

Содержание:

Введение

В данной работе рассматриваются языки гипертекстовой разметки.

На сегодняшний день Интернет и одна из его служб, Всемирная Паутина, являются неотъемлемой частью нашей жизни. Всего лишь пятнадцать лет назад для того, чтобы, например, узнать телефон ближайшей закусочной с доставкой пиццы, большинство из нас стало бы звонить в справочную службу по телефону, либо смотреть в справочнике “желтые страницы”.

Теперь мы просто открываем нашу любимую поисковую систему и вводим название того, что хотим найти. Миллионы приложений (то есть программ в сети Интернет) делают нашу жизнь проще, быстрее, интересней, безопасней. Мы привыкли пользоваться Интернетом всегда и всюду, с помощью мобильного телефона, планшетного компьютера или, например, телевизора. Большинство информационных ресурсов в Интернете представляют собой особый вид текстового документа, в котором используется специальная разметка HTML.

Современный веб-сайт представляет собой приложение, которое может, например, дать ответ на интересующий вопрос, поможет заказать услугу или товар, найти новых друзей. Большую популярность приобретают образовательные сайты, например Courseara, пользователи которого могут обучаться онлайн у лучших преподавателей по самым интересным образовательным курсам ведущих мировых Университетов. Содержимое веб-страниц может создаваться с помощью различных технологий, но оно всегда формируется с помощью языков гипертекстовой разметки, который и интерпретируется браузером.

Понятие о гипертекстовой разметке и гипертекстовом документе должен иметь любой современный грамотный человек.

Цель написания курсовой работы — рассмотреть основные технологии создания веб-страниц, основанные на использовании языков гипертекстовой разметки.

Объектом исследования являются языки гипертекстовой разметки. Предметом исследования являются особенности применения различных технологий создания

веб-страниц с использованием языков гипертекстовой разметки.

В работе использованы книги известных авторов, официальные сайты консорциума Всемирной Паутины, издания ведущих мировых университетов. Поэтому литературу, на основании которой написана работа, можно считать надежной.

В курсовой работе решаются следующие задачи:

- Определить основные понятия, связанные с языками гипертекстовой разметки
- Описать синтаксис и область применения языка HTML
- Описать область применения и синтаксис составляющих DHTML
- Сравнить возможности применения различных технологий создания веб-страниц.

Глава 1. Понятие и назначения языков гипертекстовой разметки

1.1 Веб-страница как гипертекстовый инструмент

В начале определим основные понятия, используемые в работе.

Гипертекст представляет собой специальным образом размеченный текст, который может содержать гиперссылки.

HTTP – протокол передачи гипертекста. Именно по этому протоколу передаются веб-страницы, которые затем обрабатываются браузером (специальной программой для просмотра документов, передаваемых по протоколу HTTP и еще некоторым другим протоколам).

Интернет – всемирная компьютерная сеть, объединяющая пользователей, использующих различные технологии передачи данных, по всему миру. Часто можно услышать, как люди отождествляют понятия Интернет и WWW. На самом деле, это различные понятия. WWW является одной из служб Интернет, скажем, как электронная почта, телеконференции и др [1].

Появившись как система обмена научной информацией, система WWW сейчас, в основном, используется для других целей. Современный веб-сайт имеет, как правило, большой объем мультимедийного наполнения (видео, музыка) [2]. Сайты

являются интерактивными и, зачастую, содержат различные веб-приложения, т.е. программы, сходные по способу взаимодействия с традиционными программами, требующими установки на персональный компьютер.

Отметим, что вся совокупность гипертекстовых документов образует семантическую сеть, то есть документы одной предметной области оказываются связанными друг с другом[4]. Иными словами, гипертекстовая система WWW является информационно-поисковой системой, а основным инструментом поиска информации являются гиперссылки.

Любой веб-сайт представляет собой совокупность связанных веб-страниц, объединенных одной тематикой и расположенных на одном сервере. Каждый веб-сайт должен иметь уникальное имя (оно называется доменным именем). Точнее, каждому веб-сайту может принадлежать одно или несколько доменных имен[1].

Доменное имя состоит из отдельных частей называемых доменами. Каждый домен относится к определенному уровню и указывает на географическую принадлежность сайта или на принадлежность к определенному тематическому направлению[2]. Такой способ присваивания сайту определенного адреса и система адресации в Интернете с помощью доменных имен получила название DNS, служба доменных имен, рисунок 1.

Доменная система имён

DNS - Domain Name Service.

