HTML - введение

HyperText Markup Language (HTML) является стандартным языком, предназначенным для создания гипертекстовых документов в среде WEB. HTML-документы могут просматриваться различными типами WEB-броузеров. Когда документ создан с использованием HTML, WEB-броузер может интерпретировать HTML для выделения различных элементов документа и первичной их обработки. Использование HTML позволяет форматировать документы для их представления с использованием шрифтов, линий и других графических элементов на любой системе, их просматривающей.

Большинство документов имеют стандартные элементы, такие, как заголовок, параграфы или списки. Используя **тэги** HTML вы можете обозначать данные элементы, обеспечивая WEB-броузеры минимальной информацией для отображения данных элементов, сохраняя вцелом общую структуру и информационную полноту документов. Все что необходимо, чтобы прочитать HTML-документ - это WEB-броузер, который интерпретирует тэги HTML и воспроизводит на экране документ в виде, который ему придает автор.

В большинстве случаев автор документа строго определяет внешний вид документа. В случае HTML читатель (основываясь на возможностях WEB-броузера может, в определенной степени, управлять внешним видом документа (но не его содержимым). HTML позволяет отметить, где в документе должен быть заголовок или абзац при помощи тэга HTML, а затем предоставляет WEB-броузеру интерпретировать эти тэги. Например, один WEB-броузер может распознавать тэг начала абзаца и представлять документ в нужном виде, а другой не имеет такой возможности и представляет документ в одну строку. Пользователи некоторых WEB-броузеров имеют, также, возможность настраивать размер и вид шрифта, цвет и другие параметры, влияющие на отображение документа.

HTML-тэги могут быть условно разделены на две категории:

- тэги, определяющие, как будет отображаться WEB-броузером тело документа вцелом
- тэги, описывающие общие свойства документа, такие как заголовок или автор документа

Запомните, что основное преимущество HTML заключается в том, что ваш документ может быть просмотрен на WEB-броузерах различных типов и на различных платформах.

Как создаются HTML документы?

HTML-документы могут быть созданы при помощи любого текстового редактора или специализированных HTML-редакторов и конвертеров. Выбор редактора, который будет использоваться для создания HTML-документов, зависит исключительно от понятия удобства и личных пристрастий каждого автора.

Например, HTML редакторы, такие, как "Netscape Navigator Gold" компании Netscape позволяют создавать документы графически с использованием технологии WYSIWYG (What You See Is What You Get). С другой стороны, большинство традиционных средств для создания документов имеют конвертеры, позволяющие преобразовывать документы к формату HTML.

Основные положения

Все тэги HTML начинаются с "<" (левой угловой скобки) и заканчиваются символом ">" (правой угловой скобки). Как правило, существует стартовый тэг и завершающий тэг. Для примера приведем тэги заголовка, определяющие текст, находящийся внутри стартового и завершающего тэга и описывающий заголовок документа:

```
<TITLE> Заголовок документа </TITLE>
```

Завершающий тэг выглядит также, как стартовый, и отличается от него прямым слэшем перед текстом внутри угловых скобок. В данном примере тэг <TITLE> говорит WEB-броузеру об использовании формата заголовка, а тэг </TITLE> - о завершении текста заголовка.

Некоторые тэги, такие, как <P> (тэг, определяющий абзац), не требуют завершающего тэга, но его использование придает исходному тексту документа улучшенную читаемость и структурируемость.

HTML не реагирует на регистр символов, описывающих тэг, и приведенный ранее пример может выглядеть следующим образом:

```
<title> Заголовок документа </title>
```

> **Внимание!** Дополнительные пробелы, символы табуляции и возврата каретки, добавленные в исходный текст HTML-документа для его лучшей читаемости, будут проигнорированы WEB-броузером при интерпретации документа. HTML-документ может включать вышеописанные элементы только если они помещены внутрь тэгов <PRE> и </PRE>. Более подробно о тэгах <PRE> будет написано ниже.

Структура документа

Когда WEB-броузер получает документ, он определяет, как документ должен быть интерпретирован. Самый первый тэг, который встречается в документе, должен быть тэгом <HTML>. Данный тэг сообщает WEB-броузеру, что ваш документ написан с использованием HTML. Минимальный HTML-документ будет выглядеть так:

```
<HTML> ...тело документа... </HTML>
```

Заголовочная часть документа <HEAD>

Тэг заголовочной части документа должен быть использован сразу после тэга <HTML> и более нигде в теле документа. Данный тэг представляет из себя общее описание документа. Избегайте размещать какой-либо текст внутри тэга <HEAD>. Стартовый тэг <HEAD> помещается непосредственно перед тэгом <TITLE> и другими тэгами, описывающими документ, а завершающий тэг </HEAD> размещается сразу после окончания описания документа. Например:

```
<HTML>
<HEAD>
<TITLE>                Список                сотрудников                </TITLE>
```

```
</HEAD>
...
```

> **Внимание!** Технически, стартовые и завершающие тэги типа <HTML>, <HEAD> и <BODY> необязательны. Но настоятельно рекомендуется их использовать, поскольку использование данных тэгов позволяет WEB-броузеру уверенно разделить заголовочную часть документа и непосредственно смысловую часть.

Заголовок документа <TITLE>

Большинство WEB-броузеров отображают содержимое тэга <TITLE> в заголовке окна, содержащего документ и в файле закладок, если он поддерживается WEB-броузером. Заголовок, ограниченный тэгами <TITLE> и </TITLE>, размещается внутри <HEAD>-тэгов, как показано выше на примере. Заголовок документа не появляется при отображении самого документа в окне.

Комментарии

Как любой язык, HTML позволяет вставлять в тело документа комментарии, которые сохраняются при передаче документа по сети, но не отображаются броузером. Синтаксис комментария:

```
<!-- Это комментарий -->
```

Комментарии могут встречаться в документе где угодно и в любом количестве.

Тэги тела документа

Тэги тела документа идентифицируют отображаемые в окне компоненты HTML-документа. Тело документа может содержать ссылки на другие документы, текст и другую форматированную информацию.

Тело документа <BODY>

Тело документа должно находиться между тэгами <BODY> и </BODY>. Это та часть документа, которая отображается как текстовая и графическая (смысловая) информация вашего документа.

Уровни заголовков <Hx>

Когда пишется HTML-документ, текст структурно делится на просто текст, заголовки частей текста, заголовки более высокого уровня и т.д. Первый уровень заголовков (самый большой) обозначается цифрой 1, следующий - 2, и т.д. Большинство броузеров поддерживает интерпретацию шести уровней заголовков, определяя каждому из них

собственный стиль. Заголовки выше шестого уровня не являются стандартом и могут не поддерживаться броузером. Заголовок самого верхнего уровня имеет признак "1". Синтакс заголовка уровня 1 следующий:

```
<H1> Заголовок первого уровня </H1>
```

Заголовки другого уровня могут быть представлены в общем случае так:

```
<Hx> Заголовок x-го уровня </Hx>
```

где x - цифра от 1 до 6, определяющая уровень заголовка.

Тэг абзаца <P>

В отличии от большинства текстовых процессоров, в HTML-документе обычно игнорируются символы возврата каретки. Физический разрыв абзаца может находиться в любом месте исходного текста документа (для удобства его читаемости). Однако броузер разделяет абзацы только при наличии тэга <P>. Если вы не разделите абзацы тэгом <P>, ваш документ будет выглядеть как один большой абзац.

Дополнительные параметры тэга <P>:

```
<P ALIGN=left|center|right>
```

позволяют выравнивать абзац по левому краю, центру и правому краю соответственно.

Центрирование элементов документа

Вы можете центрировать все элементы документа в окне броузера. Для этого можно использовать тэг <CENTER>.

Все элементы между тэгами `<CENTER>` и `</CENTER>` будут находиться в центре окна

Тэг преформатирования <PRE>

Тэг преформатирования, <PRE>, позволяет представлять текст со специфическим форматированием на экране. Предварительно сформатированный текст заканчивается завершающим тэгом </PRE>. Внутри предварительно сформатированного текста разрешается использовать:

- перевод строки
- символы табуляции (сдвиг на 8 символов вправо)
- непропорциональный шрифт, устанавливаемый броузером

Использование тэгов, определяющих формат абзаца, таких как <Hx> или <ADDRESS>, будет игнорироваться броузером при помещении их между тэгами <PRE> и </PRE>.

Далее идет несколько более подробный пример, собранный из предыдущих:

```
 <HTML>
<HEAD>
<TITLE> Список сотрудников </TITLE>
</HEAD>
<BODY>
<H2> Список сотрудников нашей фирмы </H2>
<H3> Составлено : 30 июля 1996 года </H3>
Данный список содержит фамилии, имена и отчества
всех сотрудников нашей компании. <P>
Список может быть использован только в служебных целях. <P>
</BODY>
</HTML>
```

Вот, что вы увидите на экране броузера:

Список сотрудников нашей фирмы

Составлено : 30 июля 1996 года

Данный список содержит фамилии, имена и отчества всех сотрудников нашей компании.

Список может быть использован только в служебных целях.

> **Внимание!** Заголовок "Список сотрудников" не отображен броузером как часть документа. Он появится в заголовке окна броузера.

**Разрыв строки
**

Тэг
 извещает броузер о разрыве строки. Наилучший пример использования данного тэга - форматированный адрес или любая другая последовательность строк, где броузер должен отображать их одну под другой. Например:

```
Алексей                                 Ярцев                                   <BR>
Дмитровское                               шоссе,                                <BR>
д.9Б, офис 326 <BR>
```

Дополнительный параметр позволяет расширить вохможности тэга
.

```
<BR CLEAR=left|right|all >
```

Данный параметр позволяет выполнить не просто перевод строки, а разместить следующую строку, начиная с чистой левой (left), правой (right) или обоих (all) границ окна броузера. Например, если рядом с текстом слева встречается рисунок, то можно использовать тэг
 для смещения текста ниже рисунка:

```
<p> Как вы можете видеть, данная схема демонстрирует связь<BR CLEAR=left>
<img    src="http://www.softexpress.com/images/schema1.gif"    align=baseline>
Мастер/Деталь </p>
```

Неразрывная строка <NOBR>

Если вы не хотите, чтобы броузер автоматически переносил строку, то вы можете обозначить ее тэгами <NOBR> и </NOBR>. В этом случае броузер не будет переносить строку даже если она выходит за нраницы экрана; вместо этого броузер позволит горизонтально прокручивать окно. Например:

```
<NOBR> Данная строка является самой длинной строкой документа, которая не допускает какого-либо разбиения где бы то ни было </NOBR>
```

Если же вы хотите все же позволить разбиение данной строки на две, но в строго запланированном месте, то вставьте тэг <WBR> в это место. Например:

```
<NOBR> Данная строка является самой длинной строкой документа,<WBR> которая не допускает какого-либо разбиения где бы то ни было </NOBR>
```

```
Данная строка является самой длинной строкой документа,
которая не допускает какого-либо разбиения где бы то ни было
```

Цитата <BLOCKQUOTE>

Данный тэг предназначен для обозначения в документе цитаты из другого источника. Текст, обозначенный тэгом <BLOCKQUOTE>, отступает от левого края документа на 8 пробелов. Например:

```
На открытии данной конференции глава представительства произнес: <P>
<BLOCKQUOTE>
Сегодня один из величайших дней для нашей компании. <BR>
Мы открыли новую технологию, позволяющую нашим клиентам повысить производительность их настольных систем в несколько раз.
</BLOCKQUOTE>
```

При отображении броузером данный текст будет выглядеть так:

На открытии данной конференции глава представительства произнес:

Сегодня один из величайших дней для нашей компании.
Мы открыли новую технологию, позволяющую нашим клиентам повысить производительность их настольных систем в несколько раз.

