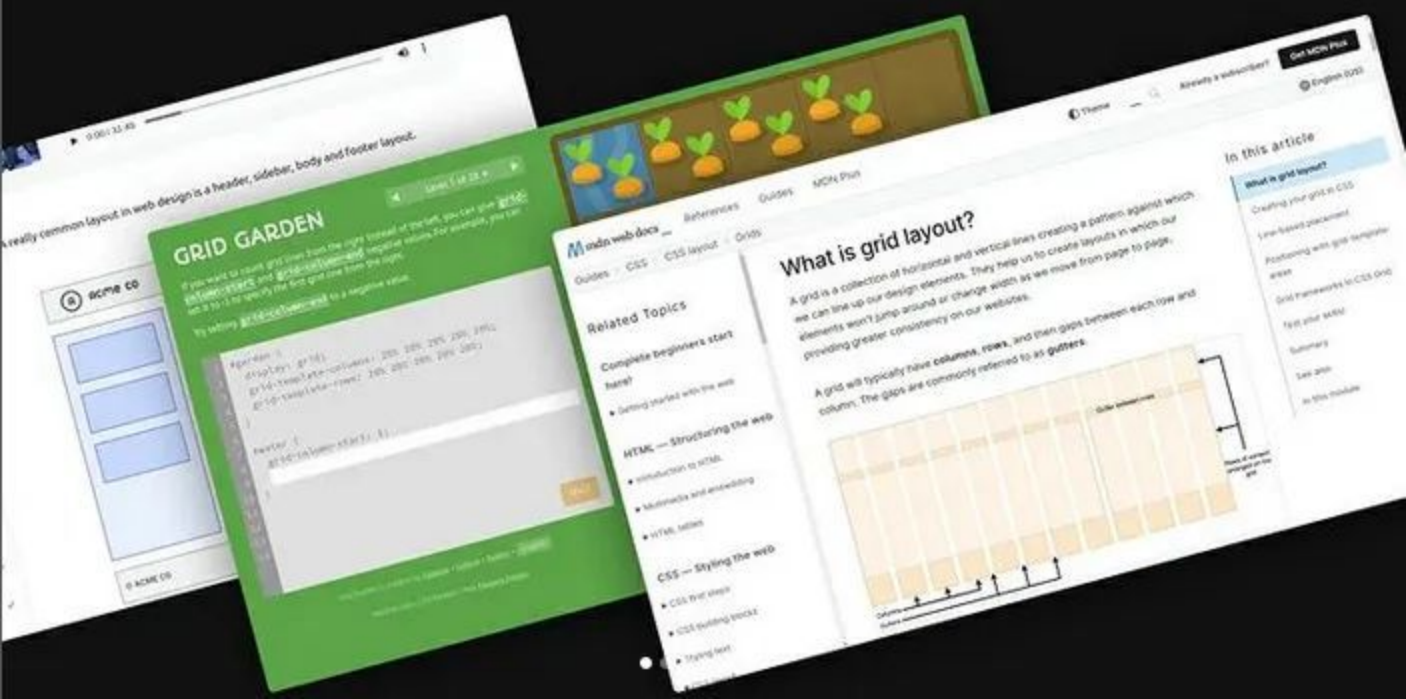




# Resources for learning CSS Grids



# Grid by web.dev

<https://web.dev/learn/css/grid/>

Learn CSS!

Search

- 000 Learn CSS ✓
- 001 Box Model
- 002 Selectors ✓
- 003 The cascade
- 004 Specificity ✓
- 005 Inheritance ✓
- 006 Color ✓
- 007 Sizing Units ✓

A really common layout in web design is a header, sidebar, body and footer layout.

ACME CO

HOME ABOUT PRICING CONTACT

## Lorem Ipsum Dolor Sit Amet

Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Donec sed odio dui. Curabitur blandit tempus porttitor. Etiam porta sem malesuada magna mollis euismod. Curabitur blandit tempus porttitor. Donec ullamcorper nulla non metus auctor fringilla.

Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Donec sed odio dui. Curabitur blandit tempus porttitor. Etiam porta sem malesuada magna mollis euismod. Curabitur blandit tempus porttitor. Donec ullamcorper nulla non metus auctor fringilla.

