Grid Layout

Cascading Style Sheets (CSS3)

Table of Content

- Grid Layout

Grid Layout

(Cascading Style Sheets 3)

Grid Layout
(Basics)

- The CSS Grid Layout Module offers a grid-based layout system with
  - rows
  - Columns
- Grid layout is used to design web pages without having to use floats and positioning

Grid Layout
(Basics)

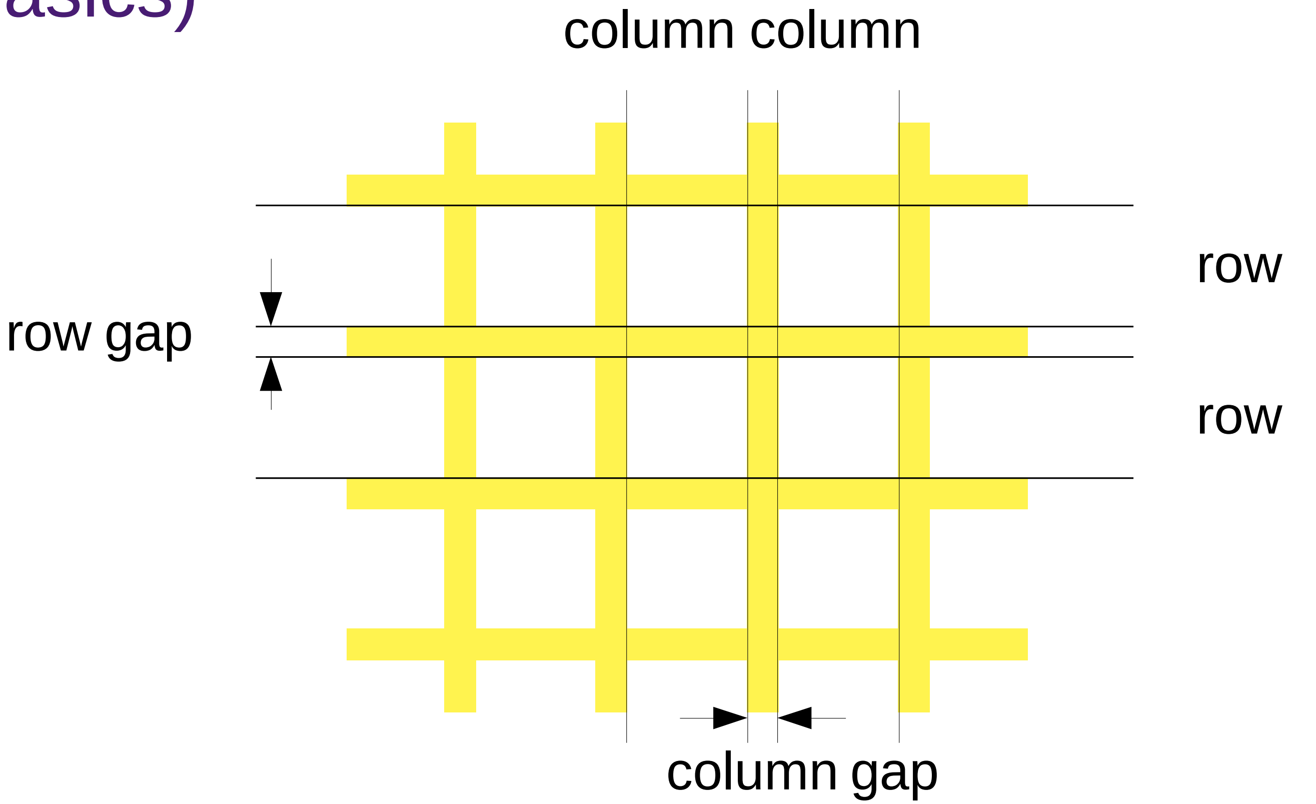
- A grid layout consists of a parent element (container), with one or more child elements (grid item)
- An HTML element becomes a grid container by setting the display property to grid, inline-grid or subgrid
- column, float, clear and vertical-align have no effect on a grid container

Syntax:

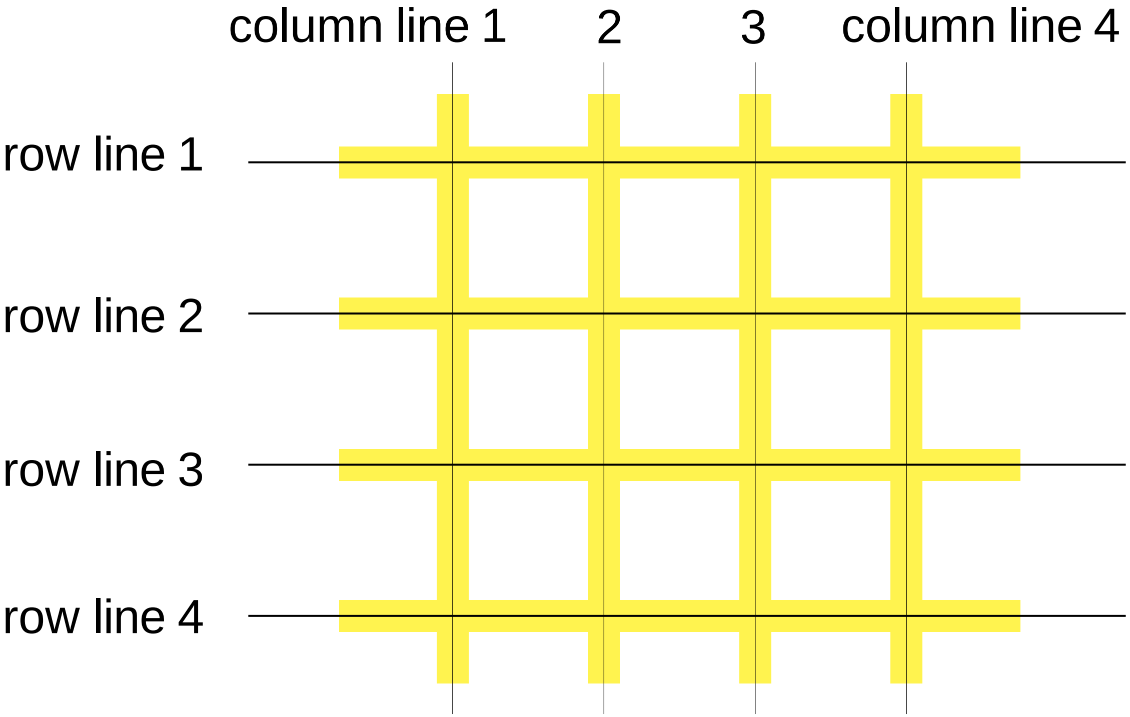
display : grid | inline-grid | subgrid;

Grid Layout

(Basics)



Grid lines



column line 1

2

3

column line 4

row line 1

row line 2

row line 3

row line 4

Grid Layout

(Basics)

- The gap size can be adjusted using one of these properties
  - grid-column-gap
  - grid-row-gap
  - grid-gap

Grid container

- Following properties shall be set for grid container

Example:

```
.grid-container {
display: grid;
grid-gap: 50px 50px; /* row-gap  column-gap */
/* 4 columns with equal size */
grid-template-columns: auto auto auto auto;
}
```

Property grid-template-column

- The grid-template-columns defines the number of columns in grid layout, and it can define the width of each column
- The value is a space-separated-list, where each value defines the length of the respective column
- Width can be specified in pixels, % or fr (flex factor)
- “auto” means all columns have the same width

Grid container

Examples:

```
/* 3 columns with equal size */
Grid-template-columns: 1fr 1fr 1fr;

/* 3 columns with unequal size */
Grid-template-columns: 1fr 2fr 1fr;
Or
Grid-template-columns: 20% 60% 20%;
Or
Grid-template-columns: 1fr 60% 1fr;
```

Grid container
(repeat function)

Examples:

```
/* 4 columns with equal size */
Grid-template-columns: repeat(4, 1fr);

/* 6 columns with repeat pattern (second is double than first) size */
Grid-template-columns: repeat(6, 1fr 2fr);
```

Property grid-template-rows

- The grid-template-rows defines the height of each row
- The value is a space-separated-list, where each value defines the height of the respective row
- Height can be specified in pixels
- “auto” means all rows have the same height
- Rows are automatically added to accommodate grid items