Доменная система имеет иерархическую структуру:

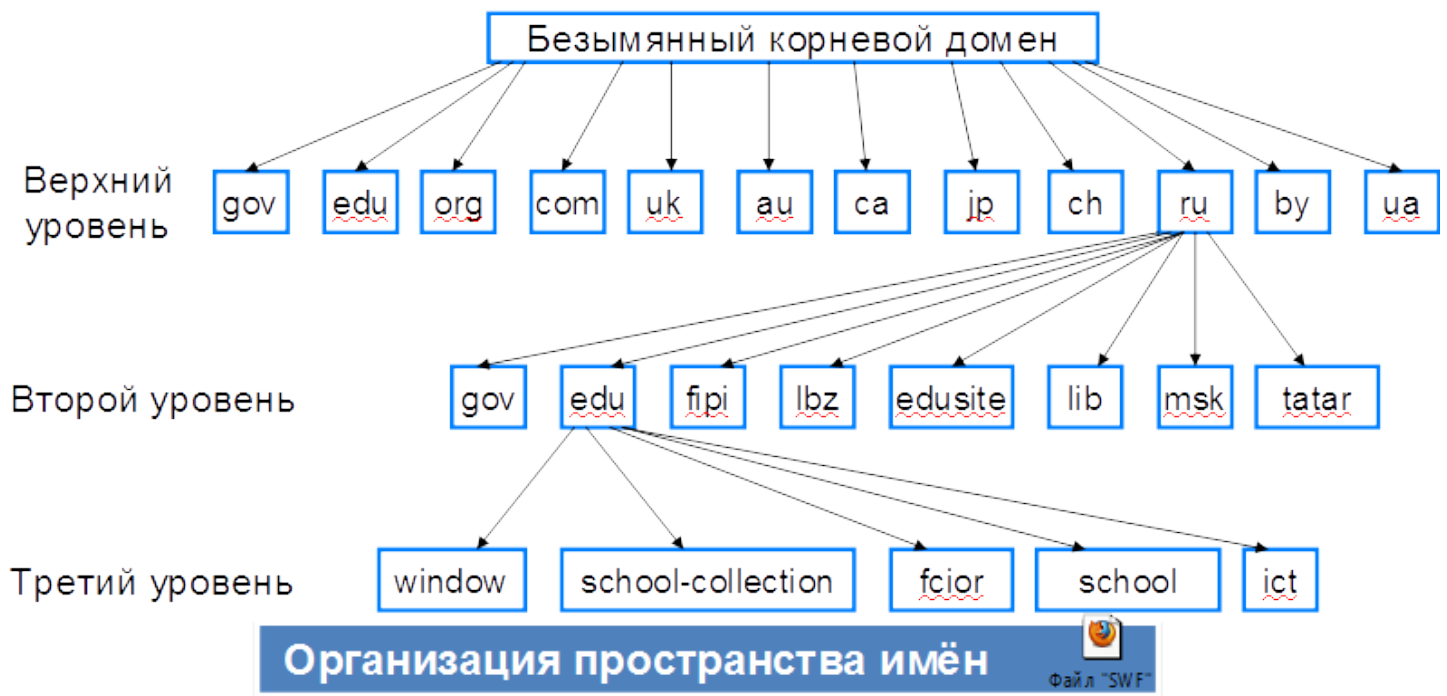


Рисунок 1 — Иерархическая структура службы доменных имен

Примером доменного имени может быть имя `www.foxford.ru`.

Любой ресурс в Интернете, то есть сайт или программа должны иметь уникальное имя[4]. Это имя называется URL, или единообразный указатель ресурса.

Например, рассмотрим URL:

`http://www.ivanov-petrov.com/company/about.html`

Данный URL указывает, что файл `about.html` находится в директории `company` на сайте с доменным именем `www.ivanov-petrov.com`. Первая часть URL указывает, что передача данных происходит по протоколу `http`.

Таким образом, структуру URL можно записать следующим образом:

протокол://доменное имя/путь к файлу на сервере/имя файла.расширение

С помощью URL задается не только адрес веб-страниц, как в указанном примере, но и адреса изображений, текстовых документов и вообще любых файлов, размещенных в сети Интернет (не только во Всемирной паутине)[1].

Уникальный адрес в сети Интернет, похож на полное имя файла на компьютере (оно тоже уникально). Под полным именем файла понимается путь к файлу начиная от логического диска, с учетом всех вложенных директорий до имени файла с расширением.

Основное отличие адреса веб-страниц от адреса других типов файлов заключается в том, что часто этот адрес определяется программой на сервере, созданной с помощью одного из серверных языков (часто такую программу называют сценарием). Веб-страницы часто формируются динамически, то есть в момент обращения к ним. Страницы собираются под управлением сценариев на сервере на основе шаблонов и данных из баз данных (БД). Таким образом, например, веб-страница товара в интернет-магазине формируется на основе ответа программы на запрос пользователя, шаблона товара и данных о конкретном товаре из БД[2].

На сервере также существует система файлов и папок, в которой могут быть размещены файлы.

Отметим, что обмен данными в сети Интернет осуществляется с указанием адресов компьютеров отправителя и получателя. Эти адреса называются ip-адреса. IP-адрес представляет собой 32-битный двоичный адрес. Для удобства записи IP-адрес обычно записывают в десятичном представлении, с помощью 4 десятичных чисел, разделенных точкой[4].

