

ББК 63.3 (2) 46

К 92

Карандин С., Колосов М., Поляков Е.

Краткий справочник по разработке Web-приложений на базе Lotus Domino R5

Настоящий справочник посвящен вопросам программирования при создании Web-приложений под управлением Lotus Domino/Notes версии 5x - программного продукта компании Lotus Development Corporation, An IBM Company.

В книге кратко рассматриваются вопросы протокола HTTP и языка HTML. Во второй части дается описание языка JavaScript, его встроенных объектов, методов, свойств, операторов и функций, а так же имеющихся ограничений и межплатформной специфики. Отдельные разделы книги посвящены вопросам использования объектов Notes в Web приложениях и практическому опыту разработки Web сайтов на основе Lotus Domino. В приложении к справочнику Вы найдете наиболее интересные проекты в российском интернете, реализованные с использованием технологии Domino.

Справочник предназначен разработчикам приложений для Lotus Notes, уже знакомым с архитектурой баз данных Notes, языками @-формул и LotusScript, которые заинтересованы в обеспечении доступа к своим приложениям не только из клиентов Notes, но и, по протоколу HTTP, из любых Web-браузеров.

Lotus, Lotus Domino и Lotus Notes являются зарегистрированными торговыми знаками фирмы IBM Company. Все другие упомянутые в данном издании товарные знаки и зарегистрированные товарные знаки принадлежат их законным владельцам.

© InterTrust Co., 2001

© Карандин С., Колосов М., Поляков Е. 2001

© Оформление обложки Бронникова О., 2001

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме, и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 5-7419-0096-8 Налоговая льгота - общероссийский классификатор
продукции ОК-005-93, том 2; 953 000 - книги, брошюры.

Предисловие авторов

Книга, которую Вы держите в руках, посвящена вопросам программирования при создании Web-приложений под управлением Lotus Domino/Notes версии 5x - программного продукта компании Lotus Development Corporation, An IBM Company.

Материал книги состоит из 5 частей. В первой части кратко рассматриваются вопросы протокола HTTP и языка HTML. Во второй части дается описание языка JavaScript, его встроенных объектов, методов, свойств, операторов и функций, а так же имеющихся ограничений и межплатформной специфики. В третьей части и четвертой части посвящены вопросам дизайна Web- приложений в Domino.

Авторы отдают себе отчет в том, что книга не лишена недостатков, и имеют к себе претензии по стилю изложения и оформления материала. Ряд ее разделов хотелось бы улучшить, дополнить и снабдить более интересными примерами. Однако в условиях необычайно быстрого развития продукта лучше скорее дать в руки читателей информацию в том виде, как она есть сейчас, чем отложить выпуск книги еще на один-два месяца. В книгу не включен раздел по конфигурированию HTTP сервиса. Этот материал подробно освещен в книге «Справочное пособие для администраторов Lotus Domino R5», выпущенной InterTrust Co. в 2000г.

Материал книги подготовили: 1 часть – С. Будылев, А. Игнатов, М. Колосов, часть 3 - - Е. Поляков; ч. 4-5 -.М. Колосов и ч.6 - С. Карандин.

Авторы будут искренне рады Вашим замечаниям и предложениям по содержанию книги и любой информации технического характера по затронутым в книге вопросам. Направляйте их по e-mail: EPolyakov@intrust.ru или EPolyako@lemon.ru (Е. Поляков), maks@adb.ru (М. Колосов) или «оставляйте» на WWW-сервере нашей компании <http://www.intertrust.ru>.

Авторы выражают искреннюю благодарность и признательность всем, кто оказывал им прямую или косвенную помощь в работе над книгой:

Е. Поляков - Бреусу Игорю Борисовичу, Поляковой Галине Дмитриевне и Полякову Виталию Григорьевичу за существенную помощь, оказанную при написании настоящей книги, а также сыну Роману и дочери Лане за терпение и сочувствие во время работы над книгой.

М.Колосов и С. Карандин – программистам компании ADB.RU Кудрявцевой Инне и Малахову Илье за разработку и тестирование кодов примеров.

