

# CSS Notes For Students By Geekster



Geekster

[www.geekster.in](http://www.geekster.in)

# Grid :

Definition :

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, makes it easier to style site without have to use floats and position .

## Grid Elements

A grid layout consists of a parent element. They have one or more child elements.

Example :

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #f4cd23;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
</head>
<body>

<h1>Grid Elements</h1>

<p>A Grid Layout must have a parent element with the <em>display</em>
property set to <em>grid</em> or <em>inline-grid</em>.</p>
```

<p>Direct child element(s) of the grid container automatically becomes grid items.</p>

```
<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>

</body>
</html>
```

Output:



### Grid Elements

A Grid Layout must have a parent element with the *display* property set to *grid* or *inline-grid*.

Direct child element(s) of the grid container automatically becomes grid items.

1	2	3
4	5	6
7	8	9

## Grid Columns

Vertical lines of grid items are called columns.

## Grid Rows

Horizontal lines of grid items are called *rows*.

## Grid Gaps

Spaces between each column/row are called *gaps*.

## Grid Lines

The lines between columns are called *column lines*.

The lines between rows are called *row lines*.

## Example

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

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  gap: 10px;
  background-color: #b4f121;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-column-start: 1;
  grid-column-end: 3;
}
</style>
</head>
<body>
```

```
<h1>Grid Lines</h1>
```

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
</div>
```

```
<p>You can refer to line numbers when placing grid items.</p>
```

```
</body>
```

```
</html>
```

Output:



1		2
3	4	5
6	7	8

You can refer to line numbers when placing grid items.

## grid-template-columns

The `grid-template-columns` property specifies the quality (and the widths) of columns in a grid layout.

The values are a area separated list, where each value specifies the dimensions of the respective column.

Example :

```
<!DOCTYPE html>
<html>
<head>
<style>

.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>The grid-template-columns Property</h1>

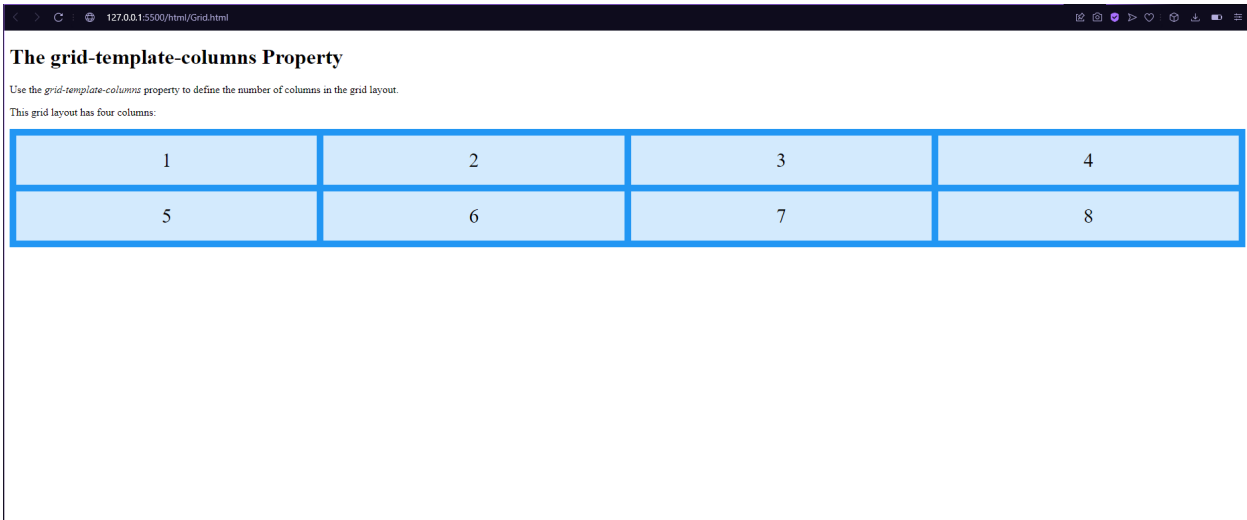
<p>Use the <em>grid-template-columns</em> property to define the number
of columns in the grid layout.</p>
<p>This grid layout has four columns:</p>

<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
```

```
<div class="item8">8</div>
</div>

</body>
</html>
```

Output:



# Grid-template-rows

The `grid-template-rows` property specifies the number of the rows in Exceedingly a grid layout.

The values are a space-separated list, where each value specifies the peak of the respective row.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto;
  grid-template-rows: 100px 300px;
  grid-gap: 10px;
  background-color: #000000;
```

```
padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>The grid-template-rows Property</h1>

<p>Use the <em>grid-template-rows</em> property to define the size of
the rows in a grid layout.</p>
<p>The first row in this grid is 100px high, and the second row is 300px
high:</p>

<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
</div>

</body>
</html>
```

Output:



## The grid-template-rows Property

Use the *grid-template-rows* property to define the size of the rows in a grid layout.

