COMP 1950

Web Development and Design 2

Day 07

Agenda

- CSS Grid
- Assignment 06

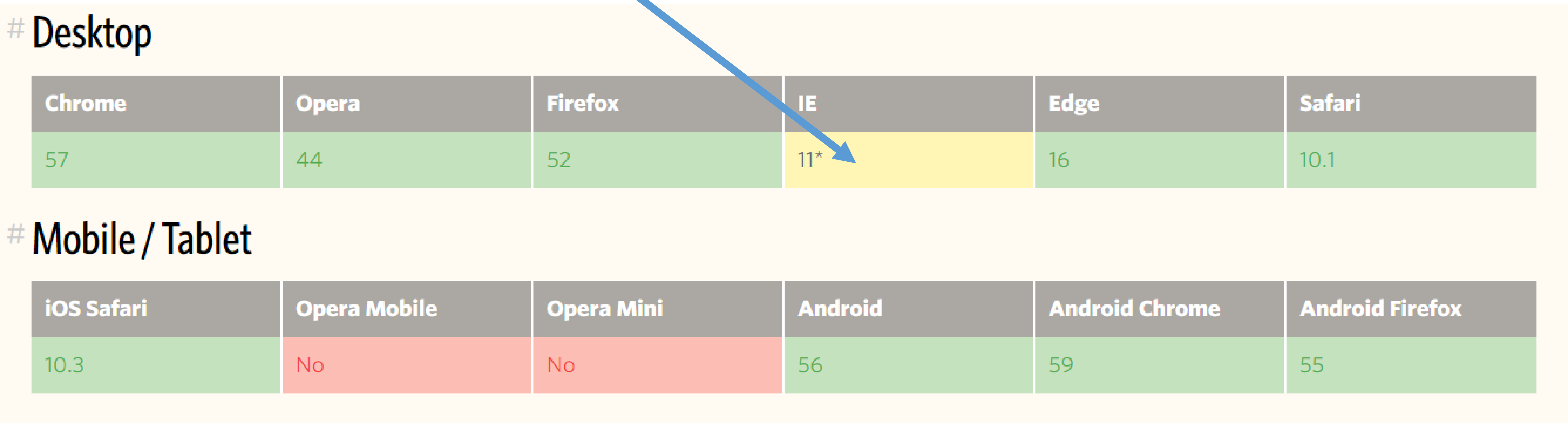
What is CSS Grid Layout?

- CSS Grid Layout (aka "Grid"), is a two-dimensional grid-based layout system[1]
- CSS Grid Layout is the most powerful layout system available in CSS. It is a 2-dimensional system, meaning it can handle both columns and rows, unlike flexbox which is largely a 1-dimensional system. You work with Grid Layout by applying CSS rules both to a parent element (which becomes the Grid Container) and to that elements children (which become Grid Items)[1]

1. https://css-tricks.com/snippets/css/complete-guide-grid/

Browser Support for Grid

*** Support is for an older outdated syntax.



Desktop

Chrome	Opera	Firefox	IE	Edge	Safari
57	44	52	11*	16	10.1

Mobile / Tablet

iOS Safari	Opera Mobile	Opera Mini	Android	Android Chrome	Android Firefox
10.3	No	No	56	59	55

Grid Terminology

Grid Container

Grid Line

Grid Area

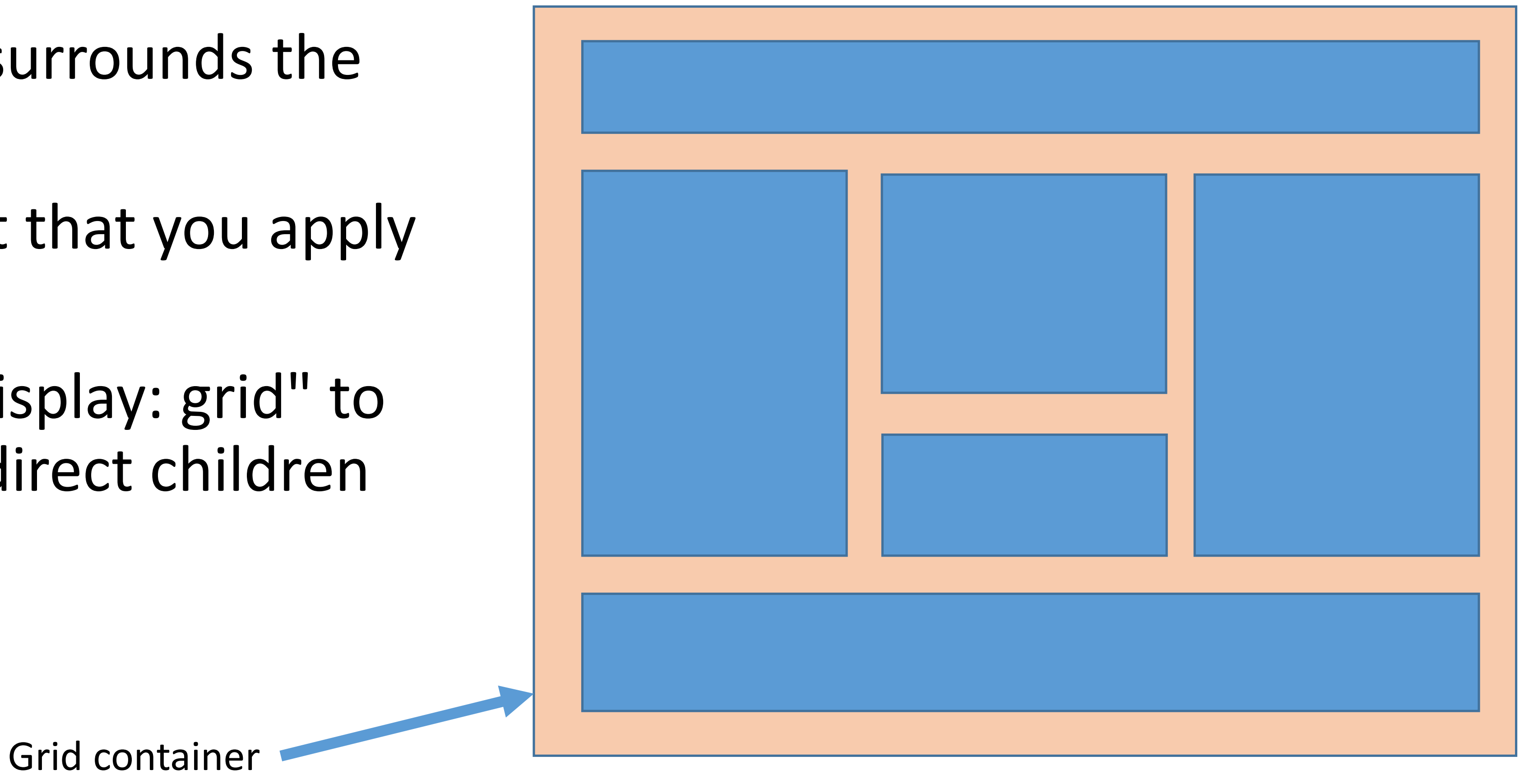
Grid Item

Grid Cell

Grid Track

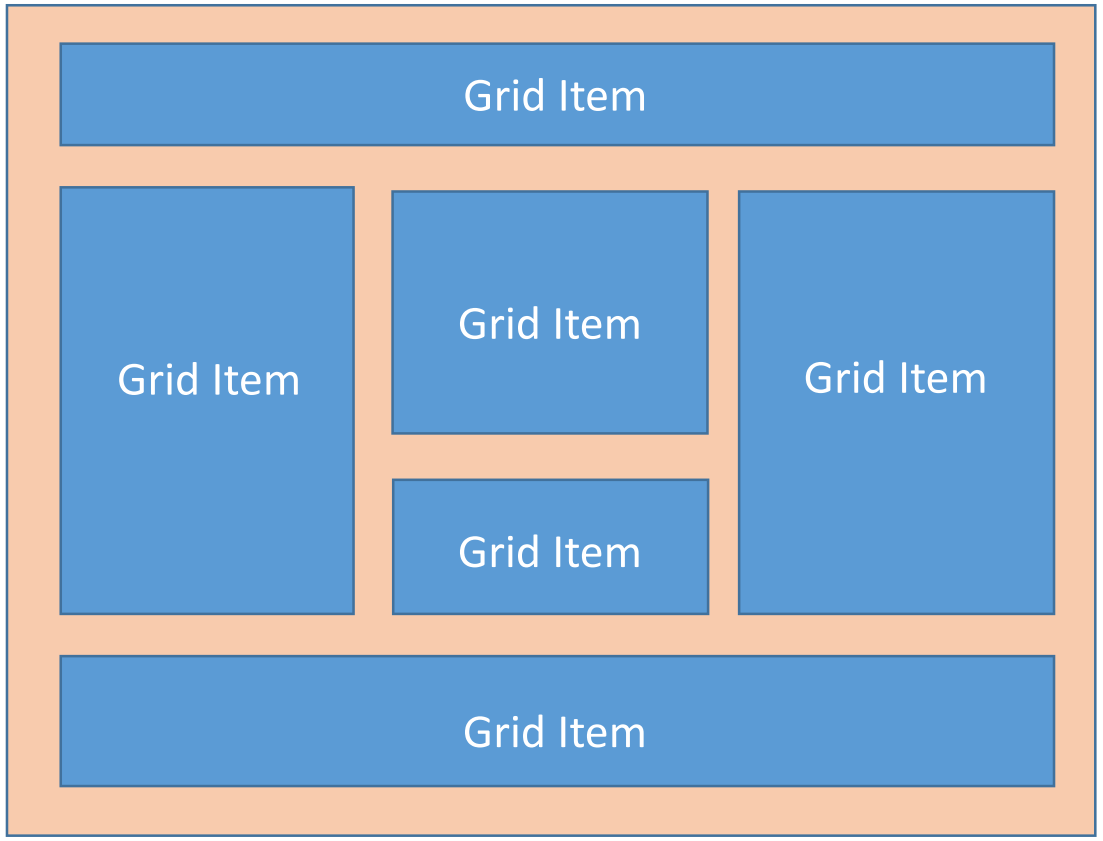
Grid Container

- The element that surrounds the grid items.
- This is the element that you apply "display: grid" to
- Once you apply "display: grid" to an element all its direct children become grid items



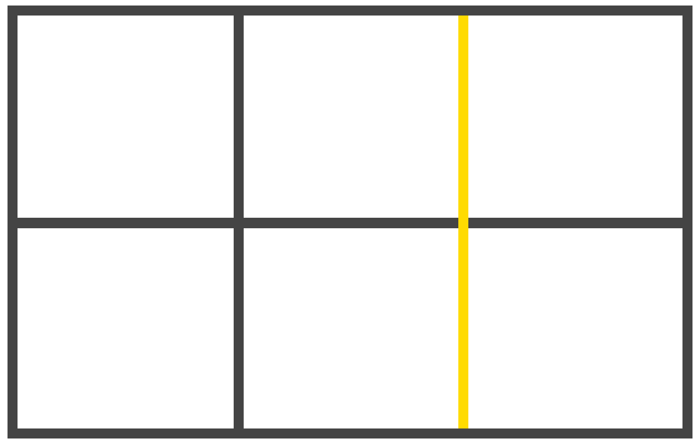
Grid Item

- The HTML elements that are direct children of a parent element that has a display property set to grid



Grid Line

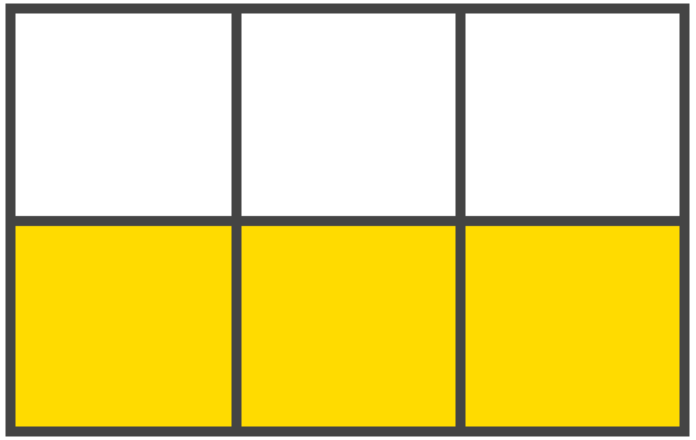
- The dividing lines that make up the structure of the grid. They can be either vertical ("column grid lines") or horizontal ("row grid lines") and reside on either side of a row or column
- Here the yellow line is an example of a column grid line



