CSS Grid

A grid can be defined as an intersecting set of horizontal lines and vertical lines. CSS Grid layout divides a page into major regions. It defines the relationship between the parts of a control built from [HTML](https://www.javatpoint.com/html-tutorial) primitives in terms of layer, position, and size. Grid property offers a grid-based layout system having rows and columns. It makes the designing of web pages easy without positioning and floating.

Similar to the table, it enables a user to align the elements into rows and columns. But compare to tables, it is easy to design layout with the [CSS](https://www.javatpoint.com/css-tutorial) grid. We can define columns and rows on the grid by using **grid-template-rows** and **grid-template-columns** properties.

The [CSS](https://www.javatpoint.com/css-full-form) **grid property** is supported in browsers such as **Google Chrome, Internet Explorer, Firefox, Safari,** and **Opera**.

Grid Container

We can define the grid container by setting the **display** property to **grid** or **inline-grid** on an element.

Grid container contains grid items that are placed inside rows and columns.

Let's see a simple example of a grid in CSS.

Example

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

        .main {

            display: grid;

            grid: auto auto / auto auto auto auto;

            grid-gap: 10px;

            background-color: black;

            padding: 10px;

        }

        .num {

            background-color: grey;

            text-align: center;

            color: white;

            padding: 10px 10px;

            font-size: 30px;

        }

**</style>**

**</head>**

**<body>**

**<div** class="main"**>**

**<div** class="num"**>**One**</div>**

**<div** class="num"**>**Two**</div>**

**<div** class="num"**>**Three**</div>**

**<div** class="num"**>**Four**</div>**

**<div** class="num"**>**Five**</div>**

**<div** class="num"**>**Six**</div>**

**<div** class="num"**>**Seven**</div>**

**<div** class="num"**>**Eight**</div>**

**</div>**

**</body>**

**</html>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=CSSGrid1)

Let's see some of the shorthand properties:

**grid-template-columns:** It is used to specify the size of the columns.

**grid-template-rows:** It is used to specify the row size.

**grid-template-areas:** It is used to specify the grid layout by using the named items.

**grid-auto-rows:** It is used to specify the automatic size of the rows.

**grid-auto-columns:** It is used to specify the automatic size of the columns.

**grid-auto-flow:** It is used to specify how to place auto-placed items and the automatic row size.

In the following example, we are including some of the above shorthand properties. Now, let's see the example of using some properties:

Example

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

        .main {

            display: grid;

            grid-template-columns: 1fr 1fr 1fr;

            grid-template-rows: 100px 250px 200px;

            background-color: black;

            grid-gap: 10px;

            padding: 20px;

        }

        .num {

            background-color: lightgrey;

            text-align: center;

            padding: 20px 10px;

            font-size: 30px;

        }

**</style>**

**</head>**

**<body>**

**<div** class="main"**>**

**<div** class="num"**>**One**</div>**

**<div** class="num"**>**Two**</div>**

**<div** class="num"**>**Three**</div>**

**<div** class="num"**>**Four**</div>**

**<div** class="num"**>**Five**</div>**

**<div** class="num"**>**Six**</div>**

**<div** class="num"**>**Seven**</div>**

**<div** class="num"**>**Eight**</div>**

**</div>**

**</body>**

**</html>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=CSSGrid2)

The justify-content property

It is used to align the entire grid within the container. It includes values such as:

**space-evenly:** It provides equal space in between or around the columns.

**space-around:** It provides equal space around the columns.

**space-between:** It gives an equal amount of space between the columns.

**center:** It is used to align the grid in the middle of the container.

**start:** It is used to align the grid at the beginning of the container.

**end:** It is used to align the grid at the end of the container.

Note: It is noted that the total width of the grid should be less than the width of the container for any effect of the justify-content property.

Let's understand this property along with the values by using an example.

Example

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

.grid-container1 {

  display: grid;

  justify-content: space-evenly;

  grid-template-columns: 50px 50px 50px; /\*Make the grid smaller than the container\*/

  grid-gap: 10px;

  background-color: #2196F3;

  padding: 10px;

}

.grid-container2 {

  display: grid;

  justify-content: space-around;

  grid-template-columns: 50px 50px 50px; /\*Make the grid smaller than the container\*/

  grid-gap: 10px;

  background-color: red;

  padding: 10px;

}

.grid-container3 {

  display: grid;

  justify-content: space-between;

  grid-template-columns: 50px 50px 50px; /\*Make the grid smaller than the container\*/

  grid-gap: 10px;

  background-color: green;

  padding: 10px;

}

.grid-container4 {

  display: grid;

  justify-content: end;

  grid-template-columns: 50px 50px 50px; /\*Make the grid smaller than the container\*/

  grid-gap: 10px;

  background-color: violet;

  padding: 10px;

}

.grid-container5 {

  display: grid;

  justify-content: start;

  grid-template-columns: 50px 50px 50px; /\*Make the grid smaller than the container\*/

  grid-gap: 10px;

  background-color: gray;

  padding: 10px;

}

.grid-container6 {

  display: grid;

  justify-content: center;

  grid-template-columns: 50px 50px 50px; /\*Make the grid smaller than the container\*/

  grid-gap: 10px;

  background-color: blue;

  padding: 10px;

}

.num {

  background-color: rgba(255, 255, 255, 0.8);

  text-align: center;

  padding: 20px 0;

  font-size: 30px;

}

**</style>**

**</head>**

**<body>**

**<b>**CONTAINER WITH SPACE-EVENLY VALUE**</b>**

**<div** class="grid-container1"**>**

**<div** class='num'**>**1**</div>**

**<div** class='num'**>**2**</div>**

**<div** class='num'**>**3**</div>**

**</div>**

**<b>**CONTAINER WITH SPACE-AROUND VALUE**</b>**

**<div** class="grid-container2"**>**

**<div** class='num'**>**1**</div>**

**<div** class='num'**>**2**</div>**

**<div** class='num'**>**3**</div>**

**</div>**

**<b>**CONTAINER WITH SPACE-BETWEEN VALUE**</b>**

**<div** class="grid-container3"**>**

**<div** class='num'**>**1**</div>**

**<div** class='num'**>**2**</div>**

**<div** class='num'**>**3**</div>**

**</div>**

**<b>**CONTAINER WITH END VALUE**</b>**

**<div** class="grid-container4"**>**

**<div** class='num'**>**1**</div>**

**<div** class='num'**>**2**</div>**

**<div** class='num'**>**3**</div>**

**</div>**

**<b>**CONTAINER WITH START VALUE**</b>**

**<div** class="grid-container5"**>**

**<div** class='num'**>**1**</div>**

**<div** class='num'**>**2**</div>**

**<div** class='num'**>**3**</div>**

**</div>**

**<b>**CONTAINER WITH CENTER VALUE**</b>**

**<div** class="grid-container6"**>**

**<div** class='num'**>**1**</div>**

**<div** class='num'**>**2**</div>**

**<div** class='num'**>**3**</div>**

**</div>**

**</body>**

**</html>**

[**Test it Now**](https://www.javatpoint.com/oprweb/test.jsp?filename=CSSGrid3)

The align-content property

It is used to align the entire grid within the container vertically.

Note: It is noted that the total height of the grid should be less than the height of the container for any effect of the align-content property.

The values of the **align-content** property are the same as the values of the **justify-content** property.