Глава 8. Объекты

Мультимедиа возможности HTML позволяют размещать на странице не только изображения, но и аплеты (программы, которые автоматически загружаются и запускаются на выполнение на машине пользователя), видеоклипы и другие документы HTML.

Большинство браузеров имеют встроенные механизмы для отображения данных обычных типов, таких как рисунки в формате GIF, цвета, шрифты и небольшое количество графических элементов. Чтобы иметь возможность отображать данные, изначально не поддерживаемые, браузеры обычно запускают внешние приложения. Элемент **object** позволяет контролировать, должны ли данные просматриваться независимо (вне браузера) или программой определённой автором для просмотра внутри браузера.

В самом общем случае, необходимо определить три типа информации:

- Реализацию (класс) включённого объекта. Например, если включаемый объект - это аплет, нужно указать, где находится исполняемый код аплета.
- Данные для отображения, например, если включаемый объект - это программа, отображающая данные шрифтов, тогда нужно указать, где находятся эти данные.
- Дополнительные значения, необходимые для объекта на этапе выполнения. Например, некоторым аплетам требуются начальные значения атрибутов.

Элемент **object** позволяет задавать данные всех трёх типов, но не всегда необходимо определять их все. Например, некоторым объектам не нужны данные (аплет, выполняющий небольшую анимацию). Другие не требуют инициализации при выполнении. Наконец, третьи могут не требовать дополнительной информации о выполнении, т.е. браузер может уже сам "знать", как отображать данные такого типа (например, рисунки формата GIF).

Чтобы определить значения во время выполнения объекта, следует использовать элемент **param**, который рассматривается в разделе инициализация объекта.

Синтаксис

```
<object id="имя" declare classid="адрес" codebase="адрес"
codetype="тип_кода" data="адрес" type="тип_данных"
archive="список_адресов" standby="подсказка"
align=выравнивание width="целое" height="целое"
border="целое" hspace="целое" vspace="целое"
usemap="#имя_карты">
Альтернативный текст
</object>
```

Для объявления объекта так, чтобы он не исполнялся после считывания браузером, необходимо указать атрибут **declare**. Одновременно нужно идентифицировать объявление установкой уникального значения атрибута **id** элемента **object**. Последующие реализации этого объекта будут ссылаться на этот идентификатор.

Объявленный **object** должен появиться до первой реализации этого объекта. Объявления объектов до их исполнения чаще всего размещаются в заголовке HTML-документа (элемент **head**).

Атрибут **classid** может использоваться для определения размещения класса объекта посредством URI. Может использоваться вместе как альтернатива или с атрибутом **data**, в зависимости от типа объекта.

Атрибут **codebase** определяет базовый путь, используемый для разрешения относительных адресов (URI), определенных атрибутами **classid**, **data** и **archive**., Значением по умолчанию является базовый URI текущего документа.

Атрибут **codetype** определяет тип содержимого данных, ожидаемых при загрузке объекта, определённого атрибутом **classid**. Этот атрибут не обязателен, но рекомендуется, если **classid** определён, поскольку он позволяет браузеру избежать загрузки информации с неподдерживаемыми типами содержимого. При отсутствии, по умолчанию принимается значение атрибута **type**.

Атрибут **data** может использоваться для определения размещения данных объекта. Если задан как относительный URI, то должен интерпретироваться относительно атрибута **codebase**.

Атрибут **type** определяет тип содержимого данных в атрибуте **data**. Атрибут не обязателен, но рекомендуется, если **data** определён, то он позволяет браузеру избежать загрузки информации с неподдерживаемыми типами содержимого.

Атрибут **archive** может использоваться для определения разделённого пробелами списка URI для архивов, содержащих ресурсы, относящиеся к объекту. Список может включать ресурсы, определённые атрибутами **classid** и **data**. Предварительная загрузка архивов, как правило, уменьшает время загрузки объектов. Архивы, определённые как относительные URI, должны интерпретироваться относительно атрибута **codebase**.