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Grid Track

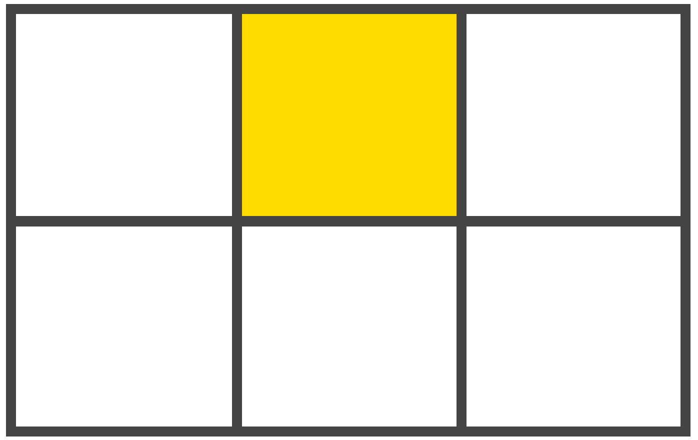
- The space between two adjacent grid lines. You can think of them like the columns or rows of the grid
- Here's the grid track between the second and third row grid lines



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Grid Cell

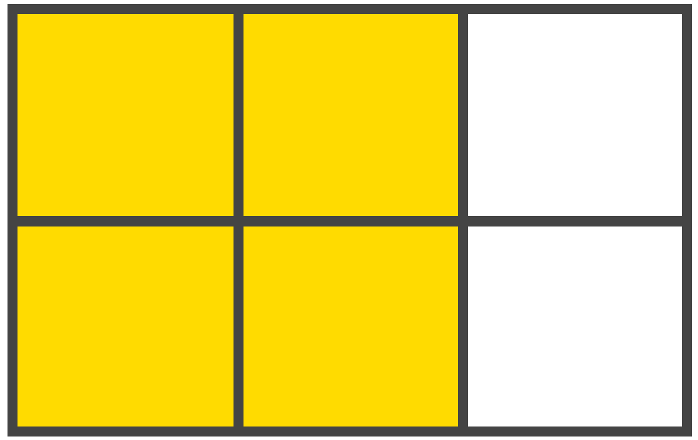
- The space between two adjacent row and two adjacent column grid lines. It's a single "unit" of the grid
- Here's the grid cell between row grid lines 1 and 2, and column grid lines 2 and 3



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

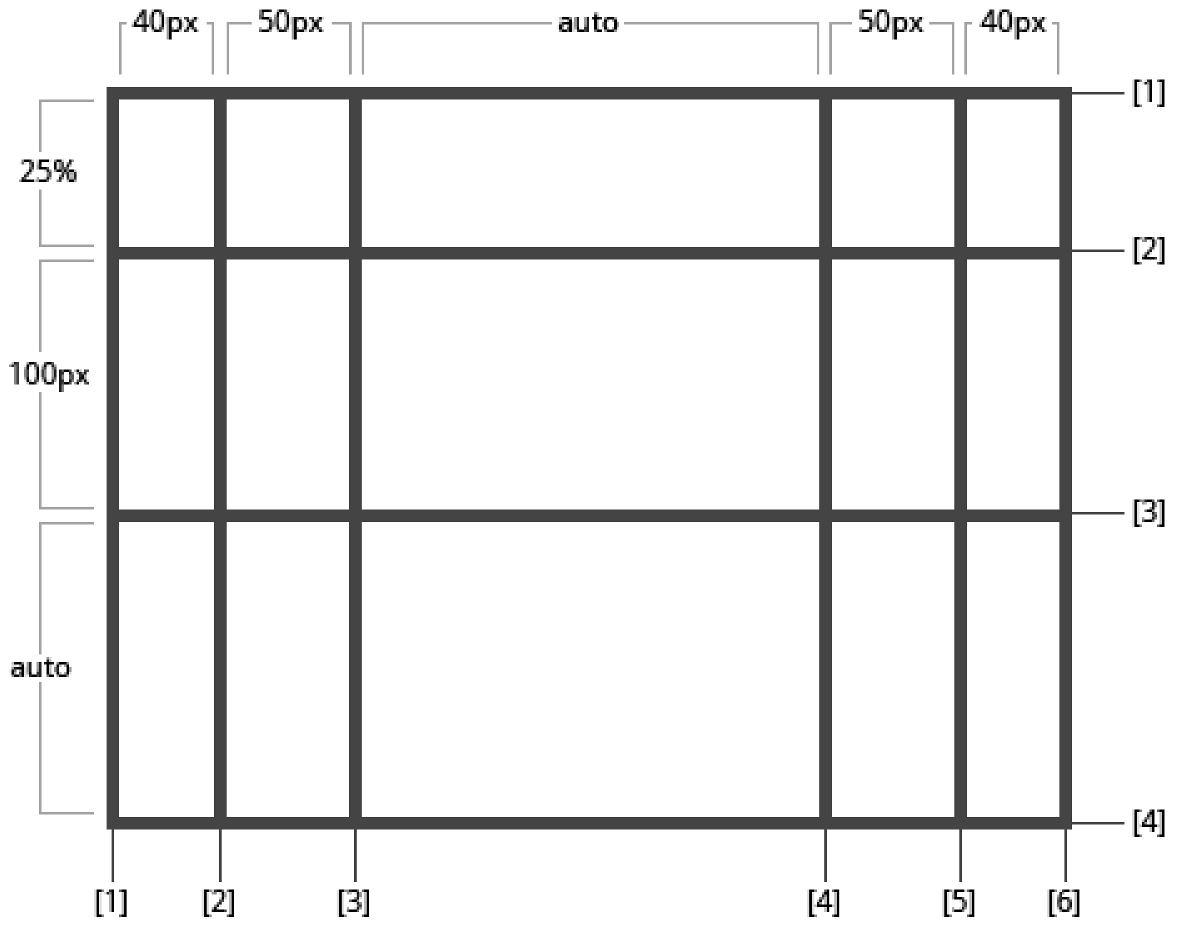
Grid Area

- The total space surrounded by four grid lines. A grid area may be comprised of any number of grid cells
- Here's the grid area between row grid lines 1 and 3, and column grid lines 1 and 3



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Defining the Grid



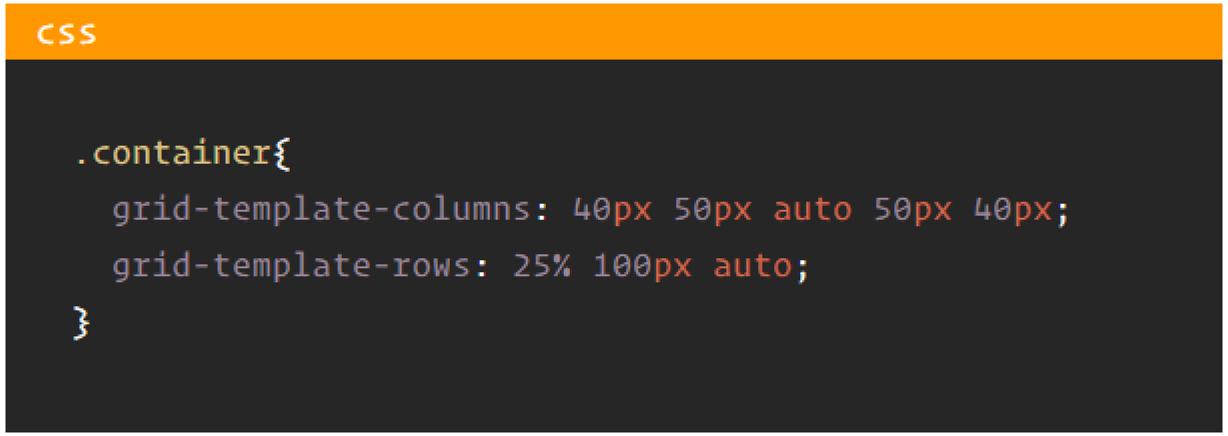
1. Image from: https://css-tricks.com/snippets/css/complete-guide-grid/

Defining the Grid

- CSS Grid will default to the default CSS layout unless you define the grid
- You can layout your grid using the following two CSS properties set on the parent element
  - "grid-template-columns"
  - "grid-template-rows"

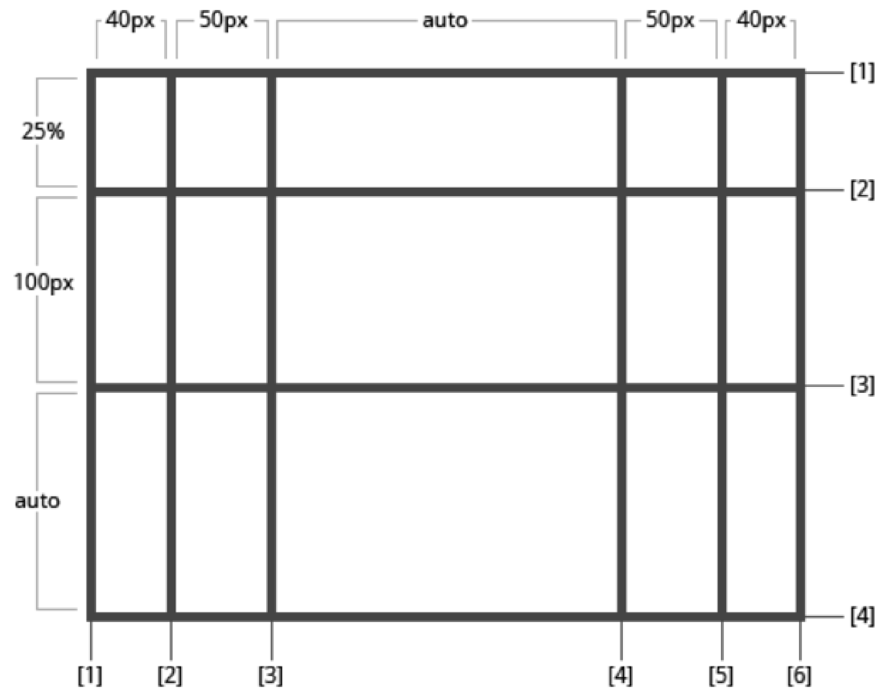
Defining the Grid

- The syntax below lays out a two dimensional grid with columns that are 40px, 50px, [auto-width], 50px and 40px wide. The rows have heights of 25%, 100px and [auto-height]



css

```
.container{
  grid-template-columns: 40px 50px auto 50px 40px;
  grid-template-rows: 25% 100px auto;
}
```



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Defining the Grid – repeating items

- If your grid contains several grid items of equivalent size than you can use the repeat() function instead of writing them out individually

css

```css
.container {
  grid-template-columns: repeat(3, 60px) ;
}
```

Which is equivalent to:

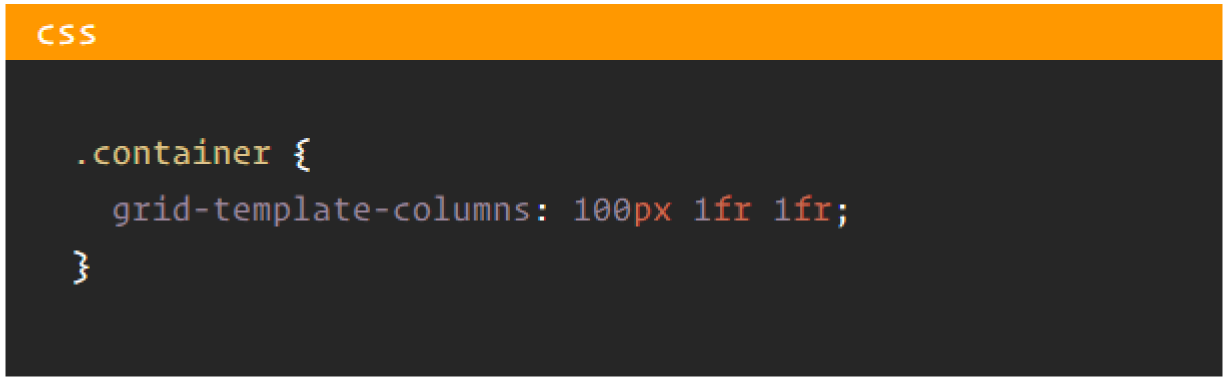
css

```css
.container {
  grid-template-columns: 60px 60px 60px ;
}
```

