


[Technologies](#) ▼[References & Guides](#) ▼[Feedback](#) ▼[Sign in](#) 

CSS: Cascading Style Sheets

Cascading Style Sheets (CSS) is a [stylesheet](#) language used to describe the presentation of a document written in [HTML](#) or [XML](#) (including XML dialects such as [SVG](#), [MathML](#) or [XHTML](#)). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is one of the core languages of the **open web** and is standardized across browsers according to the [W3C specification](#). Developed in levels, CSS1 is now obsolete, CSS2.1 is a recommendation, and [CSS3](#), now split into smaller modules, is progressing on the standardization track.

CSS Introduction

If you're new to web development, be sure to read our [CSS basics](#) article to learn what CSS is and how to use it.

CSS Tutorials

Our [CSS learning area](#) contains a wealth of tutorials to take you from beginner level to proficiency, covering all the fundamentals.

CSS Reference

Our [exhaustive CSS reference](#) for seasoned Web developers describes every property and concept of CSS.

Tutorials

Our [CSS Learning Area](#) features multiple modules that teach CSS from the ground up — no previous knowledge required.

Introduction to CSS

This module starts with the basics of how CSS works, including selectors and properties, writing CSS rules, applying CSS to HTML, how to specify length, color, and other units in CSS, cascade and inheritance, box model basics, and debugging CSS.

Styling text

This module discusses text styling fundamentals, including setting fonts, boldness, italics, line and letter spacing, text drop shadows, and other text properties. This module finishes with applying custom fonts to your page, and styling lists and links.

Styling boxes

This module looks at styling boxes, one of the fundamental steps towards laying out a web page. In this module we recap the box model, then look at controlling box layouts by setting margins, borders, and padding, custom background colors, images and other features, and fancy features such as drop shadows and filters on boxes.

CSS layout

At this point we've already looked at CSS fundamentals, how to style text, and how to style and manipulate the boxes that your content sits inside. Now it's time to look at how to place your boxes in the right place in relation to the viewport, and one another. We have covered the necessary prerequisites so you can now dive deep into CSS layout, looking at different display settings, traditional layout methods involving float and positioning, and newfangled layout tools like flexbox.

Reference

- [CSS reference](#): This exhaustive reference for seasoned Web developers describes every property and concept of CSS.
- CSS key concepts:
 - The [syntax and forms of the language](#)
 - [Specificity, inheritance and the Cascade](#)
 - [Box model](#) and [margin collapse](#)
 - The [containing block](#)
 - [Stacking](#) and [block-formatting](#) contexts
 - [Initial](#), [computed](#), [used](#), and [actual](#) values
 - [CSS shorthand properties](#)
 - [CSS Flexible Box Layout](#)
 - [CSS Grid Layout](#)
 - [Media queries](#)

Cookbook

The [CSS layout cookbook](#) aims to bring together recipes for common layout patterns, things you might need to implement in your own sites. In addition to providing code you can use as a starting point in your projects, these recipes highlight the different ways layout specifications can be used, and the choices you can make as a developer.

Tools for CSS development

- You can use the [W3C CSS Validation Service](#) to check if your CSS is valid. This is an invaluable debugging tool.
- [Firefox Developer Tools](#) lets you view and edit a page's live CSS via the [Inspector](#) and [Style Editor](#) tools.
- The [Web Developer extension](#) for Firefox lets you track and edit live CSS on watched sites.
- The Web community has created various other [miscellaneous CSS tools](#) for you to use.

Meta bugs [↗](#)

- Firefox: [bug 1323667](#)
-

See also [↗](#)

- [CSS demos](#): Get a creative boost by exploring examples of the latest CSS technologies in action.
- Web languages to which CSS is often applied: [HTML](#), [SVG](#), [MathML](#), [XHTML](#), and [XML](#).
- Mozilla technologies that make extensive use of CSS: [XUL](#), [Firefox](#), and [Thunderbird extensions](#) and [themes](#).

Join the Web layout community [↗](#)

**Choose your preferred method
for joining the discussion:**

Stack Overflow: [↗ CSS topics](#)

[↗ Mailing list](#) [↗ Newsgroup](#)

[↗ RSS feed](#)

