

MDN Plus now available in your country! Support MDN and make it your own. [Learn more](#) 🌟

Installing basic software

In *Installing basic software*, we show you what tools you need to do simple web development and how to install them properly.

What tools do I actually need, right now?

In this article, we'll set you up with the bare minimum — a text editor and some modern web browsers.

Installing a text editor

You probably already have a basic text editor on your computer. By default Windows includes [Notepad](#) and macOS comes with [TextEdit](#). Linux distros vary; Ubuntu comes with [gedit](#) by default.

For web development, you can probably do better than Notepad or TextEdit. We recommend starting with [Visual Studio Code](#), which is a free editor, that offers live previews and code hints.

Installing modern web browsers

For now, we'll install a couple of desktop web browsers to test our code in. Choose your operating system below and click the relevant links to download installers for your favorite browsers:

- Linux: [Firefox](#), [Chrome](#), [Opera](#), [Brave](#).

- Windows: [Firefox](#) , [Chrome](#) , [Opera](#) , [Internet Explorer](#) , [Microsoft Edge](#) , [Brave](#) (Windows 10 comes with Edge by default; if you have Windows 7 or above, you can install Internet Explorer 11; otherwise, you should install an alternative browser).
- macOS: [Firefox](#) , [Chrome](#) , [Opera](#) , [Safari](#) , [Brave](#) (macOS and iOS come with Safari by default).

Before going on, you should install at least two of these browsers and have them ready for testing.

Note: Internet Explorer is not compatible with some modern web features and it may not be able to run your project. You usually don't need to worry about making your web projects compatible with it, as very few people still use it — certainly don't worry too much about it while you are learning. You might sometimes run into a project that requires support for it.

Installing a local web server

Some examples will need to be run by a web server to work successfully. You can find out how to do this in [How do you set up a local testing server?](#)

What tools do the professionals use?

The following looks like a scary list, but fortunately, you can get started in web development without knowing anything about most of these.

- **A computer.** Maybe that sounds obvious to some people, but some of you are reading this article on your phone or a library computer. For serious web development, it's better to invest in a desktop or laptop computer running Windows, macOS or Linux.
- **A text editor,** to write code in. This could be a text editor (e.g. [Visual Studio Code](#) , [Notepad++](#) , [Sublime Text](#) , [Atom](#) , [GNU Emacs](#) , or [VIM](#)), or a hybrid editor (e.g. [Dreamweaver](#) or [WebStorm](#)). Office document editors are not suitable for this use, as they rely on hidden elements that interfere with the rendering engines used by web browsers.

- **Web browsers**, to test code in. Currently, the most-used browsers are [Firefox](#) , [Chrome](#) , [Opera](#) , [Safari](#) , [Internet Explorer](#) and [Microsoft Edge](#) . You should also test how your site performs on mobile devices and on any old browsers your target audience may still be using (such as IE 8–10). [Lynx](#) , a text-based terminal web browser, is great for seeing how your site is experienced by visually-impaired users.
- **A graphics editor**, like [GIMP](#) , [Figma](#) , [Paint.NET](#) , [Photoshop](#) , [Sketch](#) or [XD](#) , to make images or graphics for your web pages.
- **A version control system**, to manage files on servers, collaborate on a project with a team, share code and assets and avoid editing conflicts. Right now, [Git](#) is the most popular version control system along with the [GitHub](#) or [GitHub](#) hosting service.

 mdn web docs

- **An automation system**, like [Webpack](#) , [Grunt](#) , or [Gulp](#) to automatically perform repetitive tasks, such as minifying code and running tests.
- Libraries, frameworks, etc., to speed up writing common functionality. A library tends to be an existing JavaScript or CSS file that provides ready-rolled functionality for you to make use of in your code. A framework tends to take this idea further, offering a complete system with some custom syntaxes for you to write a web app on top of.
- More tools besides!

In this module

- [Installing basic software](#)
- [What will your website look like?](#)
- [Dealing with files](#)
- [HTML basics](#)
- [CSS basics](#)
- [JavaScript basics](#)
- [Publishing your website](#)
- [How the web works](#)

Last modified: May 22, 2022, [by MDN contributors](#)