С уважением, С. Карандин, М. Колосов и Е. Поляков

1 Введение

Эта книга замышляется как справочник для разработчика Web-приложений на Lotus Domino R5 (Domino). В первую очередь она предназначена разработчикам приложений Notes, которым нужно разрабатывать Web-приложения на Domino.

В качестве источников информации будет использован собственный опыт авторов, подсказки коллег, общедоступный форум по Lotus Notes на сайте компании ИнтерТраст (www.intertrust.ru), R5 Domino Designer Help и т.д.

Несмотря на смену версии, перед разработчиком стоит прежняя задача: получать, обрабатывать и отображать данные. В Domino версии 5 фирма Lotus приложила немало усилий для того, чтобы любая база данных Notes без малейших доработок правильно отображалась в Web-браузере. Тем не менее, победу праздновать рано. Приложение по-прежнему по-разному выглядит в Notes и в Web-браузере (и по-разному работает).

На Domino R5 действительно можно создать Web-приложение, не написав ни одного тега HTML, но при этом Вам, скорее всего, придется смириться с глупейшими мелочами типа лишних переводов строки, от которых невозможно избавиться никакими средствами.

Необходимость использования вставок на HTML возникает, как только Вы решаете воспользоваться возможностями Web-браузера, отсутствующими в клиенте Notes.

Таким образом, разработка приложений для Web на Domino по-прежнему требует от разработчика на Notes специальных знаний, касающихся представления объектов Notes сервером Domino в Web-браузере, HTML, CGI и JavaScript.

Мы надеемся, что данное справочное пособие поможет разработчикам приложений для Web на Domino быстрее и успешнее решать стоящие перед ними задачи.

2 Введение в HTML

2.1 HTTP

Данный раздел справочного пособия рассчитан на программистов Lotus Notes, которые не имеют достаточных знаний и опыта в разработке Web-страниц, и мы постараемся сделать ее кратким справочником по HTML. В крупных проектах дизайн и верстку HTML делают профессиональные дизайнеры и вебмастера, и программисту Domino, как правило, можно и не знать HTML в совершенстве, так же как и дизайнер может не подозревать о том, что его проект будет работать на Domino. Для разработки небольших проектов программисту должно быть достаточно информации, изложенной в этом разделе.

Как правило, для получения страницы с Web сервера используется протокол HTTP (Hyper Text Transfer Protocol - протокол передачи гипертекста).

HTTP является протоколом прикладного уровня Internet, разработанным для передачи информации между WWW-сервером и клиентом (браузером). Взаимодействие происходит следующим образом. Браузер выдает запрос серверу на прием или передачу информации. Сервер обрабатывает данный запрос и возвращает браузеру результат обработки.

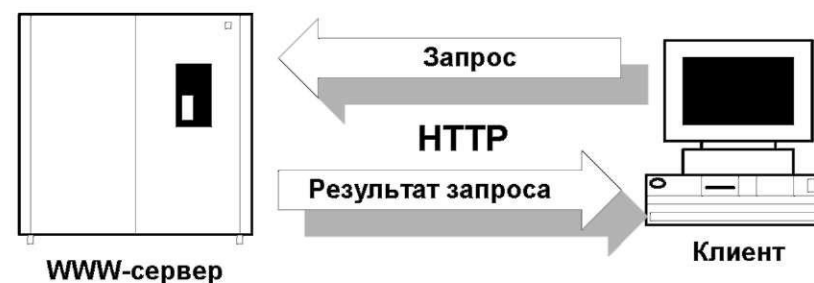


Рис. 1.1 Протокол HTTP: запрос клиента и ответ сервера

Сначала браузер устанавливает соединение с сервером по протоколу TCP/IP. Затем браузер отправляет на сервер запрос. Сервер обрабатывает запрос и возвращает браузеру результат операции. Затем браузер считывает с сервера данные. После этого соединение разрывается.

Необходимо помнить, что WWW-сервер не хранит информацию о предыдущих запросах и при каждом новом запросе устанавливается новое соединение, а после выполнения запроса это соединение разрывается.

В запросе браузера содержится следующая информация.