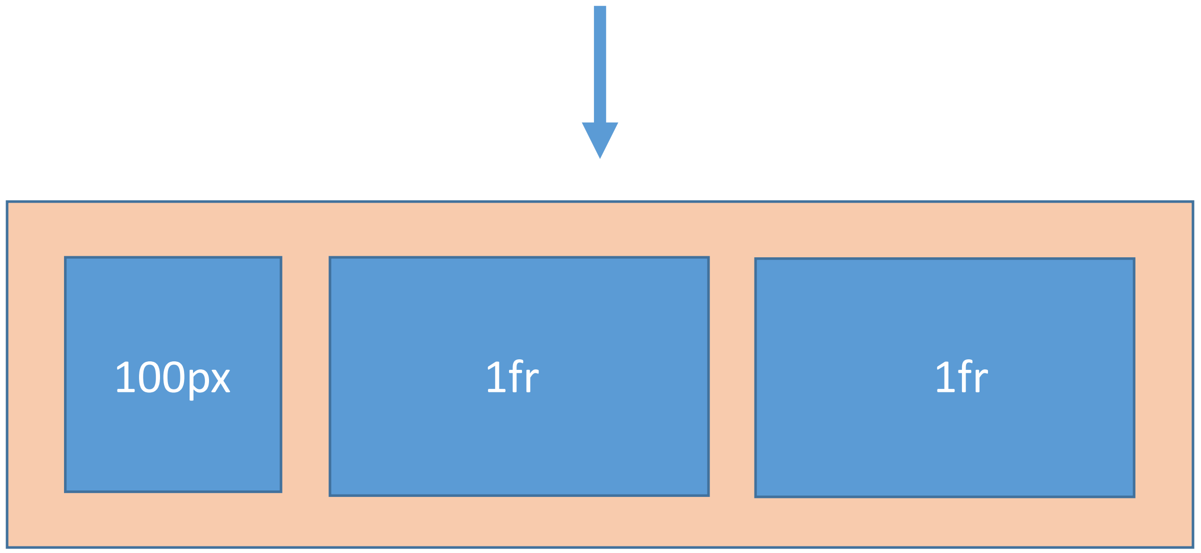
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

fr Units

- The fr unit allows you to set the size of a track as a fraction of the free space of the grid container
- For example, this will set each item to one half the width of the remaining space in the grid container (100px – (remaining space / 2))



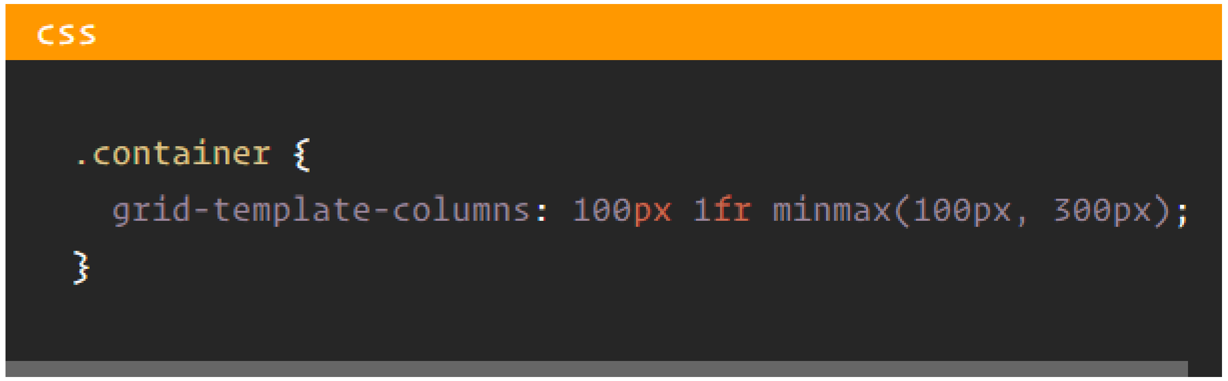
```css
.container {
  grid-template-columns: 100px 1fr 1fr;
}
```



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

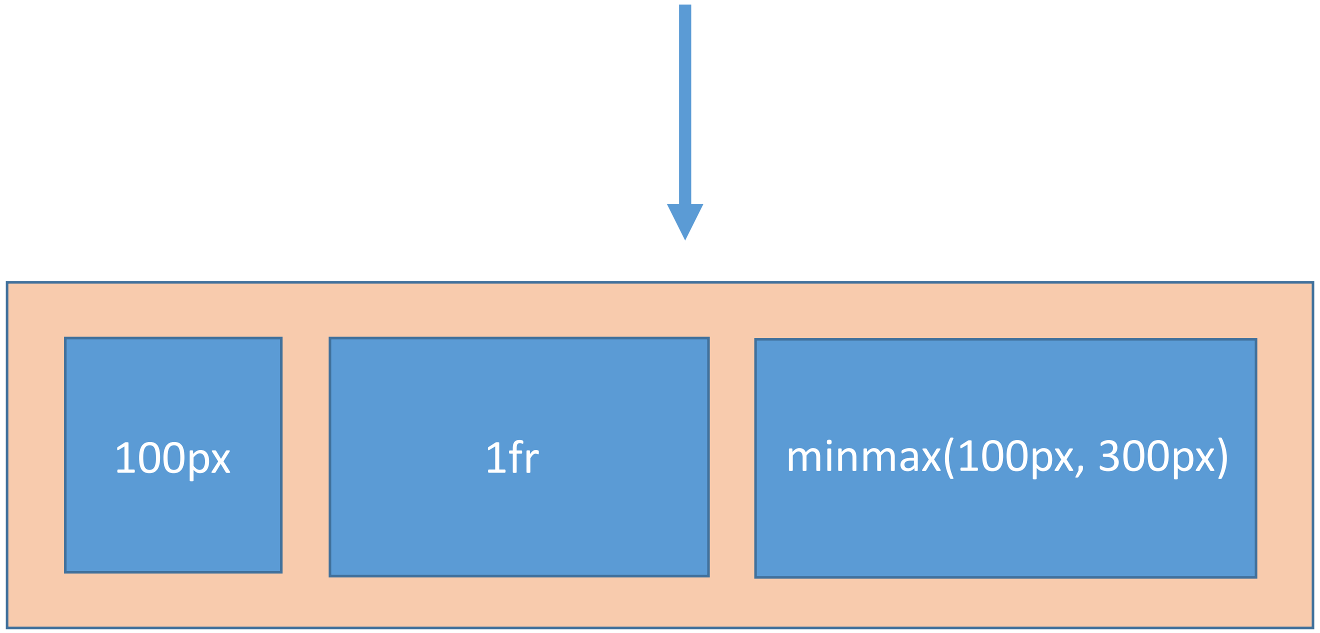
minmax() value

- The minmax() method allows you set width or height of a grid track that can vary between a minimum value and a maximum value



css

```
.container {
  grid-template-columns: 100px 1fr minmax(100px, 300px);
}
```

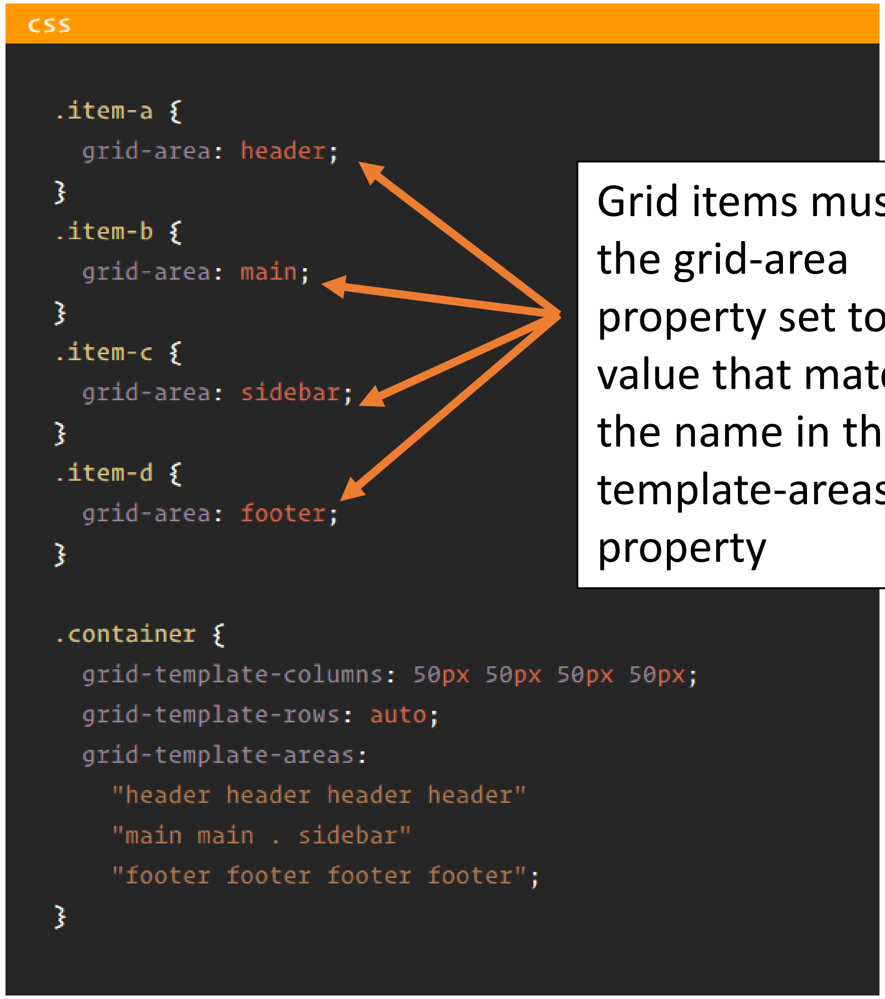


Grid Template Areas

- Defines a grid template by referencing the names of the grid areas which are specified with the grid-area property
- Repeating the name of a grid area causes the content to span those cells
- A period signifies an empty cell
- The syntax itself provides a visualization of the structure of the grid

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Grid Template Areas



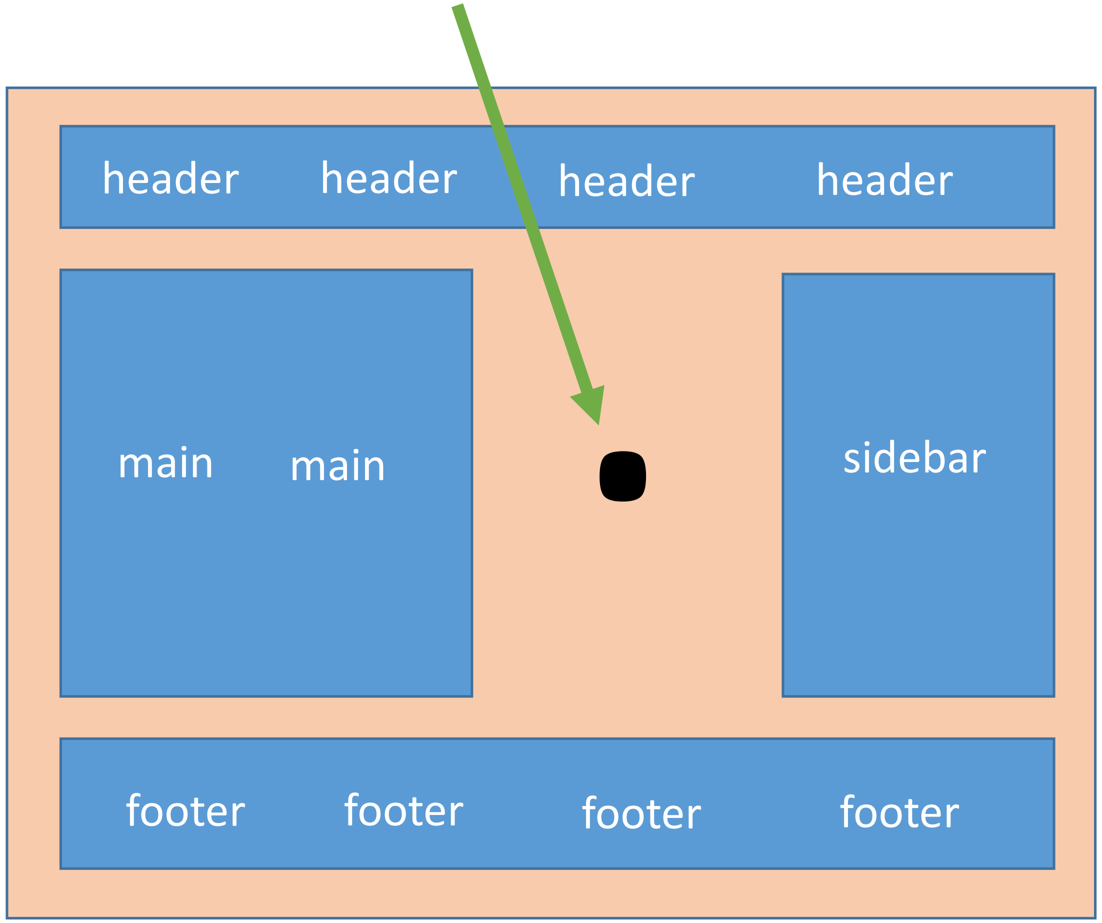
css

```css
.item-a {
  grid-area: header;
}
.item-b {
  grid-area: main;
}
.item-c {
  grid-area: sidebar;
}
.item-d {
  grid-area: footer;
}

.container {
  grid-template-columns: 50px 50px 50px 50px;
  grid-template-rows: auto;
  grid-template-areas:
    "header header header header"
    "main main . sidebar"
    "footer footer footer footer";
}
```

