

# CSS Grid Intro

- description : CSS Responsive Web Design- Images
- author :
- email : shlim@repia.com
- lastupdate : 2021-07-02 Fri

## The source of this article

[CSS Grid Intro](#)

## CSS Grid Layout Module



## Grid Layout

CSS Grid Layout Module (rows) (columns), floats positioning

## Grid Elements

가

### Example

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color:#2196F3;
```

```
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 display
property
set to grid or inline-grid.</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>
```



# Display Property

HTML      display      grid      inline-grid      가      .

## Example

```

<style>
  .grid-container {
    display: grid;
    grid-template-columns: auto auto auto; /* 3 ( ) ,
3 (가 )가 */
    background-color: #2196F3;
    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>The display Property:</h1>
<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>

<p>
  Set the <em>display</em> property to <em>grid</em> to make a block-
  level grid container.
</p>
</body>

```



## Example

```

<style>
  .grid-container {
    display: inline-grid;

```

```
grid-template-columns: auto auto auto; /* 3 ( ) ,
3 (가 )가 */
background-color: #2196F3;
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>The display Property:</h1>
<div class="grid-container">
  <!-- .grid-item*9 -->
  <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>

<p>
  Set the <em>display</em> property to <em>inline-grid</em>
  to make a block-level grid container.
</p>
</body>
```



# Grid Columns



( ) (columns) .

ref : The source of this png file is [CSS Grid Intro](#)

# Grid Rows

( ) (rows)



ref : The source of this png file is [CSS Grid Intro](#)

# Grid Gaps

/ (gaps)



- grid-column-gap
- grid-row-gap
- grid-gap

# Example

grid-column-gap ( )

```

<style>
  .grid-container {
    display: grid;
    grid-column-gap: 50px;
    grid-template-columns: auto auto auto;
    background-color: #2196F3;
    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>The grid-column-gap Property:</h1>

```
<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>
```

<p>Use the <em>grid-column-gap</em> property to adjust the **space** between the columns.</p>  
</body>



## Example

grid-row-gap (rows, 가 ) .

```
<style>
  .grid-container {
    display: grid;
    grid-template-columns: auto auto auto;
    background-color: #2196f3;
    padding: 10px;
    grid-row-gap: 50px;
  }

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

```
<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>
```

<p>Use the <em>grid-row-gap</em> property to adjust the **space** between the rows.</p>
</body>



## Example

grid-gap      grid-row-gap      grid-column-gap      (shorthand property)

```
<style>
  .grid-container {
    display: grid;
    grid-gap: 50px 100px; /* rows and columns in order */
    grid-template-columns: auto auto auto;
    background-color: #2196F3;
    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>The grid-gap Property:</h1>
```

```
<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>
```

<p>Use the <em>grid-gap</em> property to adjust the **space** between the columns <em>and</em>the rows.</p>

```
</body>
```



## Example

grid-gap

:

```
<style>  
  .grid-container {  
    display: grid;  
    grid-gap: 50px; /* one value for both rows and columns */  
    grid-template-columns: auto auto auto;  
    background-color: #2196F3;  
    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>The grid-gap Property:</h1>  
  
<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>

```

Use the `grid-gap` property to adjust the **space** between the columns and the rows.

```
</body>
```



## Grid Lines

(column lines)



### Example

1 , 3

```

<style>
  .grid-container {
    display: grid;
    grid-template-columns: 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-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>
```



## Example

1 , 3 .

```
<style>
  .grid-container {
    display: grid;
    grid-template-columns: 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: 1;
    grid-row-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>
```



## All CSS Grid Properties

„ [CSS](#), [Grid](#), [Layout](#), [Module](#)

From:

<http://rwiki.repia.com/> -

. - 2023.12

Permanent link:

[http://rwiki.repia.com/doku.php?id=wiki:css:css\\_note:grid\\_intro&rev=1625209051](http://rwiki.repia.com/doku.php?id=wiki:css:css_note:grid_intro&rev=1625209051)



Last update: **2022/03/10 19:52**