Список базовых тэгов HTML

Стартовый	Завершающий	Описание
<HTML>	</HTML>	Обозначение HTML-документа
<HEAD>	</HEAD>	Заголовочная часть документа
<TITLE>	</TITLE>	Заголовок документа
<BODY>	</BODY>	Тело документа
<H1>	</H1>	Заголовок абзаца первого уровня
<H2>	</H2>	Заголовок абзаца второго уровня

<H3>	</H3>	Заголовок абзаца третьего уровня
<H4>	</H4>	Заголовок абзаца четвертого уровня
<H5>	</H5>	Заголовок абзаца пятого уровня
<H6>	</H6>	Заголовок абзаца шестого уровня
<P>	</P>	Абзац
<PRE>	</PRE>	Форматированный текст
 		Перевод строки без конца абзаца
<BLOCKQUOTE>	</BLOCKQUOTE>	Цитата

Описанные ранее тэги - это все, что необходимо вам для того, чтобы начать работать с HTML.
С использованием описанных тэгов вы можете создать простой HTML-документ. Однако, следующие разделы позволят вам существенно улучшить внешний вид ваших документов и опишут новые возможности HTML.

Тэги списков

Существует три основных вида списков в HTML-документе:

- **пронуменрованный**
- **непронуменрованный**
- **список описаний**

Вы можете создавать вложенные списки, используя различные тэги списков или повторяя одни внутри других. Для этого просто необходимо разместить одну пару тэгов (стартовый и завершающий) внутри другой. Будут ли элементы вложенного списка иметь те же маркеры, обозначающие элемент списка - зависит от броузера. Более подробно смотри в разделе "Вложенные списки".

Пронумерованные списки

В пронумерованном списке броузер автоматически вставляет номера элементов по порядку. Это означает, что если вы удалите один или несколько элементов пронумерованного списка, то остальные номера автоматически будут пересчитаны.

Пронумерованный список начинается стартовым тэгом <OL> и завершается тэгом </OL>. Каждый элемент списка начинается с тэга <LI>. Например:

```
<OL>
<LI>                                                    Программирование
<LI>                                                      Алгоритмизация
<LI>                                                      Проектирование
</OL>
```

1. Программирование
2. Алгоритмизация

3. Проектирование

Тэг <OL> может иметь параметры:

```
<OL TYPE=A|a|I|i|1 START=n>
```

где:

TYPE

Вид счетчика:

- A - большие латинские буквы (A,B,C...)
- a - маленькие латинские буквы (a,b,c...)
- I - большие римские цифры (I,II,III...)
- i - маленькие римские цифры (i,ii,iii...)
- 1 - обычные цифры (1,2,3...)

START=n

Число, с которого начинается отсчет

Например:

```
<OL                          TYPE=I                          START=15>
<LI>                                                  Программирование
<LI>                                                    Алгоритмизация
<LI>                                                    Проектирование
</OL>
```

XV. Программирование
XVI. Алгоритмизация
XVII. Проектирование

Непронумерованные списки.

Для непронумерованных списков броузер обычно использует маркеры для пометки элемента списка. Вид маркера, как правило, настраивает пользователь броузера.

Пронумерованный список начинается стартовым тэгом <UL> и завершается тэгом </UL>. Каждый элемент списка начинается с тэга <LI>. Например:

```
<UL>
<LI>                                                  Программирование
<LI>                                                    Алгоритмизация
<LI>                                                    Проектирование
</UL>
```

- Программирование
- Алгоритмизация
- Проектирование

Тэг <UL> может иметь параметр:

```
<UL TYPE=disc|circle|square>
```

Тип тэга <UL> определяет внешний вид маркера как вид по умолчанию (disc), круглый (circle) или квадратный (square). Например:

```
<UL                                                    TYPE=square>
<LI>                                                Программирование
<LI>                                                  Алгоритмизация
<LI>                                                  Проектирование
</UL>
```

- Программирование
- Алгоритмизация
- Проектирование

Вложенные списки

Дадим пример вложенных списков:

```
<HTML>
<HEAD>
<TITLE>                Список                сотрудников                </TITLE>
</HEAD>
<BODY>
<H2>         Список         сотрудников         нашей         фирмы         </H2>
<H3>       Составлено       :       30       июля       1996       года       </H3>
Данный список содержит фамилии, имена и отчества всех сотрудников нашей
компании.                                                                  <P>
Список может быть использован только в служебных целях.   <P>
<OL>
<LI>                                                                 Дирекция
<UL>
<LI>                                  Иванов                                И.И.
<LI>                                  Петров                                К.В.
</UL>
<LI>                                Отдел                              маркетинга
<UL>
<LI>                                Варшавская                            Е.Л.
<LI>                                 Самсонов                             Д.М.
</UL>
</OL>
</BODY>
</HTML>
```

Вот, что вы увидите на экране броузера:

Список сотрудников нашей фирмы

Составлено : 30 июля 1996 года

Данный список содержит фамилии, имена и отчества всех сотрудников нашей компании.

Список может быть использован только в служебных целях.

1. Дирекция
   - Иванов И.И.
   - Петров К.В.
2. Отдел маркетинга
   - Варшавская Е.Л.
   - Самсонов Д.М.

Элемент списка <LI>

Тэг <LI> может иметь параметры:

```
<OL TYPE=disc|circle|squade> или <OL TYPE=A|a|I|i|1 VALUE=n>
```

в зависимости от того, в списке какого вида находится данный элемент.

TYPE

Вид маркера (см. <UL>) или счетчика (см.OL)

VALUE=n

Значение для элемента пронумерованного списка (его номер). Все дальнейшие номера элементов списка будут отсчитываться от этого номера.

Например:

```
<OL                              TYPE=I                              START=15>
<LI>                                                          Программирование
<LI              TYPE=i                VALUE=25>                Алгоритмизация
<LI>                                                          Проектирование
</OL>
```

XV. Программирование
XVI. Алгоритмизация
XVII. Проектирование

Список определений

Список определений начинается с тэга <DL> и завершается тэгом </DL>. Данный список служит для создание списков типа "термин"-"описание". Каждый термин начинается тэгом <DT> , а описание - тэгом <DD>. Например:

```
<DL>
<DT>            <B>            Отдел            маркетинга            </B>
<DD>  Данный  отдел  занимается  продвижением  продуктов  и  услуг
<DT>            <B>            Финансовый            отдел            </B>
<DD>  Данный  отдел  занимается  всеми  финансовыми  операциями
<DT>            <B>            Отдел            кадров            </B>
```

```
<DD> Данный отдел занимается учетом и набором новых сотрудников фирмы,
распределением отпусков, отслеживанием больничных листов и т.д.
</DL>
```

Отдел маркетинга
: Данный отдел занимается продвижением продуктов и услуг

Финансовый отдел
: Данный отдел занимается всеми финансовыми операциями

Отдел кадров
: Данный отдел занимается учетом и набором новых сотрудников фирмы, распределением отпусков, отслеживанием больничных листов и т.д.

Гипертекстовые ссылки

Гипертекстовые ссылки являются ключевым компонентом, делающим WEB привлекательным для пользователей. Добавляя гипертекстовые ссылки (далее - ссылки), вы делаете набор документов связанным и структурированным, что позволяет пользователю получать необходимую ему информацию максимально быстро и удобно.

Ссылки имеют стандартный формат, что позволяет броузеру интерпретировать их и выполнять необходимые функции (вызывать методы) в зависимости от типа ссылки. Ссылки могут указывать на другой документ, специальное место данного документа или выполнять другие функции, например запрашивать файл по FTP-протоколу для отображения его броузером. URL может указывать на специальное место по абсолютному пути доступа, или указывать на документ в текущем пути доступа, что часто используется при организации больших структурированных WEB-сайтов.

> **Внимание!** Вы можете использовать ссылки как для перемещения по документу, так и для перемещения от одного документа к другому. Однако, HTML не поддерживает возврат на предыдущую ссылку, если перемещение происходило внутри документа. Если вы используете ссылки внутри документа, а затем нажимаете на клавишу Back, то вы не перейдете на предыдущую ссылку, а вернетесь на ту часть документа, которую вы просматривали до этого.

URL

HTML использует URL (Uniform Resource Locator) для представления гипертекстовых ссылок и ссылок на сетевые сервисы внутри HTML-документа. Первая часть URL (до двоеточия) описывает метод доступа или сетевой сервис. Другая часть URL (после двоеточия) интерпретируется в зависимости от метода доступа. Обычно, два прямых слэша после двоеточия обозначают имя машины:

```
method://machine-name/path/foo.php
```

Следующий пример представляет собой вызов HTML-документа `index.php` с сервера `www.softexpress.com` с использованием HTTP протокола:

```
http://www.softexpress.com/index.php
```

Uniform Resource Locator имеет следующий формат:

method://servername:port/pathname#anchor

Опишем каждый из компонентов URL:

METHOD

Имя операции, которая будет выполняться при интерпретации данного URL. Наиболее часто используемые методы:

file:

чтение файла с локального диска. Имя файла интерпретируется для локальной машины пользователя. Данный метод используется для отображения какого-либо файла, находящегося на машине пользователя. Например:
`file:/home/alex/index.php` - отобажает файл `index.php` из каталога `/home/alex` на пользовательской машине

http:

доступ к WEB-странице в сети с использованием HTTP-протокола. (Это наиболее часто используемый метод доступа к какому-либо HTML-документу в сети). Например: `http://www.softexpress.com/` - доступ к Home-странице компании SoftExpress

ftp:

запрос файла с анонимного FTP-сервера. Например: `ftp://hostname/directory/filename`

mailto:

активизирует почтовую сессию с указанным пользователем и хостом. Например:
`mailto:info@softexpress.com` - активизирует сессию посылки сообщения пользователю `info` на машине softexpress.com, если броузер поддерживает запуск электронной почты. Заметьте, что метод `mailto:` не требует указание слэшей после двоеточия (как правило, после двоеточия сразу идет электронный адрес абонента)

telnet:

обращение к службе telnet

news:

вызов службы новостей, если броузер ее поддерживает. Например: `news:relcom.www.support`

SERVERNAME

Необязательный параметр, описывающий полное сетевое имя машины. Например:

`www.softexpress.com` - полное сетевое имя сервера фирмы СофтСервис.

Если имя сервера не указано, то ссылка считается локальной, и полный путь, указанный далее в URL вычисляется на той машине, с которой взят HTML-документ, содержащий данную ссылку. Вместо символьного имени машины может быть использован IP-адрес, однако это не рекомендуется из-за возможного пересечения с фиксированными локальными адресами внутренней сети.

PORT

Номер порта TCP на котором функционирует WEB-сервер. Если порт не указан, то "по умолчанию" используется порт 80. Данный параметр (port) не используется в подавляющем большинстве URL.

