



Client-side web APIs

Looking to become a front-end web developer?

When writing client-side JavaScript for web sites or applications, you will quickly encounter **Application Programming Interfaces (APIs)**. APIs are programming features for manipulating different aspects of the browser and operating system the site is running on, or manipulating data from other web sites or services. In this module, we will explore what APIs are, and how to use some of the most common APIs you'll come across often in your development work.

We have put together a course that includes all the essential information you need to work towards your goal.

[Get started](#)

Prerequisites

To get the most out of this module, you should have worked your way through the previous JavaScript modules in the series ([First steps](#), [Building blocks](#), and [JavaScript objects](#)). Those modules typically involve simple API usage, as it is often difficult to write client-side JavaScript examples without them. For this tutorial, we will assume that you are knowledgeable about the core JavaScript language, and we will explore common Web APIs in a bit more detail.

Basic knowledge of [HTML](#) and [CSS](#) would also be useful.

Note: If you are working on a device where you don't have the ability to create your own files, you could try out (most of) the code examples in an online coding program such as [JSBin](#) or [Glitch](#).

Guides

Introduction to web APIs

First up, we'll start by looking at APIs from a high level — what are they, how do they work, how do you use them in your code, and how are they structured? We'll also take a look at what the different main classes of APIs are, and what kind of uses they have.

Manipulating documents

When writing web pages and apps, one of the most common things you'll want to do is manipulate web documents in some way. This is usually done by using the Document Object Model (DOM), a set of APIs for controlling HTML and styling information that makes heavy use of the [Document](#) object. In this article, we'll look at how to use the DOM in detail, along with some other interesting APIs that can alter your environment in interesting ways.

Fetching data from the server

Another very common task in modern websites and applications is retrieving individual data items from the server to update sections of a webpage without having to load an entirely new page. This seemingly small detail has had a huge impact on the performance and behavior of sites. In this article, we'll explain the concept, and look at technologies that make it possible, such as [XMLHttpRequest](#) and the [Fetch API](#).

Third party APIs

The APIs we've covered so far are built into the browser, but not all APIs are. Many large websites and services such as Google Maps, Twitter, Facebook, PayPal, etc. provide APIs allowing developers to make use of their data (e.g. displaying your twitter stream on your blog) or services (e.g. displaying custom Google Maps on your site, or using Facebook login to log in your users). This article looks at the difference between browser APIs and 3rd party APIs and shows some typical uses of the latter.

Drawing graphics

The browser contains some very powerful graphics programming tools, from the Scalable Vector Graphics ([SVG](#)) language, to APIs for drawing on HTML `<canvas>` elements, (see [The Canvas API](#) and [WebGL](#)). This article provides an introduction to the Canvas API, and further resources to allow you to learn more.

Video and audio APIs

HTML5 comes with elements for embedding rich media in documents — `<video>` and `<audio>` — which in turn come with their own APIs for controlling playback, seeking, etc. This article shows you how to do common tasks such as creating custom playback controls.

Client-side storage

Modern web browsers feature a number of different technologies that allow you to store data related to web sites and retrieve it when necessary allowing you to persist data long term, save sites offline, and more. This article explains the very basics of how these work.

Last modified: Dec 30, 2020, by [MDN contributors](#)