Каждое из таких чисел должно находиться в диапазоне от 0 до 255, например:

12.0.1.255

Назначение доменной системы имен состоит в замене неудобного для человека цифрового адреса на осмысленное буквенное обозначение.

1.2. Язык HTML

Язык гипертекстовой разметки HTML (HyperText Markup Language) был предложен группой ученых из Европейского Центра ядерных Исследований (CERN) под руководством Тима Бернерсом-Ли в 1989 году в качестве одного из компонентов технологии разработки распределенной гипертекстовой системы World Wide Web [3]. Система WWW (World Wide Web, всемирная паутина) разрабатывалась как система обмена научными данными.

Именно гиперссылки были основным новшеством, введенным Бернерсом-Ли и его командой, обеспечивающим быструю навигацию по документу и возможность перехода к другим документам.

Язык HTML, как и XHTML представляет собой набор элементов. Для задания большинства элементов необходимо использовать открывающий и закрывающий теги, между которыми может находиться содержимое[1].

1.3 Понятие XHTML

Язык гипертекстовой разметки XHTML (extensible hypertext markup language) является расширением возможностей языка HTML версии 4.0. В данный момент разработка стандарта приостановлена. Несмотря на то, что HTML является самым популярным языком гипертекстовой разметки в мире, ему присущи ряд недостатков, которые было решено ликвидировать с помощью XHTML и расширив его функциональность[2]. Например, добавлена поддержка языков MathML, SMIL, SVG . Последняя версия XHTML 2.0. Разработка данной версии прекратилась в 2010 году. Использование данной версии не рекомендовано консорциумом Всемирной Паутины (W3C) [3].

Предыдущие версии XHTML 1.0 и XHTML 1.1 имеют рекомендации W3C.

Основное отличие между HTML и XHTML состоит в правилах обработки документов[4].

Стандарт XML предполагает более строгие правила обработки документов. Браузеры HTML работают с любыми входными данными, корректными и некорректными, и пытаются их интерпретировать. Схема обработки ошибок очень усложняет разработку программ, особенно если учесть, что предполагается, что реакция всех браузеров на некорректные данные должна быть одинаковой [5]. Кроме того, это означает, что огромное число документов HTML содержат ошибки,

но нормально отображаются браузерами, и авторы не подозревают о том, что код HTML некорректен. Вот почему необычайно сложно создать новый пользовательский агент, пока документы, выдаваемые за HTML, часто очень плохи.

Таким образом, документ, написанный на языке XHTML, в случае если он содержит ошибку, не будет показан. Встретив ошибку, браузеры должны выдавать сообщения об ошибке и не отображать его содержимое, по крайней мере такие рекомендации дает консорциум W3C[5].

В случае обработки HTML документа браузер должен отобразить содержимое даже в случае наличия ошибок в синтаксисе.

Основные различия синтаксиса языка XHTML в сравнении с HTML :

- все элементы должны быть закрыты. Теги, отвечающие пустым элементам, должны иметь на конце знак «/» (например, элемент означающий конце строки, `
` или разделительная черта `<hr/>`);
- логические атрибуты всегда должны быть записаны в развёрнутой форме. Например, следует писать `<option selected="selected">` или `<td nowrap="nowrap">`;
- имена тегов и атрибутов должны быть записаны только строчными буквами, в то время как стандарт HTML допускает применения как строчных, так и заглавных букв[4].

1.4 Понятие DHTML

Под DHTML (Dynamic HTML, динамический HTML) понимается методика создания веб-страниц с помощью совокупности технологий статического языка веб-разметки (HTML или XHTML), модели документов DOM, каскадных таблиц стилей CSS, скриптовых языков программирования (например, JavaScript). [6]

Основная идея технологии состоит в изменении содержимого, отображающегося в браузере без перезагрузки веб-страницы.

DHTML позволяет сделать веб-страницы интерактивными[4]. Он 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 таблицами стилей (CSS).

Каскадные таблицы стилей позволяют задавать внешний вид элемента DOM с помощью селекторов, свойств объектов и значений свойств[2].

Каждой гиперссылке, заголовку или текстовому параграфу можно присвоить имя, атрибуты стиля или цвета текста и указать это имя в сценарии, имеющемся на данной странице. Сценарий может быть написан на любом существующем скриптовом языке, чаще всего используется JavaScript. Элементы страницы впоследствии могут изменяться в результате определенного события, например при наведении курсора мыши, при щелчке, нажатии клавиши на клавиатуре либо после истечения заданного промежутка времени. Изменяться может не только стиль и цвет текста, но и весь объект, текст или рисунок.

Отметим, что в настоящее время DHTML является самым популярным методом создания веб-страниц. Связано это, прежде всего, с тем, что большинство сайтов являются интерактивными. Пользователи уже давно привычны к тому, что веб-сайты являются полноценными веб-приложениями и предлагают удобный интерфейс, огромное количество мультимедийного материала, включая анимацию и видео.