- **Метод запроса.** Описывает, что необходимо сделать с объектом.
- **URL объекта,** с которым необходимо произвести соответствующую операцию.
- Версия протокола HTTP поддерживаемая браузером.
- В ответе сервера содержатся следующая информация.
- Версия протокола HTTP, поддерживаемая сервером.
- **Код статуса.** Трехзначное число, содержащее результат выполнения запроса.
- **Описание.** Текстовая строка, содержащее краткое описание результата выполнения запроса.
- Информация о запрашиваемом объекте.
- MIME-заголовок запрашиваемого объекта.

Ниже мы более подробно рассмотрим запросы браузера и ответы сервера

2.1.1 Метод запроса

Метод **GET** протокола HTTP запрашивает объект, URL которого задан в заголовке запроса. После исполнения запроса браузер получает запрошенный объект, тип которого задан в MIME-заголовке.

Метод **HEAD** используется аналогично методу GET, с той лишь разницей, что сервер не возвращает по данному запросу объект. Данный метод используется для проверки нахождения данного ресурса на сервере.

Метод **POST** предназначен для передачи объекта на сервер. В результате выполнения такого запроса браузер передает серверу указанный объект. Сервер обрабатывает данный объект и может сохранить его в качестве ресурса или не сохранять. В случае сохранения сервер посылает браузеру ответный объект, которым

подтверждает обработку запроса. Код статуса в этом случае имеет значение 201. В качестве ответного документа обычно выступает документ HTML. Если же сервер не сохраняет объект, то код статуса равен 204 – «нет содержимого».

2.1.2 URL объекта

Местонахождение объекта в Internet, его «адрес» определяет **URL** (Uniform Resource Locator – универсальный указатель ресурса). URL однозначно идентифицирует объект среди множества других объектов. Различают абсолютные и относительные URL. Общий вид URL для протокола HTTP следующий:

http://сервер:порт/путь?параметры.

Абсолютный URL содержит полное описание доступа к объекту. Он состоит из двух основных частей, идентифицирующих данный объект - протокола и пути - в формате **протокол: путь.**

Относительный URL содержит только **путь** к объекту относительно текущего объекта. Из URL текущего объекта берется протокол, сервер, порт и путь к текущему объекту. Далее к нему добавляется относительный URL

2.1.3 Версия протокола, поддерживаемая браузером или сервером

Текстовая строка вида HTTP 1.0 или HTTP 1.1 используется для предотвращения ошибок, связанных с применением сервером и браузером различных версий этого протокола. Другими словами, перед тем как начать передачу информации, сервер должен убедиться, что браузер сможет ее правильно представить

2.1.4 Код статуса

Код статуса имеет следующие группы значений:

Код статуса	Описание	Дополнительная информация
1xx	Информационная	Зарезервировано
2xx	Запрос выполнен успешно	Запрос был успешно обработан
3xx	Запрос	Требуются

	перенаправлен	дополнительные действия
4xx	Ошибка браузера	Сервер не может выполнить запрос, потому что запрос содержит ошибки
5xx	Ошибка сервера	Запрос не содержит ошибок, но сервер не может его выполнить

2.1.5 MIME-заголовок

MIME-заголовок (Multipurpose Internet Mail Extension) используются для указания типа объекта. Заголовок состоит из основного типа и расширения: **тип/расширение**. Тип определяет основной тип ресурса, а расширение описывает его более подробно. Например, тип может быть image (изображение), а расширение - gif (изображение в формате GIF), jpeg (изображение в формате JPEG) или tiff (изображение в формате TIFF). Приведем наиболее употребительные заголовки MIME.

text/html Документ HTML.
text/plain Текстовый документ.
image/gif Изображение в формате GIF.
image/jpeg Изображение в формате JPEG.
video/mpeg Видео в формате MPEG.

2.2 HTML

Аббревиатура HTML расшифровывается как Hyper Text Markup Language – язык гипертекстовой разметки документов. С помощью HTML можно создавать гипертекстовые документы, не зависящие от программно-аппаратной платформы и от конкретной программы просмотра (браузера).

Документ HTML представляет собой текстовый файл (обычно с расширением .html или .htm), который может быть создан и изменен в любом текстовом редакторе. В документе HTML описывается, **какая** информация (текст, графика, апплеты Java и т.п.) будет «видна» при просмотре этого документа в браузере, и, с помощью соответствующих

элементов (тегов), задается, **как** будет «видна» эта информация (форматирование текста, размер шрифта и т.п.). Все дополнительные объекты (изображения, апплеты и т.п.) хранятся в отдельных файлах, а в документе HTML содержатся лишь ссылки на них.

