# slide 2 - Introduction

CSS Grid Layout, is a two-dimensional grid-based layout system that aims to do nothing less than completely change the way we design grid-based user interfaces

CSS has always been used to lay out our web pages, but it’s never done a very good job of it. First, we used tables, then floats, positioning and inline-block, but all of these methods were essentially hacks and left out a lot of important functionality.

Flexbox helped out, but it’s intended for simpler one-dimensional layouts, not complex two-dimensional ones

Grid is the very first CSS module created specifically to solve the layout problems

# slide 4- Properties for the Parent (Grid Container)

## slide 4.1 Display

DISPLAY property defines the element as a grid container and establishes a new grid formatting context for its contents. Values:

* **grid** – generates a block-level grid
* **inline-grid** – generates an inline-level grid

## slide 4.2 grid-template-columns grid-template-rows

Defines the columns and rows of the grid with a space-separated list of values.

## slide 4.3 grid-template-columns grid-template-rows

The fr unit allows you to set the size of a track as a fraction of the free space of the grid container. For example, this will set each item to one third the width of the grid container:

## slide 4.3 grid-template-areas

Defines a grid template by referencing the names of the grid areas which are specified with the [grid-area](https://css-tricks.com/snippets/css/complete-guide-grid/#prop-grid-area) property

Repeating the name of a grid area causes the content to span those cells.

A period signifies an empty cell.

The syntax itself provides a visualization of the structure of the grid.

That’ll create a grid that’s four columns wide by three rows tall. The entire top row will be composed of the **header** area. The middle row will be composed of two **main** areas, one empty cell, and one **sidebar** area. The last row is all **footer**.

## Slide 4.4 - column-gap, row-gap, gap

Specifies the size of the grid lines. You can think of it like setting the width of the gutters between the columns/rows. The gutters are only created between the columns/rows, not on the outer edges.

Gap - A shorthand for [row-gap](https://css-tricks.com/snippets/css/complete-guide-grid/#prop-grid-column-row-gap) and [column-gap](https://css-tricks.com/snippets/css/complete-guide-grid/#prop-grid-column-row-gap)

## Slide 4.5 - justify-items

Aligns grid items along the  row axis

* **start** – aligns items to be flush with the start edge of their cell
* **end** – aligns items to be flush with the end edge of their cell
* **center** – aligns items in the center of their cell
* **stretch** – fills the whole width of the cell (this is the default)

## Slide 4.6 - align-items

Aligns grid items along the column axis. This value applies to all grid items inside the container.

* **start** – aligns items to be flush with the start edge of their cell
* **end** – aligns items to be flush with the end edge of their cell
* **center** – aligns items in the center of their cell
* **stretch** – fills the whole height of the cell (this is the default)

## Slide 4.7 - justify-content

Sometimes the total size of your grid might be less than the size of its grid container.

This could happen if all of your grid items are sized with non-flexible units like px.

In this case you can set the alignment of the grid within the grid container.

This property aligns the grid along the row axis

* **start** – aligns the grid to be flush with the start edge of the grid container
* **end** – aligns the grid to be flush with the end edge of the grid container
* **center** – aligns the grid in the center of the grid container
* **stretch** – resizes the grid items to allow the grid to fill the full width of the grid container
* **space-around** – places an even amount of space between each grid item, with half-sized spaces on the far ends
* **space-between** – places an even amount of space between each grid item, with no space at the far ends
* **space-evenly** – places an even amount of space between each grid item, including the far ends

## Slide 4.7 - align-content

This property aligns the grid along the *column* axis

* **start** – aligns the grid to be flush with the start edge of the grid container
* **end** – aligns the grid to be flush with the end edge of the grid container
* **center** – aligns the grid in the center of the grid container
* **stretch** – resizes the grid items to allow the grid to fill the full height of the grid container
* **space-around** – places an even amount of space between each grid item, with half-sized spaces on the far ends
* **space-between** – places an even amount of space between each grid item, with no space at the far ends
* **space-evenly** – places an even amount of space between each grid item, including the far ends