PATHNAME

Частичный или полный путь к документу, который должен вызваться в результате интерпретации URL. Различные WEB-сервера сконфигурированы по разному для интерпретации пути доступа к документу. Например, при использовании CGI скриптов (исполняемых программ), они обычно собираются в одном или нескольких выделенных каталогах, путь к которым записан в специальных параметрах WEB-сервера. Для данных каталогов WEB-сервером выделяется специальный логический путь, который и используется в URL. Если WEB-сервер видит данный путь, то запрашиваемый файл интерпретируется как исполняемый модуль. В противном случае, запрашиваемый файл интерпретируется просто как файл данных, даже если он является исполняемым модулем. Например:

```
http://www.softexpress.com/cgi-win/handle.exe
```

В данном примере HTTP-сервер должен вызвать CGI-скрипт с именем `handle.exe`, который находится на машине с сетевым именем `www.softexpress.com`. Путь к данному скрипту - `/cgi-win/` - в действительности является виртуальным путем (выделенным сервером для исполняемых модулей). Заметьте, что при описании пути используется UNIX-подобный синтаксис, где, в отличии от DOS и Windows используются прямые слэши вместо обратных. Если после сетевого имени машины сразу идет имя документа, то он должен находиться в корневом каталоге на удаленной машине или (что чаще) в каталоге, выделенном WEB-сервером в качестве корневого. Если же URL закагчивается сетевым именем машины, то в качестве документа запрашивается документ из корневого каталога удаленной машины с именем, установленным в настройках WEB-сервера (как правило, это `index.php`).

#ANCHOR

Данный элемент является ссылкой на строку (точку) внутри HTML-документа. Большинство броузеров, встречая после имени документа данный элемент, размещают документ на экране таким образом, что указанная строка документа помещается в верхнюю строку рабочего окна броузера. Точки, на которые ссылается #anchor, указываются в документе при помощи тэга NAME, как это будет описано далее.

Структура ссылок в HTML-документе

Пока что мы рассмотрели только внешний вид URL. Для того, чтобы броузер отобразил ссылку на URL, необходимо отчетить URL специальными тэгами в HTML-документе. Синтаксис HTML, позволяющий жто сделать - следующий:

```
<A HREF="URL"> текст-который-будет-подсвечен-как-ссылка </A>
```

Тэг **<A HREF="*URL*">**открывает описание ссылки, а тэг </A> - закрывает его. Любой текст, находящийся между данными двумя тэгами подсвечивается специальным образом Web-броузером. Обычно этот текст отображается подчеркнутым и выделенным синим (или другим заданным пользователем) цветом. Текст, обозначающий URL, не отображается броузером, а используется только для выполнения предписанных им действий при активизации ссылки (обычно при щелчке мыши на подсвеченном или подчеркнутом тексте).

Ссылки на точки внутри документа

Вы можете делать ссылки на различные участки или разделы одного и того же документа, используя специальных скрытый маркер для этих разделов. Это позволяет быстро переходить от раздела к разделу внутри документа, не используя скроллирование экрана. Как только вы щелкнете на ссылке, броузер переместит вас на указанный раздел документа, а строка, в которой стоит маркер данного раздела (обычно, первая строка раздела или заголовок раздела) будет ращзмещена на первой строке окна броузера (если данная строка не присутствует уже на экране броузера).

Для создания такой ссылки необходимо выполнить следующие шаги:

1. Создайте маркер раздела. Синтаксис данного маркера следующий:

```
<A NAME="named_anchor"> Текст-который-отобразится-в-первой-строке-броузера
</A>
```

2. Создайте ссылку на данный маркер:

```
<A HREF="#named_anchor"> Текст </A>
```

Например:

```
<p><b>Список разделов</b></p>
<ul> <li><a href="#ex1">Раздел 1</a></li>
<li><a href="#ex2">Раздел 2</a></li> </ul>
<p><a name="ex1"></a>Раздел 1</p>
<ul> <p>Текст раздела 1</p> </ul>
<p><a name="ex2"></a>Раздел 2</p>
<ul> <p>Текст раздела 2 <br></p>
```

Список разделов

- Раздел 1
- Раздел 2

Раздел 1

Текст раздела 1

Раздел 2

Текст раздела 2

Символы "#ex1" сообщает вашему броузеру, что необходимо найти в данном HTML-документе маркер с именем "ex1".

Когда пользователь щелкнет мышью на строке "Раздел 1", броузер перейдет сразу к разделу 1.

> **Внимание!** Как ранее было показано в синтаксисе URL, маркер раздела может быть поставлен как в том же документе, который просматривается в текущий момент, так и в другом документе. Во втором случае броузер осуществит подгрузку другого документа и перейдет к указанному для него разделу.

Графика внутри HTML-документа

Одна из наиболее привлекательных черт Web - возможность включения ссылок на графические и иные типы данных в HTML-документ. Делается это при помощи тэга <IMG...ISMAP>. Использование данного тэга позволяет значительно улучшить внешний вид и функциональность документов.

Существует два способа использования графики в HTML-документах. Первый - это внедрение графических образов в документ, что позволяет пользователю видеть изображения непосредственно в контексте других элементов документа. Это наиболее используемая техника при проектировании документов, называемая иногда "inline image". Синтаксис тэга:

```
<IMG SRC="URL" ALT="text" HEIGHT=n1 WIDTH=n2 ALIGN=top|middle|bottom|texttop
ISMAP>
```

Опишем элементы синтаксиса тэга:

URL

> Обязательный параметр, имеющий такой же синтаксис, как и стандартный URL. Данный URL указывает броузеру где находится рисунок. Рисунок должен храниться в графическом формате, поддерживаемом броузером. На сегодняшний день форматы GIF и JPG поддерживаются большинством броузеров.

ALT="text"

> Данный необязательный элемент задает текст, который будет отображен броузером, не поддерживающим отображение графики или с отключенной подкачкой изображений. Обычно, это короткое описание изображения, которое пользователь мог бы или сможет увидеть на экране. Если данный параметр отсутствует, то на месте рисунка большинство броузеров выводит пиктограмму (иконку), активизировав которую, пользователь может увидеть изображение. Тэг

ALT рекомендуется, если ваши пользователи используют броузер, не поддерживающий графический режим, например Lynx.

HEIGTH=n1

Данный необязательный параметр используется для указания высоты рисунка в пикселах. Если данный параметр не указан, то используется оригинальная высота рисунка. Это параметр позволяет сжимать или растягивать изображения по вертикали, что позволяет более четко определять внешний вид документа. Однако, некоторые броузеры не поддерживают данный параметр. С другой стороны, экранное разрешение у вашего клиента может отличаться от вашего, поэтому будьте внимательны при задании абсолютной величины графического объекта.

WIDTH=n2

Параметр также необязателен, как и предыдущий. Позволяет задать абсолютную ширину рисунка в пикселах.

ALIGN

Данный параметр используется, чтобы сообщить броузеру, куда поместить следующий блок текста. Это позволяет более строго задать расположение элементов на экране. Если данный параметр не используется, то большинство броузеров располагает изображение в левой части экрана, а текст справа от него.

ISMAP

Этот параметр сообщает броузеру, что данное изображение позволяет пользователю выполнять какие-либо действия, щелкая мышью на определенном месте изображения. Данная возможность является расширением HTML и будет обсуждена нами позже.

Приведем пример использования данного тэга:

```
<IMG SRC="http://www.softexpress.com/images/nekton.jpg" ALT="СофтСервис лого"
ALIGN="top" ISMAP>
```

С версии HTML 2.0 у тэга <IMG> появились дополнительные параметры:

```
<IMG SRC="URL" ALT="text" HEIGHT=n1 WIDTH=n2 ALIGN=top|middle|bottom|texttop|
absmiddle|baseline|absbottom BORDER=n3 VSPACE=n4 HSPACE=n5 ISMAP>
```

Новые параметры:

BORDER

Данный параметр позволяет автору определить ширину рамки вокруг рисунка.

VSPACE

Позволяет установить размер в пикселах пустого пространства над и под рисунком, чтобы текст не наезжал на рисунок. Особенно это важно для динамически формируемых изображений, когда нельзя заранее увидеть документ.

HSPACE

То же самое, что и VSPACE, но только по горизантали.

Фоновые рисунки

Большинство броузеров позволяет включать в документ фоновый рисунок, который будет матрицироваться и отображаться на фоне всего документа. Некоторые пользователи любят фоновую графику, некоторые нет. Ненавязчивый полупрозрачный рисунок (обои) обычно хорошо выглядит в качестве фона для большинства документов.

Описание фонового рисунка включается в тэг BODY и выглядит следующим образом:

```
<BODY BACKGROUND="picture.gif">
```

Задание стандартных цветов

Многие HTML-авторы любят использовать заранее предопределенные цвета фона документа, обычного текста и ссылок. Чтобы задать эти цвета, необходимо включить в тэг <BODY> дополнительные параметры:

```
<BODY BGCOLOR="#XXXXXX" TEXT="#XXXXXX" LINK="#XXXXXX">
```

где каждый из параметров определяет цвет того или иного элемента. Опишем эти параметры:

BGCOLOR

Цвет фона документа

TEXT

Цвет простого текста документа

LINK

Цвет ссылки

Цвет задается шестизначным числом в шестнадцатиричном формате по схеме RGB (Red, Green, Blue). Цвет #000000 соответствует черному, а цвет #FFFFFF - белому. Например:

```
<BODY BGCOLOR="#000000" TEXT="#FFFFFF" LINK="#9690CC">
```

Данная строка определяет белый цвет фона документа, черный текст и серебристые ссылки.

Горизонтальная линия

Используя тэг <HR> вы можете разделить текст горизонтальной чертой.

Формат тэга:

<HR SIZE=number WIDTH=*number*|*percent* ALIGN=*left*|*right*|*center* NOSHADE**>**

Параметры тэга:

SIZE

Толщина линии в пикселах.

WIDTH

Ширина линии в пикселах или процентах от ширина окна броузера.

ALIGN

Расположение на экране (слева | по центру | справа).

NOSHADE

По умолчанию линия представлена в 3D виде с тенью. NOSHADE позволяет представить линию просто однотонной темной полоской.

Добавление стилей в HTML документ

HTML позволяет использовать различные стили шрифтов для выделения текстовой информации в ваших документах. Вот короткий список стилей, поддерживаемых большинством броузеров:

- bold (жирный)
- italic (наклонный)
- mono spaced (type writer - с использованием фиксированных шрифтов)

Вы можете комбинировать различные виды стилей, например жирный и наклонный.

Стиль	***Элемент или тэг***	***Результат***
Bold	<B> Этот текст жирный </B>	**Этот текст жирный**
Italic	<I> Этот текст наклонный </I>	*Этот текст наклонный*
Mono spaced	<TT> Этот текст с непроп. шрифтом </TT>	`Этот текст с непроп. шрифтом`

Комбинирование стилей позволяет вам отображать в одной строке несколько элементов различными стилями, например:

```
<b>Жизнь</b> - <i>это <b>песня!</b></i>
```

Жизнь - *это **песня!***

Внимание! Добавление большого количества стилей и их комбинаций приводит к затруднению чтения текста!

