What is a media experience?

- The technical requirements
- Displaying video on the web
- What we're going to learn

Users like media, especially videos; they can be fun and informative. On mobile devices, videos can be an easier way to consume information than text. For a good user experience, videos should not need more than the available bandwidth. Users should be able to use them no matter what device they're viewing them with. Users should never need to wait for media to download. Nobody likes it when they press play and nothing happens.

You've no doubt consumed video on your own device, which means that nothing in that last paragraph surprises you. Now you need to learn how to put a video or other media file on your own website. The technical requirements of adding media should be in service to the user experience.

The technical requirements

- Versions of a media file are in common web-friendly formats containing both audio and video streams.
- The resolution is appropriate for your users' devices.
- The bitrate doesn't overload your users' network bandwidth.
- The result is viewable on all major browsers using appropriate technologies.
- The file is encrypted (Optional).

The media section of this website will help you achieve these technical requirements. Don't worry if these concepts are still a bit abstract. We'll explain them progressively throughout all the articles. In the first section you will learn about the basic concepts of media, then how to add media to the web in the second section, and in the final section the practical applications, with some advanced techniques, of using media on the web.

Displaying video on the web

There are four approaches you can take when displaying video on a site.

- Upload your videos to a media hosting provider such as YouTube or Vimeo. These options require you to embed their players within your site.
- Basic self-hosted embedding using the HTML `<video>` and `<audio>` elements.
- More full-featured embedding using a video library such as Shaka Player, JW Player, or Video.js.
- Building your own media server and streaming application.

This site mainly covers the basics of embedding media. However, we do cover some more advanced topics to get you started on the path towards building your own media streaming application. The effort to do this is not trivial, so we've built a Media PWA with offline support to use as a reference, which should both show you ways this can be accomplished and just how much effort it requires. The application is by no means a production ready offering or competitor to services like YouTube or Vimeo, but it will provide you with a starting point to learn something challenging and new.

Frankly, building a competitor to hosted media services would require a team of expert engineers and thousands of human-hours of work. Unless your goal is to enter that market as a competitor, you're better off using one of the other methods. It's good to understand the technology and while you may not roll out your own application or video player, there is utility in exploring and experimenting on the cutting edge of browser support, or at the very least using one of the existing video libraries.

What we're going to learn

The media collection has three parts. In this first section, we'll provide concepts and prerequisite information to adding media to your site. This includes explaining how media files are put together, basics about the applications you'll need to prepare your files for the web, and streaming concepts. The second section explains how to prepare your files and convert them to various formats and optionally add encryption. In the last section, we'll show you how to embed a media file in a web page, discuss autoplay best practices, using media frameworks, taking videos offline, and making your videos accessible.

There's a lot of ground to cover, so let's get started with Media file basics.

□□□ #2
□ □□□ □□□ 2026-10-31 22:56:47 UTC
□ □□□ □□□ 2026-10-31 22:56:48 UTC