

HTML5 - Extra

- Dr Nick Hayward

A brief overview of some of the extras available with HTML5.

Contents

- Intro
- Media elements
 - `<video>` element
 - example usage - `<video>` element
 - `<audio>` element
 - example usage - `<audio>` element
- `<canvas>` element
 - example usage - parts 1 and 2
- References

Intro

In addition to the semantic HTML5 elements, there are also some very interesting and useful new additions. Some struggle from a lack of browser compatibility, but many of these new elements are useful and fun to use. For example, we have HTML5 graphics elements

```
canvas
svg
```

and HTML5 media elements. There are also HTML5 APIs for

- App Cache
- Drag/Drop
- Geolocation
- Local Storage
- ...

Again, each of these will be dependent upon browser support and compatibility. For example, we may use the following online check for browser support of HTML5 elements

- [Can I Use_____?](#)
 - e.g. [Can I Use Drag and Drop?](#)

Media elements

The two primary elements for HTML5 media allow us to add audio and video playback to a web page.

`<video>` element

Until the introduction of HTML5, online video playback was reliant upon custom browser plugins, such as Adobe's Flash player. However, the tide has now started to turn, and more and more web sites are using HTML5's `<video>` element to embed video content.

In fact, in early November 2011, Adobe announced they would be ceasing development of their mobile Flash plugin, instead deciding to focus upon development of tools for HTML 5. It's still in its final death throes, just, but browser

support is finally starting to wane and end.

So, to embed video using HTML5 we can use the `<video>` element.

We can also supply attributes to specify the height and width of our video, and default controls for playback. Now, this may sound like plain sailing, but there is a small caveat. Not all modern browsers support all video codecs. However, there is the option to specify multiple sources for a video, which allows a developer to offer streaming video to support most of the major browsers.

Providing your video is available in either *MP4* (or H.264), *WebM*, and *OGG* you'll be able cover all the major modern browsers. You'll also find good support for HTML5 videos in mobile browsers such as Chrome on Android, and Safari on iOS.

We can check our browser's support for `<video>` using the following tool,

- [Can I use ____ video?](#)

example usage - `<video>` element

`<video>` - a quick example might be as follows,

```
<video width="300" height="240" controls>
  <source src="media/video/movie.mp4" type="video/mp4">
  <source src="media/video/movie.webm" type="video/webm">
  Your browser does not support the video tag.
</video>
```

`<audio>` element

HTML5 also supports a standardised element for embedding audio. There are 3 main, supported codecs for use with the `<audio>` elements.

- MP3 and mp4
- WAV
- OGG Vorbis
- 3GP
- m4a

Again, before using this element on your website you'll need to check that it is supported by your required browsers, and use fall back codecs as well. Test URL is as follows,

- [Can I use ____ audio?](#)
- [HTML5 Audio](#)

example usage - `<audio>` element

`<audio>` - a quick example might be as follows,

```
<audio controls>
  <source src="media/audio/audio.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```

<canvas> element

HTML5's new graphics elements are particularly fun to use. We may use them to create interesting and useful graphics renderings, games, and visualisations. In effect, we can now easily draw various graphics on a web page.

The `<canvas>` element acts as a placeholder, or canvas in effect, to allow us to draw with JavaScript. We can draw lines, a circle, text, add gradients, and so on. For example, we can draw a rectangle on the canvas, and then use any of the pixels within this defined space.

example usage - `<canvas>` element - part 1

So, our canvas will be created as follows,

```
<canvas id="canvas1" width="200" height="100">
  Your browser does not support the canvas element.
</canvas>
```

and then we use JavaScript to add the required drawing to this new canvas element

```
<script type="text/javascript">
var can1 = document.getElementById("canvas1");
var context1 = can1.getContext("2d");
context1.fillStyle="#000000";
context1.fillRect(0,0,150,75);
</script>
```

example usage - `<canvas>` element - part 2

Now, if we wanted to create a square instead we would need to modify our JavaScript as follows

```
<script type="text/javascript">
function draw() {
/*black square*/
var can1 = document.getElementById("canvas1");
var context1 = can1.getContext("2d");
context1.fillStyle="#000000";
context1.fillRect(0,0,50,50);
}
</script>
```

This type of drawing can be modified for many different shapes and patterns, including simple lines, circles, gradients, images, and so on.

References

- [HTML5 Audio formats](#)
- [HTML5 Test](#)
- [W3C](#)
 - [HTML5 Documentation](#)
- [W3 Schools](#)
 - [W3Schools - HTML5 Semantic Elements](#)