Дополнительные стили:

- big (юольшой)
- small (маленький)
- sub (подстрочник)
- sup (надстрочник)

Стиль	***Элемент или тэг***	***Результат***
Big	Этот текст <BIG> большой </BIG>	Этот текст большой
Small	Этот текст <SMALL> маленький </SMALL>	Этот текст маленький
Sub	Этот текст _{подстрочник}	Этот текст подстрочник
Sup	Этот текст ^{надстрочник}	Этот текст надстрочник

Размер шрифта <FONT SIZE>

Вы можете изменять размер шрифта при помощи тэга:

```
<FONT SIZE=+|- n>
```

Шрифт может иметь размер от 1 до 7. Вы можете прямо указать размер шрифта цифрой, или указать смещение относительно базового значения (по умолчанию - 3) в положительную или отрицательную сторону. Базовое значение можно изменить при помощи тэга:

```
<BASEFONT SIZE=n>
```

Например:

```
<p>и
<font                SIZE=+1>з</font><font                SIZE=+2>м</font>
<font                SIZE=+3>е</font><font                SIZE=+4>н</font>
<font                SIZE=+3>е</font><font                SIZE=+2>н</font>
<font                                                     SIZE=+1>и</font>
е</p>
```

и змеНени е

Цвет шрифта <FONT SIZE>

Вы можете изменить цвет шрифта при помощи тэга:

```
<FONT COLOR="#xxxxxx>
```

Цвет указывается в RGB-формате (Red-Green-Blue) посредством указания размерности каждой компоненты цвета в шестнадцатиричном формате. Например, белый цвет обозначается "000000", черный - "FFFFFF", синий - "0000FF" и т.п.

```
<FONT                                                    COLOR="#FF0000">
Красный                                                          </FONT>
<FONT                                                    COLOR="#00FF00">
Зеленый                                                          </FONT>
<FONT                                                    COLOR="#0000FF">
Синий </FONT>
```

Красный Зеленый Синий

Специальные тэги HTML

Следующие тэги позволят вам сделать ваш HTML-документ более функциональным.

Тэг адреса <ADDRESS>

Тэг <ADDRESS> используется для выделения автора документа и его дреса (например, e-mail). Синтаксис:

```
<ADDRESS> Адрес-автора </ADDRESS>
```

Escape-последовательности

Некоторые символы являются управляющими символами в HTML и немогут напрямую использоваться в документе:

- левая угловая скобка "<"
- правая угловая скобка ">"
- амперсанд "&"
- двойные кавычки """

Чтобы использовать данные символы в документе, необходимо заменить их escape-последовательностями:

<	**<**
>	**>**
&	**&**
"	**"**

Существует большое количество escape-последовательностей для обозначения специальных символов, например "©" для обозначения знака ┗ и ® для значка ┛, появившихся в HTML 2.0. Одной из особенностей является замена символов во 2-ой части символьной таблицы (после 127-ого символа) на escape-последовательности для передачи текстовых файлов с национальными языками по 7-битным каналам.

> **Внимание!** Escape-последовательности чувствительны к регистру: НЕЛЬЗЯ использовать < вместо <.

HTML формы

Некоторые WWW browser позволяют пользователю, заполнив специальную форму, возвращающую полученное значение, выполнять некоторые действия на вашем WWW-сервере. Когда форма интерпретируется WEB-броузером, создается специальные экранные элементы GUI, такие, как поля ввода, checkboxes, radiobuttons, выпадающие меню, скроллируемые списки, кнопки и т.д. Когда пользователь заполняет форму и нажимает кнопку "Подтверждение" (SUBMIT - специальный тип кнопки, который задается при описании документа), информация, введенна пользователем в форму, посылается HTTP-серверу для обработки и передаче другим программам, работающим под сервером, в соответствии с CGI (Common Gateway Interface) интерфейсом.

Когда вы описываете форму, каждый элемент ввода данных имеет тэг <INPUT>. Когда пользователь помещает данные в элемент формы, инфоромация размещается в разделе VALUE данного элемента.

Синтаксис форм

Все формы начинаются тэгом <FORM> и звершаются тэгом </FORM>.

```
 <FORM                    METHOD="get|post"                    ACTION="URL">
Элементы_формы_и_другие_элементы_HTML</FORM>
```

METHOD

Метод посылки сообщения с данными из формы. В зависимости от используемого метода вы можете посылать результаты ввода данных в форму двумя путями:

- **GET**: Информация из формы добавляется в конец URL, который был указан в описании заголовка формы. Ваша CGI-программа (CGI-скрипт) получает данные из формы в виде параметра переменной среды QUERY_STRING. Использование метода GET не рекомендуется.
- **POST**: Данный метод передает всю информацию о форме немедленно после обращения к указанному URL. Ваша CGI-программа получает данные из формы в стандартный поток ввода. Сервер не будет пересылать вам сообщение об окончании пересылки данных в стандартный поток ввода; вместо этого используется переменная среды CONTENT_LENGTH для определения, какое количество данных вам необходимо считать из стандартного потока ввода. Данный метод рекомендуется к использованию.

ACTION

ACTION описывает URL, который будет вызываться для обработки формы. Данный URL почти всегда указывает на CGI-программу, обрабатывающую данную форму.

Тэги формы

TEXTAREA

Тэг <TEXTAREA> используется для того, чтобы позволить пользователю вводить более одной строки информации (свободный текст). Вот пример использовани тэга <TEXTAREA>:

```
 <TEXTAREA NAME="address" ROWS=10 COLS=50>
Москва,
Дмитровкое шоссе,
д.9Б, офис 448
</TEXTAREA>
```

Атрибуты, используемые внутри тэга <TEXTAREA> описывают внешний вид и имя вводимого значения. Тэг </TEXTAREA> необходим даже тогда, когда поле ввода изначально пустое. Описание атрибутов:

- **NAME** - имя поля ввода
- **ROWS** - высота поля ввода в символах
- **COLS** - ширина поля ввода в символах

Если вы хотите, чтобы в поле ввода по умолчанию выдавался какой-либо текст, то необходимо вставить его внутри тэгов <TEXTAREA> и </TEXTAREA>.

INPUT

Тэг <INPUT> используется для ввода одной строки текста или одного слова. Атрибуты тэга:

- **CHECKED -** означает, что CHECKBOX или RADIOBUTTON будет выбран.
- **MAXLENGTH** - определяет количество символов, которое пользователи могут ввести в поле ввода. При превышении количества допустимых символов броузер реагирует на попытку ввода нового символа звуковым сигналом и не дает его ввести. Не путать с атрибутом SIZE. Если MAXLENGTH больше чем SIZE, то в поле осуществляется скроллинг. По умолчанию значение MAXLENGTH равно бесконечности.
- **NAME** - имя поля ввода. Данное имя используется как уникальный идентификатор поля, по которому, впоследствии, вы сможете получить данные, помещенные пользователем в это поле.
- **SIZE** - определяет визуальный размер поля ввода на экране в символах.
- **SRC** - URL,. указывающий на картинку (используется совместно с атрибутом IMAGE).
- **TYPE** - определяет тип поля ввода. По умолчанию это простое поле ввода для одной строки текста. Остальные типы должны быть явно указаны:

  CHECKBOX
  Используется для простых логических (BOOLEAN) значений. Значение, ассоциированное с именем данного поля, которое будет передаваться в вызываемую CGI-программу, может принимать значение ON или OFF.

  HIDDEN
  Поля данного типа не отображаются броузером и не дают пользователю

изменять присвоенные данному полю по умолчанию значение. Это поле используетс для передачи в CGI-программу статической информации, как то ID прользователя, пароля или другой информации.

IMAGE
Данный тип поля ввода позволяет вам связывать графический рисунок с именем поля. При нажатии мышью на какую-либо часть рисунка будет немедленно вызвана ассоциированная форме CGI-программа. Значения, присвоенные переменной NAME будут выглядеть так - создается две новых переменных: первая имеет имя, обозначенное в поле NAME с добавлением .x в конце имени. В эту переменную будет помещена X-координата точки в пикселах (считая началом координат левый верхний угол рисунка), на которую указывал курсор мыши в момент нажатия, а переменная с именем, содержащимся в NAME и добавленным .y, будет содержать Y-координату. Все значения атрибута VALUE игнорируются. Само описание картинки осуществляется через атрибут SRC и по синтаксису совпадает с тэгом <IMG>.

PASSWORD
То же самое, что и атрибут TEXT, но вводимое пользователем значение не отображается броузером на экране.
RADIO
Данный атрибут позволяет вводить одно значение из нескольких альтернатив. Для создания набора альтернатив вам необходимо создать несколько полей ввода с атрибутом TYPE="RADIO" с разными значениями атрибута VALUE, но с одинаковыми значениями атрибута NAME. В CGI-программу будет передано значение типа NAME=VALUE, причем VALUE примет значение атрибута VALUE того поля ввода, которое в данный момент будет выбрано (будет активным). При выборе одного из полей ввода типа RADIO все остальные поля данного типа с тем же именем (атрибут NAME) автоматически станут невыбранными на экране.
RESET
Данный тип обозначает кнопку, при нажатии которой все поля формы примут значения, описанные для них по умолчанию.
SUBMIT
Данный тип обозначает кнопку, при нажатии которой будет вызвана CGI-программа (или URL), описанная в заголовке формы. Атрибут VALUE может содержать строку, которая будет высвечена на кнопке.
TEXT
Данный тип поля ввода описывает однострочное поле ввода. Используйте атрибуты MAXLENGTH и SIZE для определения максимальной длинны вводимого значения в символах и размера отображаемого поля ввода на экране (по умолчанию принимается 20 символов).

- **VALUE** - присваивает полю значение по умолчанию или значение, которое будет выбрано при использовании типа RADIO (для типа RADIO данный атрибут обязателен)

Меню выбора в формах

Под меню выбора в формах понимают такой элемент интерфейса, как LISTBOX. Существует три типа тэгов меню выбора для форм:

- **Select** - пользователь выбирает одно значение из фиксированного списка значений, представленных тэгами OPTION. Данй вид представляется как выпадающий LISTBOX.
- **Select single** - то же самое, что и Select, но на экране пользователь видит одновременно три элемента выбора. Если их больше, то предоставляется автоматический вертикальный скроллинг.
- **Select multiple** - позволяет выбрать несколько элементов из LISTBOX.

SELECT

Тэг SELECT позволяет пользователю выбрать значение из фиксированного списка значений. Обычно это представлено выпадающим меню.

Тэг SELECT имеет один или более параметр пежду стартовым тэгом <SELECT> и завершающим </SELECT>. По умолчанию, первый элемент отображается в строке выбора. Вот пример тэга <SELECT>:

```
 <FORM>
<SELECT NAME=group>
<OPTION> AT 386
<OPTION> AT 486
<OPTION> AT 586
</SELECT>
</FORM>
```

SELECT SINGLE

Тэг SELECT SINGLE - это то же самое, что и Select, но на экране пользователь видит одновременно несколько элементов выбора (три по умолчанию). Если их больше, то предоставляется автоматический вертикальный скроллинг. Количество одновременно отображаемых элементов определяется атрибутом SIZE. Пример:

```
 <FORM>
<SELECT SINGLE NAME=group SIZE=4>
<OPTION> AT 386
<OPTION> AT 486
<OPTION> AT 586
<OPTIONS> Pentium PRO
</SELECT>
</FORM>
```

SELECT MULTIPLE

Тэг SELECT MULTIPLE похож на тэг SELECT SINGLE, но пользователь может одновременно выбрать более чем один элемент списка. Атрибут SIZE определяет количество одновременно видимых на экране элементов, атрибут MULTIPLE - максимальное количество одновременно выбранных элементов. Пример:

```
 <FORM>
<SELECT SINGLE NAME=group SIZE=4 MULTIPLE=2>
<OPTION> AT 386
<OPTION> AT 486
<OPTION> AT 586
<OPTIONS> Pentium PRO
</SELECT>
</FORM>
```

Если выбрано одновременно несколько значений, то серверу передаютс соответствующее выбранному количество параметров NAME=VALUE с одинаковыми значениями NAME, но разными VALUE.

