

Содержание:

ВВЕДЕНИЕ

К сегодняшнему дню Глобальная Сеть, помимо того, что объединила людей со всего мира, еще и утвердилась как неотъемлемая часть технологий практически каждого персонального компьютера. В самом деле, каждый пользователь, приобретающий и устанавливающий, к примеру, такой привычный пакет Microsoft Office, автоматически причисляется к обладателям Internet-технологий. Самое интересное, что данный факт не зависит от наличия подключения персонального компьютера к Сети Интернет.

Иначе говоря, средства и платформы, которые изначально планировались и внедрялись как средства работы с Сетью, теперь применяются и в иных целях, которые не имеют к Сети никакого отношения. Кроме того, к сегодняшнему дню программы для работы с Сетью заняли свое почетное, даже несколько обособленное место на рынке программного обеспечения.

Результатом подобных изменений явился тот факт, что большая часть пользователей стала иметь отношение к Сети, что также не зависит от желания самих пользователей.

Кроме того, на сегодняшний день стало популярным средством выражения собственного мнения и жизненной позиции размещение в Глобальной Сети персональных страниц. В то же время, большая часть современных государственных и коммерческих организаций стала приверженцем использования Сети не только для оптимизации работы сотрудников и организации в целом, но и для рекламы и продвижения организации.

Нельзя не отметить, что на рынке труда также произошли существенные изменения благодаря использованию технология Сети – пользователи получили возможность составлять резюме и вакансии в виде отдельных веб-страниц, доступных для просмотра всей Сети.

Подобный прогресс не был бы достигнут без такого базового явления, как разметки гипертекста. Так, для того чтобы проанализировать принципы работы и построения языков разметки, необходимо выполнить следующую цель – изучение языков

гипертекстовой разметки.

Объектом исследования будут выступать сами языки гипертекстовой разметки.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Рассмотреть понятие «язык разметки гипертекста»
2. Изучить основные характеристики гипертекстовой разметки
3. Проанализировать язык разметки гипертекста HTML
4. Охарактеризовать язык разметки гипертекста Dynamic HTML
5. Дать анализ языка разметки гипертекста XHTML
6. Привести выводы по проделанной работе.

В структуру работы включены такие элементы, как Введение, Заключение и Список использованных источников. Кроме того, работа содержит две главы, первая из которых состоит из двух параграфов, а вторая – из трех.

Основной теоретической базой исследования выступают работы современных исследователей проблемы, в их числе – Гончаров А., Макарова Н., Морозевич А., Зеневич А. и Шафран Э.

Язык разметки гипертекста. Понятие. Основные характеристики

Понятие «язык разметки гипертекста»

На сегодняшний день место основы современных технологий, применяемых в сети Internet заняли языки разметки гипертекста и написания сценариев, апплетов и скриптов публикаций WWW (World Wide Web). В современном мире уже разработано и продолжает разрабатываться большое количество разнообразных программных языков, либо специально ориентированных на использование в Web, либо включающих функции, предназначенные для Internet.

Язык разметки гипертекста в компьютерной терминологии представляет собой набор символов или последовательностей, которые вставляются в текст для передачи информации о его выводе или строении. Язык гипертекстовой разметки можно отнести к классу компьютерных языков. Текстовый документ, который был

написан при помощи языка разметки, содержит не только сам текст (как последовательность слов и знаков препинания), но и дополнительную информацию о различных его участках – например, указание на заголовки, выделения, списки и т. д.

В более затруднительных случаях язык разметки предоставляет возможность интегрировать в документ интерактивные элементы и содержание других документов[1].

Термин «разметка» (как результат одноименного процесса, англ. markup) происходит от английского словосочетания «marking up» («разметка (как процесс)», букв. «помечивание, размечивание»), которые было заимствовано из традиционной издательской практики проставления специальных условных пометок на полях и в тексте рукописи или корректуры перед передачей её в печать. Таким образом «разметчики» (markup men) указывали гарнитуру, стиль и размер шрифта для каждой части текста. На сегодняшний день разметкой текста занимаются редакторы, корректоры, графические дизайнеры — и, конечно же, сами авторы[2].

Необходимо обратить внимание на то, что согласно Тьюрингу язык разметки гипертекста не является полным и его нельзя назвать языком программирования. Несмотря на это, фактически, исходя из соображений логики, язык разметки гипертекста не может не быть языком программирования. Это объясняется тем, что любые элементы языка программирования представляют собой неотъемлемую в программировании часть всего языка[3].

В 1986 г. Международная организация по стандартизации (ISO) приняла первый стандарт ISO-8879, озаглавленный «Standard Generalized Markup Language (SGML)». Этот стандарт был посвящен описанию языка SGML – обобщенного метаязыка, позволяющего строить системы логической и структурной разметки любых разновидностей текстов. Слово структурная в данном контексте означает, что управляющие коды, вносимые в текст при такой разметке, не несут никакой информации о внешнем виде документа, а лишь указывают границы и соподчинение его составных частей, т.е. задают его логическую структуру.