Grid items must have the grid-area property set to a value that matches the name in the grid-template-areas property

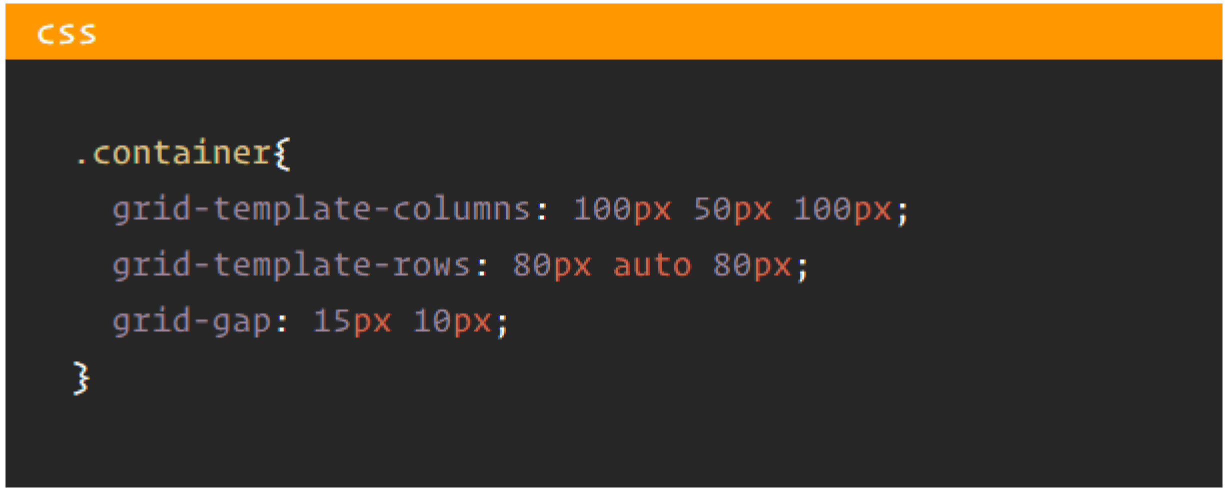
Notice the dot. It represents an empty cell or placeholder



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

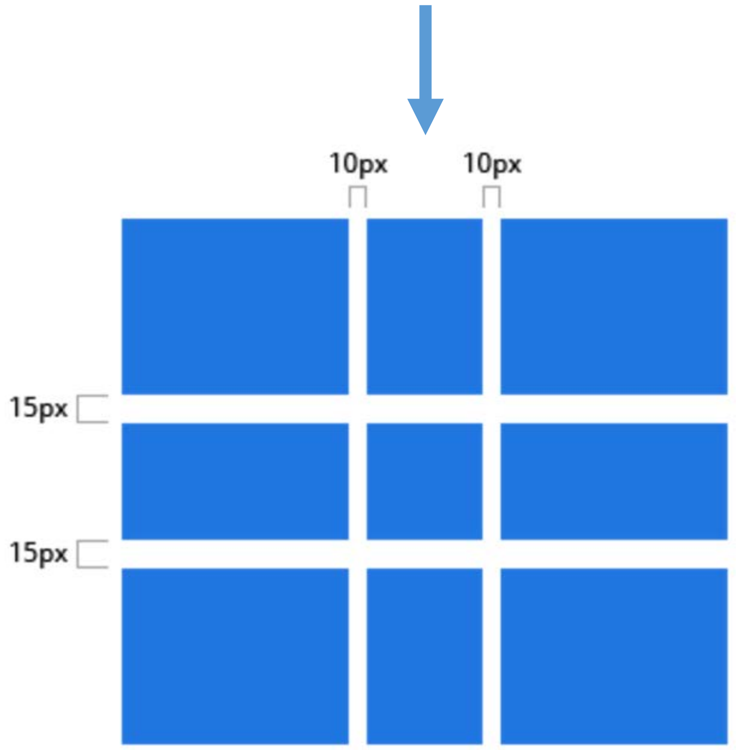
Grid Gap

- Specifies the size of the grid lines
- You can think of it like setting the width of the gutters between the columns/rows
- The first value sets the row gap, the second value sets the column gap



css

```css
.container{
  grid-template-columns: 100px 50px 100px;
  grid-template-rows: 80px auto 80px;
  grid-gap: 15px 10px;
}
```



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Box Alignment Properties

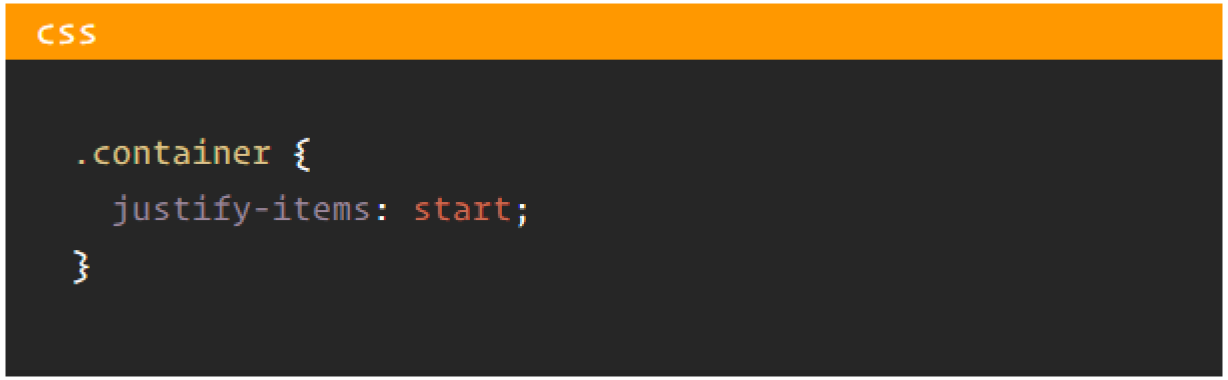
- Grid uses similar box alignment properties as flex-box
- The following are box alignment properties used by grid
  - Set on the grid container element
    - justify-items
    - justify-content
    - align-items
    - align-content
  - Set on a grid item
    - justify-self
    - align-self

Justify Items

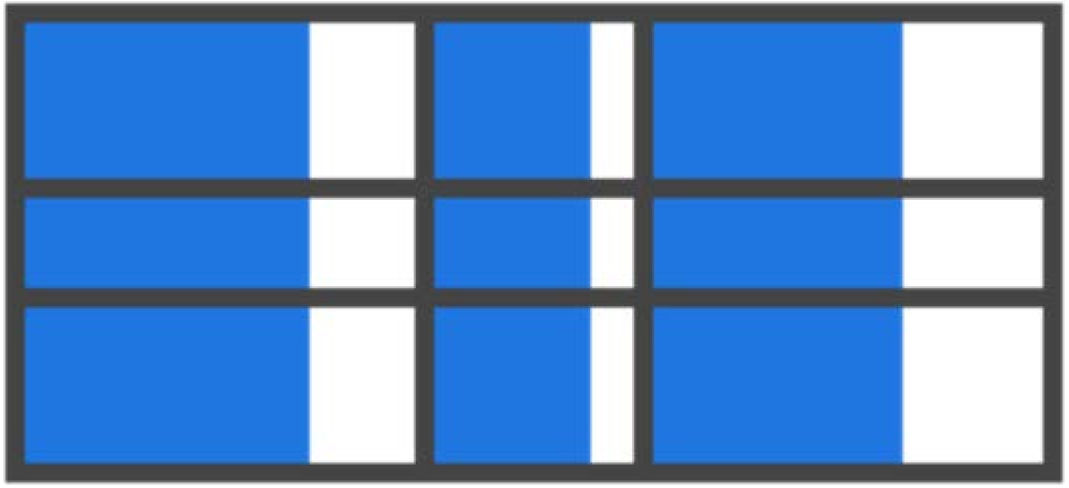
- Aligns the content inside a grid item along the row axis (as opposed to align-items which aligns along the column axis). This value applies to all grid items inside the container
- Values:
  - **start** - aligns the content to the left end of the grid area
  - **end** - aligns the content to the right end of the grid area
  - **center** - aligns the content in the center of the grid area
  - **stretch** - fills the whole width of the grid area (this is the default)

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Items - start

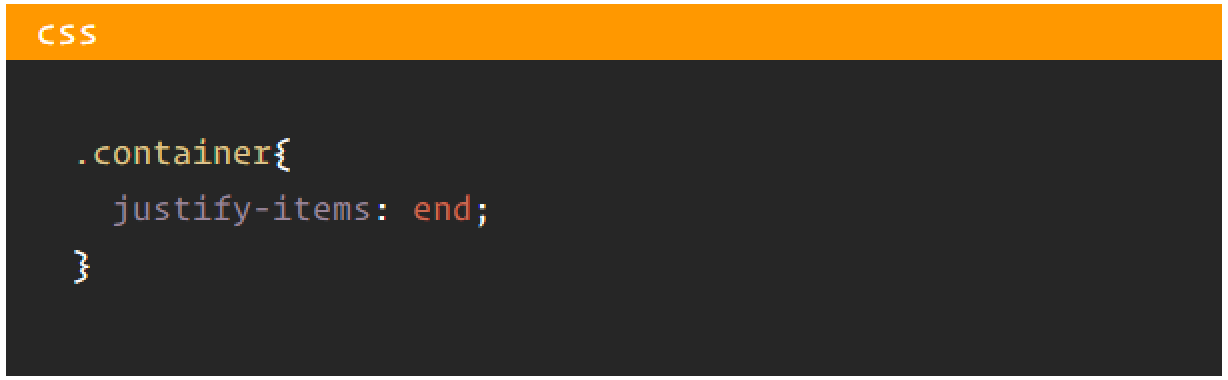


```
css
.container {
  justify-items: start;
}
```

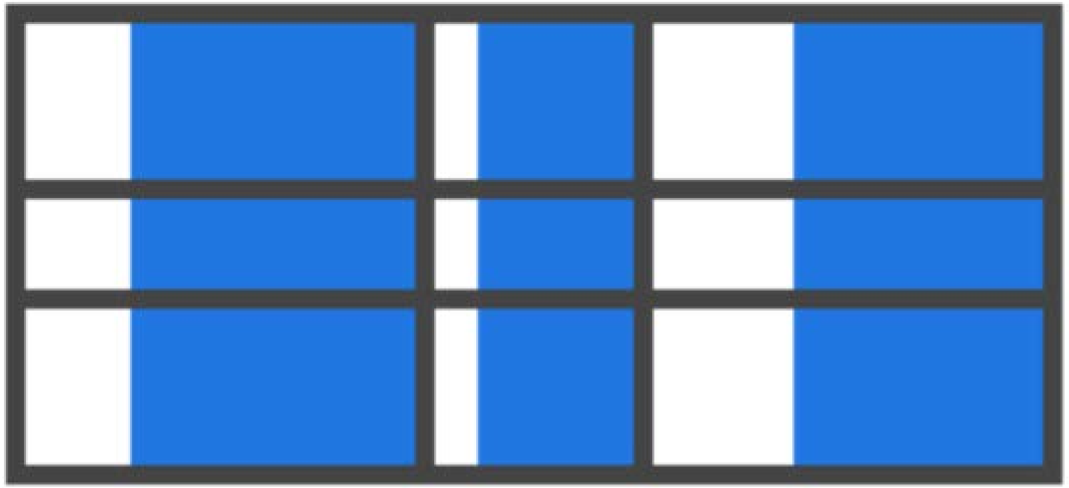


1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Items - end

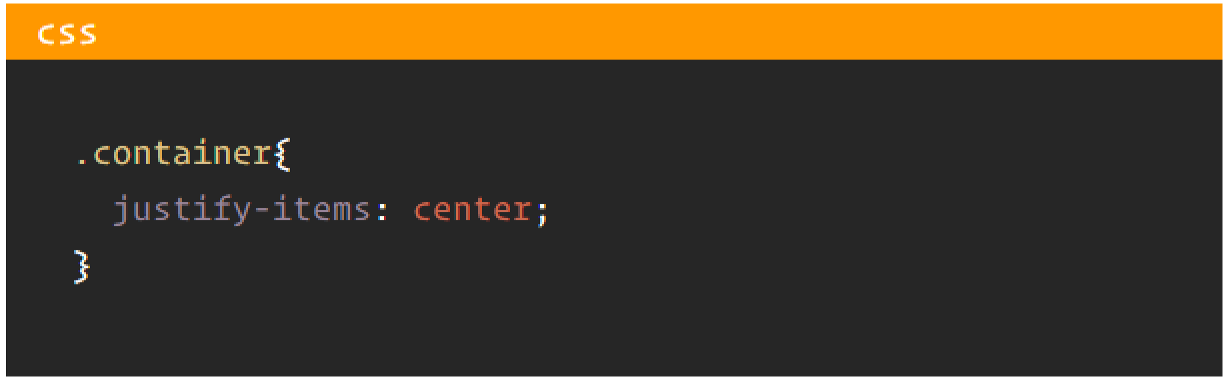


```css
.container{
  justify-items: end;
}
```

