



## Introduction

In this video, we are going to add subtitles to a video using a new technology that's still being developed, but shows great promise. It is called **WebVTT (Web Video Text Tracks)**. With WebVTT, we can provide subtitles, chapters and more - all within a very simple format that's basically a text file, just like HTML, CSS and JavaScript.

Open the *videosubtitles.html* file in the editor and in the browser before proceeding.

## Video Software

The first step is to create the WebVTT file: call it *subtitles.vtt*. One of the requirements for the VTT file is having

WEBVTT

at the beginning. Skip one line after this one.

You can easily add comments

NOTE This is a comment.

or multiline comments

NOTE

Multi-line comment.

Multi-line comment.

We can add cues beginning with an ID like this: 1. Then on the next line provide the start and end times for the subtitle or caption:

```
00:00:01.000 --> 00:00:10.000
```

Below it, the subtitle itself:

Pouring coffee into a mug...

This creates a subtitle that should appear from the first second to the tenth second of the media.

## Custom Video Controls

We now have to load this file and attach it to the video. That's done using the track tag:

```
<div class="video">
  <video controls preload="auto" poster="media/poster.png">
    <source src="media/video.webm" type='video/webm'>
    <source src="media/video.mp4" type='video/mp4'>
    <source src="media/video.ogv" type='video/ogg'>

    <track src="subtitles.vtt" kind="subtitles" srclang="en" label="English" default>

    <p>
      Your browser doesn't support HTML5 video playback.
      You can <a href="media/video.mp4">download the video instead</a>.
    </p>
  </video>
</div>
```

The src attribute points to file like in other HTML tags.

The kind attribute tells the browser the kind of file we are attaching with the VTT format. Available kinds are subtitles, captions, chapters, description and metadata.

- Subtitles and captions provide text in other languages and are for deaf and hard-of-hearing viewers.
- Chapters provides a chapters list for the video. Description can be used for description of the video for search engines and for blind users.
- Metadata contains some system data that cannot be seen by the user.

For now, the most important kinds are subtitles and captions. If no kind is specified, the browser will assume subtitles.

The srclang attribute indicates the language used in the file, employing the standard code for language.

The label attribute assigns a label to the file, like “English” or “English subtitles”.

The default Boolean attribute can be used in one of the track tags to specify that the default subtitles or captions to be shown.

With the code we typed in earlier, the subtitles should already be working in modern browsers. However I personally couldn't make this work in Chrome using a local file like this, so if you are using that browser, try uploading the file to a web server - then it should work with no issues.

To add another cue, we just add two new lines, and repeat the process:

```
2
00:00:20.000 --> 00:00:27.000
Done!
Enjoy your coffee!
```

Note the line break here - it's going to affect the subtitle itself.

## Media Element

We can add tags like **b**, *i* and u, like this:

```
2
00:00:20.000 --> 00:00:27.000
<b>Done!</b>
Enjoy your coffee!
```

This has support in Google Chrome and Firefox.

The special **c** tag can be used to add a custom CSS class like this:

```
2
00:00:20.000 --> 00:00:27.000
<b>Done</b>!
Enjoy your <c.example>coffee</c>!
```

Another special tag is the **v** tag used to specify the voice of a character. For example:

```
1
00:00:01.000 --> 00:00:10.000
<v Narrator>Pouring coffee into a mug...</v>
```

We can also set cue settings - this is beginning to have support in some browsers, like Firefox, Chrome and Safari. With cue settings, we can change the size and position of the subtitles. The settings come right after the end title of the cue, separated by a space, like this:

```
1
00:00:01.000 --> 00:00:10.000 line:1 position:20% size:30% align:start
<v Narrator>Pouring <i>coffee</i> into a mug...</v>
```

Test this in the browser. You will see that the first subtitle appears in a different position than the default one.

The **line** setting sets the vertical position of the subtitle. Positive numbers indicate spacing from the top down; negative numbers from the bottom up.

The **position** setting indicates the horizontal position of the text. It's expressed in percentage: 0% being the left of the video area and 100% the right of the video.

The **size** setting actually controls the width of the subtitles. For example width of 30% will set the subtitles to occupy 30% of the video's width.

The **align** setting will set the alignment inside the size we just configured, or within the video, if it's not set. Possible values are **start** or the left of the video; **middle** or center; and, **end** or the right of the video or defined size.

**vertical** is used with languages that are written vertically, which would change the behavior of the settings we've just discussed.