CSS grid | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/css-and-styling/css-grid.html> |
| Date de scraping | 2025-04-08T15:53:09.614760+00:00 |

## CSS Grid ​

## Why use CSS grids ​

CSS grids are a powerful tool to build complex, responsive layouts on the web.  
For example, designing a complex layout like this would be complex with a flex display but is fairly easy with a grid display:

### Images

<https://docs.weweb.io/assets/grid-example.WkZxu6ra.png>

## CSS grid basics ​

## Grid container ​

In the Style tab of WeWeb containers, you will find grid and inline-grid under the Display property options:  
  
When a container has a grid or inline-grid display, all the children of that container become grid items.  
Learn more about the grid container .

### Images

<https://docs.weweb.io/assets/grid1.DebIipi0.png>

## Grid direction ​

Grids can have a horizontal ("row") or vertical ("columns") direction:  
  
In the example above, we have 3 containers that can be displayed as rows or columns.

### Images

<https://docs.weweb.io/assets/grid-direction.6SOGMC_J.gif>

## Grid column & row size ​

By default, each column or row will have the same width and height respectively.  
You can, however, be explicit and set a fixed width or height to create more complex layouts:  
  
In CSS, this corresponds to the grid-template-rows and grid-template-columns properties.  
In WeWeb, you have a couple of options to work with these properties:  
use the no-code interface of WeWeb as shown in the columns example above, or bind the Rows or Columns settings in WeWeb to CSS values as shown in the rows example below.  
  
In the example above, you can see we have an array of 5 values that defines the height of the 5 rows in our grid.  
Learn more about explicit grid templates

### Images

<https://docs.weweb.io/assets/grid-column-width.X6B4f4eV.png>

<https://docs.weweb.io/assets/grid-custom-rows._K91OLfK.png>

## Grid gaps ​

Grid gaps (also called "gutters") are an elegant way of handling spacing:  
  
Learn more about grid gaps .

### Images

<https://docs.weweb.io/assets/grid-column-gap.B4FgFhzF.gif>

## Grid items span & position ​

For each item in a grid, you can define its:  
span: how many row or column units does it span, position: where should it be placed in the row or column  
Let's look at a few examples.

## Example 1 ​

In both cases, the grid container is configured as follows:  
Direction : Row Columns : auto-generated Rows : auto-generated Width : 100% Height : 500px  
The grid on the left has no additional settings on the grid items. The grid on the right contains grid items where we specified the span or position.

### Images

<https://docs.weweb.io/assets/grid-span-position.DZE1FfTP.png>

## Example 2 ​

In all three cases, the grid container is configured as follows:  
Direction : Column Columns : 3 columns of 1fr Rows : 3 rows of 1fr Width : 100% Height : auto  
The grid at the top has no additional settings on the grid items.  
Grid item A of the middle grid has a Column position of 2/4 .  
Grid item A of the bottom grid has a Row span of 2 and a Column position of 2/4 .

### Images

<https://docs.weweb.io/assets/grid-column-position.BYW1pfL2.png>

## Troubleshooting ​

As mentioned above, this article aims to explain how to use CSS grid properties in the WeWeb editor.  
If you're new to CSS grids, we recommend you take the time to understand how CSS grids work in general.  
We have selected 4 free resources that can help you with this:  
Learn CSS Grid helpfully lists and illustrates all available grid CSS properties CSS Grid Garden is an fun, interactive way to understand CSS grid properties MDN's docs explains the basic concepts of grid layouts Grid by Example includes numerous examples of grid layouts with the underlying HTML & CSS to reverse-engineer how each was built  
If you're still stuck after this, don't hesitate to reach out in the WeWeb Community ! Make sure to include visuals that show:  
the end result you're trying to achieve, and the styles settings you have applied to your grid container and grid items.  
We'll do our best to help 🙂