

## МУЛЬТИМЕДИА-ОБЪЕКТЫ В HTML-ДОКУМЕНТАХ

Работа с изображениями. Элемент IMG

Выравнивание

Размеры изображения. Атрибуты width, height

Размещение объектов мультимедиа на Web-странице

Элемент OBJECT и элемент EMBED

Элемент PARAM

Элемент EMBED

# МУЛЬТИМЕДИА-ОБЪЕКТЫ В HTML-ДОКУМЕНТАХ

Возможности HTML 4.01 позволяют значительно повысить привлекательность Web-страницы благодаря включению в нее различных мультимедиа-объектов: изображений, видеоклипов, звуковых фрагментов, апплетов, других HTML-документов. Универсальным HTML-элементом, осуществляющим вставку объектов мультимедиа, является элемент OBJECT, появившийся в последних версиях языка HTML. Наряду с ним применяется элемент IMG, вставляющий в HTML-документ изображения и элемент APPLET, который вставляет апплеты и уже является нежелательным, т.к. поддерживает только апплеты, написанные на языке Java. Элемент OBJECT имеет общий характер и может производить включение в Web-страницу любых существующих объектов (в том числе изображения и апплеты) и объектов, которые будут появляться в будущем. Более упрощенный версией элемента OBJECT является элемент EMBED.

## Работа с изображениями. Элемент IMG

Не вызывает сомнений, что использование изображений является основным способом, благодаря которому любой документ делается привлекательным и более понятным. Практически ни одна публикация как в обычной печати, так и в Web, не обходится без иллюстраций. Они позволяют разработчику лучше и доходчивее передать содержимое документа, сделать его более наглядным. Декоративные изображения делают работу с документами приятной для глаз. Однако используя изображения в своих публикациях, не следует забывать о чувстве меры. Большое количество графической информации может не только сделать вашу страницу безвкусной по дизайну, но и беспричинно увеличить длительность ее загрузки, что всегда приводит к нежелательным затратам времени на ожидание и отпугивает посетителей. Следует помнить, что излишняя пестрота даже неискушенного пользователя начинает очень скоро раздражать, что не способствует популярности вашего Web-узла.

В HTML-документах различают изображения, встраиваемые в качестве иллюстраций, и фоновые изображения.

Работа с последними абсолютно идентична работе с фоновыми изображениями в Windows при помещении их на Desktop. Единственное отличие заключается в том, что фоновые изображения в HTML-документах заполняют все окно браузера. В том случае, если геометрические размеры фонового изображения меньше окна браузера, все пространство окна заполняется по принципу мозаики. Отсюда следует, что при использовании маленьких картинок для задания фона, они должны быть выполнены таким образом, чтобы при мозаичном расположении границы между ними были незаметны.

В качестве фоновых изображений рекомендуется использовать небольшие декоративные картинки (текстуры) спокойного, мягкого цвета с ненавязчивым рисунком. Однако могут использоваться и большие по размерам изображения. Только в этом случае рекомендуется как можно лучше оптимизировать его, чтобы это изображение занимало как можно меньший объем памяти.

Задание фонового изображения в HTML, как уже упоминалось, осуществляется атрибутом Background элемента BODY. При загрузке документа первоначально отображается его текстовая часть, и только потом загружаются изображения, в том числе и фоновые. При этом текст отображается на фоне, цвет которого установлен (или не установлен) атрибутом BGCOLOR и только потом выводится фоновое изображение. Поэтому настоятельно рекомендуется указывать в качестве фонового цвет, который наиболее близок к основному цвету фонового изображения.

*Пример (РИС. 3.95)*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<HTML>

<HEAD>

<TITLE>

Документ с заданным фоновым изображением

</TITLE>

</HEAD>

<BODY background="lv.jpg">

<H2 align=center>Пример документа с фоновым изображением</H2>

<HR color=darkgray size=20>

<P>Деньги правят миром. И сильнее тот, у кого их больше.</P>

<I>Цитата из фильма "Брат 2"</I>

<HR color=darkgray size=20>

</BODY>

</HTML>
```

Браузер Internet Explorer поддерживает использование атрибута `bgproperties`, который позволяет фиксировать фоновое изображение и не прокручивать его вместе с содержимым документа. Особенно красочно это выглядит при использовании фоновых изображений, заполняющих весь экран. Тогда текст документа (если он, конечно, не помещается целиком в экране) при прокрутке живописно проплывает над изображением, как титры в кино после окончания фильма.

*Пример задания:*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<HTML>

<HEAD>

<TITLE>

Документ с заданным фоновым изображением </TITLE>

</HEAD>

<BODY background="backgroundimage.jpg">

<H2 align=center>Пример документа с закрепленным фоновым изображением</H2>

<HR color=darkgray size=20> текст документа

текст документа текст документа текст документа

</BODY>

</HTML>
```

Внедрение иллюстративных изображений в HTML осуществляется посредством элемента `IMG`, имеющего следующие атрибуты:

**src** — обязательный атрибут, задающий URL-адрес (полный или относительный) расположения изображения.

**align** — выравнивание.

**width, height** — геометрические размеры изображения, задаваемые либо в пикселах, либо в процентах видимого пространства окна браузера.