В данной главе дается введение в HTML, без привязки к конкретной версии, с той степенью подробности, которая необходима для понимания последующих глав книги. Различия в представлении данных между Internet Explorer и Netscape Navigator будут указываться отдельно.

2.2.1 Тэги HTML

Тег (tag) является элементом, с помощью которого производится форматирование документа. Тег – это некоторый текст, заключенный в угловые скобки, наподобие **<TAG>**. Теги делятся на одинарные и парные. Если тег парный, то для каждого открывающего элемента **<TAG>** необходим закрывающий **</TAG>**. Для одинарного тега закрывающий тег не нужен. Теги могут содержать атрибуты, которые задают различные свойства для тега, наподобие **<TAG ATTRIB="value">**.

В каких случаях используются одинарные теги, а в каких двойные? Если вам необходимо произвести над документом однократное действие, например, вставить горизонтальную линию – используется одинарный тег **<HR>**. А если какое либо действие необходимо произвести над фрагментом документа, например, установить размер шрифта, тогда необходим двойной тег **<H1> ... </H1>**.

Необходимо помнить, что браузер игнорирует в документе символы перевода строки, множественные пробелы и неизвестные ему теги.

2.2.2 Структура документа HTML

Простейший документ HTML выглядит следующим образом.

```

<HTML>
  <HEAD>
    <TITLE>
      Заголовок
    </TITLE>
  </HEAD>
  <BODY>
    Тело
    документа
  </BODY>
</HTML>

```



Рис. 1.2 Вид простейшего HTML-документа в браузере

Документ заключен между парными тегами `<HTML> ... </HTML>`. Между парными тегами `<HEAD> ...</HEAD>` может быть расположена служебная информация о документе: заголовок, мета-теги и т.п. Внутри тегов `<TITLE>...</TITLE>` находится заголовок документа, который будет показан в заголовке окна браузера. Само же описание документа содержится внутри тегов `<BODY>...</BODY>`.

2.2.3 Заголовок (<HEAD>)

Заголовок играет очень важную роль в документе, т. к. его текст выводится в заголовке страницы. Также, заголовок используется в качестве имени при помещении документа в «закладки» браузера. Не пренебрегайте этим параметром.

2.2.4 Тег <META> - служебная информация в заголовке документа

Как уже говорилось, теги используются для форматирования документа. Однако существует тег `<META>`, используемый для хранения в документе некоторой служебной информации и управления браузером.

```

<META HTTP-EQUIV="name" CONTENT="value">
<META NAME="name" CONTENT="value">

```

META-теги должны находиться в заголовке HTML-документа между `<HEAD>` и `</HEAD>`.

Использование атрибута **HTTP-EQUIV** в теге `<META>` эквивалентно HTTP-заголовкам - обычно атрибут применяется для управления действиями браузеров. Теги `<META>` с атрибутом **NAME** используются для хранения в документе некоторой служебной информации.

Значения атрибута HTTP-EQUIV

- **Content-type** - тип документа. В атрибуте **CONTENT** указывается MIME-тип документа и используемая кодировка, например:


```

<META HTTP-EQUIV="Content-type"
      CONTENT="text/html; charset=ISO-2022-JP">

```
- **Content-language** - язык документа. Язык описывается парой значений «язык-диалект». Например, «Английский-Великобритания»:


```

<META HTTP-EQUIV="Content-language" CONTENT="en-GB">

```
- **Refresh** - задержка в секундах, после которой браузер автоматически обновляет документ. Дополнительная возможность - автоматическая загрузка другого документа. В атрибуте **CONTENT** вначале указывается время, после которого будет произведен переход к URL, а после запятой - этот URL. Если URL опущен, то обновляется текущая страница. Например:


```

<META HTTP-EQUIV="Refresh"

```

```

Content="3;URL=http://www.inttrust.ru/index.html">

```

Значения атрибута NAME

- **Description** - краткая аннотация содержания документа. Например:


```

<META NAME="Description" CONTENT="Краткое описание документа">