The first row in this grid is 100px high, and the second row is 300px high:

1	2	3	4
5	6	7	8

# CSS Units

CSS are several different unit for expressing a length.

Many CSS properties like "length" values, such as width ,margin , padding ,font-size etc.

Length is a number followed by a length unit, such as 10px, 2em etc.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  font-size: 60px;
}

p {
  font-size: 25px;
  line-height: 50px;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
```

```
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

Output:

< > C 127.0.0.1:5500/html/Grid.html

# This is heading 1

**This is heading 2**

This is a paragraph.

This is another paragraph.

## Column-gap

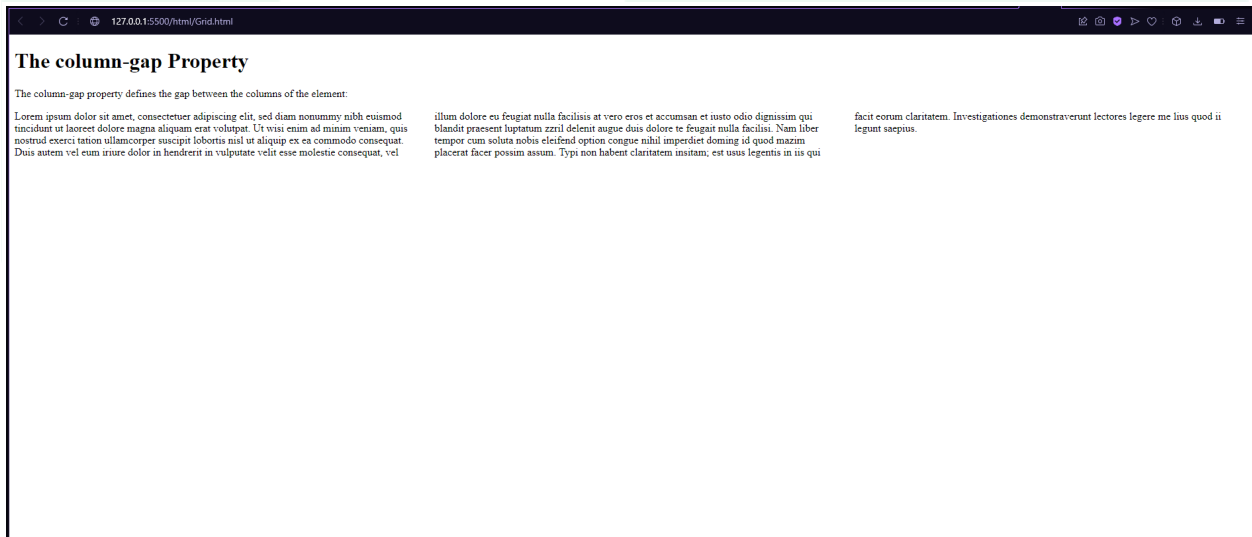
The column-gap property specifies the gap between columns.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 .newspaper {
6   column-count: 3;
7   column-gap: 40px;
8 }
9 </style>
10 </head>
11 <body>
12
13 <h1>The column-gap Property</h1>
14 <p>The column-gap property defines the gap between the columns of the element:</p>
15
16 <div class="newspaper">
17   Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod
18   tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exer
19 </div>
20 </body>
21 </html>
22

```

Output:



# Grid-row-gap

The grid-row-gap property defines the size of the gap between rows in a grid layout.

Example:

```

<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;

```

```

    grid-template-columns: auto auto auto auto;
    grid-row-gap: 50px;
    grid-column-gap: 10px;
    background-color: #2196F3;
    padding: 10px;
}

.grid-container > div {
    background-color: rgba(255, 255, 255, 0.8);
    text-align: center;
    padding: 20px 0;
    font-size: 30px;
}
</style>
</head>
<body>

