**Глава 5. ЭЛЕМЕНТ AUDIO**

**5.1. Принципы применения элемента AUDIO**

Элемент <audio> – новый элемент языка HTML5, предназначенный для воспроизведения аудиозаписей. По своей структуре и принципу применения элемент практически не отличается от <video>: в большинстве своем совпадают атрибуты, события, свойства и методы. Отличие, только в специфических для элемента <video> моментах. Например, не применяются атрибуты width, height (размеры кадра), poster (начальная картинка) и пр.

**5.2. Форматы аудиофайлов**

Основные форматы аудиофайлов, которые может воспроизводить элемент <audio> представлены в табл. 5.1. Как и в случае с видеофайлами, аудиофайл представляет собой закодированные с целью уменьшения размера данные, предназначенные для воспроизведения специальным оборудованием компьютера .

Таблица 5.1. Наиболее распространённые форматы

аудиофайлов

|  |  |
| --- | --- |
| Аудио-формат | Тип медиа |
| mp3 | audio/mpeg |
| ogg | audio/ogg |
| wav | audio/wav |

СпецификацияHTML5 допускает применение любого аудио-формата. В реальности, все зависит от реализации браузера, которая определяет перечень поддерживаемых им форматов (табл. 5.2).

Таблица 5.2. Поддержка аудио-форматов браузерами

|  |  |  |  |
| --- | --- | --- | --- |
| Браузер | mp3 | ogg | wav |
| Internet Explorer | + | - | - |
| Chrome | + | + | + |
| Firefox | - | + | + |
| Safari | + | - | + |
| Opera | - | + | + |

В теле элемента <audio> может быть указано несколько элементов <source>, задающих альтернативные источники входных данных (аналогично элементу <video>). Браузер проверяет их в порядке перечисления и воспроизводит первый, который может.

Как и в случае с <video> в элементе <audio> источник данных может быть задан в формате URL. В этом случае доставка содержимого видеофайла будет осуществляться через сеть по протоколу HTTP.

**5.3. Применение элемента AUDIO и Audio API**

Ввиду схожести элемента <audio> с элементом <video> ограничимся кратким разбором простого одностраничного web-приложения, применяющего элемент <audio>. Возьмем за основу html-страницу, представленную на рис. 5.1 и 5.2.



Рис. 5.1. Html-страница, демонстрирующая применение элемента <audio>

Не сложно заметить, что структура html-страницы (рис. 5.1) очень похожа на структуру html-страницы (рис. 4.10), а синтаксис <audio> и <video> практически аналогичен. Применяемый на странице JavaScript-код располагается в файле Audio.js, который загружается элементом <script>, расположенным в начале html-страницы.



Рис. 5.2. Отображение браузером html-страницы,

представленной на рис. 5.1

На рис. 5.3 представлен фрагмент файла Audio.js, содержащий объявление анонимной функции, выполняющейся сразу после загрузки html-страницы (событие load объекта window).

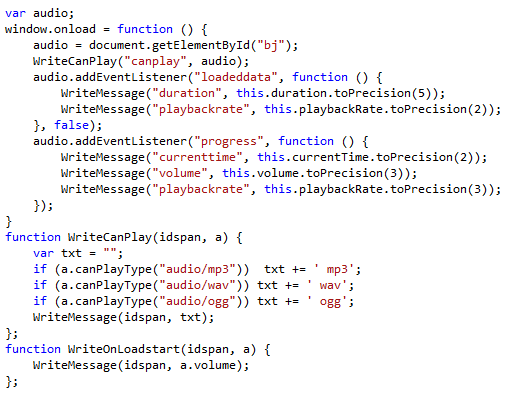


Рис. 5.3. Фрагмент файла Audio.js: анонимная функция –

обработчик события load объекта window

Код анонимной функции практически не отличается от кода на рис. 4.12: сначала вызывается функция WriteCanPlay, которая с помощью метода canPlayType объекта <audio> проверяет возможность применения нескольких аудио-форматов, затем задаются две функции, предназначенные для обработки событий loadeddata (при каждой загрузке аудиоданных) и progress (при загрузке каждой порции аудиоданных).

Атрибутом onloadstart элемента <audio> задается функция – обработчик WriteOnLoadstart соответствующего события. Функция вызывается один раз в момент начала первой загрузки аудиоданных и записывает значение свойства volume (коэффициент уровня громкости) в тело заданного первым параметром (идентификатор) элемента <span>.

Работа трех заданных обработчиков событий элемента <audio> ничем не отличается от одноименных обработчиков элемента <video>, уже рассмотренных в 4.5: с помощью функции WriteMessage они отображают значение свойств duration (продолжительность воспроизведения), playbackRate (коэффициент скорости воспроизведения) currenttime (текущая позиция воспроизведения) и volume (коэффициент громкости звука).

На рис. 5.4 представлен фрагмент файла Audio.js, содержащий функции, вызываемые при нажатии клавиш, отображаемых элементами <input type=”button”> (рис. 5.1, 5.2).

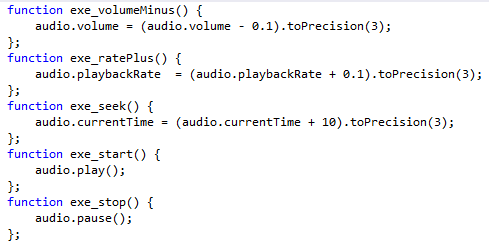


Рис. 5.4. Фрагмент файла Audio.js: функции, предназначенные для обработки события click элементов <input type=”buttom”> (рис. 5.4)

Первые три клавиши, вызывают функции меняющие свойства объекта элемента <audio>: нажатие клавиши «-Громкость» приведет к уменьшению значения свойства volume на 0.1 (уменьшению уровня громкости звука); клавиши «+Скорость» – к увеличению значения свойства playbackRate на 0.1 (увеличение скорости воспроизведения) и клавиши «Смещение» – к увеличению значения свойства playbackRate на 10 (смещение вперед по аудиоданным на 10 сек. воспроизведения).

Следующие две клавиши позволяют запустить (клавиша «Пуск») и остановит воспроизведение (клавиша «Стоп»). При нажатии клавиши «Пуск» вызывается функция exe\_start, которая в свою очередь вызывает метод play, запускающий воспроизведение аудиофайла. При нажатии клавиши «Стоп» вызывается функция exe\_stop, которая в свою очередь вызывает метод pause, останавливающий воспроизведение аудиофайла. Заметим, что методы play и pause могут применяться и для элемента <video>.