Отправление файлов при помощи форм

Формы можно использовать для отправки не только небольших информационных сообщений ввиде параметров, а также и для отправки файлов.

> **Внимание!** Поскольку данная возможность требует поддержки получения файлов WEB-сервером, то, соответственно, необходимо, чтобы сервер поддерживал получение файлов!

Например:

```
<FORM ENCTYPE="multipart/form-data" ACTION="url" METHOD=POST>
Отправить данный файл: <INPUT NAME="userfile" TYPE="file">
<P>
<INPUT TYPE="submit" VALUE="Отправить файл">
</FORM>
```

Отправить данный файл:

Отправить файл

HTML фреймы

Испольльзуя фрэймы, позволяющие разбивать Web-страницы на множественные скроллируемые подокна, вы можете значительно улучшить внешний вид и функциональность информационных систем и Web-приложений. Каждое подокно, или фрэйм, может иметь следующие свойства:

- Каждый фрэйм имеет свой URL, что позволяет загружать его независимо от других фрэймов
- Каждый фрэйм имеет собственое имя (параметр NAME), позволяющее переходить к нему из другого фрэйма
- Размер фрэйма может быть изменен пользователем прямо на экране при помощи мыши (если это не запрешено указанием специального параметра)

Данные свойства фрэймов позволяют создавать продвинутые интерфейсные решения, такие как:

- Размещение статической информации, которую автор считает необходимым постоянно показывать пользователю, в одном статическом фрэйме. Это может быть графический логотип фирмы, copyright, набор управляющих кнопок
- Помещение в статическом фрэйме оглавления всех или части WEB-документов, содержащихся на WEB-сервере, что позволяет пользователю быстро находить интересующую его информацию

- Создавать окна результатов запросов, когда в одном фрэйме находится собственно запрос, а в другом результаты запроса
- Создавать формы типа "мастер-деталь" для WEB-приложений, обслуживающих базы данных

Синтаксис фреймов

Формат документа, использующего фрэймы, внешне очень напоминает формат обычного документа, только вместо тэга BODY используется контейнер FRAMESET, содержащий описание внутренних HTML-документов, содержащий собственно информацию, размещаемую во фрэймах.

```
<HTML>
<HEAD>...</HEAD>
<FRAMESET>...</FRAMESET>
</HTML>
```

Однако, фрэйм-документ является специфичным видом HTML-документа, поскольку не содержит элемента BODY и какой-либо информационной нагрузки соответственно. Он описывает только фрэймы, которые будут содержать информацию (кроме случая двойного документа, который мы рассмотрим позже).

Представим общий синтаксис фрэймов:

```
<FRAMESET COLS="value" | ROWS="value">

      <FRAME                                                    SRC="url1">
      <FRAME                                                          ...>
      ...

</FRAMESET>
```

Общий контэйнер FRAMESET описывает все фрэймы, на которые делится экран. Вы можете разделить экран на несколько вертикальных или несколько горизонтальных фрэймов. Тэг FRAME описывает каждый фрэйм в отдельности. Рассмотрим более детально каждый компонент.

FRAMESET

```
<FRAMESET [COLS="value" | ROWS="value"]>
```

Тэг <FRAMESET> имеет завершающий тэг </FRAMESET>. Все, что может находиться между этими двумя тэгами, это тэг <FRAME>, вложенные тэги <FRAMESET> и </FRAMESET>, а также контейнер из тэгов <NOFRAME> и </NOFRAME>, который позволяет строить двойные документы для броузеров, поддерживающих фрэймы и не поддерживающих фрэймы.

Данный тэг имеет два взаимоисключающих параметра: ROWS и COLS.

ROWS="список-определений-горизонтальных-подокон"

> Данный тэг содержит описания некоторого количества подокон, разделенные запятыми. Каждое описание представляет собой числовое значение размера

подокна в пикселах, процентах от всего размера окна или связанное масштабное значение. Количество подокон определяется количеством значений в списке. Общая сумма высот подокон должна составлять высоту всего окна (в любых измеряемых величинах). Отсутствие атрибута ROWS определяет один фрэйм, величиной во все окно броузера.

Синтаксис используемых видов описания величин подокон:

value

Простое числовое значение определяет фиксированную высоту подокна в пикселах. Это далеко не самый лучший способ описания высоты подокна, поскольку различные броузеры имеют различный размер рабочего поля, не говоря уже о различных экранных разрешениях у пользователя. Если вы, все же, используете данный способ описания размера, то настоятельно рекомендуется сочетать его с каким-либо другим, чтобы в результате вы точно получили 100%-ное заполнение окна броузера вашего пользователя.

value%

Значение величины подокна в процентах от 1 до 100. Если общая сумма процентов описываемых подокон превышает 100, то размеры всех фрэймов пропорционально уменьшаются до суммы 100%. Если, соответственно, сумма меньше 100, то размеры пропорционально учеличиваются.

value*

Вообще говоря, значение value в данном описании является необязательным. Символ "*" указывает на то, что все оставшееся место будет принадлежать данному фрэйму. Если указывается два или более фрэйма с описанием "*" (например "*,*"), то оставшееся пространство делится поровну между этими фрэймами. Если перед звездочкой стоит цифра, то она указывает пропорцию для данного фрэйма (во сколько раз од будет больше аналогично описанного чистой звездочкой). Например, описание "3*,*,*", говорит, что будет создано три фрэйма с размерами 3/5 свободного пространства для первого фрэйма и по 1/5 для двух других.

COLS="список-определений-горизонтальных-подокон"

То же самое, что и ROWS, но делит окно по вертикали, а не по горизонтали.

Внимание! Совместное использование данных параметров может привести к непредствазуемым результатам. Например, строка: <FRAMESET ROWS="50%,50%" COLS "50%,50%"> может привести к ошибочной ситуации.

Примеры:

<FRAMESET COLS="50,*,50"> - описывает три фрэйма, два по 50 точек справа и слева, и один внутри этих полосок.

<FRAMESET ROWS="20%,3*,*"> - описывает три фрэйма, первый из которых занимает 20% площади сверху экрана, второй 3/4 оставшегося от первого фрэйма места (т.е. 60% всей площади окна), а последний 1/4 (т.е. 20% всей площади окна.

<FRAMESET ROWS="*,60%,*"> - аналогично предыдущему примеру.

Тэги <FRAMESET> могут быть вложенными, т.е. например:

```
<FRAMESET ROWS="50%,50%">

      <FRAMESET COLS="*,*"

      </FRAMESET>

</FRAMESET>
```

Результат данного примера мы рассмотрим позже.

FRAME

<FRAME SRC="*url*" [NAME="frame_name"] [MARGINWIDTH="nw"] [MARGINHEIGHT="nh"] [SCROLLING=yes|no|auto] [NORESIZE]>

Данный тэг определяет фрэйм внутри контейнера FRAMESET.

SRC="url"

Описывает URL документа, который будет отображен внутри ланного фрэйма. Если он отсутствует, то будет отображен пустой фрэйм.

NAME="frame_name"

Данный параметр описывает имя фрэйма. Имя фрэйма может быть использовано для определения действия с данным фрэймом из другого HTML-документа или фрэйма (как правило, из соседнего фрэйма этого же документа). Имя обязательно должно начинаться с символа. Содержимое поименованных фрэймов может быть задействовано из других документов при помощи специального атрибута TARGET, описываемого ниже.

MARGINWIDTH="value"

Это атрибут может быть использован, если автор документа хочет указать величину разделительных полос между фрэймами сбоку. Значение *value* указывается в пикселах и не может быть меньше единицы. По умолчанию данное значение зависит от реализации поддержки фрэймов используемым клиентом броузером.

MARGINHEIGHT="value"

То же самое, что и MARGINWIDTH, но для верхних и нижних величин разделительных полос.

SCROLLING="yes | no | auto"

Этот атрибут позволяет задавать наличие полос прокрутки у фрэйма. Параметр *yes* указывает, что полосы прокрутки будут в любом случае присутствовать у фрэйма, параметр *no* наоборот, что полос прокрутки не будет. *Auto* определяет наличие полос прокрутки только при их необходимости (значение по умолчанию).

NORESIZE

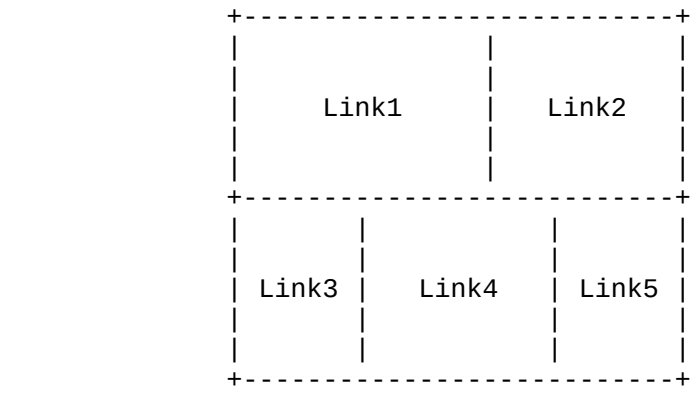
Данный атрибут позволяет создавать фрэймы без возможности изменения размеров. По умолчанию, размер фрэйма можно изменить при помощи мыши так же просто, как и размер окна Windows. NORESIZE отменяет данную возможность. Если у одного фрэйма установлен атрибут NORESIZE, то у соседних фрэймов тоже не может быть изменен размер со стороны данного.

NOFRAMES

Данный тэг используется в случае, если вы создаете документ, который может просматриваться как броузерами, поддерживающими фрэймы, так и броузерами, их не поддерживающими. Данный тэг помещяется внутри контейнера FRAMESET, а все, что находится внутри тэгов <NOFRAMES> и </NOFRAMES> игнорируется броузерами, поддерживающими фрэймы.