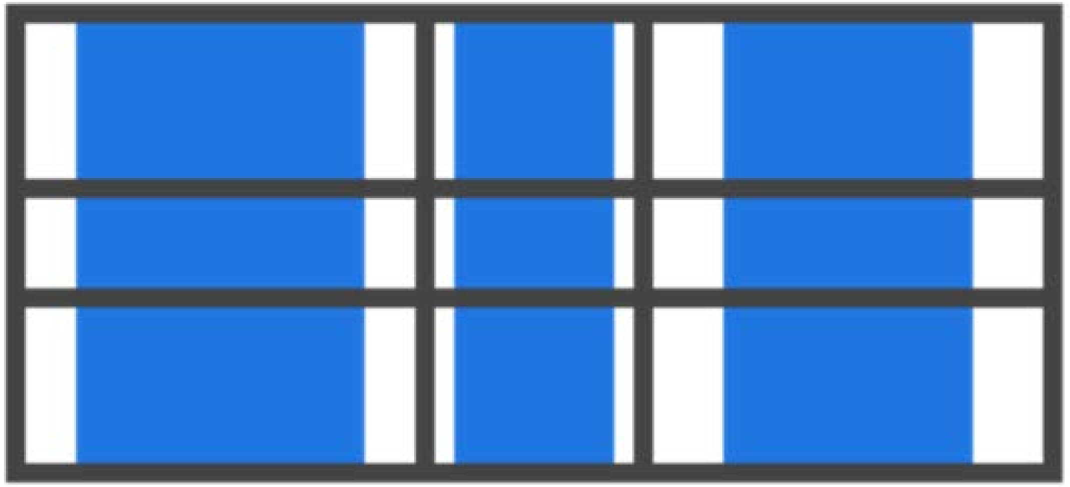


1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Items - center

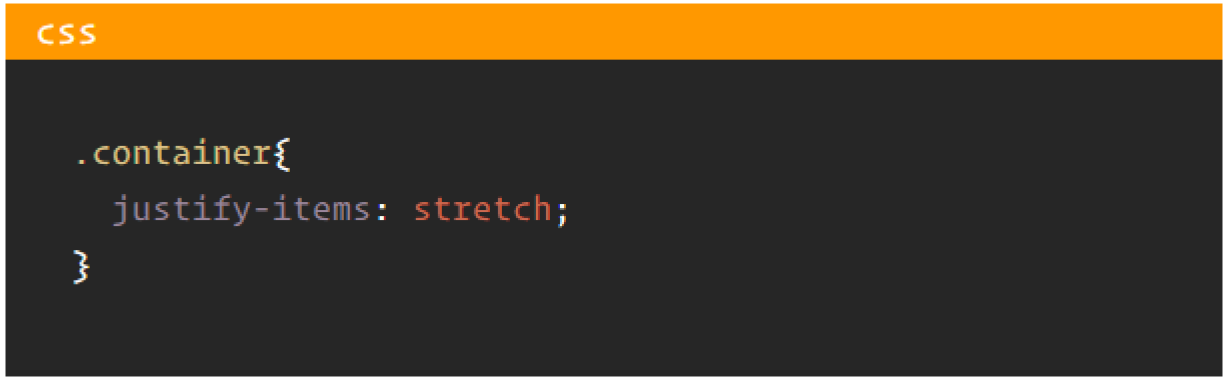


```css
.container{
  justify-items: center;
}
```



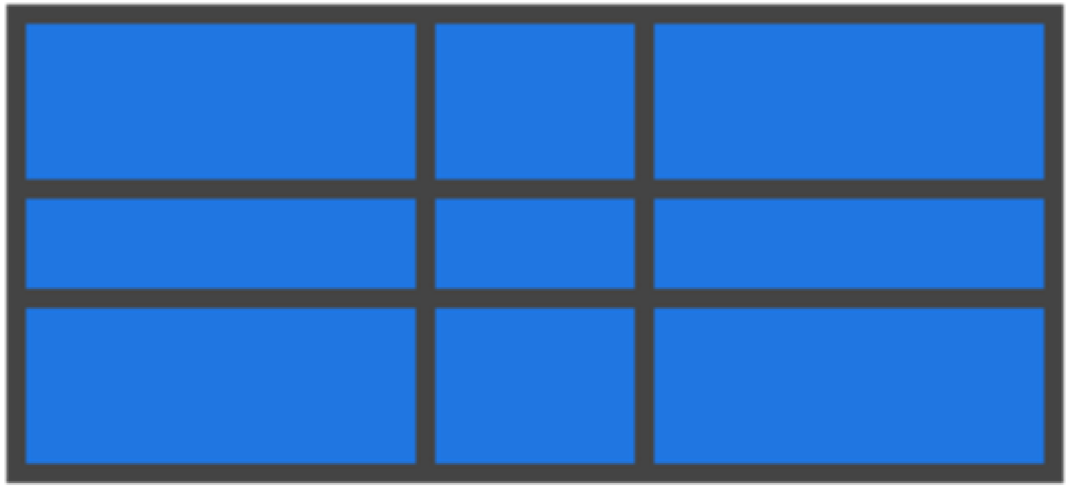
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Items - stretch



css

```css
.container{
  justify-items: stretch;
}
```



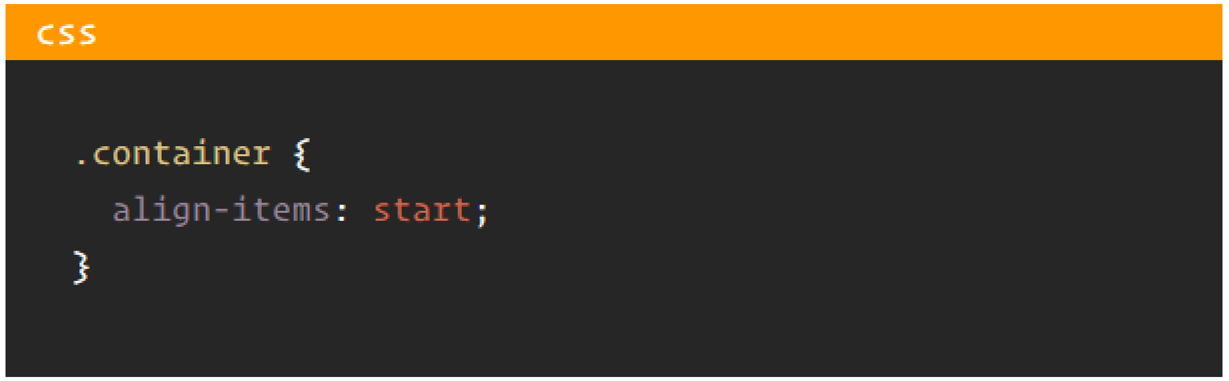
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Items

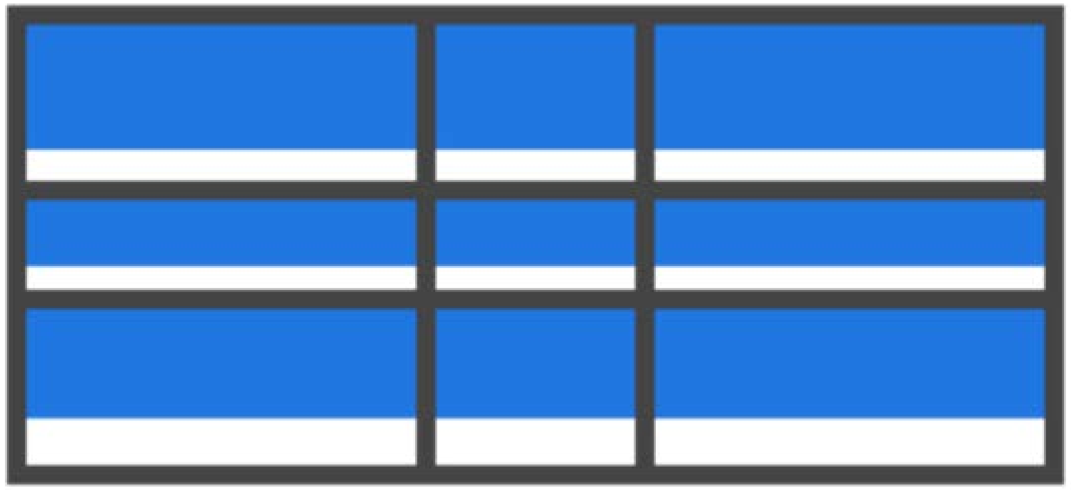
- Aligns the content inside a grid item along the column axis (as opposed to justify-items which aligns along the row axis). This value applies to all grid items inside the container
- Values:
  - **start** - aligns the content to the top of the grid area
  - **end** - aligns the content to the bottom of the grid area
  - **center** - aligns the content in the center of the grid area
  - **stretch** - fills the whole height of the grid area (this is the default)

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Items - start

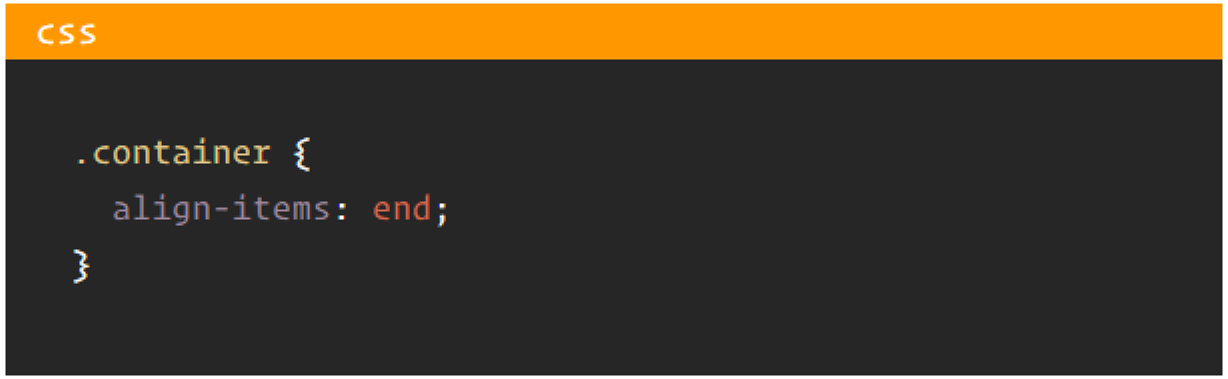


```css
.container {
  align-items: start;
}
```



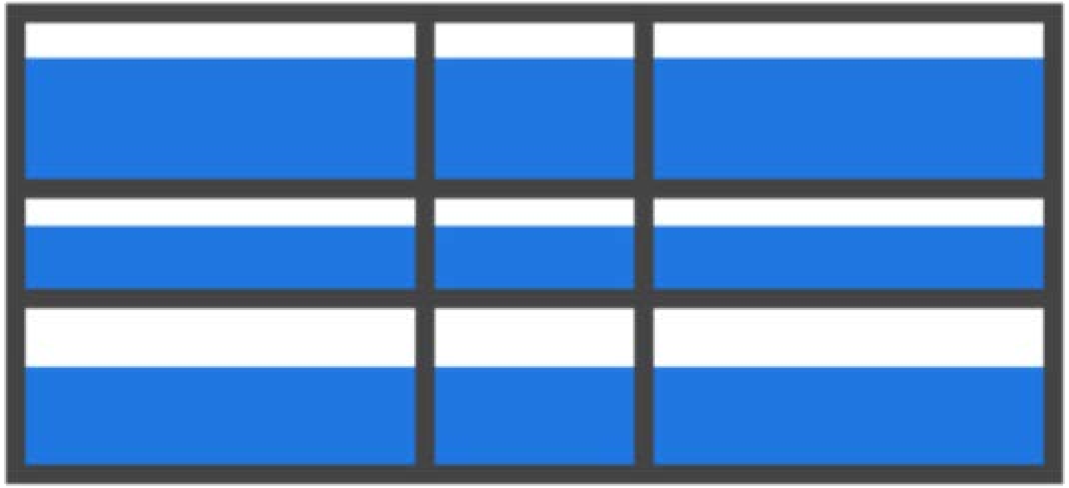
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Items - end