Данный стандарт был разработан для стандартизации обмена данными между разнородными программами и платформами. Язык SGML являлся не только мощным, но и очень сложным средством. Кроме того, SGML-документ было трудно интерпретировать без определений языка разметки, который хранится в

определении типа документа DTD (document type definition). DTD необходимо было посылать вместе с SGML-документом или включать в документ, для того чтобы можно было распознать теги, созданные пользователем. Под тегами понимались специальные управляющие коды, записываемые в виде текста, ограниченного угловыми скобками[4].

Необходимо отметить, что основной проблемой того времени оставалась проблема корректного отображения ранее созданных страниц текста ввиду достаточно большого разнообразия программ создания текстовых документов и различных аппаратно-программных платформ.

В связи с этим разработчики SGML стремились создать программно- и аппаратно-независимые языки, с тем чтобы размеченный текст мог без труда интерпретироваться любой программой, работающей с любым устройством вывода.

Фактически SGML представлял собой не готовую систему разметки текста, а лишь удобный метаязык, позволяющий строить такие системы для конкретных обстоятельств. Стандарт SGML определял лишь синтаксис записи элементов разметки текстов – тегов и их атрибутов, а также содержал правила определения новых тегов и указания структурных отношений между ними. Для практической же разметки документов использовалось приложение SGML: набор определенных в соответствии со стандартом тегов, являющийся, по сути, формальным описанием структуры документа[5].

Основные характеристики языков гипертекстовой разметки

В данном параграфе будут рассмотрены основные характеристики языков гипертекстовой разметки.

Различают логическую и визуальную разметки. В первом случае речь идет только о том, какую роль играет данный участок документа в его общей структуре (например, «данная строка является заголовком»). Во втором — определяется, как именно будет отображаться этот элемент (например, «данную строку следует отображать жирным шрифтом»).

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 (например, один и тот же файл может по-разному отображаться на экране компьютера, мобильного

телефона и на печати, поскольку свойства этих устройств вывода существенно различаются). Однако 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[6].

Языки разметки используются везде, где требуется получение форматированного текста на основании одного лишь текста: в типографии (SGML, TeX, PostScript, RTF), пользовательских интерфейсах компьютеров (Microsoft Word, OpenOffice, troff), Всемирной Сети (HTML, XHTML, XML, WML, VML, PGML, SVG, XBRL).

Языки, предназначенные для простого и быстрого написания текста в простом текстовом редакторе, называются облегченными (Lightweight markup language). Особенности таких языков:

1. Минимум функций.
2. Небольшой набор поддерживаемых тегов.
3. Легки в освоении.
4. Исходный текст на таком языке читается с такой же лёгкостью, как и готовый документ.

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

1. BBCode
2. Markdown
3. reStructuredText
4. Textile
5. Различные системы автодокументирования (например, Javadoc)[7].

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.

Пока развивалась идея использовать языки разметки с текстовыми документами, это увеличило использование языков разметки в других областях, предложили использовать их для представления различных типов информации, включая плей-листы, векторную графику, веб-сервисы, интерфейсы пользователя. Большинство этих приложений основано на XML, так как это хорошо структурированный и расширяемый язык.

Использование языка XHTML также показывает, что его можно комбинировать с различными языками разметки одного профиля, например XHTML+SMIL или XHTML+MathML+SVG[8].

Основные языки гипертекстовой разметки: HTML, Dynamic HTML, XHTML

2.1 Язык разметки гипертекста HTML

Большая часть документов в системе World Wide Web хранится в формате HTML. HTML – это язык гипертекстовой разметки, используемый для кодирования документов. Язык HTML представляет собой набор команд, в соответствии с которыми браузер отображает содержимое документа, команды HTML не отображаются. В языке HTML реализован механизм гипертекстовых ссылок, который обеспечивает связь одного документа с другими. Эти документы могут находиться на том же сервере, что и страница, с которой на них делается ссылка, а могут быть размещены на другом сервере[9].

Команды в тексте HTML-документа называются тегами.

HTML-тег может содержать список атрибутов. Текст тега заключается в угловые скобки (< и >). В простейшем случае тег представляет собой имя, заключенное в угловые скобки, например <TITLE> или . Для более сложных тегов характерно наличие атрибутов, которые могут иметь конкретные значения, определяющие функции тега. Например:

```
<P ALIGN=«JUSTIFY»>
```

В данном случае <P> - это тег, ALIGN - один из его атрибутов, а «JUSTIFY» - значение этого атрибута. (Данный тег определяет начало абзаца, атрибут ALIGN

определяет способ выравнивания текста в абзаце, значение «JUSTIFY» задает выравнивание по ширине.)

Регистр символов в именах тегов не учитывается. Теги <P> и <p> будут интерпретироваться одинаково. Однако в некоторых случаях важно использовать определенный регистр символов в значениях атрибутов.

