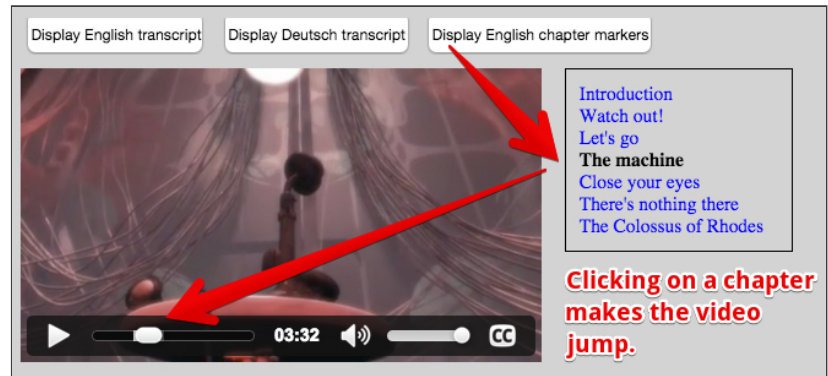


Example 4: making a simple chapter navigation menu

INTRODUCTION

You can use WebVTT files in order to define chapters. The syntax is exactly the same as for subtitles/caption .vtt files. The only difference is in the declaration of the track. Here is how we declared a chapter track in one of the previous examples (in bold in the example below):

HTML code:



```
<video id="myVideo" preload="metadata" controls crossOrigin="anonymous">
  <source src="http://...../elephants-dream-medium.mp4"
    type="video/mp4">
4.  <source src="http://...../elephants-dream-medium.webm"
5.  type="video/webm">
  <track label="English subtitles"
    kind="subtitles"
    srclang="en"
    src="http://...../elephants-dream-subtitles-en.vtt" >
  <track label="Deutsch subtitles"
    kind="subtitles"
    srclang="de"
    src="http://...../elephants-dream-subtitles-de.vtt"
    default>
  <track label="English chapters"
    kind="chapters"
    srclang="en"
    src="http://...../elephants-dream-chapters-en.vtt">
</video>
```

If we try this code in an HTML document, nothing special happens. No magic menu, no extra button!

Currently (as at November 2015), no browser takes chapter tracks into account. You can use one of the enhanced video players presented during the HTML5 Part 1 course, but as you will see in this lesson:

making your own chapter navigation menu is not complicated.

FIRST, LET'S HAVE A LOOK AT THE `.VTT` FILES FROM THIS EXAMPLE:

`elephant-dream-chapters-en.vtt`:

```
WEBVTT

chapter-1
00:00:00.000 --> 00:00:26.000
Introduction

chapter-2
00:00:28.206 --> 00:01:02.000
Watch out!

10. chapter-3
00:01:02.034 --> 00:03:10.000
Let's go

chapter-4
00:03:10.014 --> 00:05:40.000
The machine

chapter-5
20. 00:05:41.208 --> 00:07:26.000
Close your eyes

chapter-6
00:07:27.125 --> 00:08:12.000
There's nothing there

chapter-7
00:08:13.000 --> 00:09:07.500
The Colossus of Rhodes
```

We've got 7 cues (one for each chapter). Each cue id is chapter- followed by the chapter number, then we have the start and end time of the cue/chapter, and the cue content. In this case: the description of the chapter ("Introduction", "Watch out!", "Let's go", etc...).

Hmm... let's try to open this chapter track with [the example we wrote in a previous lesson - the one that displayed the clickable transcript for subtitles/captions on the right of the video](#). We need to modify it a little bit:

1. We add a "show English chapters" button with a click event listener similar to this :

```
<button disabledid="buttonEnglishChapters"onclick="loadTranscript('en',  
  'chapters');">  
  Display English chapter markers  
</button>
```