**hspace, vspace** — указывает горизонтальные и вертикальные отступы от изображения соответственно. Значение этих атрибутов задается в пикселах и позволяет вокруг изображения оставлять свободное пространство определенных размеров.

**border** — атрибут, управляющий наличием и шириной рамки вокруг изображения. Значение задается в пикселах и определяет ширину рамки. По умолчанию вокруг изображения не прорисовывается. Исключением являются случаи, когда изображение является ссылкой.

**alt** — атрибут, имеющий в качестве значения текстовую строку, которая является альтернативной текстовой информацией текущего изображения.

**usemap** — позволяет использовать изображение совместно с клиентской навигационной картой, имя которой задается в качестве значения этого атрибута.

**ismap** — позволяет использовать изображение совместно с серверной навигационной картой, имя которой задается в качестве значения этого атрибута.

**id, class** — идентификаторы в пределах документа. **style** — встроенная информация о стиле.

**onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup, onfocus, onblur** — внутренние события.

Умелое использование атрибутов позволяет осуществлять широкий спектр приемов представления изображений в HTML-документах. Остановимся на них поподробнее.

## Выравнивание

Параметры выравнивания задают расположение изображения относительно текста и других элементов в документе. Также с помощью них можно осуществлять привязку изображения к краям документа, такие и только такие изображения могут обтекаться текстом.

Для элемента **IMG** атрибут **ALIGN** может принимать следующие значения:

**left** — изображение прикрепляется к левому краю документа. Обтекание изображения текстом производится по правой стороне (рис. 3.96).

**right** — изображение прикрепляется к правому краю документа и обтекается текстом с левой стороны.

**top** — выравнивание по верхней границе. Это значит, что изображение будет вставлено таким образом, чтобы его верхняя граница находилась на уровне верхней границы самого высокого элемента в текущей строке. В этом и последующих типах выравнивания изображение не обтекается, а как бы встраивается в текст. Для наглядности можно представить, что изображение является буквой, только большого размера и воспринимается браузером как обычный элемент строки.

**texttop** — также задает выравнивание по верхней границе и является частным случаем предыдущего типа выравнивания, т.к. в данном случае верхняя граница изображения располагается на уровне верхней границы самого высокого текстового символа текущей строки.

Наглядно различия между типами выравнивания **top** и **texttop** можно увидеть на рис. 3.97.

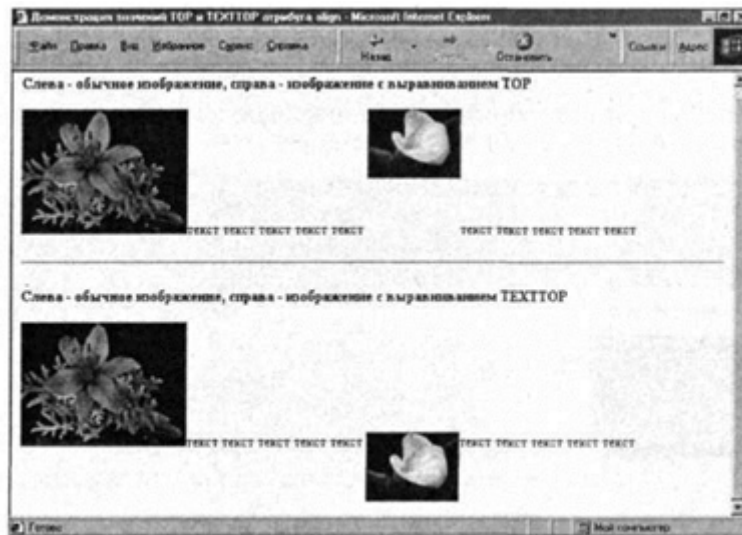


Рис. 3.97. Разница между top и texttop

В первом (верхнем) случае самым высоким элементом в строке считается большое изображение, поэтому маленькое изображение выравнивается по его верхней границе. Внизу маленькое изображение выравнивается по верхней границе текста.

```
html <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <HTML>

<HEAD> <TITLE>

Демонстрация значений TOP и TEXTTOP атрибута align </TITLE> </HEAD>

<BODY bgcolor=white>

<H4>Слева – обычное изображение,  справа – изображение с выравниванием TOP</H4>

<IMG src="577028.jpg" width=180>ТеКСТ  текст  текст  текст текст <IMG src="57703.jpg"
width=100 align=TOP>текст текст текст текст ТеКСТ<BR><BR> <HR>

<H4>Слева - обычное изображение,  справа - изображение с выравниванием TEXTTOP</H4> <IMG
src="577028.jpg" width=180>ТеКСТ текст текст текст текст

<IMG src="57703.jpg" width=100 align=TEXTTOP>текст текст текст текст текст

</BODY>

</HTML>
```

**middle** — указывает на то, что середина изображения должна находится на уровне базовой линии текущей строки.

**ABSMiddle** — указывает выравнивание центра изображения посередине текущей строки. Для браузеров данный тип выравнивания идентичен типу middle.

**BOTTOM** или **BASELINE** — при данном способе выравнивания нижняя граница изображения будет располагаться на базовой линии текущей строки.