Большинство HTML-тегов имеют парный закрывающий тег. Текст, к которому применяется форматирование, заключается между открывающим и закрывающим тегами. Закрывающий тег имеет имя, идентичное открывающему тегу, но перед именем закрывающего тега ставится косая черта. Например, чтобы отобразить текст курсивом, его нужно заключить в теги <I> и </I>:

<I> Этот текст будет отображен курсивом</I> [\[10\]](#)

Концепция создания интерактивных Web-страниц не является новой – компании Microsoft и Netscape изначально определили простую объектную модель, которая была представлена как способ создания интерактивных страниц. При более внимательном рассмотрении объектная модель оказалась более эффективной только для базовой проверки форм. Динамические документы были по большей части мифом до появления Microsoft Internet Explorer 4.0, поскольку страница могла быть изменена только при ее создании. Интерактивные документы моделировались путем полной перезагрузки страницы или внедрения объектов на страницу. Многие апплеты Java, элементы управления ActiveX и даже анимированные изображения в формате GIF были разработаны для выполнения операций, аналогичных тексту, что позволяло преодолеть статическую природу HTML. Однако использование объектов вместо стилей и элементов управления содержанием представляет собой не самое лучшее решение.

После изучения множества схем для данных объектов и анимированных изображений GIF, разработчики Internet Explorer пришли к выводу, что для манипулирования документами необходим более мощный прикладной интерфейс программирования (application programming interface, API). С помощью объектной модели, которая обеспечивает полный доступ к документу, Web-мастера могут использовать встроенные возможности размещения объектов в HTML и каскадные таблицы стилей (Cascading Style Sheets, CSS). Такая модель существенно улучшает производительность, сокращая необходимость загрузки больших анимированных изображений GIF и апплетов. Информация становится доступной мгновенно, поскольку она больше не спрятана внутри изображений или объектов [\[11\]](#).

Гипертекстовый документ – это текстовый файл, имеющий специальные метки, называемые тегами, которые впоследствии опознаются браузером и используются им для отображения содержимого файла на экране компьютера.

HTML-документ состоит из двух частей: собственно текста, т. е. данных, составляющих содержимое документа, и тегов – специальных конструкций языка HTML, используемых для разметки документа и управляющих его отображением. Теги языка HTML определяют, в каком виде будет представлен текст, какие его компоненты будут исполнять роль гипертекстовых ссылок, какие графические или мультимедийные объекты должны быть включены в документ.

Графическая и звуковая информация, включаемая в HTML-документ, хранится в отдельных файлах. Программы просмотра HTML-документов (браузеры) интерпретируют флаги разметки и располагают текст и графику на экране соответствующим образом. Для файлов, содержащих HTML-документы приняты расширения .htm или .html[12].

Самым главным из тегов HTML является одноименный тег <html>. Он всегда открывает документ, так же, как тег </html> должен непременно стоять в последней его строке. Эти теги обозначают, что находящиеся между ними строки представляют единый гипертекстовый документ. Без этих тегов браузер или другая программа просмотра не в состоянии идентифицировать формат документа и правильно его интерпретировать.

HTML-документ состоит из двух частей: заголовок (head) и тела (body), расположенных в следующем порядке:

```
<html>
```

```
<head> Заголовок документа </head>
```

```
<body> Тело документа </body>
```

```
</html>
```

Чаще всего в заголовок документа включают парный тег <title>... </title>, определяющий название документа. Многие программы просмотра используют его как заголовок окна, в котором выводят документ. Программы, индексирующие документы в сети Интернет, используют название для идентификации страницы. Хорошее название должно быть достаточно длинным для того, чтобы можно было корректно указать соответствующую страницу, и в то же время оно должно

помещаться в заголовке окна. Название документа вписывается между открывающим и закрывающим тегами.

Тело документа является обязательным элементом, так как в нем располагается весь материал документа. Тело документа размещается между тегами `<body>` и `</body>`. Все, что размещено между этими тегами, интерпретируется браузером в соответствии с правилами языка HTML позволяющими корректно отображать страницу на экране монитора.

Текст в HTML разделяется на абзацы при помощи тега `<p>`. Он размещается в начале каждого абзаца, и программа просмотра, встречая его, отделяет абзацы друг от друга пустой строкой. Использование закрывающего тега `</p>` необязательно.

Если требуется «разорвать» текст, перенеся его остаток на новую строку, при этом, не выделяя нового абзаца, используется тег разрыва строки `
`. Он заставляет программу просмотра выводить стоящие после него символы с новой строки. В отличие от тега абзаца, тег `
` не добавляет пустую строку. У этого тега нет парного закрывающего тега[\[13\]](#).

Язык HTML поддерживает логическое и физическое форматирование содержимого документа. Логическое форматирование указывает на назначение данного фрагмента текста, а физическое форматирование задает его внешний вид.

