

10

EMBEDDED MEDIA

OVERVIEW

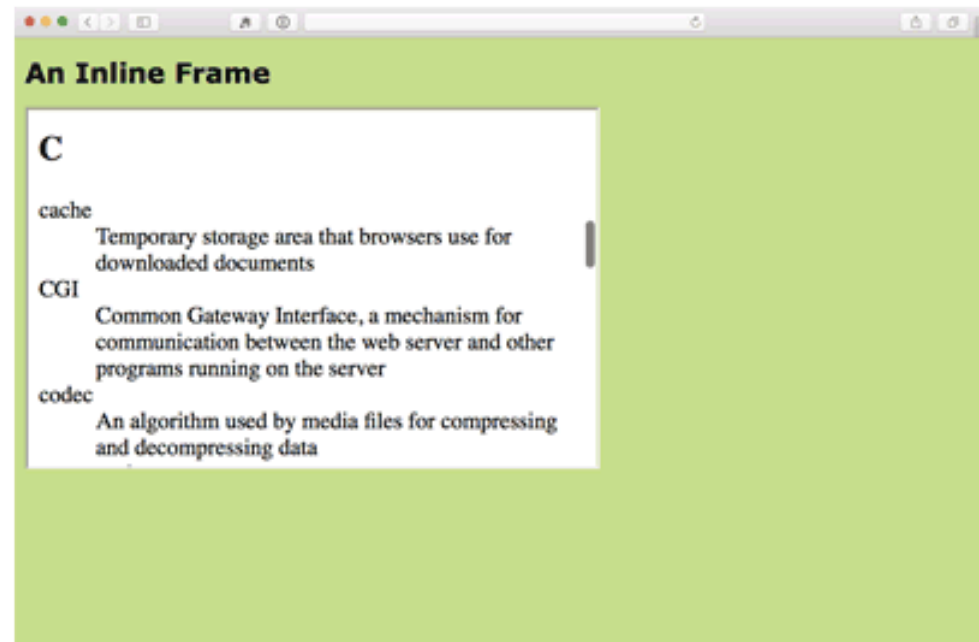
- **iframes**
- **Generic object element**
- **Video players**
- **Audio players**
- **Canvas**

Window-in-a-Window (iframe)

`<iframe> </iframe>`

- The **iframe** element lets you embed a separate HTML document on a page.
- It displays in its own window-in-a-window (called a **nested browsing context**).
- Commonly used to embed videos, maps, and advertising

iframe (cont'd)



```
<h1>An Inline Frame</h1>
```

```
<iframe src="glossary.html" width="400" height="250">
```

```
  Your browser does not support inline frames. Read  
  the
```

```
<a href="glossary.html">glossary</a>.
```

```
</iframe>
```

- The content of the **iframe** element is a fallback that displays if iframe is not supported.
- The **width** and **height** attributes specify the size of the window in pixels.

The object Element

`<object> </object>`

`<param> </param>`

```
<object data="picture.svg" type="image/svg+xml">  
    
</object>
```

- A generic element for embedding media on a web page. The media may require a plug-in to run.
- The **data** attribute points to the media file.
- The **type** attribute is the type of media (its MIME type).
- The content of the **object** element is a fallback if the media type isn't supported.

The object Element (cont'd)

```
<object type="video/quicktime" data="movies/hekboy.mov"
width="320" height="256">
  <param name="autostart" value="false">
  <param name="controller" value="true">
</object>
```

Some media formats may require that the **object** contain a number of **param** elements that set parameters specific to that type of media.

Example: Starting a video automatically or displaying player controls

Embedded Video

`<video> </video>`

HTML5 introduced the **video** element for embedding a video player on a web page. There is also an API for controlling video play.



```
<video src="windtunnel.mp4" width="320" height="262"
poster="windtunnel_still.jpg" controls autoplay>
Get the <a href="windtunnel.mp4">MP4 wind tunnel video</a>
</video>
```

Embedded Video (cont'd.)

The content in the **video** element is a fallback for browsers that don't support the HTML5 video element (e.g., providing a link to the video).

video element attributes:

- **width/height**: Specific dimension in pixels
- **poster**: Provides location of still image to show before video plays
- **controls**: Indicates that the video player controls should be visible
- **autoplay**: Makes the video start playing automatically

Video Formats

- Browsers are inconsistent about which video formats they support (see table).
- Best supported format: **MP4 (H.264)**
[MPEG-4 video container, H.264 video compression, and AAC audio compression]

Format	Type	IE	MS Edge	Chrome	Firefox	Safari	Opera	Android Browser	iOS Safari
MP4 (H.264)	video/mp4 mp4 m4v	9.0+	12+	4+	Yes*	3.2+	25+	4.4+	3.2+
WebM (VP8)	video/webm webm webmv	—	—	6+	4.0+	—	15+	2.3+	—
WebM (VP9)	video/webm webm webmv	—	14+	29+	28+	—	16+	4.4+	—
Ogg Theora	video/ogg ogv	—	—	3.0+	3.5+	—	13+	2.3+	—

Video Formats (cont'd.)

`<source> </source>`

To provide several video format options, list them in **source** elements in the **video** element.