Примеры

Рассмотрим реализацию фрэймов для подобного разбиения окна:




```
 <FRAMESET ROWS="*,*">
 <NOFRAMES>
<H1>Ваша версия WEB-броузера не поддерживает фрэймы!</H1>
</NOFRAMES>
       <FRAMESET COLS="65%,35%">
      <FRAME SRC="link1.php">
      <FRAME SRC="link2.php">
      </FRAMESET>
       <FRAMESET COLS="*,40%,*">
      <FRAME SRC="link3.php">
      <FRAME SRC="link4.php">
```

```
    <FRAME SRC="link5.php">
    </FRAMESET>
</FRAMESET>
```

Планирование фрэймов и взаимодействия между фрэймами

С появлением фрэймов сразу возникает вопрос: "*А как сделать так, чтобы нажимая на ссылку в одном фрэйме инициировать появление информации в другом?*"

Ответом на данный вопрос является **планирование взаимодействия фрэймов** (далее - планирование). Каждый фрэйм может иметь собственное имя, определяемое параметром NAME при описании данного фрэйма. Существует, также, специальный атрибут - TARGET, позволяющий определять, к какому фрэйму относится та или иная операция. Формат данного атрибута следующий:

TARGET="windows_name"

Данный атрибут может встречаться внутри различных тэгов:

TARGET в тэге **A**

Это самое прямое использование TARGET. Обычно, при активизации пользователем ссылки соответствующий документ появляется в том же окне (или фрэйме), что и исходный, в котором была ссылка. Добавление атрибута TARGET позволяет произвести вывод документа в другой фрэйм. Например:

```
<A HREF="mydoc.php" TARGET="Frame1"> Переход в фрэйм ≟ 1 </A>
```

TARGET в тэге **BASE**

Размещение TARGET в тэге BASE позволит вам не указывать при описании каждой ссылки фрэйм-приемник документов, вызываемых по ссылкам. Это очень удобно, если в одном фрэйме у вас находится меню, а в другой - выводится информация. Например:

Документ ≟ 1.

```
<FRAMESET                                                    ROWS="20,*">
<FRAME                  SRC="doc2.php"                     NAME="Frame1">
<FRAME                  SRC="doc3.php"                     NAME="Frame2">
</FRAMESET>
```

Документ ≟ 2 (doc2.php).

```
<HTML>
<HEAD>
<BASE                                                   TARGET="Frame2">
</HEAD>
<BODY>
<A          HREF="url1">          Первая          часть</A>          |
<A              HREF="url2">              Вторая              часть</A>
</BODY>
</HTML>
```

TARGET в тэге **AREA**

Таже можно включать тэг TARGET в описание ссылки при создании карты изображения. Например:

```
<AREA SHAPE="circle" COORDS="100,100,50" HREF="http://www.softexpress.com" TARGET="Frame1">
```

TARGET в тэге FORM

То же относится и к определению формы. В данном случае, после обработки переданных параметров формы результирующий документ появится в указанном фрэйме.

```
<FORM ACTION="url" TARGET="window_name">
```

Внимание! Имя окна (фрэйма) в параметре TARGET должно начинаться с латинской буквы или цифры. Также необходимо помнить, что существуют зарезервированные имена для разрешения специальных ситуаций.

Зарезервированные имена фрэймов

Зарезервированные имена фрэймов служат для разрешения специальных ситуаций. Все они начинаются со знака подчеруивания. Любые другие имена фрэймов, начинающиеся с подчеркивания будут игнорироваться броузером.

TARGET="_blank"

Данное значение определяет, что документ, полученный по ссылке будет отображаться в новом окне броузера.

TARGET="_self"

Данное значение определяет, что документ, полученный по ссылке будет отображаться в том же фрэйме, в котором находится ссылка. Это имя удобно для переопределения окна назначения, указанного ранее в тэге **BASE.**

TARGET="_parent"

Данное значение определяет, что документ, полученный по ссылке будет отображаться в родительском окне, вне зависимости от параметров **FRAMESET**. Если родительского окна нет, то данное имя аналогично "_self".

TARGET="_top"

Данное значение определяет, что документ, полученный по ссылке будет отображаться на всей поверхности окна, вне зависимости от наличия фрэймов. Использование данного параметра удобно в случае вложенных фрэймов.

Создание карты изображений

Создание карты изображения является одной из привлекательнейших возможностей HTML, позволяющей пользователю привязывать ссылки на другие документы к отдельным частям изображений. Щелкая мышью на отдельных частях изображения, пользователь может выполнять те или иные действия, переходить по той или иной ссылке на другие документы и т.п.

> **Внимание!** Если вы хотите использовать технологию картирования изображений, то вам необходимо использовать броузер, поддерживающий данную технологию!

Чтобы включить поддержку карты для изображения, необходимо ввести дополнительный параметр в тэг IMG:

```
<IMG SRC="url" USEMAP="url#map_name" >
```

Параметр USEMAP указывает, в каком месте находится карта описываемого изображения. Карта изображения определяет, какому участку изображения какой URL соответствует. Карта изображения может находиться в том же документе, что и изображение, или в другом документе. Помещение карты в другой документ позволяет собрать все карты изображений в одном документе (если у вас их несколько в различных документах), но добавляет еще одну итерацию в сети, когда за перыую итерацию выясняется местонахождение карты, а за вторую - выполнение действия, предписанного URL для данного участка изображения. Параметр `map_name` указывает имя карты для изображения, а предшествующий ему URL определяет местонахождение карты. Если данный URL отсутствует, то карта с указанным именем ищется в текущем документе.

Рассмотрим синтаксис определения карты изображения:

```
<MAP                                                    NAME="map_name">
<AREA [SHAPE=" shape "] COORDS="x,y,..." [HREF=" reference "] [NOHREF]>
</MAP>
```

Параметры:

<MAP NAME="map_name">

Данный тэг определяет начало описания карты с именем map_name.

<AREA...>

Описывает участок изображения и ставит ему в соответствие URL. Параметры:

SHAPE

Необязательный параметр, указывающий на форму определяемой области изображения. Может принимать значения:

- default - по умолчанию (обычно прямоугольник)
- rect - прямоугольник
- circle - круг
- poly - многоугольник произвольной формы

COORDS

Координаты в пикселах описываемой области. Для прямоугольника это четыре координаты левого верхнего и правого нижнего углов, для круга - три координаты (две - центр круга, третья - радиус). Для многоугольника это описание каждого угла в двух координатах - соответственно число координат равно удвоенному количеству углов.

Координаты считаются с нуля, поэтому для описания области 100 на 100 используется описание:

```
<AREA COORDS="0,0,99,99" ...>
```

HREF="url"

Описание ссылки, действия по которой будут выполняться при щелчке мыши в заданной области.

NOHREF

Параметр, указывающий, что ссылка отсутствует для данного участка. По умолчанию, если не указан параметр HREF, то считается что действует параметр NOHREF. Также, для всех неописанных участков изображения считается, что используется параметр NOHREF.

Если две описанных области накладываются друг на друга, то используется ссылка, принадлежащая первой из описанных областей.

</MAP>

Данный тэг завершает описание карты изображения.

Применение:

Технология Image Map применяется в самых различных областях. Однако наиболее часто ее применение можно увидеть при создании графических меню, когда создается одно большое изображение с элементами меню, и каждому участку изображения предписывается какое либо действие.

Так же прменять данную технологию можно при создании простых ГИС-подобных систем с картографическими возможностями.

HTML таблицы

Таблицы в HTML организуются как набор столбцов и строк. Ячейки таблицы могут содержать любые HTML-элементы, такие, как заголовки, списки, абзацы, фигуры, графику, а также элементы форм.

Основные тэги таблицы

Таблица: <TABLE>...</TABLE>

Это основноые тэги, описывающие таблицу. Все элементы таблицы должны находиться внутри этих двух тэгов. По умолчанию таблица не имеет обрамления и разделителей. Обрамление добавляется атрибутом BORDER.

Строка таблицы: <TR>...</TR>

Количество строк таблицы определяется количеством встречающихся пар тэгов <TR>..</TR>. Строки могут иметь атрибуты ALIGN и VALIGN, которые описывают визуальное положение содержимого строк в таблице.

Ячейка таблицы: <TD>...</TD>

Описывает стандартную ячейку таблицы. Ячейка таблицы может быть описана только внутри строки таблицы. Каждая ячейка должна быть пронумерована номером колонки, для которой она описывается. Если в строке отстутствует одна или несколько ячеек для некоторых колонок, то броузер отображает пустую ячейку. Расположение данных в ячейке по умолчанию определяется атрибутами ALIGN=left и VALIGN=middle. Данное расположение может быть исправлено как на уровне описания строки, так и на уровне описания ячейки.

Заголовок таблицы: <TH>...</TH>

Ячейка заголовка таблицы имеет ширину всей таблицы; текст в данной ячейке имеет атрибут BOLD и ALIGN=center.

Подпись: <CAPTION>...</CAPTION>

Данный тэг описывает название таблицы (подпись). Тэг <CAPTION> должен присутствовать внутри <TABLE>...</TABLE>, но снаружи описания какой-либо строки или ячейки. По умолчанию <CAPTION> имеет атрибут ALIGN=top, но может быть явно установлен в ALIGN=bottom. ALIGN определяет, где - сверху или снизу таблицы - будет поставлена подпись. Подпись всегда центрирована в рамках ширины таблицы.

Основные атрибуты таблицы

BORDER

Данный атрибут используется в тэге TABLE. Если данный атрибут присутствует, граница таблицы прорисовывается для всех ячеек и для таблицы вцелом. BORDER может принимать числовое значение, определяющее ширину границы, например BORDER=3.

ALIGN

Если атрибут ALIGN присутствует внутри тэгов <CAPTION> и </CAPTION>, то он определяет положение подписи для таблицы (сверху или снизу). По умолчанию ALIGN=top.

Если атрибут ALIGN встречается внутри <TR>, <TH> или <TD>, он управляет положением данных в ячейках по горизонтали. Может принимать значения left (слева), right (справа) или center (по центру).

VALIGN

Данный атрибут встречается внутри тэгов <TR>, <TH> и <TD>. Он определяет вертикальное размещение данных в ячейках. Может принимать значения top (вверху), bottom (внизу), middle (по середине) и baseline (все ячейки строки прижаты кверху).

NOWRAP

Данный атрибут говорит о том, что данные в ячейке не могут логически разбиваться на несколько строк и должны быть представлены одной строкой.

COLSPAN

Указывает, какое количество ячеек будет объединено по горизонтали для указанной ячейки. По умолчании - 1.

ROWSPAN

Указывает, какое количество ячеек будет объединено по вертикали для указанной ячейки. По умолчании - 1.

COLSPEC

Данный параметр позволяет задавать фиксированную ширину колонок либо в символах, либо в процентах, например COLSPEC="20%".

Пример таблицы

```
<TABLE BORDER=1>
<CAPTION ALIGN=bottom> Таблица ≟1 </CAPTION>
<TR><TD ROWSPAN=2></TD><TH COLSPAN=2>Среднее значение</TH></TR>
<TR><TH>Рост</TH><TH>Вес</TH></TR>
<TR><TD>Мужчины</TD><TD ALIGN=center>174</TD><TD ALIGN=center>78</TD></TR>
<TR><TD>Женщины</TD><TD ALIGN=center>165</TD><TD ALIGN=center>56</TD></TR>
</TABLE>
```

	Среднее значение	
	Рост	**Вес**
Мужчины	174	78
Женщины	165	56

Таблица ≒1		

Обзор

Большое количество World Wide Web приложений основано на использовании внешних программ, управляемых Web сервером. Использование данных программ позволяет строить Web приложения с динамически обновляемой информацией, хнанящейся в базах данных или генерирующейся в зависимости от бизнес-правил рещаемых задач. Для связи между Web сервером и вызываемыми программами широко используется Common Gateway Interface (CGI), имеющий реализации как для Windows-ориентированных программ, так и для приложений, функционирующих в среде Unix. Данный документ описывает Windows-модификацию интерфейса CG, иначе называемую Windows CGI интерфейсом.