При использовании логического форматирования текста браузером выделяются различные части текста в соответствии со структурой документа. Чтобы отобразить название, используется один из тегов заголовка. Заголовки в типичном документе разделяются по уровням. Язык HTML позволяет задать шесть уровней заголовков: h1 (заголовок первого уровня), h2, h3, h4, h5 и h6. Заголовок первого уровня имеет обычно больший размер и насыщенность по сравнению с заголовком второго уровня. Пример использования тегов заголовков:

```
<h1>I. Название главы</h1>
```

```
<h2>I.I. Название раздела</h2>
```

Теги физического форматирования непосредственно задают вид текста на экране браузера, например пара `` выделяет текст полужирным начертанием, `<u></u>` задает подчеркивание текста, `` управляет шрифтом текста.

Тег вставляет изображение в документ, как если бы оно было просто одним большим символом. Пример применения тега:

```
<img src = «picture.gif»>
```

Для создания гипертекстовой ссылки используется пара тегов <a>... .

Фрагмент текста, изображение или любой другой объект, расположенный между этими тегами, отображается в окне браузера как гипертекстовая ссылка.

Активация такого объекта приводит к загрузке в окно браузера нового документа или к отображению другой части текущей Web-страницы. Гипертекстовая ссылка формируется с помощью выражения:

```
<a href = «document.html»>ссылка на документ</a>
```

Href здесь является обязательным атрибутом, значение которого и есть URL-адрес запрашиваемого ресурса. Кавычки в задании значения атрибута href не обязательны. Если задается ссылка на документ на другом сервере, то вид гиперссылки такой:

```
<a href = «http://www.school.donetsk.ua/11.jpg»>Фотография 11-А</a>
```

С помощью различных тегов можно рисовать таблицы, форматировать текст, вставлять в документ изображения, видео-, звуковые файлы и прочее[14].

Язык разметки гипертекста Dynamic HTML

DHTML (динамический HTML) – это набор средств, которые позволяют создавать более интерактивные Web-страницы без увеличения загрузки сервера. Другими словами, определенные действия посетителя ведут к изменениям внешнего вида и содержания страницы без обращения к серверу[15].

DHTML построен на объектной модели документа (Document Object Model, DOM), которая расширяет традиционный статический HTML-документ. DOM обеспечивает динамический доступ к содержимому документа, его структуре и стилям. В DOM каждый элемент Web-страницы является объектом, который можно изменять. DOM не определяет новых тэгов и атрибутов, а просто обеспечивает возможность программного управления всеми тэгами, атрибутами и каскадными листами стилей (CSS).

Dynamic HTML (или, более сокращенно, DHTML) – способ, который дает возможность создать интерактивный веб-сайт. При этом применяется объектная модель

документа DOM, статичный язык разметки HTML, а также скриптовый язык, который выполняется браузером клиента (Javascript).

При помощи технологии DHTML изменяется информация, отображаемая на страницах сайтов, в соответствии с конкретными действиями, которые производятся пользователем[16].

Чаще всего Dynamic HTML используется с целью создания в браузере разного рода приложений. К примеру, этот способ может быть необходим для придания интерактивности форм или упрощения навигации. Нередко DHTML используется для динамического перемещения тех или иных элементов по экрану. Кроме того, Dynamic HTML служит в качестве инструмента, необходимого для создания видеоигр.

Таким образом, при помощи DHTML автор имеет возможность работать с контекстом, а также с оформлением собственной страницы. При помощи комбинации скриптов, стилей и HTML есть возможность получать столь же интересные элементы страниц, которые до того были доступны лишь с Flash-элементов или Java апплетов.

Всевозможные подсвечиваемые элементы текста, выпадающие меню, изменяющиеся кнопки реализуются непосредственно в HTML странице, чаще всего, при помощи JavaScript.

Как утверждают специалисты, технология DHTML дала множество возможностей для разработки Web-узлов. Помимо всего прочего, использование Dynamic HTML значительно снижает загрузку сервера и всей сети, ведь число запросов, на данный сервер посылаемых, значительно сокращается[17].

В самом общем виде можно сказать, что Dynamic HTML — это набор технологий, работающих на стороне клиента и призванных преодолеть статичность традиционных Веб-страниц. Точнее говоря, это технологии, которые обеспечивают

- динамическое формирование Веб-страницы в процессе ее загрузки, и
- динамическое изменение Веб-страницы в ответ на действия пользователя.

Для достижения перечисленных целей используются следующие методы:

- динамическое изменение атрибутов и стилей элементов, составляющих HTML-документ;

- динамическое извлечение данных из внешних источников и включение их в Веб-страницу;
- использование динамически загружаемых шрифтов;
- поддержка визуальных и мультимедийных эффектов при отображении страниц;
- механизмы сохранения информации на компьютере-клиенте между сессиями работы[18].

Все эти явно разнородные методы покоятся, в действительности, на двух китах. Первым из них является интеграция HTML-документа, каскадных стилей и сценариев клиента, основанная на объектной модели документа. Вторым — использование компонентного программирования, позволяющего нам встраивать однажды разработанные компоненты во вновь создаваемые документы.

На WEB-страницах можно обеспечить реакцию на определенные действия посетителя или изменения состояния документа или окна, которые вызывают определенные события.