<h1>The grid-row-gap Property</h1>

<p>Use the grid-row-gap property to specify the size of the gap
between the rows.</p>
<p>This grid has a 50px gap between rows (and a 10px gap between
columns):</p>

<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
  <div class="item9">9</div>
  <div class="item10">10</div>
  <div class="item11">11</div>
  <div class="item12">12</div>
</div>

</body>
</html>

```

Output:

## The grid-row-gap Property

Use the *grid-row-gap* property to specify the size of the gap between the rows.

This grid has a 50px gap between rows (and a 10px gap between columns):

1	2	3	4
5	6	7	8
9	10	11	12

# Grid-column

Grid-column property specifies a grid items size and location in a grid layout, and is a shorthand property .

Example :

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

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

```

</style>
</head>
<body>

<h1>The grid-column Property</h1>

<p>Use the grid-column property to specify on which column to
place an item, and how many columns the item will span.</p>
<p>This property is a shorthand property for the grid-column-start and
grid-column-end properties.</p>
<p>Item1 will start on column 1, and span 2 columns:</p>

<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
</div>

</body>
</html>

```

Output:

The grid-column Property			
Use the <i>grid-column</i> property to specify on which column to place an item, and how many columns the item will span.			
This property is a shorthand property for the <i>grid-column-start</i> and <i>grid-column-end</i> properties.			
Item1 will start on column 1, and span 2 columns:			
1	2	3	
4	5	6	7

# Grid-row

Grid-row property specifies a grid item's size and location in a grid layout, and is a shorthand property .

Example :

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-row: 1 / span 2;
}
</style>
</head>
<body>

<h1>The grid-row Property</h1>

<p>Use the <em>grid-row</em> property to specify on which row to place
an item, and how many row the item will span.</p>
<p>This property is a shorthand property for the grid-row-start and
grid-row-end properties.</p>
<p>Item1 will start on row 1, and span 2 rows:</p>

<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
```

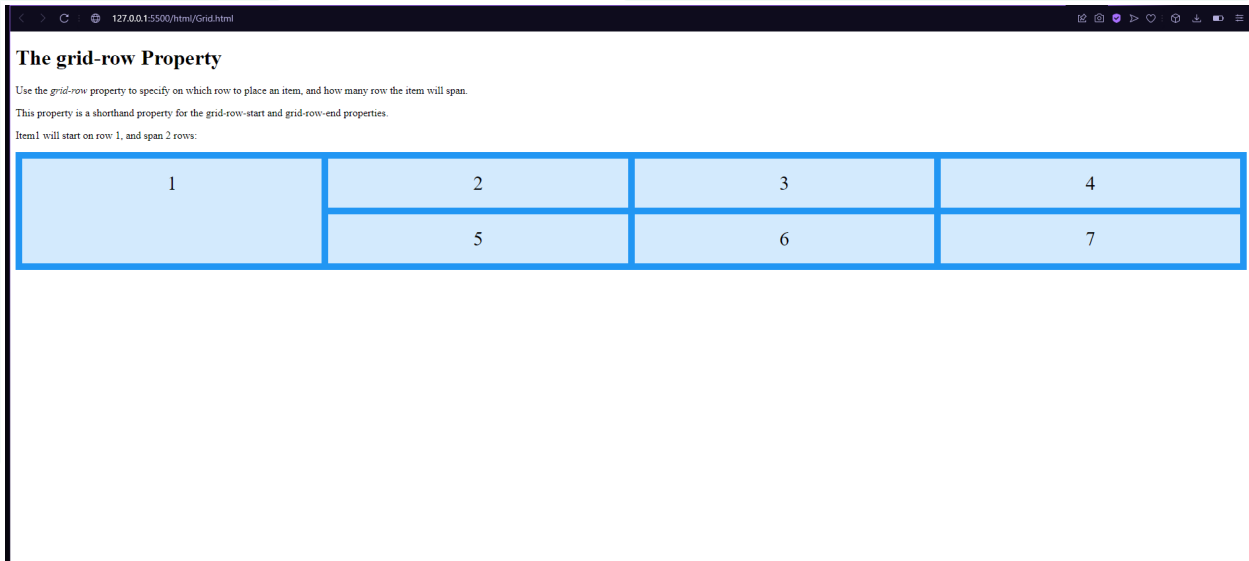
```

<div class="item6">6</div>
<div class="item7">7</div>
</div>

</body>
</html>

```

Output:



# Grid-row-start

The *grid-row-start* property defines on which row-line item will start.