Атрибут **standby** определяет сообщение, которое браузер может показывать при загрузке класса объекта и данных.

Остальные атрибуты имеют тоже значение и смысл, что и представленные в главе Графика (**usemap**, **align**, **width**, **height**, **border**, **hspace**, **vspace**)

8.1. Правила представления объектов

Браузер выполняет элемент **object** в соответствии со следующими правилами приоритета:

- браузер должен сначала попытаться отобразить объект;
- если браузер не может по каким-либо причинам отобразить объект (не сконфигурирован, недостаточно ресурсов, неверная архитектура и т.д.), он обязан попытаться отобразить содержимое объекта.

В примере ниже в документ вставлен аплет аналоговых часов с помощью элемента **object**. Аплет не требует дополнительных данных или значений для этапа выполнения:

```
<object classid="http://www.exclock.net/analogclock.py">
</object>
```

Часы будут отображены сразу, как только браузер обработает эти строки. Можно отсрочить отображение объекта, первоначально объявив его:

```
<object classid="http://www.exclock.net/analogclock.py">
Анимированные часы
</object>
```

Включить альтернативный текст в содержимое элемента **object** следует, если браузер не сможет отобразить объект в данном случае - часы.

Важным следствием дизайна элементов **object** является то, что он предоставляет механизм альтернативного представления объектов; каждое объявление внедрённого **object** может определять альтернативные типы содержимого. Если браузер не может отобразить наиболее поздние альтернативы **object**, он пытается отобразить содержимое, которое может быть другим элементом **object** и т.д.

Пример внедрения нескольких объявлений **object**, для демонстрации работы альтернативных представлений.

Браузер будет пытаться отобразить первый элемент **object**, который он сможет, в следующем порядке:

1. аплет Earth, написанный на языке Python,
2. MPEG-анимацию,
3. GIF-рисунок,
4. альтернативный текст.

```
<object   classid="http://www.observer.mars/TheEarth.py"  title="Земля  из
космоса">
<!-- В противном случае, MPEG video -->
  <object data="TheEarth.mpeg" type="application/mpeg">
<!-- В противном случае, рисунок GIF -->
    <object data="TheEarth.gif" type="image/gif">
<!-- В противном случае отображает текст -->
      Земля из космома.
    </object>
  </object>
</object>
```

Самое внешнее объявление определяет аплет, который не требует дополнительных данных или начальных значений. Второе объявление - это анимация MPEG и, поскольку не определяется размещение программы обработки MPEG, обращается к браузеру для отображения анимации (установлен атрибут **type**, чтобы браузер, не способный обработать MPEG, не загружал "TheEarth.mpeg" из сети). Третье объявление определяет размещение файла GIF и определяет альтернативный текст на тот случай, если все остальные механизмы не сработают.

Внутренние и внешние данные. Отображаемые данные могут быть получены двумя путями: внутренние (т.е. из самого документа) и из внешнего источника. Хотя предыдущий метод обычно приводит к более быстрому отображению, это не всегда бывает удобно при выводе данных большого объёма.

8.2. Инициализация объекта.

Элементы **param** определяют набор значений, которые могут потребоваться объекту на этапе выполнения. Элементы **param** могут появляться в содержимом элементов **object** в любом количестве, в любом порядке, но должны размещаться в начале содержимого элементов **object**.

Синтаксис
<object> **<param name="имя" value="значение" valuetype="тип_данных" type="тип_ресурса">** … **<param name="имя" value="значение" valuetype="тип_данных"**

```
type="тип_ресурса">

</object>
```

Атрибут **name** определяет имя этапа выполнения, принимаемого вставленным объектом. Является ли имя чувствительным к регистру, зависит от конкретной реализации объекта.

Атрибут **value** определяет значение этапа выполнения, опредленного в **name**.

Атрибут **valuetype** определяет тип атрибута **value**.