Посетитель генерирует события при передвижении мыши, нажатии кнопок мыши и клавиатуры. Изменения состояния документа генерируют события при загрузке документа, изображений или объектов, при появлении ошибки на странице или переходе фокуса от одного элемента к другому.

Модель событий DHTML связана с определенной иерархией HTML-контейнеров и основана на всплывании событий и действии по умолчанию.

Всплывание событий заключается в том, что на событие может быть получена реакция не только от элемента-источника события, но также и от всех его родительских элементов вплоть до тела документа и самого документа. Событие может быть обработано на любом уровне.

Установление связи между определенным событием и сценарием называется связыванием событий. События можно связать с помощью специальных атрибутов любого элемента или с помощью тэга SCRIPT.

Связывание событий с атрибутами удобно, но требует расширения языка HTML каждый раз при изобретении нового события. А так как HTML развивается медленно, данный подход используется только для небольшого набора встроенных событий. Как атрибуты любого элемента в DHTML представлены события мыши и клавиатуры[19].

Основных преимуществ DHTML можно выделить три:

1. Повышение конкурентоспособности узла. Узел, содержащий динамически изменяемые цвета, анимированную графику, систему выпадающих меню и другие стандартные (для современных ОС, но не для Веба) элементы пользовательского интерфейса, явно привлекательнее для пользователя, чем обычные статические страницы. Разумеется, при этом нужно иметь чувство меры и соблюдать общие принципы дизайна.
2. Легкость сопровождения. DHTML позволяет нам вынести типовые элементы формирования страниц нашего Веб-узла в отдельные таблицы стилей и сценарии, а затем включать их одинаковым образом во все страницы. Это, безусловно, упрощает сопровождение и минимизирует количество возникающих опечаток и других ошибок.
3. Уменьшение трафика. Одна из основных особенностей DHTML состоит в том, что он переносит основную тяжесть по формированию Веб-страниц с сервера на клиент. Платой за это является загрузка с сервера сценариев и программных компонентов, которые при правильном проектировании незначительны по объему. В результате динамические страницы загружаются и отображаются намного быстрее статических, что особенно важно для отечественных каналов связи, не страдающих чрезмерно высоким качеством [\[20\]](#).

Объективности ради следует остановиться и на тех недостатках DHTML, о которых постоянно говорят сторонники «Веб-страниц, одинаково отображаемых всеми обозревателями».

Dynamic HTML не оправдывает себя прежде всего потому, что он не стандартизован, и под каждый из популярных Веб-обозревателей нужно писать свою версию HTML-документа. Действительно, DHTML, как и всякая другая новая технология, не оформился пока в четкий стандарт. Тем не менее, усилия по созданию динамических страниц окупаются хотя бы по перечисленным выше причинам. Да и отсутствие стандарта никогда не мешало продвижению перспективных технологий (электроприборы, к примеру, существуют гораздо дольше, чем Интернет, но никого не смущает то, что часть из них рассчитана на 220В, а часть на 127В).

Если присмотреться внимательнее, то противодействие распространению DHTML вызвано совершенно другой причиной. Дело в том, что до сих пор создание Веб-страниц не требовало специальных знаний, кроме знакомства с HTML и умения

минимально манипулировать графикой. С появлением DHTML ситуация изменилась в корне: разработка динамической страницы требует программирования на сценарном языке, сопряженного с пониманием работы обозревателя на уровне генерации и обработки событий, владением основами пользовательского интерфейса и т. п. Иными словами, Веб-дизайн востребовал специалистов, что у многих вызывает неприятие. На самом деле, демократическая сущность WWW от этого не пострадала. По-прежнему любой желающий может создать, скажем, свою домашнюю страничку на уровне своих знаний, и DHTML этому не помеха. Другое дело, что профессиональные разработки должны соответствовать по уровню лучшим из наличествующих технологий, но на то они и называются профессиональными[21].

Краткая история DHTML

Dynamic HTML родился в процессе конкурентной борьбы между Веб-обозревателями корпораций Netscape и Microsoft. Он появился в обозревателе Netscape Navigator 4, где были введены два новых тега LAYER и IFRAME, которые позволяли создавать на Веб-страницы слои с динамически изменяемыми свойствами (положением на странице, стилями отображения и т. п.). В ответ на это Microsoft в Internet Explorer 4 предложили намного более мощную поддержку динамических страниц, которая не использовала нестандартных тегов и опиралась на единый подход к HTML-документу как к дереву объектов с динамически изменяемыми атрибутами и свойствами стилей.

Основное различие между двумя подходами состояло в парадигме самого понятия «динамическая страница». Netscape под этим понимал динамическое формирование Веб-страницы в процессе ее загрузки и последующую косметическую правку в ответ на действия пользователя без существенных изменений содержания страницы. Microsoft с самого начала занял позицию динамического формирования и изменения любых элементов страницы в любой момент ее существования[22].