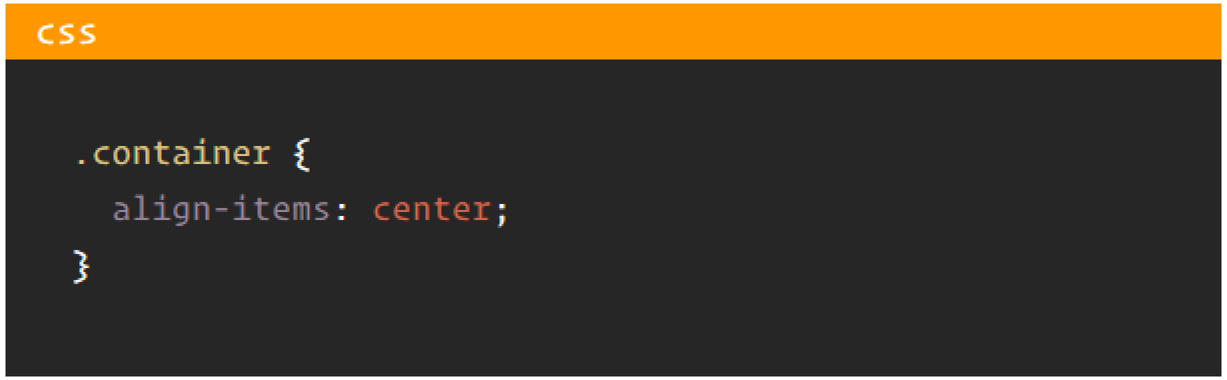
css

```
.container {
  align-items: end;
}
```



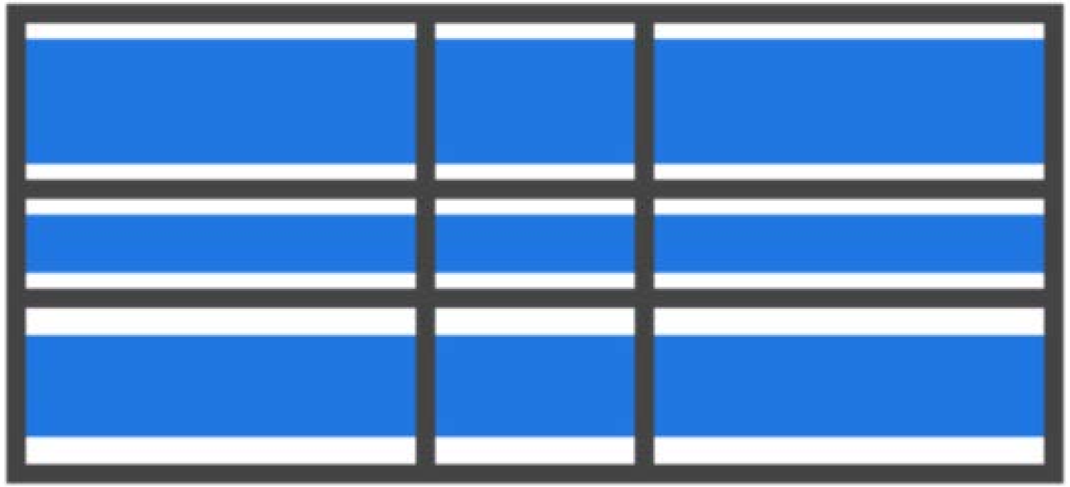
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Items - center



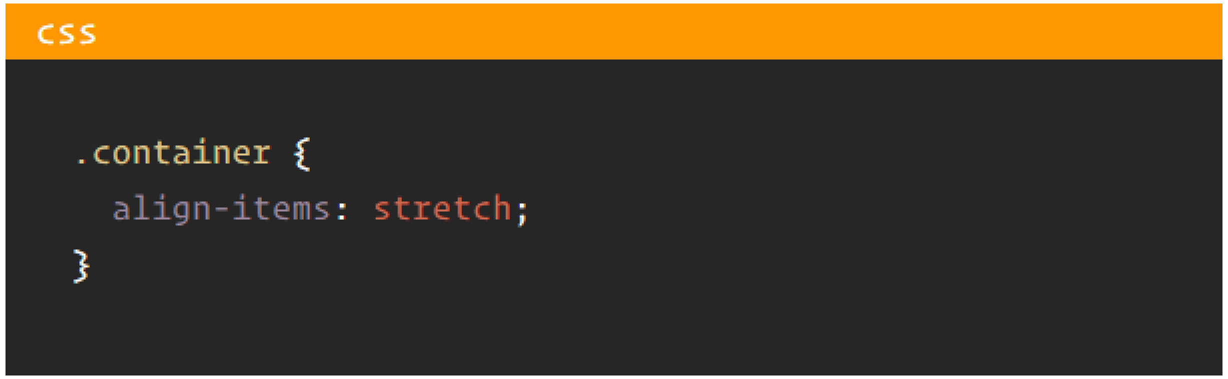
css

```
.container {
  align-items: center;
}
```

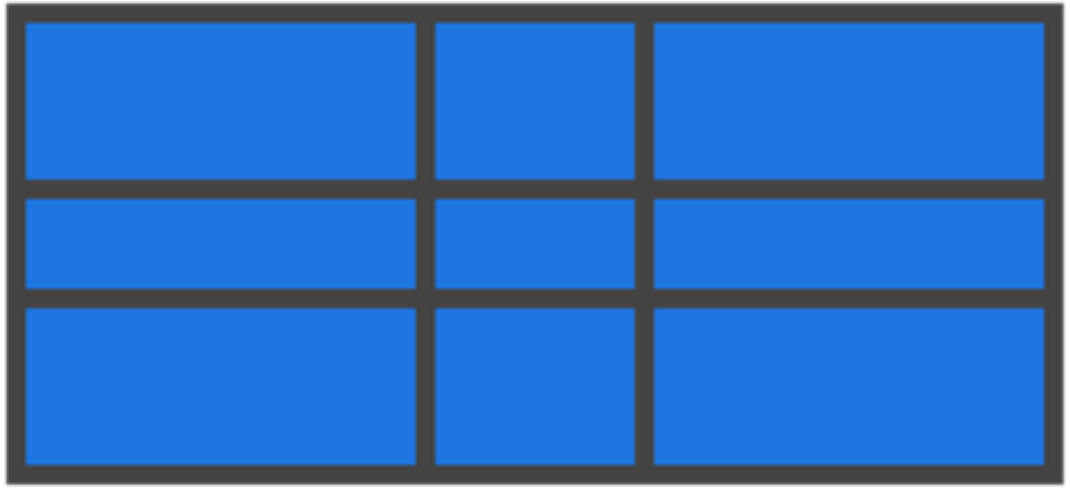


1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Items - stretch



```css
.container {
  align-items: stretch;
}
```



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

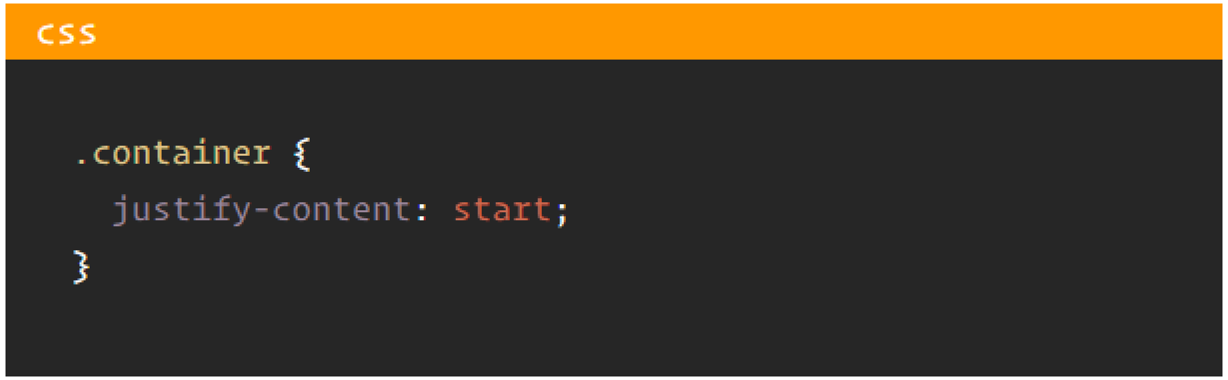
Justify Content

- Sometimes the total size of your grid might be less than the size of its grid container. This could happen if all of your grid items are sized with non-flexible units like px. In this case you can set the alignment of the grid within the grid container
- This property aligns the grid along the row axis (as opposed to align-content which aligns the grid along the column axis)

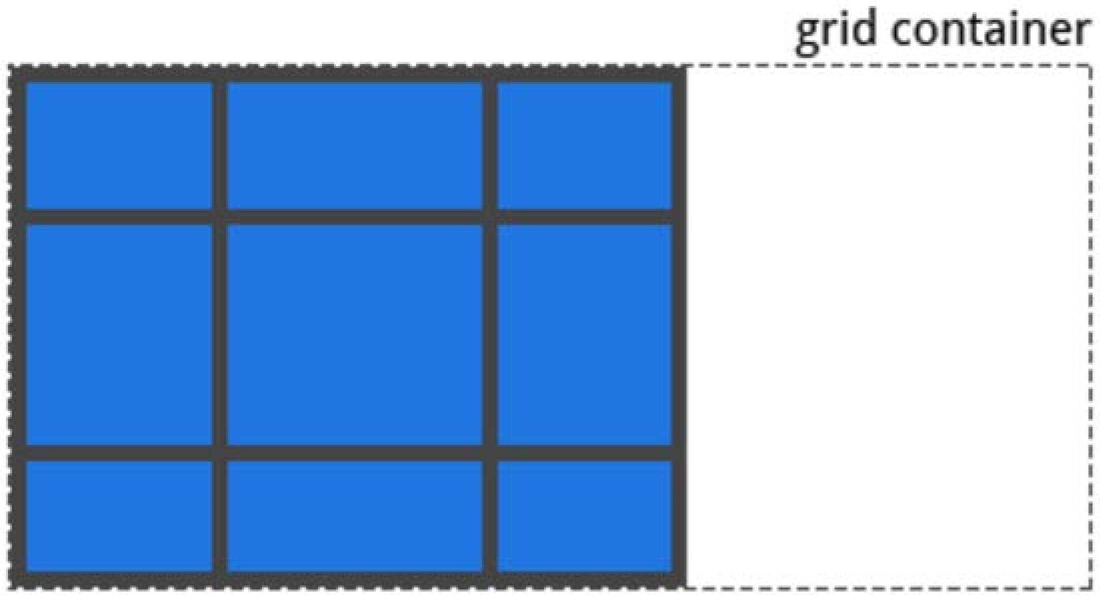
- Values:
  - **start** - aligns the grid to the left end of the grid container
  - **end** - aligns the grid to the right end of the grid container
  - **center** - aligns the grid in the center of the grid container
  - **stretch** - resizes the grid items to allow the grid to fill the full width of the grid container
  - **space-around** - places an even amount of space between each grid item, with half-sized spaces on the far ends
  - **space-between** - places an even amount of space between each grid item, with no space at the far ends
  - **space-evenly** - places an even amount of space between each grid item, including the far ends