Grid item

- Following properties shall be set for grid item

Example:

```
.grid-item {
background-color: rgba(242, 242, 242, 0.8);
border: 1px solid rgba(0, 0, 0, 0.8);
}
```

The Grid

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

Spanning rows and columns

```
.spanning-column-item
  { grid-column-start: 1;
  grid-column-end: 3;
}

.spanning-row-item
  { grid-row-start: 2;
  grid-row-end: 4;
}
```

Spanning rows and columns

```
/* starting from column line 1 and ending at column line 3 */
.spanning-column-item
  { grid-column: 1 / 3;
}
/* starting from row line 2 and ending at row line 4 */
.spanning-row-item
  { grid-row: 2 / 4;
}
```

Spanning rows and columns

```
/* starting from column line 1 and spanning over 2 columns */
.spanning-column-item {
  grid-column: 1 / span
  2;
}
/* starting from row line 2 and spanning over 2 rows */
.spanning-row-item {
  grid-row: 2 / span
  2;
}
```

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, grid-column-end)

Grid area property

Syntax:

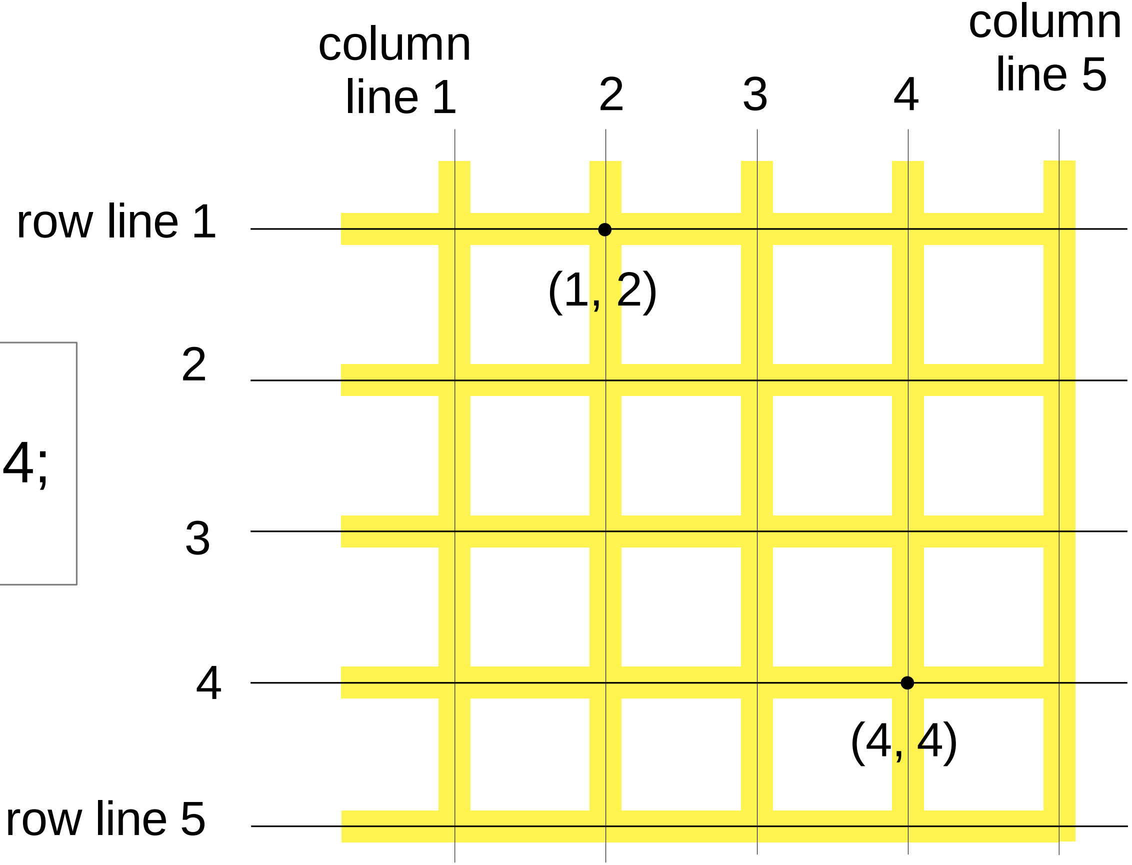
```
.grid-area-item {
  grid-area: row-start / col-start / row-end / col-end;
}


.grid-area-item {
  grid-area: row-start / col-start / span rows-count / span cols-count;
}
```