**ABSBOTTOM** — задает выравнивание нижней границы изображения по нижней границе строки. Для браузеров способ выравнивания ABSBOTTOM идентичен способу BOTTOM или BASELINE. А вообще-то для текста базовая линия и нижняя граница отличаются на размер нижней части таких символов, как p, j и т.п.

## Размеры изображения. Атрибуты width, height

Благодаря использованию атрибутов width, height могут быть явно заданы размеры изображения, с которыми оно будет отображено в окне браузера. Следует помнить, что неправильное задание этих атрибутов может привести к искажению картинки (рис. 3.99). Можно задавать оба

Задание размеров изображений позволяет достичь следующих результатов: последовательного представления документа, при котором сначала отображается текст, а потом изображения, под которые при первом проходе оставляется свободное место, с размерами, задаваемыми атрибутами width, height. Кроме этого возникает возможность работы пользователя в режиме отключенной графики. Вместо изображений будет оставляться обведенное рамкой пустое пространство (рис. 3.100). В этом случае атрибуты width, height задают размеры этого пустого пространства, что позволяет сохранить авторское форматирование документа. В противном случае вместо изображения будет выводиться маленькая иконка с крестиком, что приведет к неразберихе в документе, отформатированном с учетом размеров изображений.

Кроме этого, с помощью атрибутов width и height становятся возможными приемы, с помощью которых можно достичь экономии памяти, а значит и уменьшения времени загрузки документа. Например, очень часто на Web-страницах используются всякие горизонтальные разноцветные полосы. Так вот, благодаря наличию атрибутов width и height можно не включать в документ изображение всей полосы, а вставить маленькое изображение и растянуть его.

*Например (рис. 3.101)*

```
html <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<HTML>

<HEAD>

<TITLE>Примеры использования атрибутов width и height для создания полосок</TITLE>

</HEAD>

<BODY bgcolor="white">

<H2 align=center>Примеры использования атрибутов width и height для создания полосок</H2>

<CENTER>

<TABLE>

<TR>

<TH><IMG src="pol.gif" border=1>

<TH>Исходное изображение

</TABLE>
```

```

</CENTER>

<BR><BR>

<TABLE width=700 bgcolor=lightgrey cellpadding=5>

<CAPTION>Полоски, созданные на основе исходного изображения</CAPTION>

<TR>

<TD><IMG src="pol.gif" width=200 height=10 border=1>

<TR>

<TD><IMG src="pol.gif" width=400 height=10 border=5>

<TR>

<TD><IMG src="pol.gif" width=600 height=10 border=1>

<TR>

<TD><IMG src="pol.gif" width=200 height=20 border=1>

<TR>

<TD><IMG src="pol.gif" width=700 height=25 border=1>

</TABLE>

</BODY>

</HTML>

```

На основе такого применения атрибутов width и height, а также с использованием сценариев, можно создавать динамически изменяющиеся гистограммы на Web- страницах.

Наличие у элемента IMG атрибутов hspace и vspace позволяет управлять размерами отступов между текстом и изображением. По умолчанию текст непосредственно примыкает к изображению, что вызывает некоторый дискомфорт восприятия. Наглядно положительный эффект применения атрибутов hspace и vspace можно увидеть на рис.3.102.

Изображение в HTML-документа может быть обведено рамкой (рис. 3.103). За наличие вокруг изображения рамки и ее ширину отвечает атрибут border элемента IMG, значение которого задается в пикселах.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<HTML>