Понятно, что такое различие моделей DHTML в основных обозревателях никак не соответствует базовой концепции WWW как платформи-независимой информационной сети. Поэтому W3C был вынужден предпринять усилия по стандартизации Dynamic HTML и начал с разработки спецификации объектной модели документа (DOM), в основу которой была положена модель Microsoft,

предложенная W3C в качестве Веб-стандарта. В результате в октябре 1998 г. был принята рекомендация W3C по первому уровню DOM (DOM 1), а в ноябре 2000 г. приняты рекомендации по второму уровню DOM (DOM 2), существенно расширяющему DOM 1.

Тем временем Microsoft выпустил обозреватели Internet Explorer 5 и 5.5, которые существенно расширили возможности DHTML, причем во многом в направлениях, не предусмотренных стандартами W3C. Параллельно шла работа над обозревателем Netscape 6, который был призван вернуть часть рынка, завоеванного Microsoft (летом 2000 г. 86% пользователей WWW использовали Internet Explorer и только 14% - Netscape).

С этой целью была создана независимая группа Mozilla.org, которая опубликовала в Сети исходные тексты обозревателя Netscape и собрала группу независимых разработчиков для приведения его в соответствие стандарту. Результатом их работы стало ядро обозревателя Gecko, почти в полной мере соответствующее стандартам HTML 4.0, CSS1 и DOM 1 (поддержка стандартов CSS2 и DOM 2 реализована частично) и одновременно сохранившее некоторую совместимость с моделью Netscape 4. Gecko является основой обозревателя Netscape 6, который после двух с половиной лет обещаний вышел, наконец, в свет в конце 2000 г. Выпущенная версия Netscape 6 пока очень сырая, и в ближайшее время следует ожидать появления ее новых сборок, которые должны будут заменить Netscape 4 [\[23\]](#).

Таким образом, сегодня на рынке присутствуют:

1. обозреватели с ядром MSHTML корпорации Microsoft. Прежде всего это Microsoft Internet Explorer 5.x, динамическая модель которого близка к DOM 1 и DOM 2, но не совпадает с ними. Это обозреватель, который установлен на абсолютном большинстве клиентских компьютеров, подключенных к Вебу, и который предлагает самую богатую на сегодня палитру технологий DHTML.
2. Netscape 4.x (наиболее стабильная версия 4.7), поддерживающий исходную динамическую модель Netscape, которую следует признать на сегодня морально устаревшей.
3. Обозреватели с ядром Gecko (Mozilla/5.0, Netscape 6). Эти обозреватели приведены в соответствие со стандартами W3C, но еще не приобрели товарного вида [\[24\]](#).

2.3 Язык разметки гипертекста XHTML

XHTML – это основанный на XML язык разметки гипертекста, максимально приближенный к текущим стандартам HTML. XHTML отличается от HTML строгостью написания кода. Если HTML позволял писать практически любые конструкции и браузер их корректно распознавал, то теперь, с появлением XHTML, это стало невозможным. Последний требует строгого соблюдения всех правил, предъявляемых W3C. Строгие требования к оформлению XHTML-кода позволяют избежать многих ошибок ещё на стадии написания и отладки[25].

XHTML – это новый язык, который пришёл на смену старому HTML. Новых версий HTML больше не будет. В итоге все браузеры, как предполагается, перейдут на XHTML (очевидно, что при этом сохранится совместимость со старым HTML, но не более того). Так же хотим привести цитату из русского перевода спецификации XHTML от W3C: «Семейство XHTML создавалось с учетом общей совместимости пользовательских агентов. С помощью нового механизма профилирования пользовательских агентов и документов серверы, прокси и пользовательские агенты смогут преобразовывать содержимое наилучшим образом. В конечном счете, станет возможной разработка соответствующего XHTML содержимого, пригодного для любого соответствующего XHTML пользовательского агента».

XHTML совместим с HTML при соблюдении некоторых правил, описание которых можно найти в стандарте. Это означает, что даже самые старые браузеры, которые понимают HTML, будут работать и с XHTML[26].

Для проверки правильности написания XHTML-кода существуют программы-валидаторы. Кроме того, пользовательские агенты, которые поддерживают XHTML, сами будут вам сообщать об ошибках в синтаксисе, если таковые возникнут.

XHTML представляет собой семейство имеющихся на данный момент и могущих появиться в будущем типов документов и модулей, являющихся копиями, подмножествами или расширениями языка HTML 4 [HTML]. Семейство типов документов XHTML базируется на XML и предназначено для работы с пользовательскими агентами на базе. Более подробную информацию об этом семействе и его эволюции можно найти в разделе «Направления развития».