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Content - start

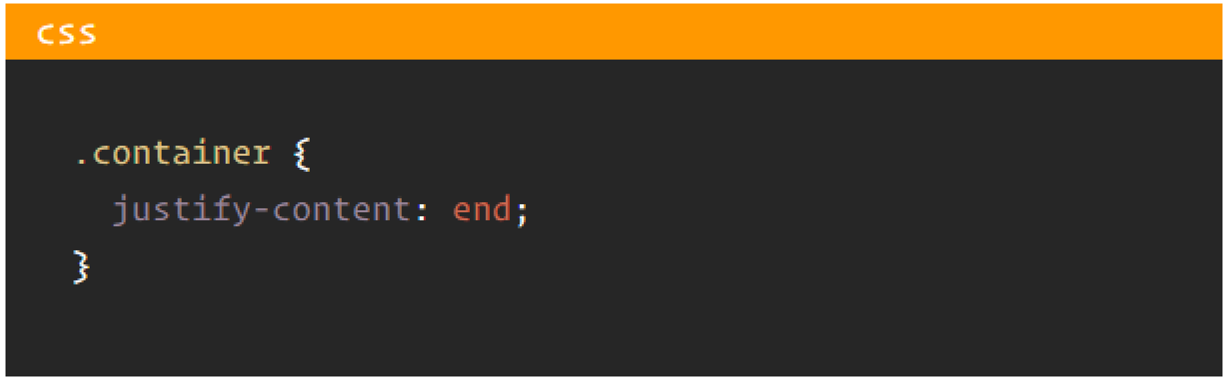

css

```css
.container {
  justify-content: start;
}
```



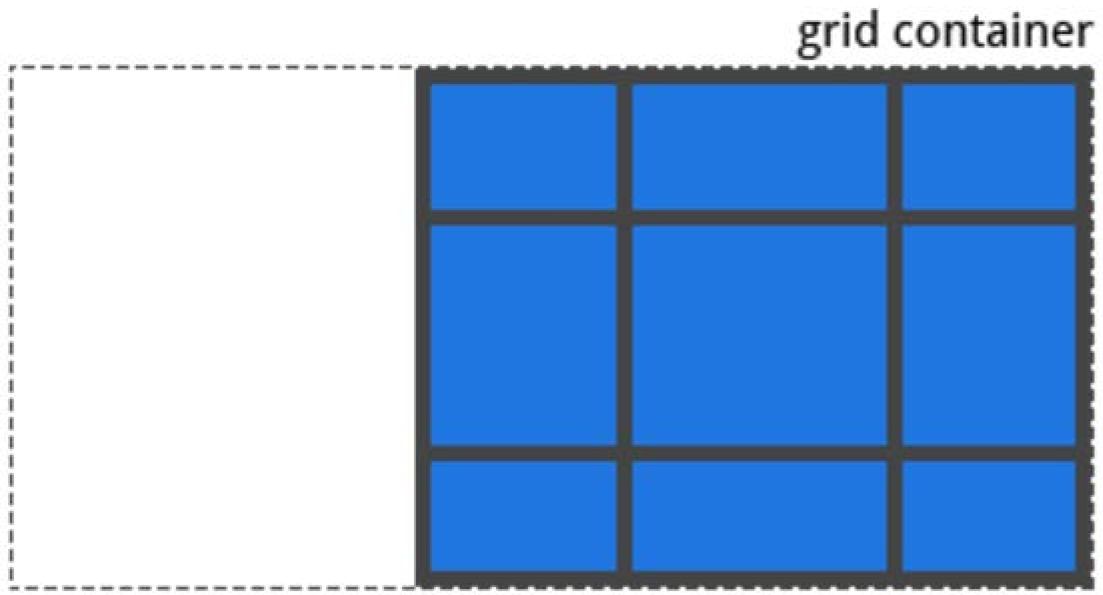
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Content - end



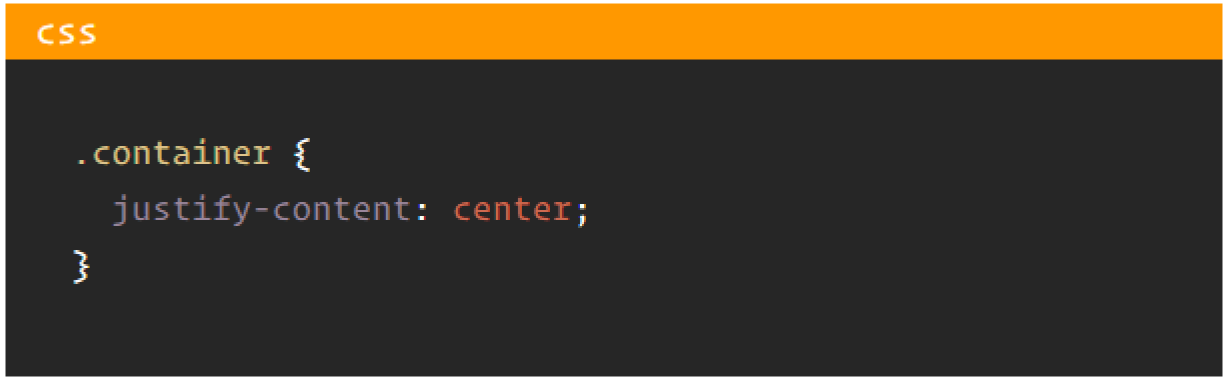
css

```
.container {
  justify-content: end;
}
```

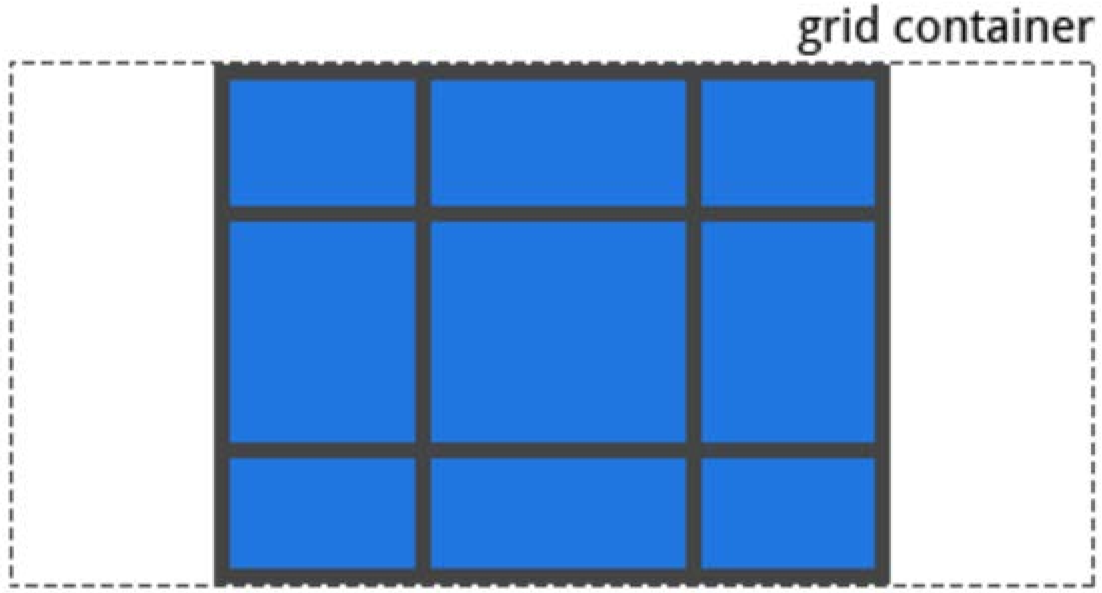


1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Content - center

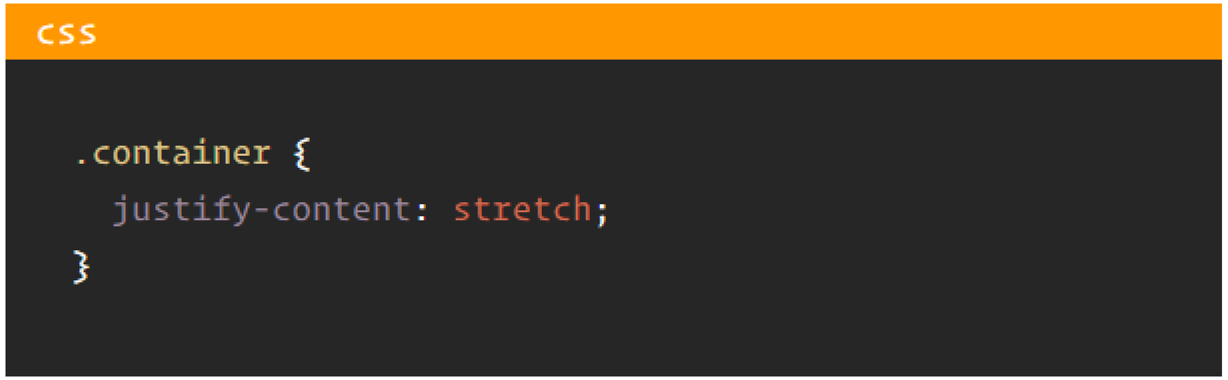


```css
.container {
  justify-content: center;
}
```



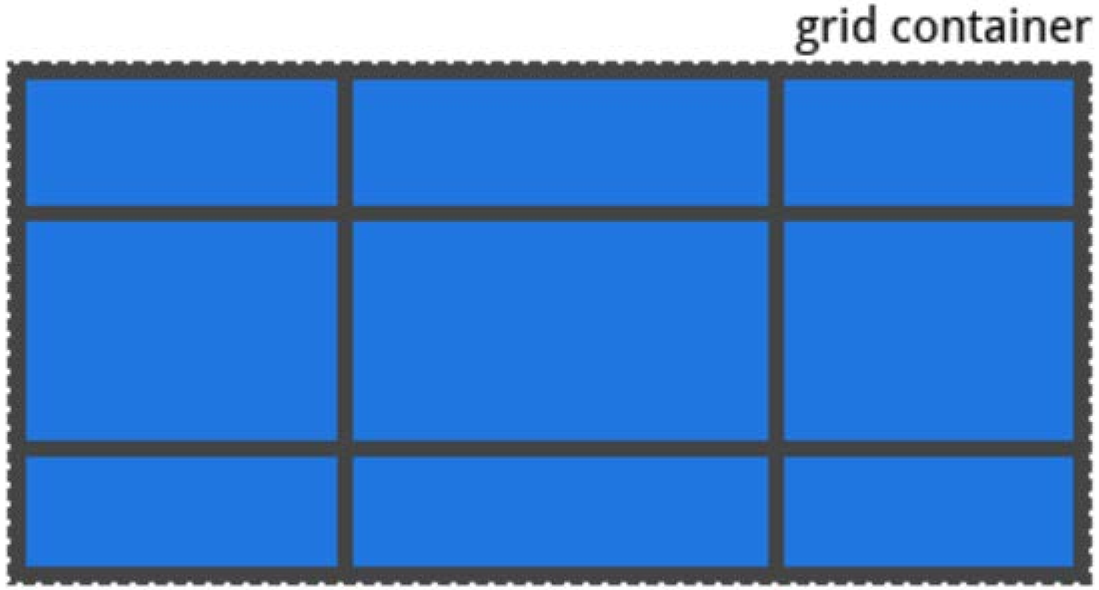
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Content - stretch



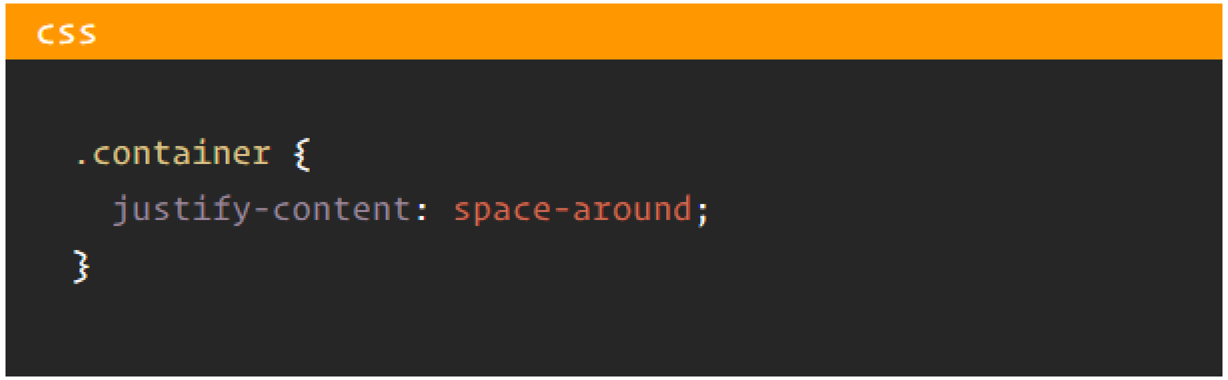
css

```css
.container {
  justify-content: stretch;
}
```