<HEAD>

<TITLE>Примеры задания атрибута border</TITLE>

</HEAD>

<BODY bgcolor="white">

```

```

<H2 align=center> Примеры задания атрибута border</H2>

<CENTER>

<TABLE>

<TR>

<TH><IMG src="Img0001.jpg" width=200>

<TH><IMG src="Img0001.jpg" width=200 border=1>

<TR>

<TH><IMG src="Img0001.jpg" width=200 border=5>

<TH><IMG src="Img0001.jpg" width=200 border=15>

</TABLE>

</CENTER>

</BODY>

</HTML>

```

Альтернативный текст, заданный в качестве значения атрибута alt, отображается при наведении курсора на изображение или вместо изображения, когда оно, ввиду каких-либо причин, не может быть показано. При этом возможны следующие варианты,

наглядно представленные на рис. 3.104.

Компанией Netscape для элемента IMG предложен и поддерживается браузером Netscape Communicator (и только им) атрибут Lowsrc, который указывает URL-адрес предварительного изображения, отображаемого на первом проходе при последовательном предоставлении документа. В качестве предварительного изображения рекомендуется использовать быстрозагружаемую черно-белую версию основного изображения с меньшим разрешением.

Следует помнить, что основное изображение выводится поверх предварительного с размерами, задаваемыми размерами предварительного изображения. Это значит, что при несоответствии размеров предварительного и основного изображений, размеры основного будут приводиться к размерам предварительного, что может привести к его искажению.

## Размещение объектов мультимедиа на Web-странице

В общем случае, в зависимости от вида мультимедиа-объектов, существует три технологии включения их в HTML-документы: силами самого браузера с его библиотекой подключаемых модулей, с помощью элементов Active X (только для браузера Internet Explorer) и с помощью Java-апплетов.



Использование подключаемых модулей заключается в следующем: при запуске браузера он проверяет наличие подключенных к нему модулей, расположенных в определенном каталоге. Каждому модулю ставится в соответствие MIME-тип, который показывает, какие файлы могут быть открыты с его помощью. Когда в окне браузера открывается документ с включенным в него мультимедиа-файлом, браузер определяет его MIME-тип загружает соответствующий ему подключаемый модуль и передает файл ему. При этом, исходя из своих особенностей и установок разработчика, мультимедиа- объект либо отображается в контексте остального содержимого документа, либо занимает все окно браузера, либо не отображается вовсе (последний вариант обычно используется при обработке звуковых файлов). Именно с помощью подключаемых модулей в HTML внедряются звуковые и видеофайлы. Этому и будет уделено основное внимание в данном разделе.

***Примечание.** Для Internet Explorer каталогом, в котором хранят-ся подключаемые модули, является каталог Plugins в каталоге Internet Explorer. Для Netscape Communicator такой каталог также называется Plugins, а располагается он в одном из подкаталогов каталога Netscape.*

В том случае, если браузер сам не в состоянии определить соответствующий какому-либо объекту подключаемый модуль, он игнорирует этот объект. Если браузер определил, какому модулю соответствует объект, но его нет в библиотеке подключаемых модулей, то он пытается, с разрешения пользователя, подгрузить необходимый модуль.

## Элемент OBJECT и элемент EMBED

---

Хоть в начале этого раздела и было сказано, что размещение объектов на Web- странице осуществляется элементом OBJECT, на самом деле сейчас для этих целей могут использоваться два HTML-элемента: OBJECT и EMBED.

Элемент OBJECT официально включен в HTML только в спецификации HTML 4.0. А раньше для этих целей использовался HTML-элемент EMBED, который сейчас поддерживается всеми браузерами. Какой из них предпочесть, большой разницы не имеет. Возможно, эта разница появится в будущем, так как элемент OBJECT все еще разрабатывается, а элемент EMBED уже исключен из спецификации HTML 4.0. На данный момент имеют место следующие особенности каждого из них:

- Элемент **EMBED** предпочтительнее применять для включения в документ стандартных объектов, таких, как аудио- и видеофайлы обычных форматов, так как

этот элемент проще в использовании. Дополнительным преимуществом элемента EMBED является то, что он появился и используется уже достаточно давно (он поддерживается даже браузером Netscape Communicator 2.0). Это может послужить гарантией того, что ваш документ будет совместим с наибольшим количеством различных браузеров.

- Элемент **OBJECT** является более универсальным по сравнению с элементом EMBED. Этот элемент предпочтительнее применять для использования новейших HTML- технологий, включения в Web-страницу каких-либо специфических объектов и их параметров. Кроме этого, элемент OBJECT позволяет задавать альтернативное содержимое объекта. В более глобальных масштабах использование этого элемента является предпочтительным, так как по замыслу консорциума W3C (World Wide Web Consortium), за ним будущее.

С элементом OBJECT связаны некоторые тонкости, на которых следует остановиться поподробнее. Этот элемент не просто вставляет в страницу файл мультимедиа. Элемент OBJECT создает внутри HTML-документа объект, содержимое которого находится во внешнем мультимедиа-файле. Таким образом, с точки зрения HTML сам мультимедиа-файл не является объектом. Он является содержимым объекта. Аналогично обстоит дело и с элементами EMBED, IMG, APPLET, да и вообще со всеми элементами, просто их содержимое задано в том же

файле, что и они сами. Объект, будь то видеоклип или изображение, существует, вообще говоря, независимо от его содержимого.

Например, если для элемента `IMG` указать несуществующий графический файл, то от этого сам он не перестанет существовать. Просто у него не будет содержимого. Аналогично и для элемента `OBJECT`.

Это была теория. Тому, кто ее не понял, сначала достаточно просто считать что элемент `IMG` вставляет изображения, элемент `APPLET` — апплеты, элемент `IFRAME` — фреймы, элементы `EMBED` и `OBJECT` — объекты типа видеоклипов и звуковых фрагментов.

При задании объекта указывается тип его содержимого, в соответствии с которым браузер инициирует соответствующий подключаемый модуль, и месторасположение файла, который должен быть передан этому модулю. Дополнительные параметры файла задаются элементом `PARAM` внутри элемента `OBJECT`. Все остальные элементы, указанные между начальным и конечным тегами элемента `OBJECT`, являются его альтернативным содержимым и используются только тогда, когда корректная обработка объекта является невозможной. Причиной последнего может быть невозможность доступа к файлу, либо отсутствие соответствующего ему подключаемого модуля. В качестве альтернативного содержимого элемента `OBJECT` могут использоваться любые HTML-элементы, в том числе и сам элемент `OBJECT`, заданный один внутри другого. Задание начального и конечного тегов для элемента `OBJECT` является обязательным.

*Список атрибутов элемента `OBJECT`:*

**data** — указывает URL-адрес файла, в котором содержатся данные объекта, **classid**

— используется в двух вариантах:

- для указания URL-адреса подключаемого модуля ActiveX, и позволяет пользователю, если у него этот модуль отсутствует, скачать его из сети и проинсталлировать у себя на компьютере;