Grid area property

```
/* starting from row line 1 and ending at row line 4
   starting from col line 2 and ending at col line 4 */
.grid-area-item {
  grid-area: 1 / 2 / 4 / 4;
}
/* starting from row line 1 and spanning over 3 rows
   starting from col line 2 and spanning over 2 cols */
.grid-area-item {
  grid-area: 1 / 2 / span 3 / span 2;
}
```

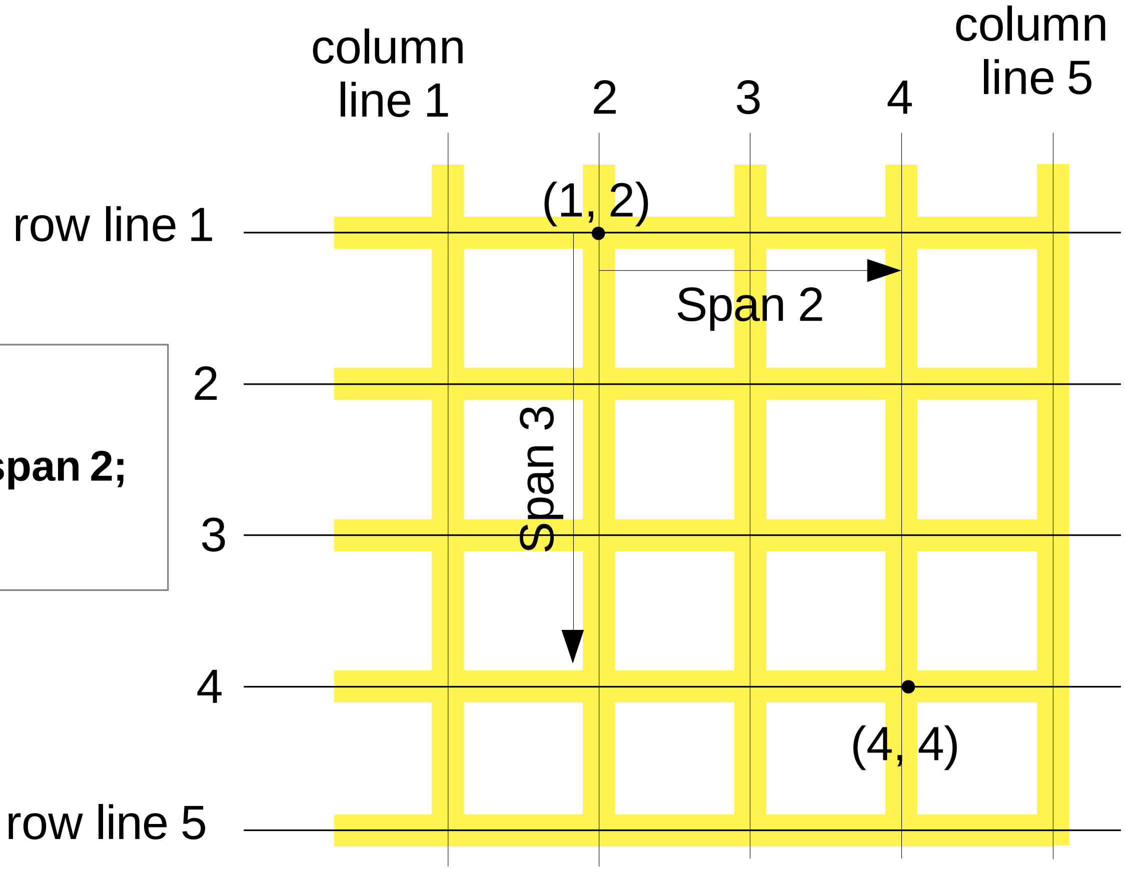
Grid Area

```
.grid-area-item {
  grid-area: 1 / 2 / 4 / 4;
}
```



