**CSS**

## Grid Elements =>(resource) w3schools

## <https://www.w3schools.com/css/css_grid.asp>

A grid layout consists of a parent element, with one or more child elements.

### Example

<div class="grid-container">  
  <div class="grid-item">1</div>  
  <div class="grid-item">2</div>  
  <div class="grid-item">3</div>  
  <div class="grid-item">4</div>  
  <div class="grid-item">5</div>  
  <div class="grid-item">6</div>  
  <div class="grid-item">7</div>  
  <div class="grid-item">8</div>  
  <div class="grid-item">9</div>  
</div>

Display Property

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

Example

.grid-container {  
  display: grid;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_display_grid)

Example

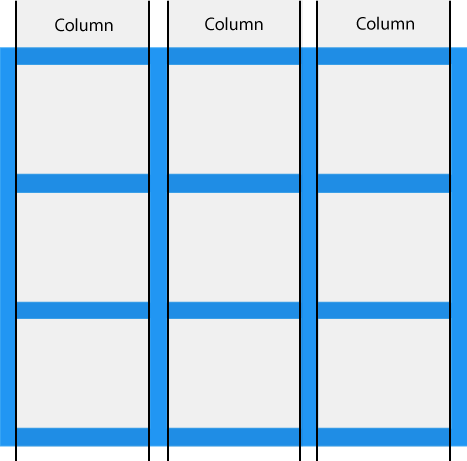
.grid-container {  
  display: inline-grid;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_display_inline-grid)

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

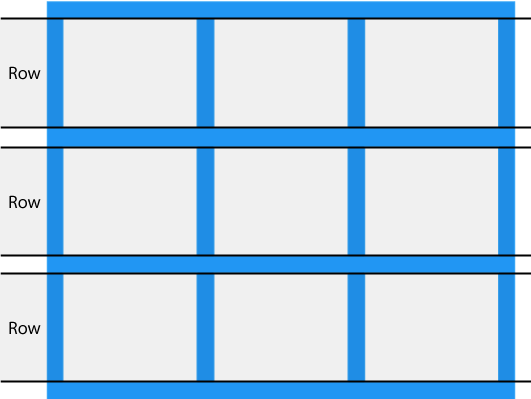
Grid Columns

The vertical lines of grid items are called *columns*.



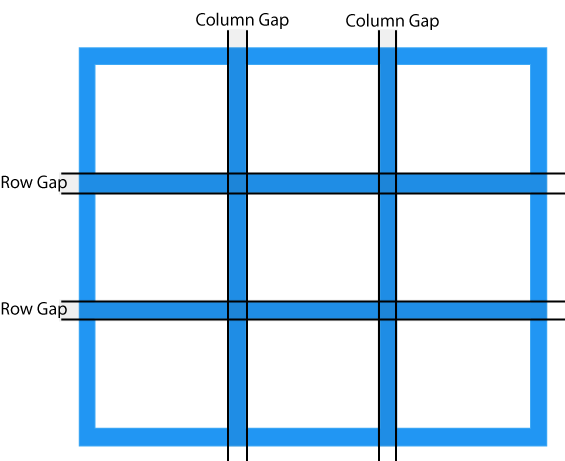
Grid Rows

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

Example

The column-gap property sets the gap between the columns:

.grid-container {  
  display: grid;  
**column-gap: 50px;**  
}

Example

The row-gap property sets the gap between the rows:

.grid-container {  
  display: grid;  
**row-gap: 50px;**}

Example

The gap property is a shorthand property for the row-gap and the column-gap properties:

.grid-container {  
  display: grid;  
**gap: 50px 100px;**}

Example

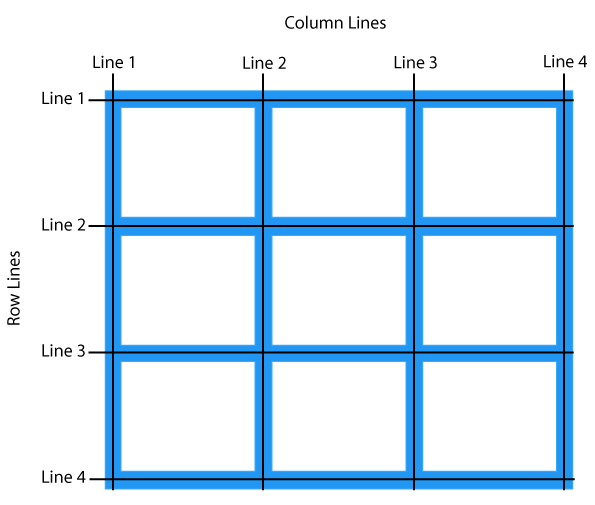
The gap property can also be used to set both the row gap and the column gap in one value:

.grid-container {  
  display: grid;  
**gap: 50px;**}

Grid Lines

The lines between columns are called *column lines*.