- либо служит для подключения уже имеющегося управляющего элемента ActiveX, указывая его идентификационное имя, которое является уникальным. В этом случае значением атрибута **classid** является шестнадцатичный код, который использует Windows для идентификации управляющих элементов ActiveX. Информация о всех установленных элементах ActiveX содержится в системном реестре Windows.

**codebase** — указывает базовый URL-адрес для текущего объекта, относительно которого будут браться все относительные ссылки. Иными словами, задание атрибута **codebase** для объекта аналогично заданию элемента `BASE` для HTML-документа. Таким образом, если в значении атрибута **data** (или атрибута **classid**) указана относительная ссылка, то она будет браться относительно URL-адреса, указанного атрибутом **codebase**. Если этот атрибут отсутствует, то по умолчанию в качестве базового адреса используется URL текущего документа.

**codetype** — своим значением указывает тип содержимого данных, получения

которых следует ожидать при загрузке объекта, задаваемого атрибутом **classid**. Этот атрибут не является обязательным, но рекомендуется, поскольку он позволяет браузеру не загружать информацию, тип содержимого который он не поддерживает. Если этот атрибут отсутствует, то по умолчанию используется значение атрибута **type**.

**type** — значением этого атрибута указывается MIME-тип содержимого мульти- медиа-файла. Этот атрибут является необязательным, но его использование настоятельно рекомендуется. Задание атрибута **type** позволяет браузеру вообще не загружать мультимедиа-файлы, обработать которые он не в состоянии. Иначе он сначала загрузит этот файл, а потом попытается определить его MIME-тип.

**border** — атрибут, задающий толщину рамки в пикселах вокруг объекта. По умолчанию рамка вообще отсутствует.

**height** — задает вертикальный размер прямоугольной области в пикселах, отводимой под данный объект в документе.

**width** — ширина прямоугольной области в пикселах, отводимой под данный объект в документе.

***Примечание,** отображаемая информация в объекте при этом изменяется таким образом, чтобы полностью заполнить отведенную под него прямоугольную область. Если атрибуты *height* и *width* для объекта не заданы, браузер сам автоматически определяет их, исходя из размеров содержимого объекта.*

**hspace** — атрибут, указывающий размеры в пикселах горизонтальных отступов между объектом и остальным содержимым документа.

**vspace** — атрибут, указывающий размер в пикселах вертикальных отступов между объектом и остальным содержимым документа.

**align** —выравнивание объекта в документе.Как и для элемента **IMG**, может принимать следующие значения:

**left** — объект прикрепляется к левому краю документа;

**right** — объект прикрепляется к правому краю документа;

**top** — выравнивание по верхней границе текущей строки текста;

**middle** — середина объекта должна находиться на уровне базовой линии строки текста;

**bottom** — выравнивание нижней границы объекта производится по базовой линии текущей строки. Это значение используется по умолчанию.

**id** — задает имя объекта в пределах документа;

**tabindex** — номер объекта в последовательности перехода по клавише "**Tab**"; **style** — стилевые установки объект;

**usemap** — ЭТОТ атрибут связывает данный объект с навигационной картой. В качестве значения этого атрибута указывается имя навигационной карты;

**onclick**, **ondblclick**, **onmousedown**, **onmouseup**, **onmouseover**, **onmousemove**, **onmouseout**, **onkeypress**, **onkeydown**, **onkeyup** — внутренние события.

## Элемент **PARAM**

С помощью элементов **PARAM** задается список параметров и их значений, которые будут переданы подключенному модулю для задания параметров обработки содержимого объекта. Внутри одного элемента **OBJECT** может быть определено любое количество элементов **PARAM** и в любом порядке. Начальный тег элемента **PARAM** является обязательным, конечный тег — запрещен.

### *Список атрибутов*

**name** — в качестве значения этого атрибута указывается имя параметра, которое должно быть понятно подключенному модулю. Учитывает ли это имя регистр, определяется модулем, которому он предназначен.

**value** — задает значение параметра, чье имя задано атрибутом **name**. Принимаемые атрибутом **value** значения не определены в HTML, а определяются подключенным модулем.

**valuetype** — указывает тип атрибута **value**. Может принимать следующие значения:

**data** — задание этого типа говорит о том, что указанное для атрибута value значение будет определяться и передаваться подключенному модулю в виде строки. Это значение используется по умолчанию.

**ref** — указывает на то, что значением атрибута value является URL-адрес файла, в котором хранятся значения параметров.

**object** — реализует ссылку на другой объект в этом же документе. В качестве значения указывается id другого объекта.

**type** — указывает тип содержимого (content-type) ресурса, чей URL-адрес указан атрибутом value. Используется только в том случае, если для атрибута valuetype указано значение "ret".

**id** — задает имя элемента в пределах документа.

Не все браузеры поддерживают элемент OBJECT. К тому же формат данных, используемых в качестве содержимого объекта может быть неизвестен браузеру или у него может не быть требуемого подключаемого модуля. Для этих случаев разработчики HTML предусмотрели задание альтернативного содержимого объекта. Все, что задано между начальным и конечным тегами элемента OBJECT и не является элементами PARAM, воспринимается браузером как альтернативное содержимое объекта. Вообще, возможны три варианта обработки элемента OBJECT и его содержимого браузером:

- браузер может определить тип содержимого объекта и у него имеется необходимый подключенный модуль. В этом случае браузер включает объект в документ и обрабатывает все элементы PARAM, заданные в содержимом элемента OBJECT. Прочие элементы, имеющиеся в содержимом этого элемента и являющиеся его альтернативным содержимым, игнорируются.
- браузер может определить тип содержимого объекта, но у него отсутствует необходимый подключаемый модуль. В такой ситуации браузер пытается, с предварительного согласия пользователя, подгрузить требуемый модуль с адреса, указанного атрибутом classid или в установках самого браузера. Если ему это удастся, то ситуация переходит в описанную выше. Если браузеру не удастся подгрузить нужный модуль, то элемент OBJECT и содержащиеся в нем элементы PARAM игнорируются. Отображению подлежат все остальные элементы, находящиеся в содержимом элемента OBJECT и являющиеся его альтернативным содержимым.
- браузер изначально не может определить тип содержимого объекта. Здесь все совсем просто: элемент OBJECT вместе со своими элементами PARAM игнорируется, а отображается его альтернативное содержимое.

#### Пример задания

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>

<HEAD>

<TITLE>Пример использования элемента OBJECT</TITLE>

</HEAD>

<BODY bgcolor=white>

....текст документа ....

....текст документа ....

....текст документа.... <BR>

«OBJECT data="tutut.por" width=300 height=200 align=left hspace=20» <PARAM name="parametr1"
value="12"> <PARAM name="parametr2" value="#r@r">

<H2>Альтернативное содержимое</H2>
```

```

<H3>Данный объект не может быть показан браузером, вместо этого посмотрите на картинку</H3> <IMG
src="image.gif"> </OBJECT>

... текст документа ....

... текст документа ....

... текст документа.... <BR>

</BODY> </HTML>