```
- **Author** - имя автора. Например:


```

<META NAME="Author" CONTENT="Петров П.П.,
Дизайнерское бюро № 3">

```
- **Generator** - название и версия редактора или динамического генератора HTML, с помощью которого создан этот HTML-документ. Например:


```

<META NAME="Generator" CONTENT="Web Editor Ver 3.0">

```

- **Copyright** - описание авторских прав на документ в произвольном формате. Например:

```
<META NAME="Copyright" CONTENT="Читать документ  
запрещается!">
```

2.2.5 Тело (<BODY>)

Тэг <BODY> используется для установки таких параметров как цвет фона страницы, цвет текста по умолчанию, цвет гиперссылок и т.д. Ниже описаны только те параметры тэга, которые работают как в Internet Explorer, так и в Netscape Navigator:

```
<BODY ALINK="цвет активной ссылки" BACKGROUND="URL фона"  
BGCOLOR="цвет фона" LINK="цвет не просмотренной ссылки"  
TEXT="цвет текста" VLINK="цвет просмотренной ссылки">
```

У тэга <BODY> в NN и IE существует еще целый ряд параметров. Но при их использовании картинка страницы, показываемая NN и IE, будет различаться.

2.2.6 Комментарии

Текст, помещенный между тегами <!-- ... --> считается комментарием и не отображается в браузере. Комментарии часто используют для пояснений в сложных HTML-документах.

2.2.7 Форматирование текста в документе

Для форматирования текста в HTML-документе обычно используются следующие теги.

<H1>...</H1> - <H6>...</H6> - Задают размера шрифта: <H1> - самый крупный, <H6> - самый мелкий.

<P>...</P> - Все, что заключено между данными тегами, воспринимается как один абзац.

Оба эти тега могут содержать атрибут **ALIGN**, который задает выравнивание текста (**CENTER**, **LEFT**, **RIGHT**).

 ... - Текст внутри пары тэгов выделен полужирным шрифтом.

<I> ... </I> - Текст внутри пары тэгов выделен курсивом.

<TT> ... </TT> - Текст внутри пары тэгов выводится шрифтом фиксированной ширины.

<U> ... </U> - Текст внутри пары тэгов подчеркнут.

<STRIKE> ... </STRIKE> - Текст внутри пары тэгов зачеркивается.

<BIG> ... </BIG> - Текст внутри пары тэгов выводится шрифтом большего размера.

<SMALL> ... </SMALL> - Текст внутри пары тэгов выводится шрифтом меньшего размера.

_{...} - Текст внутри пары тэгов сдвигается ниже уровня строки и выводится, по возможности, шрифтом меньшего размера.

^{...} - Текст внутри пары тэгов сдвигается выше уровня строки и выводится, по возможности, шрифтом меньшего размера.

 - данный тег позволяет перейти на новую строку.

<HR> - Горизонтальный разделитель. Имеет следующие атрибуты:

ALIGN – Выравнивает разделитель по краям или центру. Допустимые значения: **LEFT**, **CENTER**, **RIGHT**

WIDTH – Устанавливает длину линии в пикселях или процентах от ширины окна браузера или ячейки таблицы. В случае указания длины в процентах, после указания количества процентов, устанавливается знак "%".

SIZE – Устанавливает ширину линии в пикселях.

NOSHADE – Отменяет рельефность линии

COLOR – Устанавливает цвет линии. Используется формат RGB или стандартное имя.

Использование горизонтальных линий должно быть логически обосновано. Нежелательно применять их для отделения заголовка от текста или применяться для выделения пустого места в Вашем документе.

<PRE>...</PRE> - Текст, находящийся между этими тегами, отображается в браузере без изменений. Множественные пробелы и символы конца строки в этом случае не игнорируются.

