CSS Grid Property:

## Display Property

An HTML element becomes a grid container when its display property is set to grid or inline-grid.

All direct children of the grid container automatically become grid items.

## Grid Columns

The vertical lines of grid items are called columns.

## Grid Rows

The horizontal lines of grid items are called rows.

## Grid Gaps

The spaces between each column/row are called gaps.

You can adjust the gap size by using one of the following properties:

* column-gap
* row-gap
* gap

## Grid Lines

The lines between columns are called column lines.

The lines between rows are called row lines.

## The grid-template-columns Property

The grid-template-columns property defines the number of columns in your grid layout, and it can define the width of each column.

The value is a space-separated-list, where each value defines the width of the respective column.

If you want your grid layout to contain 4 columns, specify the width of the 4 columns, or "auto" if all columns should have the same width.

**Note:** If you have more than 4 items in a 4 columns grid, the grid will automatically add a new row to put the items in.

The grid-template-columns property can also be used to specify the size (width) of the columns.

## The grid-template-rows Property

The grid-template-rows property defines the height of each row.

## The justify-content Property

The justify-content property is used to align the whole grid inside the container.

**Note:** The grid's total width has to be less than the container's width for the justify-content property to have any effect.

## The align-content Property

The align-content property is used to vertically align the whole grid inside the container.

**Note:** The grid's total height has to be less than the container's height for the align-content property to have any effect.

# **CSS Grid Item**

Child Elements (Items)

A grid *container* contains grid *items*.

By default, a container has one grid item for each column, in each row, but you can style the grid items so that they will span multiple columns and/or rows.

The grid-column Property:

The grid-column property defines on which column(s) to place an item.

You define where the item will start, and where the item will end.

**Note:** The grid-column property is a shorthand property for the grid-column-start and the grid-column-end properties.

To place an item, you can refer to line numbers, or use the keyword "span" to define how many columns the item will span.

## The grid-row Property:

The grid-row property defines on which row to place an item.

You define where the item will start, and where the item will end.

**Note:** The grid-row property is a shorthand property for the grid-row-start and the grid-row-end properties.

## The grid-area Property

The grid-area property can be used as a shorthand property for the grid-row-start, grid-column-start, grid-row-end and the grid-column-end properties.

## Naming Grid Items

The grid-area property can also be used to assign names to grid items.

Named grid items can be referred to by the grid-template-areas property of the grid container.

Each row is defined by apostrophes (' ')

The columns in each row is defined inside the apostrophes, separated by a space.

## The Order of the Items

The Grid Layout allows us to position the items anywhere we like.

The first item in the HTML code does not have to appear as the first item in the grid.