The lines between rows are called *row lines*.



Refer to line numbers when placing a grid item in a grid container:

Example

Place a grid item at column line 1, and let it end on column line 3:

.item1 {  
grid-column-start: 1;  
  grid-column-end: 3;  
}

Example

Place a grid item at row line 1, and let it end on row line 3:

.item1 {  
grid-row-start: 1;  
  grid-row-end: 3;  
}

## Grid Container

## <https://www.w3schools.com/css/css_grid_container.asp>

To make an HTML element behave as a grid container, you have to set the display property to grid or inline-grid.

Grid containers consist of grid items, placed inside columns and rows.

## 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.

### Example

Make a grid with 4 columns:

.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto auto;  
}

**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.

### Example

Set a size for the 4 columns:

.grid-container {  
  display: grid;  
  grid-template-columns: 80px 200px auto 40px;  
}

## The grid-template-rows Property

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

The value is a space-separated-list, where each value defines the height of the respective row:

### Example

.grid-container {  
  display: grid;  
  grid-template-rows: 80px 200px;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-template-rows)

## 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.

### Example

.grid-container {  
  display: grid;  
  justify-content: space-evenly;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_justify-content_space-evenly)

### Example

.grid-container {  
  display: grid;  
  justify-content: space-around;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_justify-content_space-around)

### Example

.grid-container {  
  display: grid;  
  justify-content: space-between;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_justify-content_space-between)

### Example

.grid-container {  
  display: grid;  
  justify-content: center;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_justify-content_center)

### Example

.grid-container {  
  display: grid;  
  justify-content: start;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_justify-content_start)

### Example

.grid-container {  
  display: grid;  
  justify-content: end;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_justify-content_end)

## 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.

### Example

.grid-container {  
  display: grid;  
  height: 400px;  
  align-content: center;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_align-content_center)

### Example

.grid-container {  
  display: grid;  
  height: 400px;  
  align-content: space-evenly;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_align-content_space-evenly)

### Example

.grid-container {  
  display: grid;  
  height: 400px;  
  align-content: space-around;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_align-content_space-around)

### Example

.grid-container {  
  display: grid;  
  height: 400px;  
  align-content: space-between;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_align-content_space-between)

### Example

.grid-container {  
  display: grid;  
  height: 400px;  
  align-content: start;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_align-content_start)

### Example

.grid-container {  
  display: grid;  
  height: 400px;  
  align-content: end;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_align-content_end)

<https://www.w3schools.com/css/css_grid_item.asp>

## 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.

### Example

Make "item1" start on column 1 and end before column 5:

.item1 {  
  grid-column: 1 / 5;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-column_line)

### Example

Make "item1" start on column 1 and span 3 columns:

.item1 {  
  grid-column: 1 / span 3;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-column)

### Example

Make "item2" start on column 2 and span 3 columns:

.item2 {  
  grid-column: 2 / span 3;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-column2)

## 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.

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

### Example

Make "item1" start on row-line 1 and end on row-line 4:

.item1 {  
  grid-row: 1 / 4;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-row1)

### Example

Make "item1" start on row 1 and span 2 rows:

.item1 {  
  grid-row: 1 / span 2;  
}

## 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.

### Example

Make "item8" start on row-line 1 and column-line 2, and end on row-line 5 and column line 6:

.item8 {  
  grid-area: 1 / 2 / 5 / 6;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-area1)

### Example

Make "item8" start on row-line 2 and column-line 1, and span 2 rows and 3 columns:

.item8 {  
  grid-area: 2 / 1 / span 2 / span 3;  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-area2)

## 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.

### Example

Item1 gets the name "myArea" and spans all five columns in a five columns grid layout:

.item1 {  
  grid-area: myArea;  
}  
.grid-container {  
  grid-template-areas: 'myArea myArea myArea myArea myArea';  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-area_named)

Each row is defined by apostrophes (' ')

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

**Note:** A period sign represents a grid item with no name.

### Example

Let "myArea" span two columns in a five columns grid layout (period signs represent items with no name):

.item1 {  
  grid-area: myArea;  
}  
.grid-container {  
  grid-template-areas: 'myArea myArea . . .';  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-area_named1)

To define two rows, define the column of the second row inside another set of apostrophes:

### Example

Make "item1" span two columns and two rows:

.grid-container {  
  grid-template-areas: 'myArea myArea . . .' 'myArea myArea . . .';  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-area_named2)

### Example

Name all items, and make a ready-to-use webpage template:

.item1 { grid-area: header; }  
.item2 { grid-area: menu; }  
.item3 { grid-area: main; }  
.item4 { grid-area: right; }  
.item5 { grid-area: footer; }  
  
.grid-container {  
  grid-template-areas:  
    'header header header header header header'  
    'menu main main main right right'  
    'menu footer footer footer footer footer';  
}

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_grid-area_named3)

## 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.

### Example

.item1 { grid-area: 1 / 3 / 2 / 4; }  
.item2 { grid-area: 2 / 3 / 3 / 4; }  
.item3 { grid-area: 1 / 1 / 2 / 2; }  
.item4 { grid-area: 1 / 2 / 2 / 3; }  
.item5 { grid-area: 2 / 1 / 3 / 2; }  
.item6 { grid-area: 2 / 2 / 3 / 3; }

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_flexible_order)

You can re-arrange the order for certain screen sizes, by using media queries:

### Example

@media only screen and (max-width: 500px) {  
  .item1 { grid-area: 1 / span 3 / 2 / 4; }  
  .item2 { grid-area: 3 / 3 / 4 / 4; }  
  .item3 { grid-area: 2 / 1 / 3 / 2; }  
  .item4 { grid-area: 2 / 2 / span 2 / 3; }  
  .item5 { grid-area: 3 / 1 / 4 / 2; }  
  .item6 { grid-area: 2 / 3 / 3 / 4; }  
}

All CSS Grid Properties :

|  |  |
| --- | --- |
| **Property** | **Description** |
| [column-gap](https://www.w3schools.com/cssref/css3_pr_column-gap.asp) | Specifies the gap between the columns |
| [gap](https://www.w3schools.com/cssref/css3_pr_gap.asp) | A shorthand property for the *row-gap* and the *column-gap* properties |
| [grid](https://www.w3schools.com/cssref/pr_grid.asp) | A shorthand property for the *grid-template-rows, grid-template-columns, grid-template-areas, grid-auto-rows, grid-auto-columns*, and the *grid-auto-flow* properties |
| [grid-area](https://www.w3schools.com/cssref/pr_grid-area.asp) | Either specifies a name for the grid item, or this property is a shorthand property for the *grid-row-start*, *grid-column-start*, *grid-row-end*, and *grid-column-end* properties |
| [grid-auto-columns](https://www.w3schools.com/cssref/pr_grid-auto-columns.asp) | Specifies a default column size |
| [grid-auto-flow](https://www.w3schools.com/cssref/pr_grid-auto-flow.asp) | Specifies how auto-placed items are inserted in the grid |
| [grid-auto-rows](https://www.w3schools.com/cssref/pr_grid-auto-rows.asp) | Specifies a default row size |
| [grid-column](https://www.w3schools.com/cssref/pr_grid-column.asp) | A shorthand property for the *grid-column-start* and the *grid-column-end* properties |
| [grid-column-end](https://www.w3schools.com/cssref/pr_grid-column-end.asp) | Specifies where to end the grid item |
| [grid-column-gap](https://www.w3schools.com/cssref/pr_grid-column-gap.asp) | Specifies the size of the gap between columns |
| [grid-column-start](https://www.w3schools.com/cssref/pr_grid-column-start.asp) | Specifies where to start the grid item |
| [grid-gap](https://www.w3schools.com/cssref/pr_grid-gap.asp) | A shorthand property for the *grid-row-gap* and *grid-column-gap* properties |
| [grid-row](https://www.w3schools.com/cssref/pr_grid-row.asp) | A shorthand property for the *grid-row-start* and the *grid-row-end* properties |
| [grid-row-end](https://www.w3schools.com/cssref/pr_grid-row-end.asp) | Specifies where to end the grid item |
| [grid-row-gap](https://www.w3schools.com/cssref/pr_grid-row-gap.asp) | Specifies the size of the gap between rows |
| [grid-row-start](https://www.w3schools.com/cssref/pr_grid-row-start.asp) | Specifies where to start the grid item |
| [grid-template](https://www.w3schools.com/cssref/pr_grid-template.asp) | A shorthand property for the *grid-template-rows*, *grid-template-columns* and *grid-areas* properties |
| [grid-template-areas](https://www.w3schools.com/cssref/pr_grid-template-areas.asp) | Specifies how to display columns and rows, using named grid items |
| [grid-template-columns](https://www.w3schools.com/cssref/pr_grid-template-columns.asp) | Specifies the size of the columns, and how many columns in a grid layout |
| [grid-template-rows](https://www.w3schools.com/cssref/pr_grid-template-rows.asp) | Specifies the size of the rows in a grid layout |
| [row-gap](https://www.w3schools.com/cssref/css3_pr_row-gap.asp) | Specifies the gap between the grid rows |