Пример 1. Форматирование документа.

```

<HTML>
<HEAD>
<TITLE>Пример
форматирования
документа</TITLE>
</HEAD>
<BODY>
<H1
ALIGN=CENTER>Вырав
нение по
центру</H1>
<P ALIGN=RIGHT>
Абзац, выровненный
вправо
</P>
<P ALIGN=LEFT>
Первая строка
<BR>
Вторая строка
после разрыва
</P>
<HR WIDTH=80%>
Разделитель

```

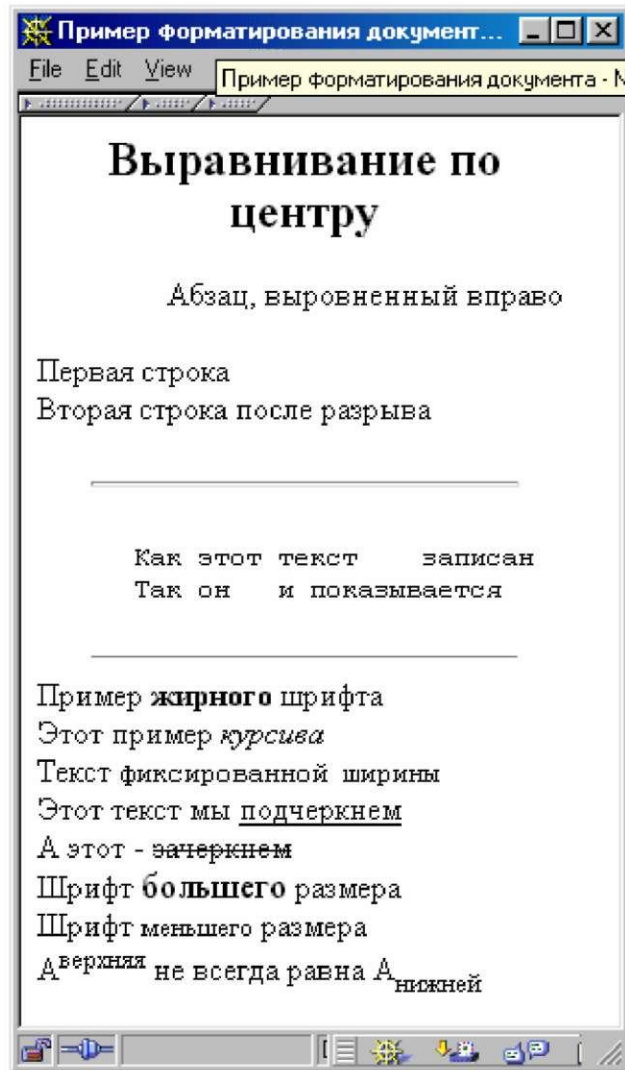


Рис. 1.3 Пример форматирования текста

```

<PRE>
        Как этот текст    записан
        Так он    и показывается
</PRE>
Пример <B>жирного</B> шрифта<BR>
Этот пример <I>курсива</I><BR>
Текст <TT>фиксированной ширины</TT><BR>
Этот текст мы <U>подчеркнем</U><BR>
А этот - <STRIKE>зачеркнем</STRIKE><BR>
Шрифт <BIG>большого</BIG> размера<BR>
Шрифт <SMALL>меньшего</SMALL> размера<BR>
A<SUP>верхняя</SUP> не всегда равна A<SUB>нижней</SUB>
</BODY>
</HTML>

```

2.2.8 Шрифты

Для того чтобы текст в документе был максимально удобочитаем, в спецификации HTML было введен парный тэг , которые обеспечивают работу со шрифтами.

Атрибуты:

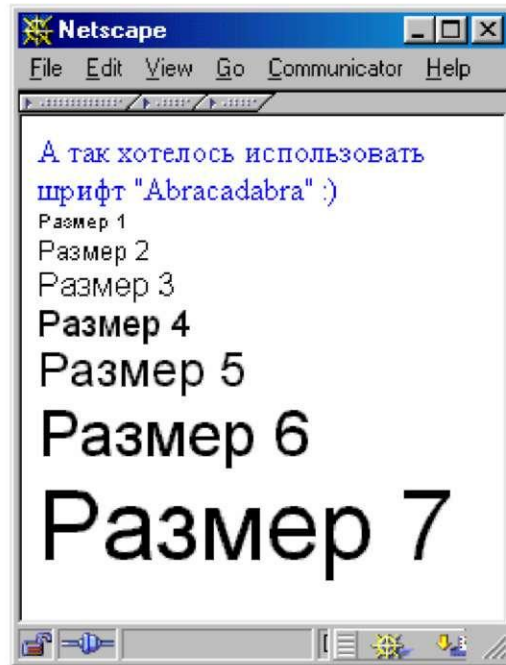
COLOR – Цвет шрифта в стандарте RGB или по имени. При этом не учитывается функциональное назначение текста, т.е. простой текст, активная ссылка, ссылка на которой вы еще не были, и просмотренная ссылка будут показываться одним цветом.