XHTML 1.0 представляет первый тип документов семейства XHTML. В ней три типа документов HTML 4 переформулируются в терминах XML 1.0. Она предназначена для использования в качестве языка содержимого, как соответствующего XML, так

и, если соблюдены некоторые простые требования, работающего в конформных пользовательских агентах HTML 4. Разработчики, переносящие свои документы в XHTML 1.0, получают следующие преимущества:

1. Документы XHTML соответствуют XML. Как таковые они без труда просматриваются, редактируются и проверяются на корректность стандартными средствами XML.
2. Документы XHTML могут работать лучше, чем они работали в существующих пользовательских агентах, соответствующих HTML 4, а также в новых пользовательских агентах, соответствующих XHTML 1.0.
3. Документы XHTML могут использовать прикладные программы (например, скрипты и апплеты), базирующиеся на HTML Document Object Model или XML Document Object Model [DOM][27].

По мере расширения семейства XHTML документы, соответствующие XHTML 1.0, будут с большей вероятностью совместимы с различными средами XHTML.

Семейство XHTML является следующим шагом в эволюции Интернет. Переходя сегодня на XHTML, разработчики содержимого (контента) могут вступить в мир XML со всеми его преимуществами, сохраняя при этом совместимость содержимого с более старыми и более новыми версиями.

Если говорить обобщенно, то XML является языком разметки для описания данных, в котором все обязано быть размечено синтаксически правильно.

HTML является языком разметки, который используется для отображения данных в браузере.

XHTML является языком, который комбинирует сильные стороны XML и HTML. Он создан для отображения в браузере правильно оформленных данных[28].

Существует несколько требований, которые разработчик обязан выполнять:

- в начале документа должен указываться один из возможных DTD (Document Type Definition)
- в теле XHTML-документа должны обязательно присутствовать следующие тэги: html, head, title и body;
- обязательно наличие атрибута xmlns в элементе html;
- имена тегов и атрибутов должны записываться в нижнем регистре;
- все значения атрибутов должны заключаться в «кавычки»;

- все тэги должны закрываться; если у элемента нет закрывающего тэга, следует добавлять в его конец слеш (
 или
 — пробел для совместимости со старыми браузерами);
- необходимо соблюдать корректную вложенность тэгов (<i>текст</i> — неверно; следует писать <i>текст</i>);
- запрещается использовать минимизированные атрибуты (nowrap следует заменить на nowrap=«nowrap»); полный список таких атрибутов: checked, compact, declare, defer, disabled, ismap, noresize, noshade, nowrap, multiple, readonly, selected.
- на следующие элементы налагаются ограничения по включению в них других элементов:
 - a не может содержать другие элементы a;
 - form не может содержать другие элементы form;
 - label не может содержать другие элементы label;
 - pre не может содержать img, object, big, small, sub или sup;
 - button не может содержать элементы input, select, textarea, label, button, form, fieldset, iframe или isindex;
 - специальные символы в истинном значении должны заменяться на свои эквиваленты:
 - & на &
 - < на <

Кроме того, существует ряд необязательных рекомендаций, которые разработчик не обязан выполнять в версии XHTML 1.0, но в последующих версиях этого языка возможно рекомендации перерастут в требования:

- декларация XML-документа в самом начале кода перед DTD (<?xml version=«1.0» encoding=«windows-1251»?>);
- следование элемента title сразу после открытия тега head;
- использование атрибута id вместо name (name считается устаревшим атрибутом);
- наличие атрибута type в элементах подключаемых файлов (таблиц стилей и скриптов);
- отказ от использования атрибута target[\[29\]](#).

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была достигнута поставленная цель, а именно – изучены языки гипертекстовой разметки.

Для достижения данной цели были выполнены следующие задачи:

1. Рассмотрено понятие «язык разметки гипертекста»
2. Изучены основные характеристики гипертекстовой разметки
3. Проанализирован язык разметки гипертекста HTML
4. Охарактеризован язык разметки гипертекста Dynamic HTML
5. Дан анализ языка разметки гипертекста XHTML

На момент проведения исследования проблему нельзя назвать полностью раскрытой и изученной. Это, в первую очередь, связано с непрерывным совершенствованием компьютерных технологий и с ежедневными открытиями в мире сетей, информатики и коммуникаций. Следовательно, языки разметки гипертекста также продолжают развиваться и модернизироваться соответственно потребностям времени и новых технологий. Языки, рассмотренные в данной работе, на сегодняшний день еще не изжили себя и являют собой достаточно интересный объект исследования.

Кроме выполнения поставленных задач и достижения цели, также была значительно расширена база теоретических знаний, которая была пополнена благодаря изучению выбранной темы. Данные знания будут полезными при дальнейшем изучении дисциплины и смежных с ней.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гончаров А. HTML в примерах. С.-Пб.: Питер, 2003.
2. Гончаров А. Самоучитель HTML. С.-Пб: Питер, 2000.
3. Дрога А. А., Жукова П. Н., Копонев Д. Н., Лукьянов Д. Б., Прокопенко А. Н. Информатика и математика. - Минск, 2008.
4. Информатика. Базовый курс. Учебник для Вузов/под ред. С.В. Симоновича. СПб.: Питер, 2000.
5. Информатика. Компьютерная техника. Компьютерные технологии. / Пособие под ред. О.И.Пушкаря. Киев Издательский центр «Академия». 2001.
6. Информатика: Учебник/под ред. Н.В. Макаровой. - М.: Финансы и статистика, 2000. - 768 с.