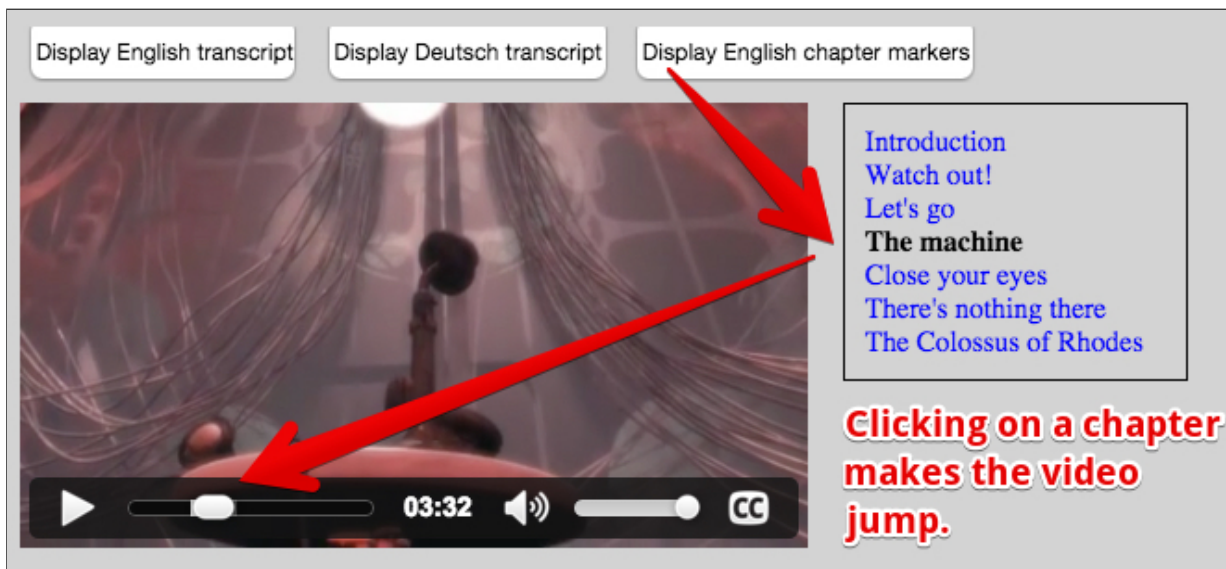
2. We modify the `loadTranscript` function from the previous example, so that it matches both the `srcLang` and the `kind` attribute of the track.

Here is a new version: in bold are the source code lines we modified.

```
function loadTranscript(lang, kind){  
  ...  
  // Locate the track with lang and kind that match the  
  parameters  
5.   for(var i = 0; i < tracks.length;i++) {  
    ...  
    if((track.language === lang)&& (track.kind === kind)) {  
      // display it contents...  
    }  
  }  
}
```

FIRST VERSION : CHAPTERS AS CLICKABLE TEXT ON THE RIGHT OF THE VIDEO

[Try it on JSBin](#); this version includes the modifications we presented earlier - nothing more. Notice that we kept the existing buttons to display a clickable transcript:



Look at the JavaScript and HTML tab of the JSBin example to see the source code. It's the same as in the clickable transcript example, except for the small changes we explained earlier.

This sort of navigation, illustrated in the video player below, is fairly popular.

HTML5 video accessibility and the WebVTT file format

Using CSS for richer styling

```

WebVTT
1
00:00:13,000 --> 00:00:16,100
I heard about this
2
00:00:16,100 --> 00:00:20,100
- and I said 'Wow! a lot of people are starting to talk about this.
I should check it out.'
::cue pseudo-element
CSS
@keyframes .ardaino {
  color: red;
  text-transform:
  font-family: 'Helvetica Neue';
  font-weight: lighter;
  }

```

that track works is in this way.

that track works is in this way.

Generated using the track JavaScript API

SPACE = play / pause toggle
ENTER = navigate to chapter
TAB = navigate chapters
CTL-ALT-DOWNARROW = navigate text elements

Navigation to specific time points.

Click on link to navigate to video fragment. Press space to toggle video play/pause.

- [Slide 1: Title Slide](#)
- [Slide 2: Introduction by Naomi Black](#)
- [Slide 3: Impact of Captions on the Web](#)
- [Slide 4: Requirements of a Video text format](#)
- [Slide 5: Simple WebVTT file](#)
- [Slide 6: Styled WebVTT file](#)
- [Slide 7: Internationalized WebVTT file](#)
- [Slide 8: Positioning of WebVTT cues](#)
- [Slide 9: Speaker semantics in cues](#)
- [Slide 10: Audio descriptions in WebVTT](#)
- [Slide 11: Navigation through chapters](#)
- [Slide 12: HTML markup for captions](#)
- [Slide 13: CSS for rich styling](#)
- [Slide 14: Demo of roll-up captions](#)
- [Slide 15: Example with paint-on captions](#)
- [Slide 16: Concluding remarks](#)
- [Slide 17: Question on line numbers](#)
- [Slide 18: Question on hyperlinks](#)
- [Slide 19: Questions on mixing kinds in a file](#)

In addition to the clickable chapter list, this one displays an enhanced progress bar created using a canvas. The small squares are drawn depending on the chapter cues start and end time, etc. You can try to modify the provided example in order to add such an enhanced progress indicator.

However, we will see how we can do better by using JSON objects as cue contents. This is the topic of the two next lessons!