FACE – Позволяет указать тип шрифта, которым программа просмотра выводит Ваш текст. Параметром атрибута служит название шрифта, которое должно точно совпадать с названием шрифта у пользователя. Если нужного шрифта нет – программа проигнорирует запрос, и будет использовать шрифт, установленный у пользователя по умолчанию. Атрибут FACE позволяет указывать как один, так и несколько шрифтов, что весьма полезно, т.к. почти идентичный шрифт на платформе Windows и Macintosh называется по-разному. Список будет просмотрен слева направо, и первый из имеющихся у пользователя шрифтов будет подставлен для вывода документа.

SIZE – Этот атрибут служит для указания размера шрифта в условных единицах от 1 до 7. По умолчанию, нормальный шрифт имеет размер 3. Можно указывать как абсолютный (SIZE=2), так и относительный (SIZE=+1) размер.

Пример. Использование шрифтов.

```
<HTML>
<HEAD>
<TITLE>
Пример использования
шрифтов.
</TITLE>
</HEAD>
<BODY>
<FONT
FACE="Abracadabra",
"Arial", "Helvetica"
COLOR=blue>
А так хотелось
использовать шрифт
"Abracadabra" :)
</FONT><BR>
```



```
<FONT FACE= "Helvetica" SIZE=1>
Размер 1</FONT><BR>
<FONT FACE= "Helvetica" SIZE=2>
Размер 2</FONT><BR>
<FONT FACE= "Helvetica" SIZE=3>
Размер 3</FONT><BR>
<FONT FACE= "Helvetica" SIZE=4>
Размер 4</FONT><BR>
```

```
<FONT FACE= "Helvetica" SIZE=5>
Размер 5</FONT><BR>
<FONT FACE= "Helvetica" SIZE=6>
Размер 6</FONT><BR>
<FONT FACE= "Helvetica" SIZE=7>
Размер 7</FONT>
</BODY>
</HTML>
```

2.2.9 Специальные символы

Для лучшего восприятия текста желательно использовать специальные символы.

Ниже приведена краткий список наиболее употребимых спецсимволов и их коды:

Код	Наименование	Вид
§	Параграф	§
©	Copyright	©
®	Registered	®
°	знак градуса	°
«	левая кавычка	«
»	правая кавычка	»
…	Троеточие	...
‘	одиночная откр. кавычка	'
’	одиночная закр. кавычка	'
“	откр. кавычка	“
”	закр. кавычка	”
•	жирная точка	•
–	среднее тире	–

—	длинное тире	—
™	Trademark	™
№	русский знак номера	№
±	плюс-минус	±

2.2.10 Цвета

В элементах FONT и BODY некоторые конструкции HTML могут быть использованы, чтобы специфицировать цвета: например, цвет фона, цвет текста по умолчанию или цвет текста связи.

Так как все мониторы и программы могут отображать цвет, действительное отображение может быть черно-белым или с различными оттенками серого.

Для спецификации цвета, как значения атрибута, существуют два варианта:

Символьная нотация RED.

Black (Черный)	Green (Зеленый)	Silver (Серебро)	Lime (Известь)
Gray (Серый)	Olive (Оливковый)	White (Белый)	Yellow (Желтый)
Maroon (Бордовый)	Navy (Темно-синий)	Red (Красный)	Blue (Синий)
Purple (Фиолетовый)	Teal (Чирок)	Fuchsia (Фуксия)	Aqua (Аква)

Имена цветов можно записывать в любом регистре с кавычками или без.

Цифровое обозначение в шестнадцатеричной записи, например, "#FF0000", которая контролирует, каким образом цвет формируется из основных цветов - красного, зеленого и голубого - в так называемое RGB цветовое пространство, при этом обозначение должно быть взято в кавычки.