Grid Area

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



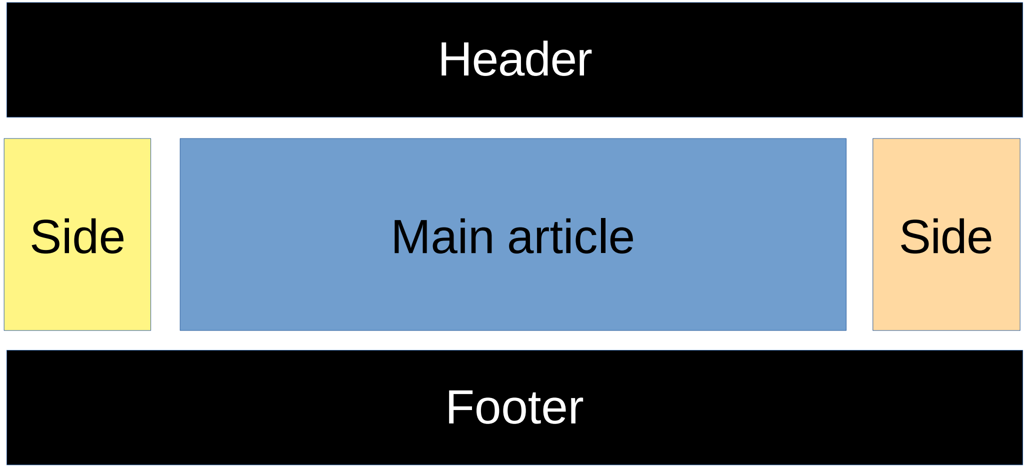
Grid area property

- 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

```
.grid-item {
  grid-area: itemname;
}
.grid-container {
  grid-template-areas: 'itemname itemname itemname itemname’;
}
```

Class work

- Design following layout using grid properties
- Use media queries to alter the layout for mobile and desktop



Header

Side

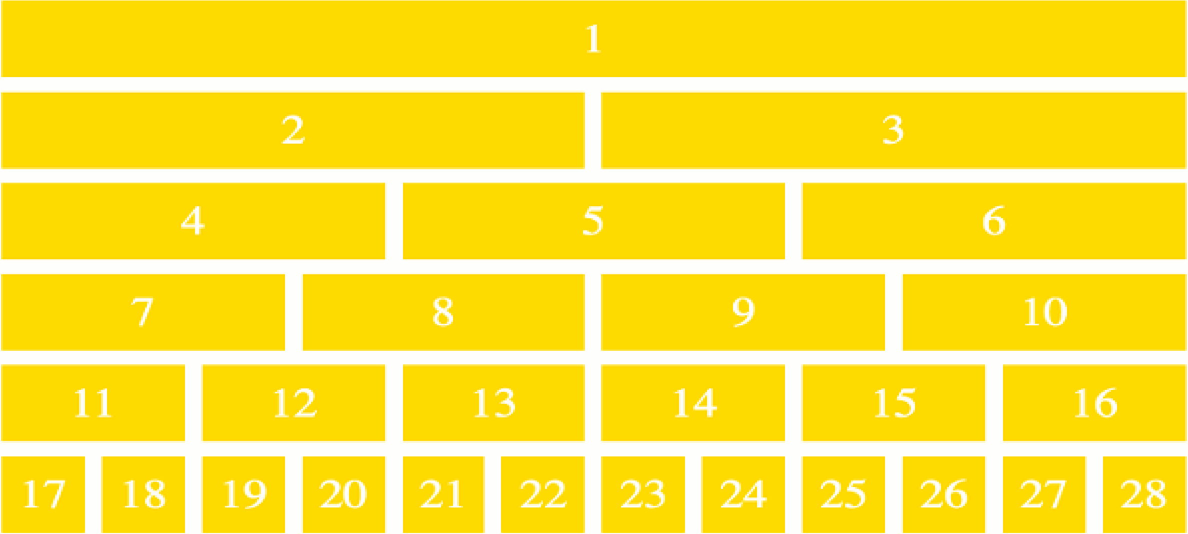
Main article

Side

Footer

Class work

- Design following layout using grid properties
- Use media queries to alter the layout for mobile and desktop



1

2 3

4 5 6

7 8 9 10

11 12 13 14 15 16

17 18 19 20 21 22 23 24 25 26 27 28

Web Stack Academy (P) Ltd

#83, Farah Towers,
1st floor,MG Road,
Bangalore –560001

M: +91-80-41289576
T: +91-98862 69112
E: info@www.webstackacademy.com