Возможные значения:

тип_данных	Пояснение
data	Значение по умолчанию. Означает, что определённое в **value** значение будет вычислено и передано в реализацию объекта как строка.
ref	Значением **value** является URI, указывающий на ресурс, где хранятся значения этапа выполнения. Это позволяет поддерживать утилиты идентификации URI, заданного в качестве атрибута. URI должен быть передан объекту как есть, т.е. без обработки.
object	Значением **value** является идентификатор, ссылающийся на объявление **object** в этом же документе.

Таблица 23. Значения атрибута valuetype

Атрибут **type** определяет тип содержимого ресурса, на который указывает атрибут **value**, только в том случае, когда **valuetype** установлен в "ref". Этот атрибут, таким образом, устанавливает для браузера тип значений, которые будут найдены по URI, указанному в **value**.

Например, если аплет может принять два атрибута этапа выполнения, определяющих его начальные ширину и высоту, то можно установить начальные размеры 40x40 пикселов с помощью двух элементов **param**:

```
<object classid="http://www.exclock.net/analogclock.py">
<param name="height" value="40" valuetype="data">
<param name="width" value="40" valuetype="data">
Данный браузер не может отобразить аплет.
</object>
```

Если элемент **object** отображён, браузер должен искать содержимое только тех элементов **param**, которые являются прямыми потомками и обеспечить ими **object**.

Так, в следующем примере, если "obj_1" отображён, "param_1" применяется к "obj_1". Если "obj_1" не отображён, а "obj_2" отображён, "param_1" игнорируется, а "param_2" применяется к "obj_2". Если ни один **object** не отображён, ни один **param** не применяется:

```
<object id="obj_1">
   <param name="param_1">
   <object id="obj_2">
      <param name="param_2">
   </object>
</object>
```

8.3. Глобальные схемы именования объектов

Размещение реализации объекта задаётся в URI. Для документов HTML эта схема, чаще всего, "http". Некоторые аплеты могут применять другие схемы именования. Например, при спецификации аплета Java, можно использовать URI, начинающиеся с "java", а для аплетов ActiveX - использовать "clsid".

Пример вставки аплета Java:

```
<object classid="java:program.start">
</object>
```

При установленном атрибуте **codetype**, браузер может решить, запрашивать ли приложение Java, чтобы выполнить аплет.

```
<object codetype="application/java-archive" classid="java:program.start">
</object>
```

Некоторые схемы отображения требуют наличия дополнительной информации для идентификации конкретной реализации, и поэтому им нужно сообщить, где эту информацию искать. Путь реализации объекта можно задать с помощью атрибута **codebase**.

Например:

```
<object codetype="application/java-archive" classid="java:program.start"
codebase="http://foooo.bar.com/java/myimplementation/">
</object>
```

В следующем примере определен в атрибуте **classid** объект ActiveX в URI. Атрибут **data** локализует данные для отображения:

```
<object classid="clsid:663C8FEF-1EF9-11CF-A3DB-080036F12502"
data="http://www.acme.com/ole/clock.stm">
Это приложение не поддерживается.
</object>
```

8.4. Объявление и размещение объектов

Если документ содержит более чем одну реализацию одного и того же объекта, можно разделить объявление объекта и его исполнение. Такой подход даёт определённые преимущества:

- Данные могут быть запрошены браузером из сети только один раз при объявлении и повторно использоваться для каждого исполнения.
- Можно размещать исполнение из другого места, например, из гиперссылки.
- Можно определять объекты как данные этапа выполнения других объектов.

Объект, объявленный с атрибутом **declare**, размещается каждый раз, когда элемент, который ссылается на этот объект, запрашивает его для отображения (например, гиперссылка, ссылающаяся на него, активирована, объект, ссылающийся на него, активирован и т.д.).

В следующем примере объявляется **object** и размещается вызовом гиперссылки. Таким образом, объект может быть активирован щелчком на выделенном тексте, например:

```
<object declare id="earth.declaration" data="TheEarth.mpeg"
type="application/mpeg">
<strong>Земля</strong> - вид из космоса.
</object>
...позже в документе...
<p>Пример <a href="#earth.declaration"> анимации Земли</a>
```