Technologies ▾

References & Guides ▾

Feedback ▾

Sign in

Search

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 we can now dive deep into CSS layout, looking at different display settings, modern layout tools like flexbox, CSS grid, and positioning, and some of the legacy techniques you might still want to know about.

Prerequisites

Before starting this module, you should already:

1. Have basic familiarity with HTML, as discussed in the Introduction to HTML module.
2. Be comfortable with CSS fundamentals, as discussed in Introduction to CSS.
3. Understand how to style boxes.

> **Note**: If you are working on a computer/tablet/other 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 Thimble.

Guides

These articles will provide instruction on the fundamental layout tools and techniques available in CSS. At the end of the lessons is an assessment to help you check your understanding of layout methods, by laying out a webpage.

Introduction to CSS layout

This article will recap some of the CSS layout features we've already touched upon in previous modules — such as different `display` values — and introduce some of the concepts we'll be covering throughout this module.

Normal Flow

Elements on webpages lay themselves out according to *normal flow* - until we do something to change that. This article explains the basics of normal flow as a grounding for learning how to change it.

Flexbox

Flexbox is a one-dimensional layout method for laying out items in rows or columns. Items flex to fill additional space and shrink to fit into smaller spaces. This article explains all the fundamentals.

Grids

CSS Grid Layout is a two-dimensional layout system for the web. It lets you lay content out in rows and columns, and has many feature that make building complex layouts straightforward. This article will give you all you need to know to get started with page layout.

Floats

Originally for floating images inside blocks of text, the `float` property became one of the most commonly used tools for creating multiple column layouts on webpages. With the advent of Flexbox and Grid it has now returned to its original purpose, as this article explains.

Positioning

Positioning allows you to take elements out of the normal document layout flow, and make them behave differently, for example sitting on top of one another, or always remaining in the same place inside the browser viewport. This article explains the different `position` values, and how to use them.

Multiple-column layout

The multiple-column layout specification gives you a method of laying content out in columns, as you might see in a newspaper. This article explains how to use this feature.

Legacy layout methods

Grid systems are a very common feature used in CSS layouts, and before CSS Grid Layout they tended to be implemented using floats or other layout features. You imagine your layout as a set number of columns (e.g. 4, 6, or 12), and then fit your content columns inside these imaginary columns. In this article we'll explore how these older methods work, in order that you understand how they were used if you work on an older project.

Supporting older browsers

In this module we recommend using Flexbox and Grid as the main layout methods for your designs. However there will be visitors to your site who use older browsers, or browsers which do not support the methods you have used. This will always be the case on the web — as new features are developed, different browsers will prioritise different things. This article explains how to use modern web techniques without locking out users of older technology.

Fundamental Layout Comprehension

An assessment to test your knowledge of different layout methods by laying out a webpage.

See also

Practical positioning examples

This article shows how to build some real world examples to illustrate what kinds of things you can do with positioning.