© ACME CO

SS

# Grid Garden (game)

<https://cssgridgarden.com/>

## GRID GARDEN

◀ Level 5 of 25 ▶

If you want to count grid lines from the right instead of the left, you can give `grid-column-start` and `grid-column-end` negative values. For example, you can set it to -1 to specify the first grid line from the right.

Try setting `grid-column-end` to a negative value.

```
1 #garden {  
2   display: grid;  
3   grid-template-columns: 20% 20% 20% 20% 20%;  
4   grid-template-rows: 20% 20% 20% 20% 20%;  
5 }  
6  
7 #water {  
8   grid-column-start: 1;  
9  
10 }  
11  
12  
13  
14
```

Test

Grid Garden is created by [Quincy](#) • [Gordon](#) • [Thomas](#) • [English](#)

Want to learn CSS flexbox? [Play Flexbox Garden](#)



# Grids by MDN

[https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Grids](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Grids)

mdn web docs

References

Guides

MDN Plus

Theme

Search

Already a subscriber?

Get MDN Plus

Guides > CSS > CSS layout > Grids

English (US)

Related Topics

Complete beginners start here!

▶ Getting started with the web

HTML — Structuring the web

▶ Introduction to HTML

▶ Multimedia and embedding

▶ HTML tables

CSS — Styling the web

▶ CSS first steps

▶ CSS building blocks

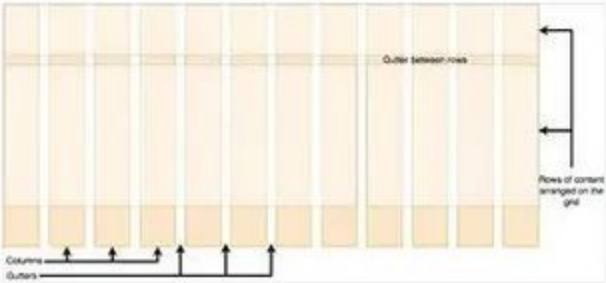
▶ Styling text

▼ CSS layout

## What is grid layout?

A grid is a collection of horizontal and vertical lines creating a pattern against which we can line up our design elements. They help us to create layouts in which our elements won't jump around or change width as we move from page to page, providing greater consistency on our websites.

A grid will typically have **columns**, **rows**, and then gaps between each row and column. The gaps are commonly referred to as **gutters**.



The diagram illustrates a grid layout with 10 vertical columns and 3 horizontal rows. The columns are separated by vertical lines, and the rows are separated by horizontal lines. The gaps between the columns are labeled 'Columns' and 'Gutters' with arrows pointing to them. The gap between the rows is labeled 'Gutter between rows' with an arrow pointing to it. The rows are labeled 'Rows of content arranged on the grid' with an arrow pointing to them.

In this article

What is grid layout?

Creating your grid in CSS

Line-based placement

Positioning with grid-template-areas

Grid frameworks in CSS Grid

Test your skills!

Summary

See also

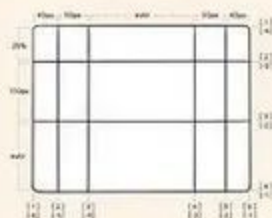
In this module

# A Complete Guide to Grid by CSS Tricks

<https://css-tricks.com/snippets/css/complete-guide-grid/>

```
100px 1fr max-content
```

Grid lines are automatically assigned positive numbers from these assignments (-1 being an alternate for the very last row).

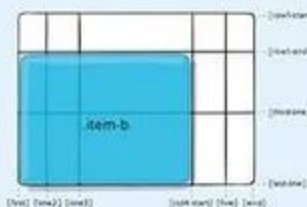


But you can choose to explicitly name the lines. Note the bracket syntax for the line names:

```
.container {  
  grid-template-columns: [first] 40px [line2] 50px [3]  
  grid-template-rows: [row1-start] 25% [row1-end] 100px  
}
```



```
.item-b {  
  grid-column-start: 1;  
  grid-column-end: span col4-start;  
  grid-row-start: 2;  
  grid-row-end: span 2;  
}
```



If no `grid-column-end`/`grid-row-end` is declared, the item will span 1 track by default.

Items can overlap each other. You can use `z-index` to control

# Learn CSS Grid

<https://learncssgrid.com/>

## Grid Container

Create a grid container by setting the `display` property with a value of `grid` or `inline-grid`. All direct children of grid containers become grid items.

`display: grid;`

Grid items are placed in rows by default and span the full width of the grid container.



`display: inline-grid;`