Разбор данных HTML-форм

Windows CGI требует, чтобы Web сервер декодировал данные из HTML форм, если они переданы при помощи POST метода запроса. Он не требует от сервера декодирования параметров, если они переданы в качестве строки запроса ("query string"), являющейся частью URL.

Существует два способа, которыми данные из форм могут быть переданы серверу броузером:

URL-Encoded
: Это наиболее используемый формат данных, передаваемых из форм. Содержимое полей формы выделяются из формы и передаются согласно спецификации HTML 1.0, а затем собираются в одну строку, где отделяются друг от друга символом амперсанда. Тип содержания сообщения устанавливается броузером в `application/x-www-form-urlencoded`.

Multipart Form Data
: Данный формат разработан для эффективной загрузки файлов на сервер с использованием форм. Содержимое полей формы передается как многостраничное MIME сообщение. Каждое поле содержится в одной странице. Тип содержания, устанавливается броузером в `multipart/form-data`.

"Грамотные" серверы должны уметь обрабатывать оба типа данных из форм.

Вызов CGI программ

Сервер использует функцию `CreateProcess()` для вызова CGI программ. Сервер синхронизируется с CGI программой, поскольку он должен определить момент завершения CGI программы. Это достигается использованием функции Win32 `WaitForSingleObject()`, ожидающей получения сигнала завершения CGI программы.

Командная строка

Сервер должен вызывать CGI программу выполняя функцию `CreateProcess()` с командной строкой следующего формата:

```
   WinCGI-exe cgi-data-file
WinCGI-exe
```
Полный путь к исполняемой CGI программе. Сервер не зависит от "текущего каталога" или переменной окружения PATH. Примите к сведению, что "исполняемая" не обязательно означает .EXE файл. Это может быть документ, ассоциирующийся с реально исполняемой программой, описанной в WIN.INI или System Registry.
```
 cgi-data-file
```

Метод вызова

Сервер использует `CreateProcess()` для запуска процесса, не имеющего главного окна. Вызванный процесс не будет отображаться каким либо образом на мониторе сервера.

Некоторые сервера поддерживают режим отладки CGI программ и скриптов, что позоволяет серверу запускать CGI программу как обычный процесс с созданием главного окна и отображением информации на мониторе сервера. Данный способ весьма удобен на стадии отладки CGI программ.

Обработка результата

CGI программа возвращает результат работы, отвечающий (явно или неявно) целям запроса. Сервер кодирует результат работы в соответствии со стандартом HTTP и использует HTTP для отправки результата клиенту. Это означает, что сервер добавляет необходимый HTTP заголовки в сообщение, формируемое CGI программой.

Результат работы CGI программы состоит из двух частей: *заголовка* и *тела сообщения.* Заголовок состоит из одной или более строк текста, отделенных от тела пустой строкой. Тело сообщения содержит данные, представленные в MIME формате, указанном в заголовке.

Сервер не изменяет тело документа, что означает, что сервер передает сформированный CGI программой ответ "как он есть".T

Специальные строки заголовка

Сервер распознает следующие строки заголовка в выходном потоке:

`Content-Type:`
Указывает на MIME тип тела сообщения. Значение этого параметра должно быть в формате *type/subtype.*

`URI:` <*value*> `(value enclosed in angle brackets)`
Данное значение указывает на полный URL или ссылку на локальный файл, сообщение из которого будет возвращено клиенту в теле сообщения. Если значение является локальным файлом, сервер отсылает его как результат запроса, как будто клиент воспользовался методом GET при генерации запроса. Если значение является полным URL, то сервер возвращает сообщение "401 redirect" для обеспечения прямой загрузки указанного объекта.

`Location:`
То же самое, что и URI, но данная форма сейчас не используется. Параметр *value* НЕ должен быть взят в угловые скобки.

Другие заголовки

Другие заголовки передаются клиенту в том виде, в котором они представлены.

Прямой возврат

Сервер позволяет конечному приложению осуществлять прямой возврат результата запроса клиенту. Это осуществляется посредством включение в заголовок возвращаемого сообщения его информационного протокола. Это позволяет CGI программам формировать непосредственный ответ клиенту с указанием HTTP заголовка без предварительной обработки его сервером..

Сервер анализирует результат запроса, помещаемый CGI программой в выходной файл (Output File), и, если первая строка "`HTTP/1.0`", он предполагает, что сообщение содержит полный HTTP ответ и отсылает его клиенту без упаковки.

Оптимизация графики для Web

На данный момент в Web используется два типа растровых файлов: в форматах **JPEG** и **GIF**.

JPEG-формат хорошо передает цветовые и тоновые раскаты, размытые границы (например, фото). JPEG-файл хорошо масштабируется в броузере.
Плохо передает ровные плоскости цвета, в компрессии уступает GIF-формату. При сохранении в JPEG-формате выбирайте качество "medium".

GIF-формат хорошо передет ровные плоскости цвета, жесткие границы (например, векторную графику, логотипы). Имеет максимальную компрессию, допускает прозрачный фон.
Плохо масштабируется в броузере, искажает цветовые и тоновые раскаты.

Используйте GIF-формат, если изображение без значительных потерь переводится в 128-цветовую гамму с включенной опцией "dithering". В противном случае лучше сохранять изображение в JPEG-формате.

Для экспорта файла в GIF-формат сначала проиндексируйте его цветовую палитру в Adobe Photoshop:

1. Подбирайте минимальное количество цветов вручную (для качественной передачи антиалиасного одноцветного изображения на одноцветном фоне достаточно 5-8 цветов, двух-трехцветного изображения - 15-25 цветов) Если исходное изображение **Grayscale**, перед индексацией переведите его в RGB-гамму.

2. По возможности избегайте включения опции **"dithering"**- она увеличивает размер файла. Эта опция необходима только если в изображении присутствует цветовой или тоновой раскат (напр. тень). Назначая прозрачный фон, после применения **"dithering"** убедитесь что фон не стал "клетчатым".

3. В сложных случаях перед индексированием выделите наиболее важные элементы изображения. Цвета внутри выделенной области индексируются корректнее остальных.

Основы CSS

Основным понятием CSS является стиль √ т. е. набор правил оформления и форматирования, который может быть применен к различным элементам страницы. В стандартном HTML для присвоения какому-либо элементу определенных свойств (таких, как цвет, размер, положение на странице и т. п.) приходилось каждый раз описывать эти свойства, даже если на одной страничке должны располагаться 10 или 110 таких элементов, ничуть не отличающихся один от другого. Вы должны были десять или сто десять раз вставить один и тот же кусок HTMLкода в страничку, увеличивая размер файла и время загрузки на компьютер просматривающего ее пользователя.

CSS действует другим, более удобным и экономичным способом. Для присвоения какому-либо элементу определенных характеристик вы должны один раз описать этот элемент и определить это описание как стиль, а в дальнейшем просто указывать, что элемент, который вы хотите оформить соответствующим образом, должен принять свойства стиля, описанного вами. Удобно, не правда ли?

Более того, вы можете сохранить описание стиля не в тексте вашей странички, а в отдельном файле √ это позволит использовать описание стиля на любом количестве Web-страниц. Потрясающе удобно. И еще одно, связанное с этим, преимущество √ возможность изменить оформление любого количества страниц, исправив лишь описание стиля в одном (отдельном) файле.

Кроме того, CSS позволяет работать со шрифтовым оформлением страниц на гораздо более высоком уровне, чем стандартный HTML, избегая излишнего утяжеления страниц графикой.

Давайте рассмотрим, как мы можем воплотить столь замечательные возможности в жизнь.

Практическое освоение CSS

Как вам уже известно, информация о стилях может располагаться либо в отдельном файле, либо непосредственно в коде Web-странички. Расположение описания стилей в отдельном файле имеет смысл в случае, если вы планируете применять эти стили к большему, чем одна, количеству страниц. Для этого нужно создать обычный текстовый файл, описать с помощью языка CSS необходимые стили, разместить этот файл на Web-сервере, а в коде Web-страниц, которые будут использовать стили из этого файла, нужно будет сделать ссылку на него. Делается это с помощью тега <LINK>, располагающегося внутри тега <BODY> ваших страниц:

```
<LINK REL=STYLESHEET TYPE="text/css" HREF="URL">
```

Первые два параметра этого тега являются зарезервированными именами, требующимися для того, чтобы сообщить броузеру, что на этой страничке будет использоваться CSS. Третий параметр √ HREF= ╚URL╝ √ указывает на файл, который содержит описания стилей. Этот параметр должен содержать либо относительный путь к файлу √ в случае, если он находится на том же сервере, что и документ, из которого к нему обращаются √ или полный URL (╚http://...╝) в случае, если файл стилей находится на другом сервере.

Второй вариант, при котором описание стилей располагается в коде Webстранички, внутри тега <BODY>, в теге <STYLE type="text/css">... </STYLE>. В этом случае вы

можете использовать эти стили для элементов, располагающихся в пределах странички. Параметр type="text/css" является обязательным и служит для указания броузеру использовать CSS.

И третий вариант, когда описание стиля располагается непосредственно внутри тега элемента, который вы описываете. Это делается с помощью параметра STYLE, используемого при применении CSS с большинством стандартных тегов HTML. Этот метод нежелателен, и понятно почему: он приводит к потере одного из основных преимуществ CSS √ возможности отделения информации от описания оформления информации. Впрочем, если необходимо описать лишь один элемент, этот вариант расположения описания стилей также вполне применим.

Давайте рассмотрим механизм, с помощью которого стили присваиваются элементам Webстраниц. Самый простой случай присвоения какому-либо элементу определенного стиля выглядит так:

```
НАЗВАНИЕ_ЭЛЕМЕНТА {свойство: значение;},
```

Где НАЗВАНИЕ_ЭЛЕМЕНТА √ имя HTMLтега (H1, P, TD, A и т. д.), а параметры в фигурных скобках √ список свойств элемента и присвоенных им значений. Более подробно команды языка CSS мы рассмотрим чуть позже.

Пример:

```
H1 {font-size: 30pt; color: blue;}
```

В этом примере всем заголовкам на странице, оформленным тегом H1, присваивается размер шрифта 30 пунктов и синий цвет.

Также элементы страниц, созданные с использованием CSS, используют механизм наследования: т. е. если вы располагаете изображение внутри тега <P>...</P>, оформленного с помощью CSS, с отступами, так, чтобы параграф занимал только определенную часть ширины страницы, изображение также унаследует значения отступов, указанные для этого параграфа.

CSS реализует возможность присваивать стили не всем одинаковым элементам страницы, а избирательно √ для этого используется параметр CLASS = "имя класса" или идентификатор ID=╚имя элемента╝, присваивающиеся любому элементу страницы. Рассмотрим эти возможности подробнее.

Параметр CLASS применяется в случае, если необходимо создать одинаковый стиль для нескольких, но не всех элементов страницы (одинаковых или разных).

Пример:

```
.b-c {font-weight: bold; text-align: center}
√ описание стиля для класса b-c
```

Все элементы класса b-c будут отображаться жирным шрифтом с выравниванием по центру страницы (или ячейки таблицы).

```
<P CLASS="b-c">Текст параграфа</P>
```

√ параграфу присвоен стиль класса b-c. <TD CLASS="b-c">текст</TD>
√ ячейке таблицы присвоен стиль класса b-c.

