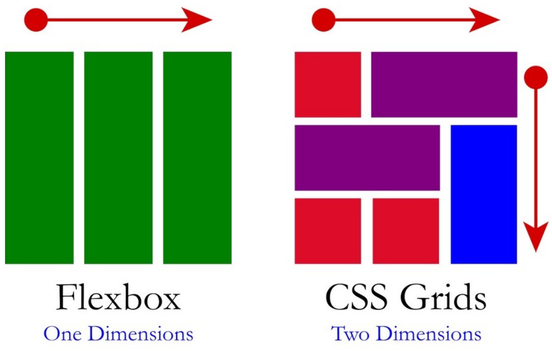
New layout techniques (OPTIONAL)

New layout techniques (OPTIONAL)

Web layout is always constrained by the limitations of CSS, but future trends will be able to make use of new tools, such as CSS Flexbox (officially: CSS Flexible Box Layout) and CSS Grid.

Since the summer of 2017, the major browsers support this module of CSS, which provides the beginnings of something professional designers have long been waiting for: a design grid.

It's been a few years now that the CSS Flexbox module gives advanced control over alignment and spacing of elements on a line or in a stack. The Grid module now allows alignment in two dimensions: to elements on the same line and to elements in the same column.



Grid doesn't replace Flexbox. There is some flexibility when placing elements in one line that you lose when you add a second dimension. At the same time there are layouts that were difficult or impossible even with Flexbox and are now easy with Grid.

The Grid module is only a first step. Level 1 has a number of limitations (just like Flexbox), due to do issues around complexity and performance. For example, you can currently only align elements that have the same parent. That said, for laying out graphical user interfaces (GUIs), Flexbox and Grid work very well.

We briefly present them below, and give a few resources at the bottom of the page so that you can explore more. Note: *this lecture is optional and there will be no questions related to it in the final exam.*

CSS Flexbox

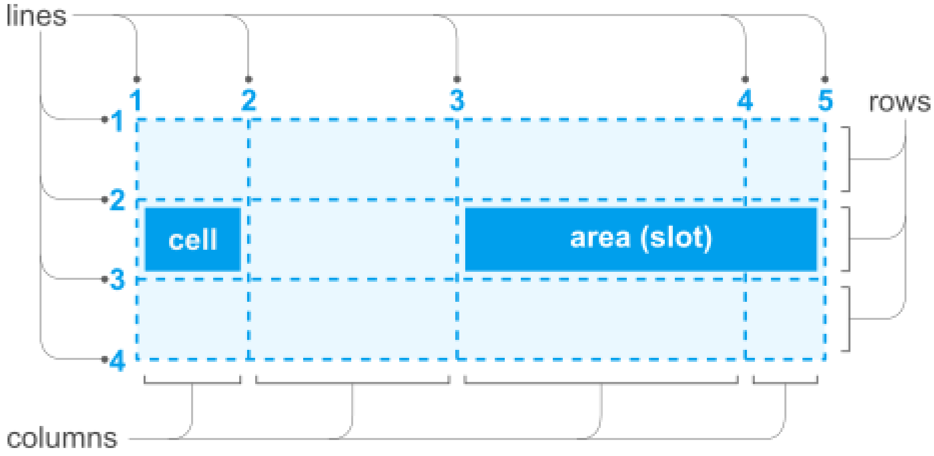
Since its introduction in 2012, CSS Flexbox has become widely used and browser support is really good. Flexbox made a lot of complex layout tasks easier, like equidistant spacing between elements, top-to-bottom layouts or the holy grail. Find out about Flexbox and how to use it in the last section of the W3C HTML5 and CSS Fundamentals course!

Despite the similarities in concept and syntax, Flexbox and Grid are not competing layout techniques. Grid arranges in two dimensions, while Flexbox lays out in one. There is synergy when using the two together.

CSS Grid

CSS Grid is a CSS module that defines a two-dimensional grid-based layout system, optimized for user interface design. In the grid layout model, the children of an element (the ‘grid container’) can be positioned into arbitrary slots in a predefined flexible or fixed-size layout grid.

If that sounds a bit too abstract, here is another way of looking at it. The idea behind the Grid module is that you split the box that makes up an element into many individual ‘slots’, arranged in a matrix, and separated from each other by (invisible) horizontal and vertical lines. You do that with a property called 'grid', which contains the desired number of rows and columns and/or their sizes. Each child element goes into a slot, so that they end up aligned as in a table. But you have full control over which slot they go into, you can change their order, they can span more than one row or column, and you can leave some slots empty.



The Grid module provides several different ways to define such a grid and to place the child elements. Too many, in fact, to present here.

Because they can refer to a previously defined grid of horizontal and vertical lines, the properties from the CSS Grid module provide more control over the alignment of elements than most other properties in CSS, such as the table-related properties or the 'float' and 'clear' properties, while also allowing elements to be displayed out of order. As such they are especially appreciated for (Web) applications with user interfaces that are made with HTML and CSS. The Grid module is not yet the ‘design grid’ that typographers want for the layout of magazines and books, but it is a first step. (E.g., one obvious thing to do, applying grid properties to an HTML table, doesn't work, because the properties do not handle nested elements yet.) Even though this is only level 1 of the module, it is well worth trying out.

The properties from the Grid module have only been available in major browsers since mid 2017 (see the status of browser support). But the ideas behind the Grid module aren't new. From the start of CSS, there have been proposals to use CSS properties to define a template or matrix to guide the layout of elements, e.g.: Frame-based layout, Advanced Layout (later called Template Layout), Grid Style Sheets and Constraint CSS. But only recently has technology become good enough to support some (not all!) of those ideas.

A few resources

If you look at the CSS Grid module, you may notice that it has rather a large number of properties: 18. That is because it tries to allow different manners of writing style sheets. There are many shorthand properties and many alternative ways to define the same grid.

In practice, most style sheet writers will select a set of just three or four properties that suits their way of working.

When considering the CSS Grid module, also look at the CSS Flexible Box module. It only provides for alignment of elements in a single row or column, but has some features that Grid doesn't have (and it has been around longer and works in older browsers). On the other hand, even for a single row or column, the Grid properties may turn out easier in some cases.

CSS Layout

- CSS Layout News - A weekly collection of tutorials, news and information on all things CSS Layout, by Rachel Andrew.

CSS grid resources

- A complete guide to Grid (CSS Tricks)
- CSS Grid, the W3C specification
- Grid by example: this site is a collection of *examples*, video and other information to help you learn CSS *Grid* Layout.

CSS flexbox resources

- A complete guide to Flexbox (CSS Tricks)
- CSS Flexible Box Layout, the W3C specification
- Flexbox froggy (game to practice CSS flexbox code)

Learn About Verified Certificates