Использование HTML для задания элементов, находящихся на веб-странице, CSS для изменения внешнего вида объектов, взаимодействие пользователя с сайтом с помощью JavaScript является стандартом современного сайтостроения.

Огромное преимущество CSS при задании внешнего вида сайта состоит в разделении описания содержимого от его внешнего вида. Большинство правил оформления при использовании DHTML содержатся в CSS, а описание всех элементов, которые изначально находятся на веб-странице, задается в файле с расширением html.

Глава 2. Описание языка HTML

2.1 Базовые конструкции

Язык HTML не является языком программирования, как многие ошибочно думают. Если основой многих языков программирования являются операторы, которые говорят исполнителю, что нужно делать, то основу языка HTML составляют элементы.

Элементы в документе обозначаются с помощью специальных тегов, двух угловых скобок, внутри которых находится название элемента. Например, чтобы выделить некоторый текст полужирным шрифтом, например слово “Информатика”, на языке HTML следует использовать элемент `b` (от английского `bold` — жирный, полужирный). Для этого в документе можно заключить выделяемый текст между двумя тегами, то есть вот так: `Информатика`. Первый из этих тегов, `` называется открывающим, потому что он открывает структуру, а второй `` называется закрывающим, потому что он закрывает эту часть текста. Между этими двумя тегами может помещаться текст, либо теги других элементов, вместе с их содержимым.

Например, отдельный параграф на языке HTML обозначается с помощью элемента `p`. Если внутри этого параграфа мы захотим выделить часть текста жирным, мы можем использовать элемент `b`. Например так:

```
<p>Язык <b>HTML</b> очень удобен для применения и легок в освоении</p>
```

В браузере данный пример будет отображаться примерно так:

Язык HTML очень удобен для применения и легок в освоении.

То есть в предложении слово “HTML” будет иметь жирное начертание[7].

Большинство элементов задаются парой тегов и содержимым, однако есть некоторые элементы, которые не могут иметь содержимого. Например, элемент `br` задается тегом `
` без использования закрывающего тега. Данный элемент создает в тексте разрыв строки, то есть перенос части текста на другую строку. Другим распространенным примером является элемент `img`, который позволяет вставлять на веб-страницу изображение. Например так

```
.
```

Здесь мы видим один тег `` и внутри него особую конструкцию имя атрибута со значением атрибута. В данном примере атрибуты являются обязательными, поэтому мы приводим этот пример вместе с атрибутами.

2.2 Атрибуты

С помощью атрибутов сообщают некоторую информацию о элементе. Часто это информация о внешнем виде, либо о поведении элемента при взаимодействии с пользователем. Атрибуты бывают как обязательными, так и необязательными. В приведенном выше примере используется атрибут `src` — указывающий имя файла в той же директории в которой находится файл `html` документа (в более общем виде здесь должен находиться путь к файлу) со значением атрибута `goodgirl.jpg` и атрибут `alt` со значением `good girl`. Значение атрибута `alt` отображается вместо изображения, если по каким-то причинам изображение не отображается (например, не удалось загрузить). На заре развития веб-технологий с помощью атрибутов полностью описывался внешний вид веб-страниц. В настоящее время атрибуты используются только там, где это необходимо, либо целесообразно. Оформление же веб-страницы формируется в основном с помощью технологии `CSS` и использования для этого отдельного файла с таблицами стилей[7]. Чаще всего в современных веб-страницах используются необязательные атрибуты `id` и `class` позволяющие выделить данный элемент среди других, либо выделить группу элементов, чтобы задать им потом одинаковое оформление, либо поведение. Общая структура введения атрибута такова:

<элемент имя атрибута="значение атрибута">

Например, с помощью записи

<p class="front">Язык HTML очень удобен для применения и легок в освоении</p>,

Мы отнесли данный параграф к классу `front`. Особенности представителей данного класса могут быть потом определены, например, с помощью технологий `CSS`, либо `JavaScript`.

2.3 Минимальная структура документа.

Документ на языке `HTML` должен отвечать правилам языка. Эти правила указывают структуру документа, то есть какие элементы обязательно должны присутствовать в документе и как их нужно располагать. Элемент, находящийся внутри содержания другого элемента называется вложенным, например

<p class="front">Язык HTML позволяет создавать веб-страницы.</p>

В этом примере элемент `em` является вложенным в элемент `p`. Говорят также, что элемент `em` является потомком элемента `p`, а элемент `p` является предком элемента `em`. Немного обговорим здесь назначение элемента `em`. Элемент `em` обычно используется для выделения некоторого элемента текста, хотя может использоваться и для других целей. Как именно будет реализовано выделение, определяется стилями, описанными для этого документа. По умолчанию содержимое данного элемента будет выделено курсивом[7].

Таким образом, в HTML одни элементы вкладываются в другие, при этом цепочка вложений может быть очень длинной. Структуру гипертекстового документа можно представлять в виде иерархической структуры. Часто можно встретить упоминание, что элементы HTML — это кирпичики, из которых состоит документ. Также структура документа похожа на русскую матрешку, где одни матрешки вкладываются в другие.

