Quick guide to implementing the HTML5 audio tag

- Step 2: Add the source reference
- Step 3: Add fallback to Flash
- Step 4: Add the default controls to show the player

Those browsers that don't recognize the audio tag will load the Flash content instead.

```
<audio>
</span><span class="old">
    <object class="playerpreview" type="application/x-shockwave-flash"
            data="player_mp3_mini.swf" width="200" height="20">
      <param name="movie" value="player_mp3_mini.swf" />
      <param name="bgcolor" value="#085c68" />
      <param name="FlashVars" value="mp3=test.mp3" />
      <embed href="player_mp3_mini.swf" bgcolor="#085c68" width="200"
             height="20" name="movie" align=""
             type="application/x-shockwave-flash" flashvars="mp3=test.mp3">
      </embed>
    </object>
</span><span class="new">
</audio>
```

Step 2: Add the source reference

We can add as many "source" lines and formats as we want. If the browser doesn't support one specific format it will fallback to the next one and so forth.

```
<span class="old"><audio></span>
  <span class="new"><source src="test.mp3" type="audio/mpeg" />
  <source src="test.ogg" type="audio/ogg" /></span><span class="old">

  <object class="playerpreview" type="application/x-shockwave-flash"
          data="player_mp3_mini.swf" width="200" height="20">
    <param name="movie" value="player_mp3_mini.swf" />
    <param name="bgcolor" value="#085c68" />
    <param name="FlashVars" value="mp3=test.mp3" />
    <embed href="player_mp3_mini.swf" bgcolor="#085c68" width="200"
           height="20" name="movie" align=""
           type="application/x-shockwave-flash" flashvars="mp3=test.mp3">
    </embed>
  </object>

</audio></span>
```

Step 3: Add fallback to Flash

To be safe, we need to add the fallback to a Flash audio player, in case the browser doesn't support any of the formats we specified. For instance, Firefox 3.5 only supports the audio tag with **Ogg** format, but we might only have the **mp3** file available.

There are also tools and online converters you can use if you want to create ogg files from your mp3 and add support for ogg too.

```
<span class="old"><audio></span>
<span class="new"><source src="test.mp3" type="audio/mpeg" /></span><span class="old">

<object class="playerpreview" type="application/x-shockwave-flash"
        data="player_mp3_mini.swf" width="200" height="20">
<param name="movie" value="player_mp3_mini.swf" />
<param name="bgcolor" value="#085c68" />
<param name="FlashVars" value="mp3=test.mp3" />
<embed href="player_mp3_mini.swf" bgcolor="#085c68" width="200"
        height="20" name="movie" align=""
        type="application/x-shockwave-flash" flashvars="mp3=test.mp3">
</embed>
</object>
```

```
</audio>
</span><span class="new">
<div id="player_fallback"></div>
<script>
if (document.createElement('audio').canPlayType) {
if (!document.createElement('audio').canPlayType('audio/mpeg')) {
    swfobject.embedSWF(
    "player_mp3_mini.swf",
    "player_fallback",
    "200",
    "20",
    "9.0.0",
    "",
    {"mp3":"test.mp3"},
    {"bgcolor":"#085c68"});
}
}
</script></span>
```

To make it easier, we are using the SWFObject library to insert the Flash player via JavaScript. To include the library you can simply use the Google AJAX Libraries API inserting these two lines in your header:

```
<span class="new"><script src="http://www.google.com/jsapi"></script>
<script>google.load("swfobject", "2.2");</script></span></pre>
```

Step 4: Add the default controls to show the player

These controls are not customizable (see examples at the end). Since these default controls will show up regardless of the supported format we will need to handle its visibility with the conditional we previously created.

```
<span class="old"><audio </span><span class="new">id="audio_with_controls"
controls</span><span class="old">></span>
<span class="old"><source src="test.mp3" type="audio/mpeg" /></span><span class="old">
```

```
<object class="playerpreview" type="application/x-shockwave-flash"
        data="player_mp3_mini.swf" width="200" height="20">
<param name="movie" value="player_mp3_mini.swf" />
<param name="bgcolor" value="#085c68" />
<param name="FlashVars" value="mp3=test.mp3" />
<embed href="player_mp3_mini.swf" bgcolor="#085c68" width="200"
        height="20" name="movie" align=""
        type="application/x-shockwave-flash" flashvars="mp3=test.mp3">
</embed>
</object>

</audio>
</span><span class="old">
<div id="player_fallback"></div>
<script>
if (document.createElement('audio').canPlayType) {
if (!document.createElement('audio').canPlayType('audio/mpeg')) {
    ... SWFObject script line here ...
    </span><span class="new">document.getElementsById('audio_with_controls').style.display =
'none';</span><span class="old">
}
}
</script></span>
```

Alternatively, you can create your own player using JavaScript and CSS.

```
<span class="old"><audio </span><span class="new">id="audio"</span><span class="old">></span>
<span class="old"><source src="test.mp3" type="audio/mpeg" /></span><span class="old">

<object class="playerpreview" type="application/x-shockwave-flash"
        data="player_mp3_mini.swf" width="200" height="20">
<param name="movie" value="player_mp3_mini.swf" />
<param name="bgcolor" value="#085c68" />
<param name="FlashVars" value="mp3=test.mp3" />
<embed href="player_mp3_mini.swf" bgcolor="#085c68" width="200"
        height="20" name="movie" align=""
        type="application/x-shockwave-flash" flashvars="mp3=test.mp3">
</embed>
</object>
```

```
</audio>
</span><span class="old">
<div id="player_fallback"></div>
<span class="new"><div id="player" style="display: none">
<button onClick="document.getElementById('audio').play()">Play</button>
<button onClick="document.getElementById('audio').pause()">Pause</button>
</div></span>

<script>
if (document.createElement('audio').canPlayType) {
if (!document.createElement('audio').canPlayType('audio/mpeg')) {
    ... SWFObject script lines here ...
} </span><span class="new">else { // HTML5 audio + mp3 support
    document.getElementById('player').style.display = 'block';
}</span><span class="old">
}
</script></span>
```

□□□ #3
□ □□□ □□□ 2026-10-31 21:38:39 UTC
□ cocobook-sync □□□ 2026-11-04 17:36:34 UTC