```

Как уже упоминалось ранее, в качестве альтернативного содержимого элемента OBJECT может быть использован другой элемент OBJECT. При этом элементы PARAM альтернативного элемента OBJECT игнорируются первичным элементом OBJECT.

---

#### Пример задания

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<HTML>

<HEAD>

<TITLE>Пример использования элемента OBJECT</TITLE>

</HEAD>

<BODY bgcolor=lightgrey>

... текст документа ....

... текст документа.... <BR>

«DBJECT data="bob.tul" width=300 height=200 align=left hspace=20>

<PARAM name="parametr1" value="12">

<PARAM name="parametr2" value="#r@r">

<H2>Альтернативное содержимое</H2>

<OBJECT data="bob.avi" align=left hspace=20>

<PARAM name=AUTOSTART value=false>

</OBJECT>

</OBJECT>

... текст документа ....

... текст документа ....

... текст документа ....

</BODY> </HTML>

```

# Элемент EMBED

Использование этого элемента является предпочтительным ввиду более корректной поддержки его наибольшим количеством браузеров, и более простой работой с ним.

При использовании элемента EMBED задание начального тега является обязательным, конечный тег запрещен.

*Список атрибутов элемента EMBED:*

**src** — указывает URL-адрес мультимедиа-файла, в котором хранятся данные объекта.

**type** — значением этого атрибута указывается MIME-тип содержимого мультимедиа-файла. Этот атрибут является необязательным, но его использование настоятельно рекомендуется. Задание атрибута type позволяет браузеру вообще загружать мультимедиа-файлы, обработать которые он не в состоянии. Иначе он сначала загрузит этот файл, а потом попытается определить его MIME-тип.

**border** — атрибут, задающий толщину рамки в пикселах вокруг объекта.

**frameborder** — логический атрибут, который указывает наличие или отсутствие рамки вокруг объекта. Может принимать значения "Yes" и "No". По умолчанию используется значение "No".

**height** — задает вертикальный размер в пикселах прямоугольной области, отводимой под данный объект в документе.

**width** — ширина в пикселах прямоугольной области, отводимой под данный объект в документе.

**hspace** — атрибут, указывающий размеры в пикселах горизонтальных отступов между объектом и остальным содержимым документа.

**vspace** — атрибут, указывающий размер в пикселах вертикальных отступов между объектом и остальным содержимым документа.

**hidden** — логический атрибут, наличие которого делает объект невидимым.

**pluginspage** — указывает URL-адрес ресурса, в котором содержатся инструкции по установке модуля.

**align** — выравнивание объекта. Принимаемые значения такие же как и для элемента OBJECT.

**id** — задает имя объекта в пределах текущего документа.

*Пример простейшего использования (рис. 3.105)*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<HTML>

<HEAD>

<TITLE>
```

Пример простейшего использования элемента EMBED

```
</TITLE>

</HEAD>

<BODY>

<H2 align=center>

Пример безрезультатной работы головой

</H2>

<HR>

<HR color=lightgrey size=20>

<CENTER>

<EMBED src="headbrake[1].avi">

</CENTER>

</BODY>

</HTML>
```

С помощью атрибутов width и height нельзя добиться искажения видеоклипа, как это можно сделать с изображением. Размеры кадра видеоклипа изменяются пропорционально друг другу, то есть, если один из них, например, увеличивается в два раза, то и второй должен увели-

читься в два раза. Если второй этого сделать не может из-за ограничения пространства, то и первый тогда увеличиваться не будет. Проще говоря, с помощью атрибутов width и height можно осуществлять только увеличение или уменьшения масштаба видеоизображения.

