов Антон Валерьевич Языки программирования: классификация, особенности, критерии выбора // Современная наука. 2015. №4. URL: <https://cyberleninka.ru/article/n/yazyki-programmirovaniya-klassifikatsiya-osobennosti-kriterii-vybora> (дата обращения: 15.05.2019).
  11. Перроуин Д. HTML: самоучитель. - СПб.: Аврора, 2018. - 377 с.
  12. Петровенко С. Б. Парадигма развития вычислительной техники в двадцатом веке. - СПб.: Питер, 2018. - 488 с.
  13. Р. С. Алиев, А. А. Копий Семантические особенности семейства языков разметки // International Journal of Open Information Technologies. 2017. №7. URL: <https://cyberleninka.ru/article/n/semanticheskie-osobennosti-semeystva-yazykov-razmetki> (дата обращения: 16.05.2019).
  14. Саянова А. П. Разработка программ и инженерное программирование: учебник. - 4 изд. - СПб.: Альфа, 2016. - 466 с.
  15. Тлюстен Валерий Шахамболетович Вариант машины Тьюринга с «Подсматривающей» головкой // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. 2014. №4 (147). URL: <https://cyberleninka.ru/article/n/variant-mashiny-tyuringa-s-podsmatrivayushey-golovkoj> (дата обращения: 15.05.2019).
  16. Что такое XML // MSiter.ru URL: <https://msiter.ru/tutorials/uchebnik-xml-dlya-nachinayushchih/chto-takoe-xml> (дата обращения: 11.05.2019)
  17. Штейнгауэр Д. Метапрограмн Т. Фреймворки языков разметки. - 1 изд. - М.: Сфера, 2018. - 204 с.
- 
1. Мальцев Д. И. Толковый словарь программиста. - М.: Проспект, 2016. - С 164. [↑](#)

2. Иванцова Е. Л., Иванцов Д. И. История HTML и перспективы развития. - М.: Флинта, 2015. - С. 23. [↑](#)
3. Иванцова Е. Л., Иванцов Д. И. История HTML и перспективы развития. - М.: Флинта, 2015. - С. 53. [↑](#)
4. Штейнгауэр Д. Метапрограммирование. - СПб.: Питер, 2017. - С 306. [↑](#)
5. HTML - язык программирования? // Metanit URL:  
<https://metanit.com/html2/tutorial/1.3.php> (дата обращения: 10.05.2019). [↑](#)
6. Общие признаки языков программирования // ProgrammingLang URL:  
[https://proglang.com/basis/info/язык\\_программирования](https://proglang.com/basis/info/язык_программирования) (дата обращения: 10.05.2019). [↑](#)
7. Общие признаки языков программирования // ProgrammingLang URL:  
[https://proglang.com/basis/info/язык\\_программирования](https://proglang.com/basis/info/язык_программирования) (дата обращения: 10.05.2019). [↑](#)
8. Орлов С. А. Программная инженерия: учебник для вузов. - СПб.: Питер, 2015. - С 97. [↑](#)
9. Грищук Д. А. Объектно-ориентированное программирование. - М.: Научный Мир, 2012. - С. 100-103 [↑](#)
10. Лавров Святослав Сергеевич Функциональное программирование // КИО. 2002. №3-4. URL: <https://cyberleninka.ru/article/n/funktsionalnoe-programmirovanie-1> (дата обращения: 16.05.2019). [↑](#)
11. Пахунов Антон Валерьевич Языки программирования: классификация, особенности, критерии выбора // Современная наука. 2015. №4. URL: <https://cyberleninka.ru/article/n/yazyki-programmirovaniya-klassifikatsiya-osobennosti-kriterii-vybora> (дата обращения: 15.05.2019). [↑](#)