Содержащийся в ячейке текст будет отображаться согласно описанию класса.

Таким образом, вы можете присвоить описанный стиль любым текстовым элементам страниц. Обратите внимание, что при написании названия классов необходимо соблюдать регистр символов, согласно тому, как вы назвали класс в описании стиля!

Присвоение стилей с помощью идентификаторов применяется в случае, если данному идентификатору соответствует только один элемент на странице. Если элементов, которым необходимо присвоить такой стиль, несколько √ это уже класс.

Свойства элементов, управляемых с помощью CSS

В настоящее время язык CSS насчитывает довольно большое количество свойств элементов HTML, которыми он может управлять. Но из-за того, что этот стандарт еще очень молод, в полном объеме его пока не поддерживают наиболее популярные броузеры (Netscape Navigator и Microsoft Internet Explorer). Последние версии этих броузеров могут работать с довольно большим количеством команд CSS, а вот 3-и версии или совсем не поддерживают его (Netscape Navigator 3), или поддерживают, но лишь частично (Microsoft IE 3). Более того, поскольку разработчики из этих компаний никак не могут договориться между собой, последние версии броузеров поддерживают неодинаковый набор свойств CSS. Все это делает малоприемлемым использование CSS в полном объеме, так как, при использовании CSS для форматирования элементов страницы и просмотре ее с помощью броузера версии ниже 4й, есть большая вероятность увидеть нечто такое, что вам не понравится. Поэтому будет разумнее воздержаться от использования CSS для форматирования основной структуры страниц до всеобщего перехода на последние версии броузеров. В то же время, применяя ╚безопасные╝, т. е. совместимые с максимальным количеством броузеров элементы CSS, вы можете сильно облегчить себе жизнь и сделать ваши Web-странички более привлекательными в плане шрифтового оформления, а пользователи, путешествующие по Internet с помощью устаревших броузеров, просто этого не увидят, но также они не увидят и тех кошмаров, которые появляются при использовании CSS для верстки страниц.

СВОЙСТВА ШРИФТА	
font-family	Используется для указания шрифта или шрифтового семейства, которым будет отображаться элемент. P {font-family: Times New Roman, sans-serif;}
font-weight	Определяет степень жирности шрифта с помощью трех параметров: lighter, bold, bolder B {font-weight: bolder;}
font-size	Устанавливает размер шрифта. Параметр может указываться как в относительной (проценты), так и абсолютной величине (пункты, пикселы, сантиметры) H1 {font-size: 200%;} H2 {font-size: 150px;} H3 {font-size: 400pt;}

font-size	Устанавливает размер шрифта. Параметр может указываться как в относительной (проценты), так и абсолютной величине (пункты, пикселы, сантиметры) H1 {font-size: 200%;} H2 {font-size: 150px;} H3 {font-size: 400pt;}
ЦВЕТ ЭЛЕМЕНТА И ЦВЕТ ФОНА	
color	Определяет цвет элемента I {color: yellow;}
background-color	Устанавливает цвет фона для элемента √ именно для элемента, а не для странички. Обратите внимание, что броузеры отображают это свойство по-разному: Microsoft IE отводит под фон элемента всю доступную ширину страницы, а Netscape Navigator √ лишь ширину, занимаемую этим элементом. Посмотрите пример (рис. 3, 4), вот его исходный код: <HTML> <HEAD> <TITLE>Пример использования CSS</TITLE> <STYLE type=■text/css■> H1 {font-size: 300%;} </STYLE> </HEAD> <BODY bgcolor=white> <H1 style=■background-color: teal; color: white;■>Cascading</H1> <H1 style=■background-color: navy; color: yellow;■>Style</H1> <H1 style=■background-color: gold; color: brown;■>Sheets</H1> </BODY> </HTML> В этом примере в разделе <STYLE> всем элементам <H1> на этой страничке был установлен размер 300 % от нормы. Затем каждому из элементов <H1> были присвоены собственные значения цвета фона и цвета символов.
СВОЙСТВА ТЕКСТА	
text-decoration	Устанавливает эффекты оформления шрифта, такие, как подчеркивание или зачеркнутый текст H4 {text-decoration: underline;} A {text-decoration: none;} .wrong {text-decoration: line-through;}
text-align	Определяет выравнивание элемента. P {text-align: justify} H1 {text-align: center}
text-indent	Устанавливает отступ первой строки текста. Чаще всего используется для

	создания параграфов с табулированной первой строкой. P {text-indent: 50pt;}
line-height	Управляет интервалами между строками текста. P {line-height: 50 %}
СВОЙСТВА ГРАНИЦ	
margin-left	Устанавливают значения отступов вокруг элемента. IMG { margin-right: 20pt} P { margin-left: 2cm}
margin-right margin-right margin-top	Устанавливают значения отступов вокруг элемента. IMG { margin-right: 20pt} P { margin-left: 2cm}
ЕДИНИЦЫ ИЗМЕРЕНИЯ	
px	Пикселы
cm	Сантиметры
mm	Миллиметры
pt	Пункты (типограф.)
%	Проценты

Итак, перейдем к изучению безопасных элементов CSS. Описание свойств элементов в CSS состоит из названия свойства с последующим присвоением ему определенного значения. Название свойства и его значение разделены двоеточием.

Указывая абсолютные, а не относительные размеры шрифтов, вы лишаете людей, просматривающих ваши странички, возможности увеличивать или уменьшать размер шрифтов с помощью специальной кнопочки в броузере в соответствии с разрешением их дисплея и зрением. Шрифты будут отображаться только такого размера, который вы указали при написании странички.

Поэтому, если использование абсолютных размеров шрифтов не обусловлено художественным замыслом или коварным умыслом, рекомендую вам использовать для этих целей указание размеров в процентах. В этом случае размер шрифта будет меньше (больше) на указанное вами количество процентов, чем при оформлении его с помощью стандартного HTML-тега.

Есть еще одна небольшая, но очень полезная хитрость √ это способ скрыть от устаревших броузеров описания стилей, располагающихся в теге <STYLE>, внутри раздела<HEAD>. Поскольку броузер был написан несколько лет назад, когда никакого CSS еще и в планах не было, он просто не поймет, что это такое написано внутри <STYLE>┘ </STYLE>, и выдаст все описания стилей на страничку, как обычный текст. Для того чтобы предотвратить это, необходимо заключить описания стилей в тег комментариев. Делается это очень просто.

```
<HEAD>
<STYLE type="text/css">
<!--
```

```
описания стилей
-- >
</STYLE>
</HEAD>

где
<!--√ тег, открывающий комментарий, а
> √ закрывающий.
```

Устаревшие броузеры посчитают все заключенное между тегами комментариев информацией неотображенной, а новые и сообразительные броузеры определят, что это ≈ описание стилей, и задействуют их.

Еще один из интересных вариантов применения CSS скрывается за, казалось бы, простой возможностью: вы можете указывать значения отступов вокруг объектов, как отрицательные величины! Это позволяет накладывать один слой текста на другой и получать весьма интересные и привлекательные результаты.

Добиться такого эффекта не очень сложно, давайте попробуем создать страничку с заголовком, который будет выглядеть трехмерным, но не будет использовать графику.

Создадим новый html-файл и составим описание стилей для трех объектов:

```
<HEAD>
<STYLE type="text/css">
BODY {font-family: Verdana; font-size: 70pt; font-weight: bold;}
.z1 { color: silver; margin-top: 100px; margin-left: 70px;}
.z2 {color: navy; margin-top: -118px; margin-left: 68px;}
</STYLE>
</HEAD>
```

В этом описании мы присвоили <BODY> (впрочем, это мог быть практически любой другой тег) размер, шрифт и начертание √ в таком стиле будут отображаться все элементы страницы. Это было сделано лишь ради стремления уменьшить размер файла странички, вместо этого можно было описать эти параметры дважды: для каждого из классов z. Далее мы описываем два стиля, которые отличаются цветом и размером отступов вокруг них: нижний слой описывается стилем z1, а верхний √ z2. Используя отрицательные значения отступов и подбирая нужное значение, мы добиваемся того, что верхний слой как бы наползает на предыдущий...

```
<BODY bgcolor=white>

<DIV class="z1">EC-NET</DIV>
<DIV class="z2">EC-NET</DIV>

</BODY>
```

Откроем наш любимый Web-редактор Notepad и создадим файл с будущим названием styles.css (название файла может быть любым). Опишем в этом файле стиль параграфа <P>, который будет использоваться на всех страничках нашего сайта:

```
P {
font-family: Times New Roman, serif;
color: #000000;
margin-left: 15%;
margin-right: 15%;
```

```
margin-top: 1pt;
margin-bottom: 1pt;
text-indent: 1cm;
text-align: justify;
}
```

Внутри описания стиля для удобства форматирования вы можете использовать любое количество пробелов и переносов строк √ при чтении стиля броузер просто отбросит все лишние пробелы.

В этом стиле мы задали, что параграфы <P> на всех страничках, которые используют это описание, будут отображаться шрифтом Times New Roman или в случае, если этот шрифт на машине не установлен, другим шрифтом, но из этого семейства (serif). Цвет шрифта мы установили черный, выравнивание √ полное (по обеим сторонам).

Также мы установили для параграфа ряд значений отступов. Это было сделано со следующей целью: по умолчанию параграф в HTML отображается равным практически 95 % страницы и с интервалами между параграфами, равными 180 % межстрочного интервала. Читать такие параграфы не очень удобно, так как интервалы между ними слишком велики, а ширина параграфа слишком большая. Посмотрите на журнал, который вы сейчас держите в руках: текст сверстан в колонки для того, чтобы его было удобнее читать. В стиле параграфа, который мы создали, установлены боковые отступы в 15 % ширины окна и вертикальные отступы в 1 пункт √ так текст статьи будет гораздо читабельнее.

Давайте так же создадим стиль для заголовков статей:

```
H2 {
font-family: Verdana, Arial Cyr, Arial;
font-weight: bold;
font-size: 14pt;
color: black;
margin-left: 20%;
margin-top: 1cm;
text-align: left;
}
```

Все заголовки наших страниц, оформленные тегом <H2>, будут отображаться жирным шрифтом Verdana или, если этот шрифт не установлен, шрифтом Arial. Размер заголовка мы установим равным 14 пунктам, цвет черный, отступ слева равен 20 % ширины страницы, а отступ сверху √ 1 см. Заголовок будет выравниваться относительно левого края страницы.

Благодаря тому, что боковые отступы заголовка и параграфа установлены нами в процентах от ширины окна броузера, при просмотре наших страничек на компьютерах с разным разрешением дисплея пропорции и расположение заголовка, основного текста и отступов будут сохранены.

Для того чтобы ╚привязать╩ созданные нами стили к нашим страничкам, во все html-файлы в разделе <HEAD> мы должны поместить строку со ссылкой на файл стилей и с указанием об использовании CSS:

```
<LINK REL=STYLESHEET TYPE="text/css" HREF="styles.css">
```

Поскольку файл со стилями будет находиться в том же каталоге сервера, что и остальные странички, параметр HREF="URL" в нашем случае будет просто именем нашего файла стилей (styles.css).