Символические нотации много легче и более понятны. С другой стороны численные обозначения дают больше возможностей.

Практически полный список названий цветов приведен на странице: <http://www.design.ru/kovodstvo/paragraphs/11.html>.

2.2.11 Представление абзацев в виде списков

Часто необходимо бывает представить текст в виде списка, например:

- Первый элемент списка
- Второй элемент списка
- Третий элемент списка

Этот вид списка называется нумерованным. Чтобы создать такой список, необходимо вставить в документ HTML следующий текст:

```
<UL>
<LI>Первый элемент списка
<LI>Второй элемент списка
<LI>Третий элемент списка
</UL>
```

Список располагается между парными тегами Каждый элемент списка начинается с одинарного тега .

Чтобы сделать список нумерованным, например, таким:

1. Первый элемент списка
2. Второй элемент списка
3. Третий элемент списка

необходимо вместо тегов ... использовать теги

Пример 2. Использование списков.

```

<HTML>
<HEAD>
<TITLE>
    Пример со
    списками
</TITLE>
</HEAD>
<BODY>
<H2>Ненумерованный
список</H2>
<H4>
<UL>
    <LI>Первый элемент
    списка
    <LI>Второй элемент
    списка
    <LI>Третий элемент
    списка

</UL>
</H4>

<HR>
<H2>Нумерованный список</H2>
<H4>

```

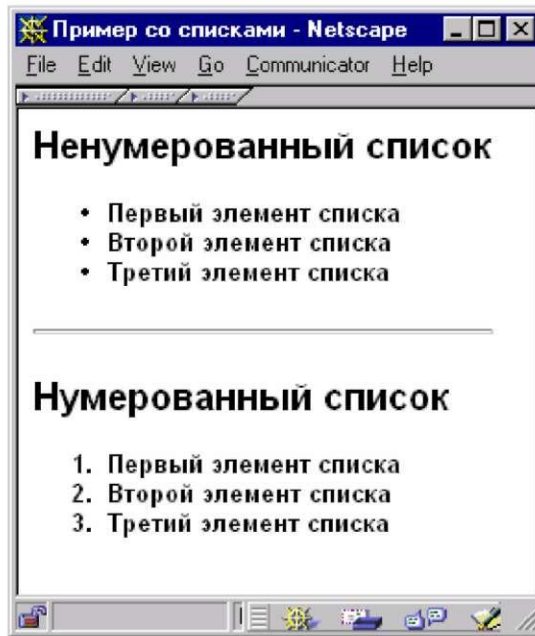


Рис. 1.4 Использование списков

```

<OL>
    <LI>Первый элемент списка
    <LI>Второй элемент списка
    <LI>Третий элемент списка
</OL>
</H4>
</BODY>
</HTML>

```

2.2.12 Изображения в документе HTML

Чтобы встроить изображение в документ HTML, используется тег ****:

```

<IMG SRC="URL изображения" ALT="Пояснение"
ALIGN=ВЫРАВНИВАНИЕ BORDER="Толщина рамки в пикселях"
HEIGHT="Высота изображения в пикселях"
HSPACE="Горизонтальная граница в пикселях" ISMAP
LOWSRC="Местоположение" NAME="Имя изображения"
USEMAP=#map VSPACE="Вертикальная граница в пикселях"
WIDTH="Ширина изображения в пикселях">

```

В документах HTML используются изображения в форматах GIF (*.gif) и JPEG (*.jpg или *.jpeg).

Атрибут **SRC** необходим для указания URL изображения.

Атрибут **ALT** задает альтернативный текст-пояснение. Пояснение будет видно пользователю до того, как браузер полностью загрузит изображение, может быть полезно, если у пользователя отключена автоматическая загрузка изображений, или в случае, когда используется старый браузер, не имеющий графических возможностей.

Атрибут **ALIGN** служит для выравнивания изображения и имеет следующие значения:

CENTER - выравнивание по центру;

LEFT - текст обтекает изображение, изображение слева от текста;

RIGHT - текст обтекает изображение, изображение справа от текста;

TOP - Выравнивает текст по верху изображения;

MIDDLE - Выравнивает текст по середине изображения;