12. Тлюстен Валерий Шахамболетович Вариант машины Тьюринга с «Подсматривающей» головкой // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. 2014. №4 (147). URL: <https://cyberleninka.ru/article/n/variant-mashiny-tyuringa-s-podsmatrivayuschey-golovkoj> (дата обращения: 15.05.2019). [↑](#)
13. Является ли язык разметки языком программирования? // habr URL: <https://habr.com/ru/post/231897/> (дата обращения: 10.05.2019). [↑](#)
14. Орлов С. А. Программная инженерия: учебник для вузов. - СПб.: Питер, 2015. - С 337. [↑](#)
15. История гипертекста: краткая и понятная // habr URL: <https://habr.com/ru/post/2341897/> (дата обращения: 10.05.2019). [↑](#)
16. Петровенко С. Б. Парадигма развития вычислительной техники в двадцатом веке. - СПб.: Питер, 2018. - С 97. [↑](#)
17. История гипертекста: краткая и понятная // habr URL: <https://habr.com/ru/post/2341897/> (дата обращения: 10.05.2019). [↑](#)
18. Петровенко С. Б. Парадигма развития вычислительной техники в двадцатом веке. - СПб.: Питер, 2018. - С 97. [↑](#)
19. Петровенко С. Б. Парадигма развития вычислительной техники в двадцатом веке. - СПб.: Питер, 2018. - С 456. [↑](#)
20. Дуглас Энгельбрат и его вклад в развитие программной инженерии // Всё обо Всём URL: <https://vsobs.com/ru/post/y332/> (дата обращения: 10.05.2019). [↑](#)
21. Лещенко Л. Д., Гасинович А. П. Метапрограммирование: истоки, корни и перспективы дальнейшего развития // Сборник докладов XII научной конференции КубГУ. - Краснодар: Издание КубГУ, 2016. - С. 114. [↑](#)

22. Лещенко Л. Д., Гасинович А. П. Метапрограммирование: истоки, корни и перспективы дальнейшего развития // Сборник докладов XII научной конференции КубГУ. - Краснодар: Издание КубГУ, 2016. - С. 118. [↑](#)
23. Алернов И. П. Алгоритмы упаковки данных в метапрограммировании. - 2 изд. - М.: Астра, 2013. - С. 45. [↑](#)
24. Экклстон Т. Фреймворки языков разметки. - 1 изд. - М.: Сфера, 2018. - С. 104. [↑](#)
25. Лещенко Л. Д., Гасинович А. П. Метапрограммирование: истоки, корни и перспективы дальнейшего развития // Сборник докладов XII научной конференции КубГУ. - Краснодар: Издание КубГУ, 2016. - С. 18. [↑](#)
26. Лещенко Л. Д., Гасинович А. П. Метапрограммирование: истоки, корни и перспективы дальнейшего развития // Сборник докладов XII научной конференции КубГУ. - Краснодар: Издание КубГУ, 2016. - С. 37. [↑](#)
27. Саянова А. П. Разработка программ и инженерное программирование: учебник. - 4 изд. - СПб.: Альфа, 2016. - С. 55. [↑](#)
28. История гипертекста: краткая и понятная // habr URL: <https://habr.com/ru/post/2341897/> (дата обращения: 10.05.2019). [↑](#)
29. Перроуин Д. HTML: самоучитель. - СПб.: Аврора, 2018. - С. 76. [↑](#)
30. Саянова А. П. Разработка программ и инженерное программирование: учебник. - 4 изд. - СПб.: Альфа, 2016. - С. 234. [↑](#)
31. Гавриленко С. Ю., Шитьков П. А. Использование языка xml для промежуточного представления программы // Вестник НТУ ХПИ. 2008. №24. URL: <https://cyberleninka.ru/article/n/ispolzovanie-yazyka-xml-dlya-promezhutochnogo-predstavleniya-programmy> (дата обращения: 16.05.2019). [↑](#)

32. Р. С. Алиев, А. А. Копий Семантические особенности семейства языков разметки // International Journal of Open Information Technologies. 2017. №7. URL: <https://cyberleninka.ru/article/n/semanticheskie-osobennosti-semeystva-yazykov-razmetki> (дата обращения: 16.05.2019). [↑](#)
33. Что такое XML // MSiter.ru URL: <https://msiter.ru/tutorials/uchebnik-xml-dlya-nachinayushchih/что-такое-xml> (дата обращения: 11.05.2019). [↑](#)
34. Перроуин Д. HTML: самоучитель. - СПб.: Аврора, 2018. - С. 76. [↑](#)
35. Лавров Святослав Сергеевич Функциональное программирование // КИО. 2002. №3-4. URL: <https://cyberleninka.ru/article/n/funktsionalnoe-programmirovaniye-1> (дата обращения: 16.05.2019). [↑](#)
36. Что такое XML // MSiter.ru URL: <https://msiter.ru/tutorials/uchebnik-xml-dlya-nachinayushchih/что-такое-xml> (дата обращения: 11.05.2019). [↑](#)
37. Иванцова Е. Л., Иванцов Д. И. История HTML и перспективы развития. - М.: Флинта, 2015. - С. 23. [↑](#)
38. Пахунов Антон Валерьевич Языки программирования: классификация, особенности, критерии выбора // Современная наука. 2015. №4. URL: <https://cyberleninka.ru/article/n/yazyki-programmirovaniya-klassifikatsiya-osobennosti-kriterii-vybora> (дата обращения: 15.05.2019). [↑](#)