The browser downloads the first file it supports, so put the videos with the smallest file sizes first.

```
<video id="video" controls poster="img/poster.jpg">  
  <source src="clip.webm" type="video/webm">  
  <source src="clip.mp4" type="video/mp4">  
  <source src="clip.ogv" type="video/ogg">  
  <a href="clip.mp4">Download the MP4 of the clip.</a>  
</video>
```

Audio Players

```
<audio> </audio>  
<source> </source>
```

The **audio** element embeds an audio player on the page, similar to **video**, but with no **height**, **width**, or **poster** attributes:

Play "Jet Fighter" by The Three O'Clock



```
<audio id="threeoclock" controls preload="auto">  
  <source src="jetfighter.mp3" type="audio/mp3">  
  <source src="jetfighter.ogg" type="audio/ogg">  
  <source src="jetfighter.webm" type="audio/webm">  
  <p>Download <a href="jetfighter.mp3">"Jet Fighter"</a></p>  
</audio>
```

Audio Formats

- Browsers are inconsistent in audio format support (see table).
- Best supported format: **MP3**

Format	Type	IE	MS Edge	Chrome	Firefox	Opera	Safari	iOS Safari	Android
MP3	audio/mpeg mp3	9.0+	12+	3.0+	22+	15+	4+	4.1	2.3+
WAV	audio/wav or audio/wave	–	12+	8.0+	3.5+	11.5+	4+	3.2+	2.3+
Ogg Vorbis	audio/ogg ogg oga	–	–	4.0+	3.5+	11.5+	–	–	2.3+
MPEG-4/AAC	audio/mp4 m4a	11.0+	12+	12.0+	–	15+	4+	4.1+	3.0+
WebM/Vorbis	audio/webm webm	–	–	6.0+	4.0+	11.5+	–	–	2.3.3+
WebM/Opus	audio/webm webm	–	14+	33+	15+	20+	–	–	–

Adding Text Tracks

`<track> </track>`

The **track** element provides text that is synchronized with the audio or video. Useful for:

- Subtitles
- Captions
- Descriptions for sight impaired
- Chapter titles
- Metadata (non-displaying)

Adding Text Tracks (cont'd.)

The **track** element goes inside the **video** or **audio** element you want to annotate.

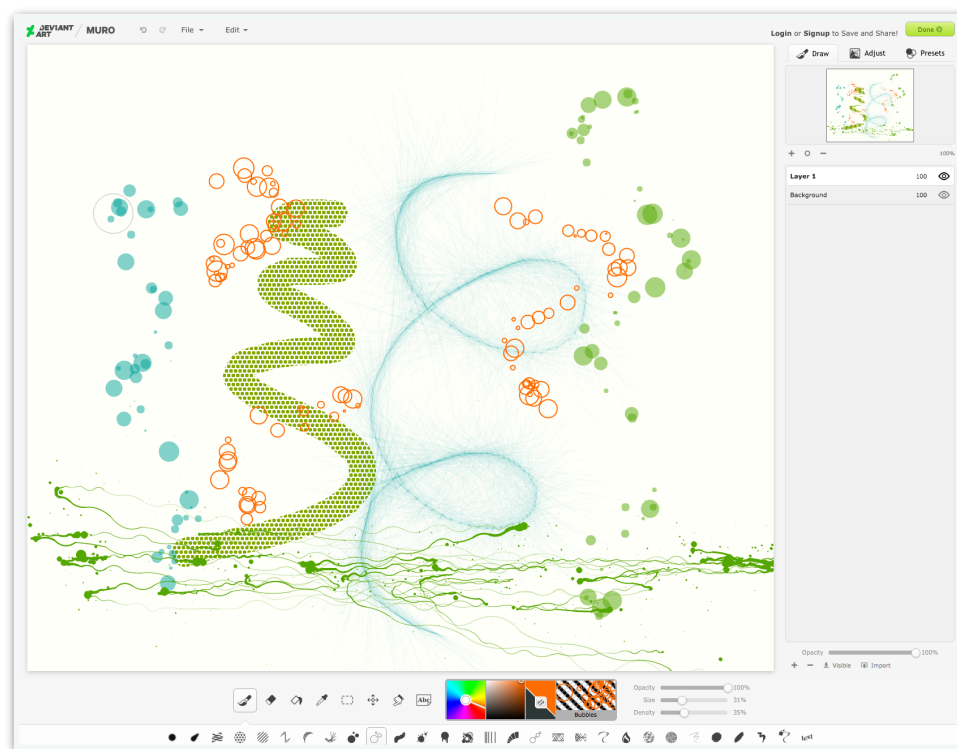
The **src** attribute points to a text file in **.vtt** format.

```
<video width="640" height="320" controls>
  <source src="japanese_movie.mp4" type="video/mp4">
  <source src="japanese_movie.webm" type="video/webm">
  <track src="english_subtitles.vtt"
        kind="subtitles"
        srclang="en"
        label="English subtitles"
        default>
  <track src="french.vtt"
        kind="subtitles"
        srclang="fr"
        label="Sous-titres en français">
</video>
```

Canvas

`<canvas>` `</canvas>`

- The **canvas** element embeds a 2-D bitmapped drawing area that is controlled by JavaScript functions.
- It is useful for games and drawing interfaces.



muro.deviantart.com



majong.frvr.com

Canvas (cont'd)

A sample of the JavaScript used to draw this simple graphic.



```
//draw face
my_canvas.beginPath();
my_canvas.arc(100, 100, 75, (Math.PI/180)*0, (Math.PI/180)*360, false);
// circle dimensions
my_canvas.strokeStyle = "black"; // circle outline is black
my_canvas.lineWidth = 3; // outline is three pixels wide
my_canvas.fillStyle = "yellow"; // fill circle with yellow
my_canvas.stroke(); // draw circle
my_canvas.fill(); // fill in circle
my_canvas.closePath();
```