Learn web development

Tools and testing

Once you've started to become comfortable programming with core web technologies (like HTML, CSS, and JavaScript), and you start to get more experience, read more resources, and learn more tips and tricks, you'll start to come across all kind of tools, from ready-rolled CSS and JavaScript, to testing and automation apps, and more besides. As your web projects become larger and more complex, you'll want to start taking advantage of some of these tools, and working out reliable testing plans for your code. This part of the learning area aims to give you what you need get started and make informed choices.

The web industry is an exciting place to work, but it is not without its complications. The core technologies we use to build web sites are fairly stable now, but new features are being added all the time, and new tools — that facilitate working with, and are built on top of these technologies — are constantly appearing. On top of that, we still need to keep cross-browser support in the forefront of our minds, and make sure that our code follows best practices that allow our projects to work across different browsers and devices that our users are using to browser the Web, and be usable by people with disabilities.

Working out what tools you should be using can be a difficult process, so we have written this set of articles to inform you of what types of tool are available, what they can do for you, and how to make use of the current industry favourites.

> **Note**: Because new tools appear and old ones go out of fashion all the time, we have deliberately written this material to be as neutral as possible — we want to focus first and foremost on the general types of tasks these tools will help you accomplish, and keep prescribing specific tools to a minimum. We obviously need to show tool usage to demonstrate specific techniques, but be clear that we do not necessarily recommend these tools as the best or only way to do things — in most cases there are other ways, but we want to provide you with a clear methodology that works.

Learning pathway

You should really learn the basics of the core HTML, CSS, and JavaScript languages first before attempting to use the tools detailed here. For example, you'll need to know the fundamentals of these languages before you start debugging problems in complex web code, or making effective use of JavaScript libraries, or writing tests and running them against your code using test runners, etc.

You need a solid foundation first.

Modules

Real world web development tools (TBD)

In this module, we explore the different kinds of web development tools available. This includes reviewing the most common kinds of tasks you may be called on to solve, how they can fit together in a workflow, and the best tools currently avaiable for carrying out those tasks.

Cross browser testing

This module looks specifically at the area of testing web projects across different browsers. Here we look identifying your target audience (e.g. what users, browsers and devices do you most need to worry about?), how to go about doing testing, the main issues that you'll face with different types of code and how to fix/mitigate those, what tools are most useful in helping you test and fix problems, and how to use automation to speed up testing.

Was this article helpful?

👍 👎

Learn the best of web development ✖

Get the latest and greatest from MDN delivered straight to your inbox.

you@example.com

SIGN UP NOW