Например, если в предыдущем примере строку `<EMBED src="headbrake[1].avi">` заменить на строку `<EMBED src="headbrake[1].avi" width=100>`, то получится следующее (рис. 3.106).

Если заменить на `<EMBED src="headbrake[1].avi" width=500>`, то следующее (рис.

3.107).

Чтобы гарантированно увеличить видеоизображение, надо задать оба атрибута: width и height.



Параметры, передаваемые подключенному модулю вместе с мультимедиа-файлом

указываются в самом теге элемента EMBED. Например:

В этом случае подключенному модулю будет передан параметр autostart со значением false. При использовании элемента OBJECT для этих же целей понадобилась бы следующая запись:

У браузеров обычно изначально есть подключенные модули для воспроизведения внедренных в HTML-документы видеоклипов и звуковых фрагментов. У Netscape Communicator это:

- Live Audio — для воспроизведения звуковых файлов; \* Live Video — для воспроизведения видеофайлов. У Internet Explorer это:
- ActiveMovie — воспроизводит аудио- и видеофайлы.

Подключаемые модули, призванные выполнять идентичные функции, обычно используют одноименные параметры.

Модулям, воспроизводящим видеофайлы могут быть переданы следующие параметры:

- **autostart** — автоматическое воспроизведение, может принимать следующие значения:
- **true** — автоматически начинать воспроизведение после открытия документа. Используется по умолчанию.
- **false** — отключить автоматическое воспроизведение, **loop** — задает многократное воспроизведение:
- **true** — установить многократное воспроизведение
- **false** — отключить многократное воспроизведение. Это значение используется по умолчанию.
- **controls** — указывает наличие или отсутствие панели управления. Принимает следующие значения:
- **true** — отображать панель управления. Используется по умолчанию.



- **false** — не отображать панель управления.

Модули, воспроизводящие аудиофайлы, используют в своей работе параметры, приведенные ниже. Далее перечислены параметры модуля LiveAudio для Netscape Communicator. Для других модулей могут использоваться такие же параметры, просто под иным названием (или даже под этими):

- **autostart** — автоматическое воспроизведение, может принимать следующие значения:
- **true** — автоматически начинать воспроизведение после открытия документа. Используется по умолчанию.
- **false** — отключить автоматическое воспроизведение, **autoload** — атрибут, задающий параметры загрузки аудиофайла.
- **true** — автоматически загружать файл при открытии документа. Используется по умолчанию.
- **false** — не загружать файл автоматически.

**starttime="mm:ss"** — задает смещение начала воспроизведения звукового фрагмента в минутах (mm) и секундах (ss) от его начала.

**endtime="mm:ss"** — указывает смещение окончания воспроизведения звукового фрагмента в минутах (mm) и секундах (ss) от его начала. По умолчанию фрагмент будет проигран до конца.

- **volume** — параметр, устанавливающий громкость воспроизведения звукового фрагмента в процентах от максимальной.
- **controls** — параметр, с помощью которого задается внешний вид панели управления (консоли). Для этого параметра определены следующие значения:
- **console** — будет отображен полный набор кнопок управления. Используется по умолчанию.
- **smallconsole** — консоль будет отображена в более компактном виде.

Кнопка "пауза" при этом будет отсутствовать.

- **playbutton** — вместо консоли будет отображена только кнопка "Воспроизведение".
- **pausebutton** — отображена только кнопка "Пауза".
- **stopbutton** — отображена только кнопка "Стоп".
- **volumelevel** — отображен будет только регулятор громкости.

Сейчас на сервере Natscape.com можно скачать следующие подключаемые модули (Plug-ins) для проигрывания звуковых фрагментов в MP3-формате:

- Apple QuickTime 4 Plug-in by Apple Computer, Inc, Crescendo by Live Update;
- Winamp MP3 Player by Nullsoft.

Стоит отметить, что атрибуты **align**, **width** и **height**, применительно к звуковым файлам, относятся к панели управления, задавая, соответственно, ее выравнивание, ширину и высоту. Однако размеры консоли являются фиксированными для каждого ее вида. Поэтому пространство, выделяемое атрибутами **width** и **height**, должно быть не меньше размеров консоли. Если больше — то просто останется свободное пространство (в Netscape Communicator). В Internet Explorer, поскольку для воспроизведения видео и аудиофайлов используется одна и та же программа

ActiveMovie, то явное задание размеров консоли может привести к показу черного экрана при воспроизведении звукового фрагмента, если указаны большие размеры. Пример использования:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

```
<HTML>
```

```
<HEAD>
```

```

<TITLE>

Демонстрация различных консолей при включении в документ аудиофайлов

</TITLE>

</HEAD>

<BODY>

<H3 align=center>Примеры консолей</H3>

<HR>

<CENTER>

<EMBED src="polet.mp3">

<HR>

<EMBED src="polet.mp3" autostart=false controls=sraallconsole>

<HR>

<EMBED src="polet.mp3" autostart=false controls=startbutton>

<H! >

<EMBED src="polet.mp3" autostart=false controls=playbotton loop=true>

<HR>

<EMBED src="polet .mp3" width=350 neight=200 autostart=false controls=playbotton>

</CENTER>

</BODY>

</HTML>

```

Надо отметить следующий важный факт. Внедрение в документ видеоклипов и больших звуковых файлов является нежелательным, так как из-за этого значительно увеличивается размер памяти, занимаемой документом, а, значит, и увеличивается время его загрузки. Гораздо эффективнее просто задавать в документе ссылки на видео- и аудиофайлы, с кратким их описанием. Такой документ загружается быстро и пользователь может сам выбрать, какой из мультимедиа-файлов загрузить.