Далее приведем основную структуру HTML документа, то есть ту структуру, которую должны иметь все документы.

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

<title>Название нашей чудесной страницы</title>

</head>

<body>

<p>Я параграф 1</p>

<p>Я параграф 2</p>

</body>

</html>

В первой строчке объявляется тип документа. Если мы после слова DOCTYPE пишем просто HTML, это означает, что мы будем использовать версию языка HTML 5, то есть последнюю версию языка. Все, что дальше будет задавать на наш документ должно располагаться между открывающим и закрывающим тегом html[7].

Все остальные элементы вкладываются в элемент html. На третьей строчке мы видим открывающий тег элемента head. В этом элементе содержится информация о документе[8]. Например, на четвертой строчке указываются тип кодировки символов, который используется в данном документе. Содержимое элемента title служит для формирования названия страницы. Именно это название отображается в качестве названия вкладки окна браузера, когда в браузере будет открыта наша веб-страница. Внутри элемента head могут располагаться и другие элементы, несущие другую информацию. Например, может описываться для поисковых систем о чем данная веб-страница, могут указываться адреса подключаемых таблиц стилей CSS, сценариев на языке JavaScript и др[8].

Внутри элемента body находится все то, что должен увидеть пользователь на нашей странице. В нашем примере внутри элемента body находятся два элемента p, то есть два параграфа текста. В конце мы видим закрывающие теги элементов body и html. В этом документе, есть обязательные элементы это html, head и body, то есть они должны присутствовать в любом html-документе.

2.4 Элементы языка.

Рассмотрим те элементы языка HTML, которые можно встретить в исходном тексте практически любой веб-страницы. Прежде всего, это элементы, которые позволяют обрабатывать текст. Помимо приведенных элементов p сюда также относятся заголовки различных уровней, элементы h1, h2, h3 и т.д. Всего язык HTML 5 предусмотрено 6 уровней заголовков, то есть заголовки с первого по шестой уровень. В качестве примера приведем использование текстовых элементов параграфа и заголовка первого уровня [8].

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My First Heading</h1>
```

<p>My first paragraph.</p>

</body>

</html>

Элемент h1 превращает текст (My First Heading) в заголовок первого уровня, который будет отображаться специальным образом, отличным от основного текста. Основной текст (My first paragraph.) является содержанием элемента p (параграфа текста).

Обратите внимание, элементы h1 и p являются дочерними по отношению к элементу body. Элементы могут быть вложены друг в друга, как матрешки.

Результат обработки такого документа в браузере будет выглядеть следующим образом:



Рисунок 2 — Отображение содержимого элементов h1 и p в браузере[8]

Часто используются на сайтах элементы img (изображение) и br, о которых мы рассказывали в предыдущих разделах. Элемент div не несет никакой смысловой нагрузки, однако является контейнером для других элементов. В современном подходе к созданию страниц без этого элемента не обходится ни один сайт.

Очень интересно применение элементов таблиц table. Помимо тех таблиц, что мы видим на сайте, на веб-страницах можно использовать таблицы для создания разметки. То есть эти таблицы не будут видны пользователю непосредственно, однако позволят расположить элементы там, где нужно. Такой способ создания разметки (верстка) широко использовался еще 7 лет назад. Однако, теперь он устарел, так как с его помощью тяжело реализовывать адекватное отображение веб-страниц на мобильных устройствах. Таблица задается следующим образом:

```
<table>
```

```
<tr>
```

```
<td>...</td>
```

```
</tr>
```

```
</table>
```

Элемент tr задает строку таблицы, а элемент td отдельную ячейку таблицы.

Для формирования нумерованного списка используется элемент ol, а для создания маркированного списка элемент ul. Например, маркированный список можно оформить так:

```
<ul>
```

```
<li>Чебурашка</li>
```

```
<li>Крокодил Гена</li>
```

```
<li>Шапокляк</li>
```

```
<li>Крыса Лариса</li>
```

```
</ul>
```

Этот пример будет отображаться в браузере так:

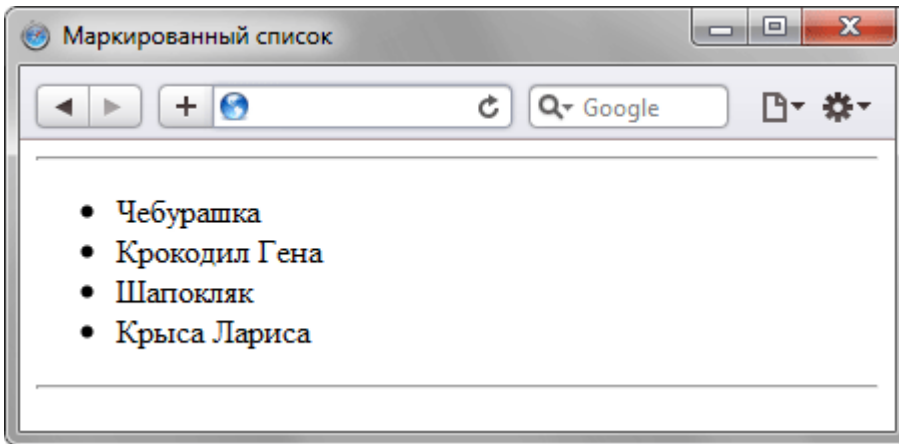


Рисунок 3 — Использование элементов ul

Элемент li служит для создания одного пункта списка. Такие элементы должны быть вложены в элемент ul, либо в элемент ol [7].

Существует много других элементов, позволяющих делать информацию в сети Интернет интересной и доступной.

Есть ряд элементов, называемых пустыми. У этих элементов нет содержимого. Точнее, оно не задается между двумя тешами.

К таким элементам относятся: hr, img, input, keygen, link, meta.

Согласно модели DOM все элементы по способу представления (расположения на странице) делятся на группы: inline, block, inline-block и др

Отнесение к определенной группе означает, каким образом задается область расположения элемента и каким образом элемент ведет себя относительно других элементов.

Инлайн-элементы — это те элементы, которые не образуют новых блоков содержимого; содержимое определяется в строке (на месте) (например, выделенные участки текста внутри параграфа, инлайн-изображения и т.п.)

Название

Назначение

Элемент a

Является одним из важных элементов HTML и предназначен для создания ссылок.

Элемент <code>b</code>	Устанавливает жирное начертание шрифта.
Элемент <code>br</code>	Перенос строки.
Элемент <code>em</code>	Предназначен для акцентирования текста.
Элемент <code>i</code>	Устанавливает курсивное начертание шрифта.
Элемент <code>strong</code>	Акцентирует текст, обычно жирным начертанием

Таблица — Элементы группы Inline. Таблица составлена по [7,9]

Элементы уровня блока это элементы документа-источника, визуально отформатированные как блоки (например, параграфы).

Название	Назначение
Элемент <code>address</code>	Предназначен для хранения информации об авторе
Элемент <code>center</code>	Выравнивает содержимое контейнера по центру относительно родительского элемента.
Элемент <code>div</code>	Универсальный блочный элемент.
Элемент <code>form</code>	Устанавливает форму на веб- странице.
Элемент <code>h1 - h6</code>	Заголовок первого уровня.

Элемент `hr` Рисует горизонтальную линию.

Элемент `menu` Отображает список меню.

Элемент `ol` Устанавливает нумерованный список.

Элемент `p` Определяет текстовый абзац.

Элемент `pre` Определяет блок предварительно форматированного текста.

Элемент `table` Создает таблицу.

Элемент `ul` Устанавливает маркированный список.

Таблица — Элементы группы Block. Таблица составлена по [7,9]

2.5 Описание DHTML

Как было указано выше, DHTML, по-сути, представляет собой набор средств, делающих страницу интерактивной. Этот термин означает совместное использование HTML, CSS, JavaScript для создания веб-страниц. Связывает все это объектная модель DOM.

В современном подходе в HTML задается содержимое веб-страниц, а оформление и поведение задается с помощью CSS и JavaScript.

Cascading Style Sheets (Таблицы Каскадных Стилей) - это язык, содержащий набор свойств для описания внешнего вида любых HTML документов. С его помощью дизайнер имеет полный контроль над стилем и расположением каждого элемента веб-страницы, что проще и гораздо функциональнее использования обычного набора HTML тегов. Приведу пример: Вам нужно создать жирный красный подчеркнутый текст [9].

ПРИМЕР HTML:

<u>

Какой-то текст

</u>

Существует три вида таблиц стилей:

- Внутренние таблицы стилей
- Глобальные таблицы стилей
- Связанные таблицы стилей.

Внутренние таблицы стилей при помощи специального атрибута `style` помещаются прямо в HTML-теги. Глобальные определяют стиль элементов во всем документе. Связанные могут быть использованы для нескольких документов сразу и хранятся во внешнем файле. Подробнее обо всем этом написано ниже [10].

Синтаксис:

селектор {свойства}

Любой элемент HTML — это возможный CSS селектор. Свойства селектора определяют стиль элемента, для которого он определен.

ПРИМЕР:

H1 {color:red; size:20pt;}

Все элементы H1 в документе будут красного цвета, размером в 20 точек (pt, point).

Классовые селекторы (Class Selectors) задаются с помощью следующего синтаксиса:

селектор.класс {свойства}

CLASS - атрибут элемента в HTML, определяющий его класс. В CSS можно описать собственные стили для различных классов одних и тех же элементов.

ПРИМЕР:

H1.blue {color:blue; size:20pt;}

Все элементы H1 с атрибутом CLASS="blue" станут синими.

Классы могут так же быть описаны без явного привязывания их к определенным элементам.

Синтаксис:

.класс {свойства}

ПРИМЕР:

.green {color:green;}

В данном случае все элементы с атрибутом CLASS="green" станут зелеными.

ID селекторы (IDSelectors):

Синтаксис:

#id {свойства}

ID - индивидуально именованный стиль. С его помощью можно создавать стилистические исключения среди элементов одного класса [9].

Индификаторы используются в основном для придания одному или нескольким элементам одного класса индивидуальных свойств. Скажем, Вы создали класс blue - синий курсив. Но Вам понадобился жирный подчеркнутый текст синим курсивом. Можно создать новый класс, но проще описать ID. Например "boldunderline". И все элементы класса blue с значением ID "boldunderline" станут жирным подчеркнутым синим курсивом. Произойдет как бы синтез свойств класса blue и идентификатора boldunderline.

ПРИМЕР:

<html>

<head>

<title>ПримерCSS</title>

</head>

<style>

.blue {color:blue; font-style:italic}

#boldunderline {text-decoration:underline; font-weight:bold}