7. Ковтанюк Ю.С., Соловьян С.В. Самоучитель работы на персональном компьютере. К.: Юниор, 2001.
8. Комягин А.О. Современный самоучитель работы на персональном компьютере. М.: ТРИУМФ, 2003.
9. Матросов А. HTML 4.0. М.: Дело, 2005.
10. Морозевич А.Н., Зеневич А.М. Информатика. Минск, 2008.
11. Основы современных компьютерных технологий: Учебное пособие / Под. ред. Хомоненко. С.-Пб.: КОРОНА, 2002.
12. Острейковский В.А. Информатика. М.: Высшая школа, 2001.
13. Симонович С.В., Евсеев Г.А., Алексеев А.Г. Специальная информатика. Учебное пособие. М.: АСТпресс, 2001.
14. Хеслоп П. HTML самого начала. С.-Пб: Санкт-Петербург, 2005.
15. Шафран Э. Создание Web страниц. С.-Пб: Питер, 2004.

1. Гончаров А. Самоучитель HTML. С.-Пб: Питер, 2000, с. 123 [↑](#)
2. Гончаров А. Самоучитель HTML. С.-Пб: Питер, 2000, с. 136 [↑](#)
3. Гончаров А. HTML в примерах. С.-Пб.: Питер, 2003, с. 42 [↑](#)
4. Ковтанюк Ю.С., Соловьян С.В. Самоучитель работы на персональном компьютере. К.: Юниор, 2001, с. 236 [↑](#)
5. Ковтанюк Ю.С., Соловьян С.В. Самоучитель работы на персональном компьютере. К.: Юниор, 2001, с. 257 [↑](#)
6. Ковтанюк Ю.С., Соловьян С.В. Самоучитель работы на персональном компьютере. К.: Юниор, 2001, с. 258 [↑](#)
7. Матросов А. HTML 4.0. М.: Дело, 2005, с. 36 [↑](#)
8. Матросов А. HTML 4.0. М.: Дело, 2005, с. 43 [↑](#)
9. Шафран Э. Создание Web страниц. С.-Пб: Питер, 2004, с. 304 [↑](#)

10. Шафран Э. Создание Web страниц. С.-Пб: Питер, 2004, с. 324 [↑](#)
11. Информатика: Учебник/под ред. Н.В. Макаровой. - М.: Финансы и статистика, 2000. - с. 596 [↑](#)
12. Информатика: Учебник/под ред. Н.В. Макаровой. - М.: Финансы и статистика, 2000. - с. 603 [↑](#)
13. Информатика: Учебник/под ред. Н.В. Макаровой. - М.: Финансы и статистика, 2000. - с. 631 [↑](#)
14. Информатика: Учебник/под ред. Н.В. Макаровой. - М.: Финансы и статистика, 2000. - с. 642 [↑](#)
15. Морозевич А.Н., Зеневич А.М. Информатика. Минск, 2008, с. 124 [↑](#)
16. Шафран Э. Создание Web страниц. С.-Пб: Питер, 2004, с. 46 [↑](#)
17. Шафран Э. Создание Web страниц. С.-Пб: Питер, 2004, с. 94 [↑](#)
18. Симонович С.В., Евсеев Г.А., Алексеев А.Г. Специальная информатика. Учебное пособие. М.: АСТпресс, 2001, с. 146 [↑](#)
19. Симонович С.В., Евсеев Г.А., Алексеев А.Г. Специальная информатика. Учебное пособие. М.: АСТпресс, 2001, с. 162 [↑](#)
20. Хеслоп П. HTML самого начала. С.-Пб: Санкт-Петербург, 2005, с. 321 [↑](#)
21. Хеслоп П. HTML самого начала. С.-Пб: Санкт-Петербург, 2005, с. 336 [↑](#)
22. Гончаров А. Самоучитель HTML. С.-Пб: Питер, 2000, с. 30 [↑](#)
23. Гончаров А. Самоучитель HTML. С.-Пб: Питер, 2000, с. 36 [↑](#)

24. Гончаров А. Самоучитель HTML. С.-Пб: Питер, 2000, с. 42 [↑](#)
25. Дрога А. А., Жукова П. Н., Копонев Д. Н., Лукьянов Д. Б., Прокопенко А. Н. Информатика и математика. - Минск, 2008, с. 406 [↑](#)
26. Дрога А. А., Жукова П. Н., Копонев Д. Н., Лукьянов Д. Б., Прокопенко А. Н. Информатика и математика. - Минск, 2008, с. 413 [↑](#)
27. Шафран Э. Создание Web страниц. С.-Пб: Питер, 2004, с. 178 [↑](#)
28. Шафран Э. Создание Web страниц. С.-Пб: Питер, 2004, с. 183 [↑](#)
29. Шафран Э. Создание Web страниц. С.-Пб: Питер, 2004, с. 185 [↑](#)