Если же видео или аудио встроены в документ, то они загружаются автоматически вместе с документом. Пример использования ссылок на объекты (рис. 3.110).

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">

<HTML>

<HEAD>

<TITLE>

Демонстрация использования ссылок на объекты

```

```

</TITLE>

</HEAD>

<BODY bgcolor=lightgrey> <H1 align=center>Видеоприколы</H1>

<H3 align=center>В данном документе представлено несколько видеоприколов .</H3>

<CENTER>

Для просмотра видеоклипа нажмите на кадр из него <TABLE cellpadding=5 bgcolor=darkgray
width=600> <TR>

<TH><A href="parking[1] .mpg" xIMG src="clip1.gif" width=180></A><BR> <I>Крутая парковка

<TH><A href="horseriding[1] .mpg" xIMG src="clip2.gif" width=180> </A> <BR>

<I>Лихой всадник <TR>

<TH><A href="kick[1] .mpg"><IMG src="clip3 .gif" width = 180x/A><BR> <I>Ох, уж эти женщины

<TH><A href = "headbrake[1] .avi"><IMG src="clip4 .gif" width=180x/A>

<BR><I>Железная голова </TABLE> </CENTER> </BODY> </HTML>

```

Кстати, все видеоприколы содержатся на прилагаемом к книге CD. Имеет смысл включать в свои документы видео и аудиофрагменты в качестве RealAudio и RealVideo. При этом использоваться будут техноло-

гии потоковой передачи данных. Это значит, что, например, показ видеоролика будет осуществляться параллельно с его загрузкой, то есть в реальном времени. Для поддержки такой возможности у пользователя должен быть установлен программный продукт под названием RealPlayer 8 Plus. Внешний вид проигрывателя показан на рис.3.111.

Этот программный продукт является условно бесплатным. Его вы можете скачать с сервера [www.real.com](http://www.real.com). и бесплатно наслаждаться им в течение 30 дней. Потом предполагается, что вы будете в таком восторге, что с радостью выложите 29.99 долларов, лишь бы с ним не расставаться.

Подробнее о потоковых технологиях можно прочитать в следующей части книги, посвященной мультимедиа в Интернете.

### *Элемент APPLET*

Этот элемент служит для включения в HTML-документ Java-аппле-тов. Использование его является нежелательным ввиду предпочтительного использования для этих целей элемента OBJECT.

*Список атрибутов:*

**codebase** — этим атрибутом задается базовый URL-адрес апплета. Если этот атрибут не указан, то по умолчанию используется базовый URL-адрес самого документа. Значениями этого атрибута могут быть только подкаталоги каталога, в котором расположен текущий документ, либо сам этот каталог.

**code** — этот атрибут задает имя файла класса, содержащего скомпилированный подкласс апплета или адрес, по которому можно получить этот класс. Значение атрибута интерпретируется с учетом кода апплета.

**name** — в качестве значения этого атрибута указывается уникальное имя аппле- та, по которому осуществляется обращение к нему.

**width** — этот атрибут задает рекомендуемую ширину области, отведенной под апплет на странице.

**height** — этот атрибут задает рекомендуемую высоту области, отведенной под апплет на странице.

**hspace, vspace** — указывает горизонтальные и вертикальные отступы от апплета соответственно. Значение этих атрибутов задается в пикселах и позволяет вокруг изображения оставлять свободное пространство определенных размеров.

**alt** — атрибут, имеющий в качестве значения текстовую строку, которая является

альтернативной текстовой информацией апплета.

**style** — встроенная информация о стиле;

**title** — заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);

**id, class** — идентификаторы в пределах документа.

*Пример задания:*

```
<APPLET code="Fire.class" width="600" height="200"> Java-апплет, рисующий интерактивный огонь
</APPLET>
```

Вместо этого разработчиками рекомендуется использовать элемент OBJECT:

```
<OBJECT codetype="application/java" classid="Java:fire.class"
width="600" height="200">
Java-апплет, рисующий интерактивный огонь
</OBJECT>
```

Параметры для апплета, также как и для элемента OBJECT, задаются с помощью вспомогательного элемента PARAM.

```
<APPLET code="Fire.class" width="600" height="200"> <PARAM name="hyt" value="23"> <PARAM
name="blink" value="yes"> Java-апплет, рисующий интерактивный
огонь </APPLET>
```

Так же, как и для элемента OBJECT, все, что находится между начальным и конечным тегами элемента APPLET и не является элементом PARAM, воспринимается как альтернативное содержимое.