```
</style>
```

```
<body>
```

```
<pclass="blue"> Здравствуйте, это моя домашняя страница. </p>
```

```
<pclass="blue" id="boldunderline"> Пока еще в стадии разработки ... </p>
```

```
<p id="boldunderline">... Но скоро откроется</p>
```

```
</body>
```

```
</html>
```

Как видно из примера, атрибут ID может быть использован без указания класса (последний параграф примера. Тогда параграф будет обладать только свойствами ID"boldunderline" (в примере - жирный, подчеркнутый текст) [10].

Далее рассмотрим возможности включения кода, написанного на языке программирования JavaScript в содержимое веб-страницы

При загрузке страницы просматриваются сверху вниз, тем самым последовательно происходит компоновка страницы и ее отображение в окне браузера. А это означает, что и объектная модель страницы также формируется последовательно, по мере ее обработки. Поэтому невозможно обратиться из сценария, расположенного ранее какой-либо формы на странице, к элементам этой формы.

Всегда следует помнить о том, что браузер последовательно сверху вниз интерпретирует содержимое HTML-страницы. Еще один аспект работы с объектами языков сценариев заключается в том, что нельзя изменить свойства объектов. Браузер обрабатывает страницу только один раз, компоная и отображая ее. Поэтому попытка в сценарии изменить свойство отображенного элемента страницы, обречена на провал. Только повторная загрузка страницы приведет к желаемому результату[6].

Размещение операторов языка JavaScript на странице. Встроить сценарий JavaScript в HTML-страницу можно несколькими способами.

1. Задание операторов языка внутри тэга <script>, внутри которого могут располагаться операторы языка JavaScript. Обычно браузеры, не поддерживающие какие-нибудь тэги HTML, просто их игнорируют, анализируя, однако, содержимое пропускаемых тэгов с точки зрения синтаксиса языка HTML, что может приводить к

ошибкам при отображении страницы. Во избежание подобной ситуации следует помещать операторы языка JavaScript в контейнер комментария `<!-- -->`. Параметр `language` задает используемый язык сценариев. В случае языка JavaScript его значение задавать не обязательно, так как этот язык используется браузерами по умолчанию.

Пример:

```
<script language="JavaScript" src="http://url/file.js">
```

операторы javascript

```
</script>
```

2. Элементы JavaScript в параметрах тэгов HTML

Переменные и выражения JavaScript можно использовать в качестве значений параметров тэгов HTML. Эта процедура аналогична процедуре встраивания числовых или символьных примитивов HTML. Элементы JavaScript также располагаются между амперсандом (&) и точкой с запятой (;), но должны заключаться в фигурные скобки {} и использоваться только в качестве значений параметров тэгов HTML.

Пусть определена переменная `barwidth`, и ей присвоено значение 75. Следующий тэг нарисует горизонтальную линию длиной в 75% от горизонтального размера окна браузера:

```
<HR WIDTH="{barWidth}%" ALIGN="LEFT">
```

Нужно учитывать следующее. Нельзя использовать элементы JavaScript в тексте HTML. Они интерпретируются только тогда, когда расположены справа от параметра и задают его значение. Например, попытка использовать значение переменной `myVar` в следующем фрагменте

```
<H4> {myVar}; </H4>
```

обречена на провал. Вместо ожидаемого значения переменной `myVar` браузер отобразит строку `myVar[6]`.

3. Обработчики событий.

Для совместимости с языками сценариев в некоторые тэги HTML были введены специальные параметры обработки возникающих событий. Значениями этих параметров могут быть операторы языка JavaScript. Обычно в

качестве значения задается имя функции, которая вызывается, когда происходит соответствующее событие, определяемое параметром обработки события. Имя параметра начинается с приставки `on`, за которым следует имя самого события. Например, параметр обработки события `click` (щелчок кнопкой мыши) будет иметь имя `onclick` [13].

События в основном связаны с действиями, производимыми пользователем с элементами форм HTML. Поэтому чаще всего перехват и обработка событий задается в параметрах элементов форм, что позволяет проверить введенную информацию перед ее отправкой на обработку CGI-сценарием.

Здесь кратко остановимся на функциях JavaScript. Функция или процедура — это именованная последовательность операторов, которая выполняет определенную задачу и может возвращать некоторое значение. Функция определяется оператором `function`, имеющем следующий синтаксис:

```
function имя_функции([параметры]) {
```

```
[операторы JavaScript]
```

```
[return значение]
```

```
}
```

Параметры, передаваемые функции, разделяются запятыми. Необязательный оператор `return` в теле функции (блок операторов, заключенный в фигурные скобки), определяет возвращаемое функцией значение.

Следует четко понимать различие между объявлением функции и ее вызовом. Объявление функции только задает ее имя и определяет, что она будет делать при ее вызове. Непосредственное выполнение функции осуществляется, когда в сценарии вызывают ее и передают действительные параметры.

Определение необходимых функций следует осуществлять в тэге `<HEAD>`, так как все определенные в нем операторы сценария интерпретируются до отображения страницы, и, таким образом, будут известны в процессе отображения всей страницы[11].

2.6 Выводы по главе

Таким образом, современный подход к разработке веб-страниц предполагает совместное использование технологий HTML, CSS, JavaScript.

Только такой подход позволяет сделать веб-страницы по-настоящему интерактивными.

Использование таблиц стилей имеет следующие преимущества:

- возможность подключения одного набора стилей к нескольким документам, что особенно актуально, так как веб-страницы в пределах одного сайта, как правило, имеют схожее оформление.
- возможность редактирования стилей документа отдельно от содержимого
- возможность подключения специальных стилей, например версии для слабовидящих.

JavaScript позволяет не только изменять внешний вид элементов веб-страницы в зависимости от действий пользователя (событий), но также выводит интерактивность веб-страниц на новый уровень, включая анимацию и онлайн-игры.

Заключение

Мы определили основные понятия, связанные с языками гипертекстовой разметки. Описали синтаксис и область применения языка HTML, область применения и синтаксис составляющих DHTML, сравнили возможности применения различных технологий создания веб-страниц.

Содержимое веб-старниц чаще всего формируется с помощью языка HTML. Даже если, для формирования содержимого используются серверные языки программирования или скриптовые сценарии на стороне клиента, вывод элементов веб-страницы все равно основан на элементах HTML или XHTML.

XHTML используется, когда необходимо применить также языки разметки основанные на XML.

Наиболее современным подходом к созданию веб-сайтов является DHTML, позволяющий разделить элементы содержания веб-страниц, их оформление (внешний вид) и скриптовую часть. Кроме того, использование DHTML позволяет сделать сайты интерактивными.

Для того, чтобы создать современный профессиональный сайт необходимо на высоком уровне владеть всеми составляющими DHTML: HTML, CSS и JavaScript, понимать принцип их взаимодействия с помощью DOM.

Библиография

1. Дакетт Д. Основы веб-программирования с использованием HTML, XHTML и CSS / Д. Дакетт, 2-е издание. -М.: Эксмо, 2010.
2. Агулар Р. HTML и CSS. Основа любого сайта / Р. Агулар. -М.: Эксмо, 2010.
3. A history of HTML // URL: <https://www.w3.org/People/Raggett/book4/ch02> . (Дата обращения 03.01.2018)
4. Фримен Э. Изучаем HTML, XHTML и CSS / Э. Фримен. - СПб.: Питер, 2017. -720 с.
5. Пэмбертон С. Ответы на часто задаваемые вопросы по XHTML и HTML/ С. Пэмбертон / URL: <https://www.w3.org/2006/06/xhtml-faq-ru.html#need> (Дата обращения 03.01.2018)
6. Зудилова Т.В., Буркова М.Л.. Web-программирование JavaScript /Зудилова Т.В., Буркова М.Л. - СПб: НИУ ИТМО, 2012. - 68 с
7. htmlbook.ru | Для тех, кто делает сайты //URL: <http://htmlbook.ru> (Дата обращения 03.01.2018)
8. HTML Tutorial // URL: <https://www.w3schools.com/html/default.asp> (Дата обращения 03.01.2018)
9. НОУ ИНТУИТ | Лекция | Модель визуального форматирования<
<https://www.intuit.ru/studies/courses/90/90/lecture/28343?page=2>
10. Лекция 15. Каскадные таблицы стилей (CSS) // URL:
http://infofiz.ru/joom1/index.php?option=com_content&view=article&id=217:lk15&catid=4
(Дата обращения 04.01.2018)
11. Учебники по созданию сайтов, создание сайтов, веб-дизайн, интернет программирование // URL: <http://www.des4web.ru/Html4.0/gl9.html> (Дата обращения 04.01.2018)
12. Дубовиченко С.Б., Шмыгалева Т.А. Web-программирование. Основы языка JavaScript. / С.Б. Дубовиченко, Т.А. Шмыгалева.- Т. 1- учебник для ВУЗов. :- Алматы: Қазақ университеті, 2011 г. - 265 с.