Example :

```

<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

```



```
.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-row-start: 2;
}
</style>
</head>
<body>

<h1>The grid-row-start Property</h1>

<p>Use the grid-row-start property to specify on which row to
start displaying the item.</p>
<p>Item1 will start on row 2:</p>

<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
</div>

</body>
</html>
```

Output :

## The grid-row-start Property

Use the *grid-row-start* property to specify on which row to start displaying the item.

Item1 will start on row 2:

2	3	4	5
1	6		

# Grid-row-end

The grid-row-end property defines how many rows an item will span, or on which row-line item will end

Example :

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  grid-gap: 10px;
  background-color: #8321f3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
```

```
    grid-row-end: span 3;
}
</style>
</head>
<body>

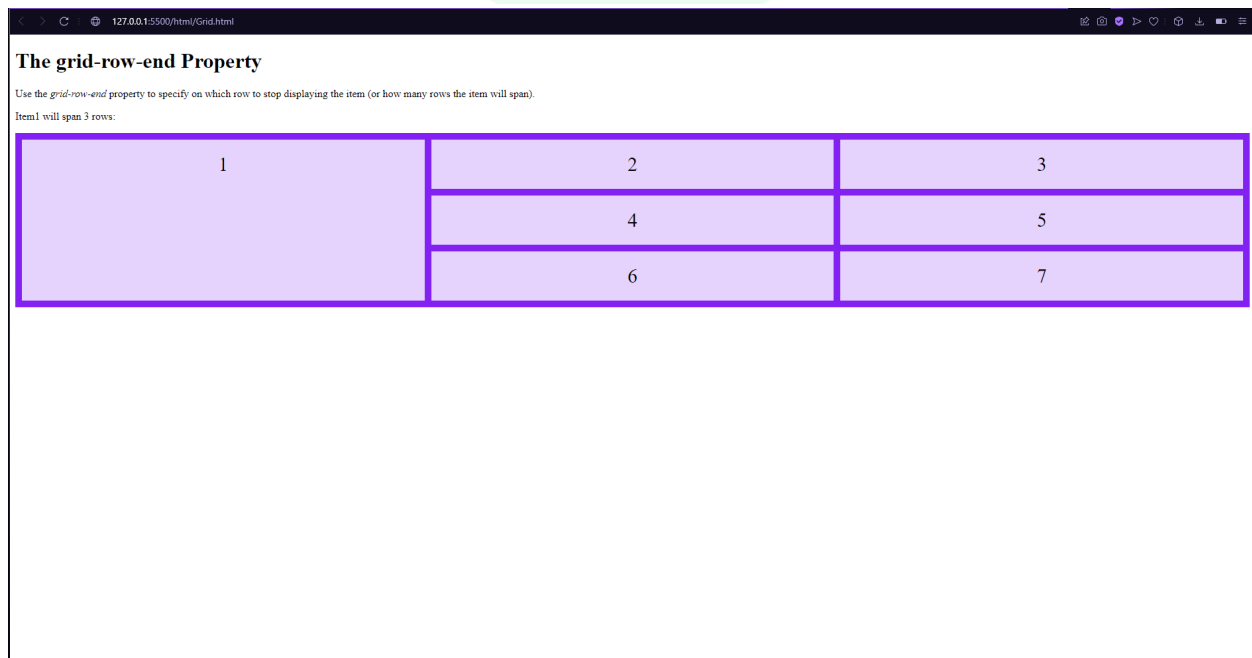
<h1>The grid-row-end Property</h1>

<p>Use the grid-row-end property to specify on which row to
stop displaying the item (or how many rows the item will span).</p>
<p>Item1 will span 3 rows:</p>

<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
</div>

</body>
</html>
```

## Output:



1	2	3
	4	5
	6	7

## Reference Links :

1. [https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)
2. [https://www.w3schools.com/css/css\\_grid\\_container.asp](https://www.w3schools.com/css/css_grid_container.asp)
3. [https://www.w3schools.com/css/css\\_grid\\_item.asp](https://www.w3schools.com/css/css_grid_item.asp)
4. [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout)
5. <https://www.youtube.com/watch?v=BNmxUzPRYdw>
6. <https://www.youtube.com/watch?v=9zBsdzdE4sM>