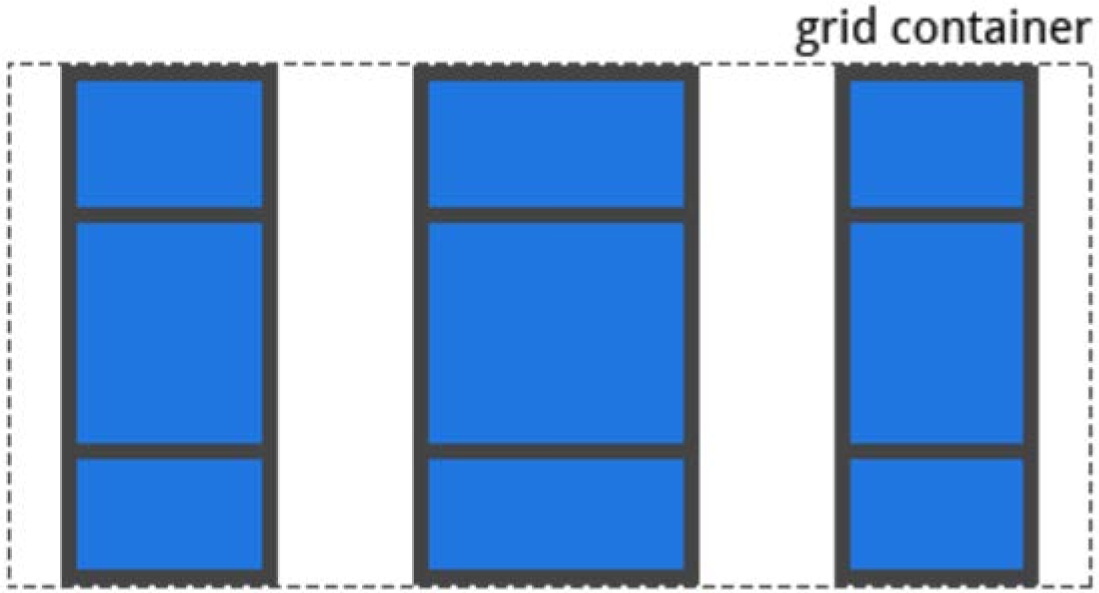


1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Content – space-around

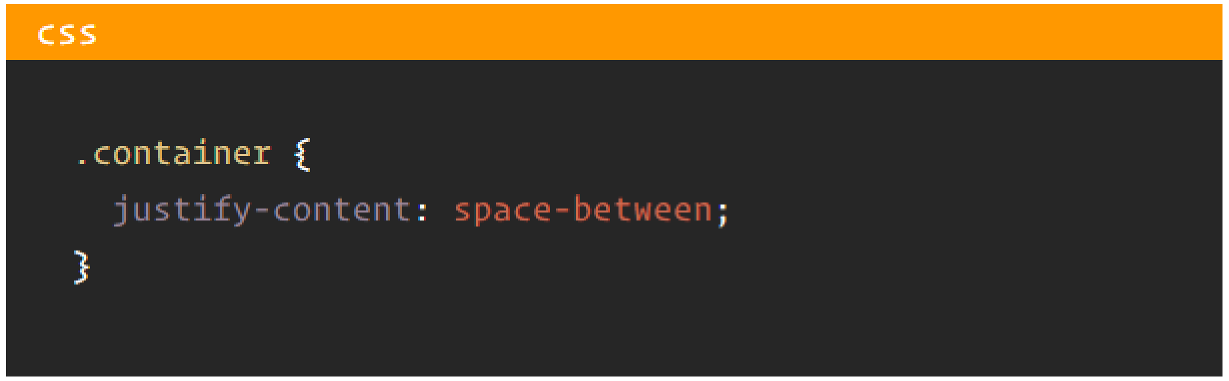


```css
.container {
  justify-content: space-around;
}
```



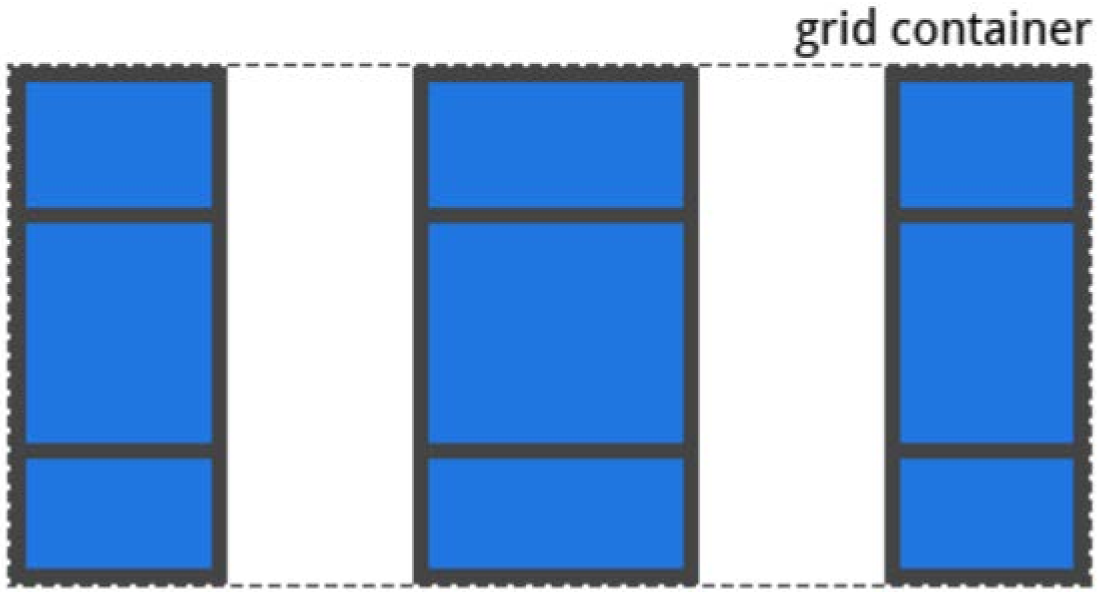
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Content – space-between



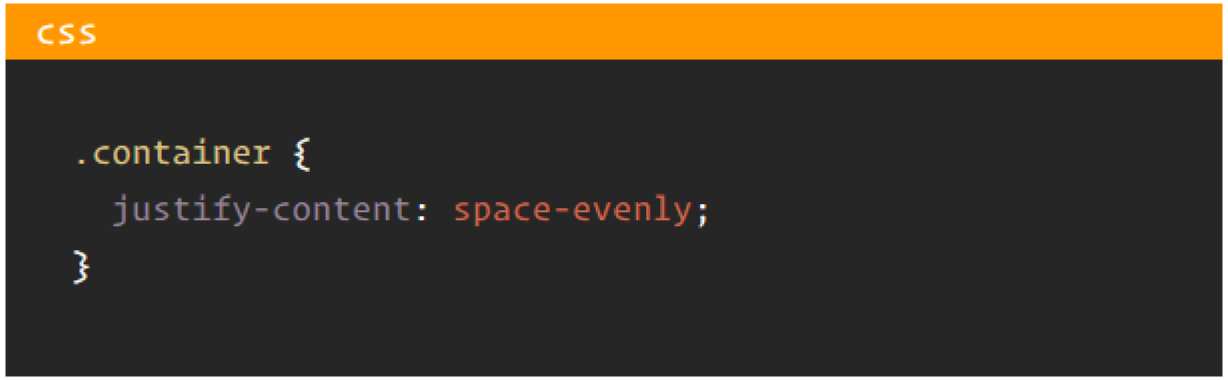
css

```css
.container {
  justify-content: space-between;
}
```

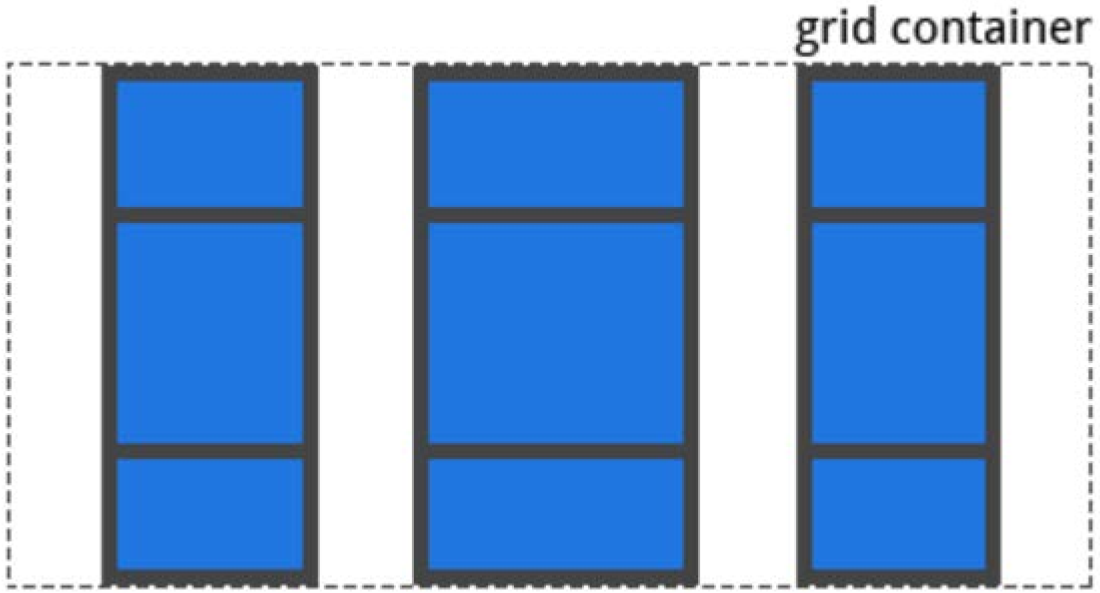


1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Justify Content – space-evenly



```css
.container {
  justify-content: space-evenly;
}
```



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Content

- Sometimes the total size of your grid might be less than the size of its grid container. This could happen if all of your grid items are sized with non-flexible units like px. In this case you can set the alignment of the grid within the grid container
- This property aligns the grid along the column axis (as opposed to justify-content which aligns the grid along the row axis)

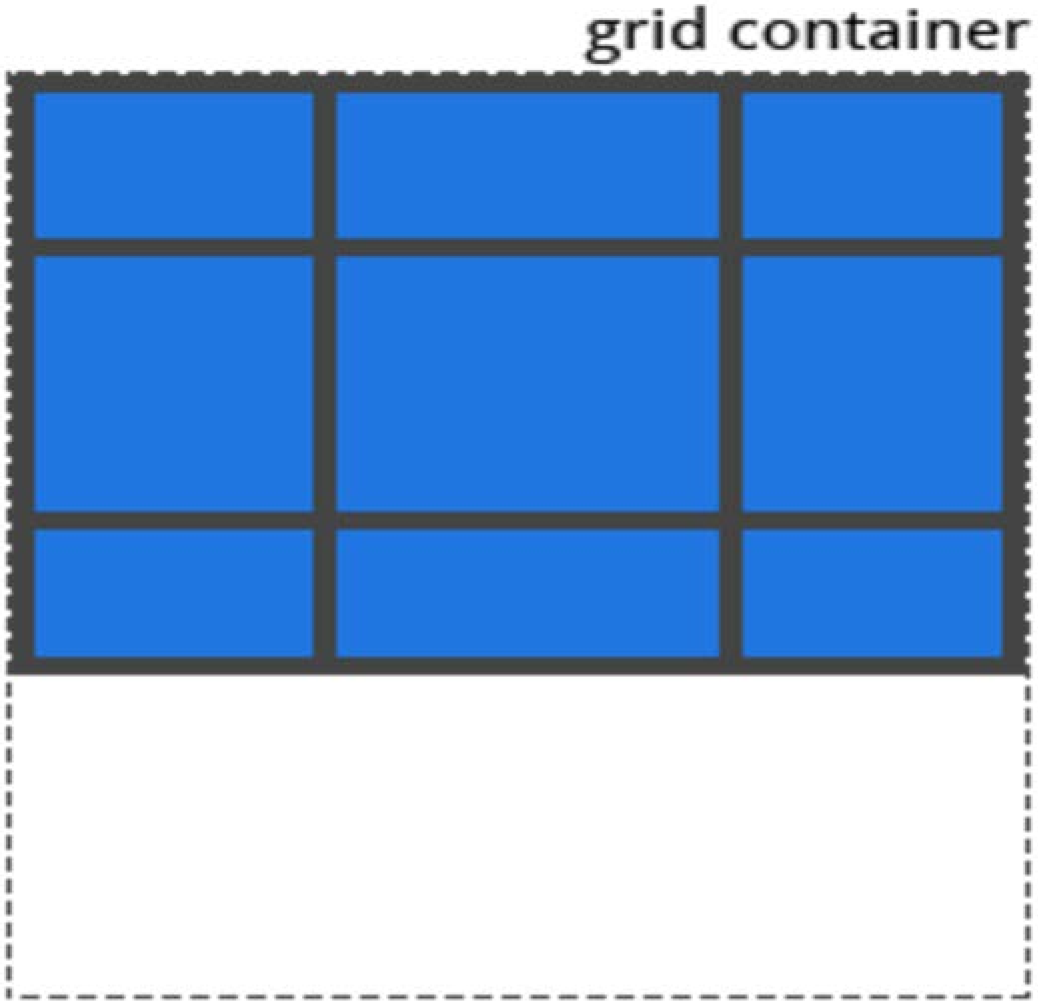
- Values:
  - **start** - aligns the grid to the top of the grid container
  - **end** - aligns the grid to the bottom of the grid container
  - **center** - aligns the grid in the center of the grid container
  - **stretch** - resizes the grid items to allow the grid to fill the full height of the grid container
  - **space-around** - places an even amount of space between each grid item, with half-sized spaces on the far ends
  - **space-between** - places an even amount of space between each grid item, with no space at the far ends
  - **space-evenly** - places an even amount of space between each grid item, including the far ends

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Content - start

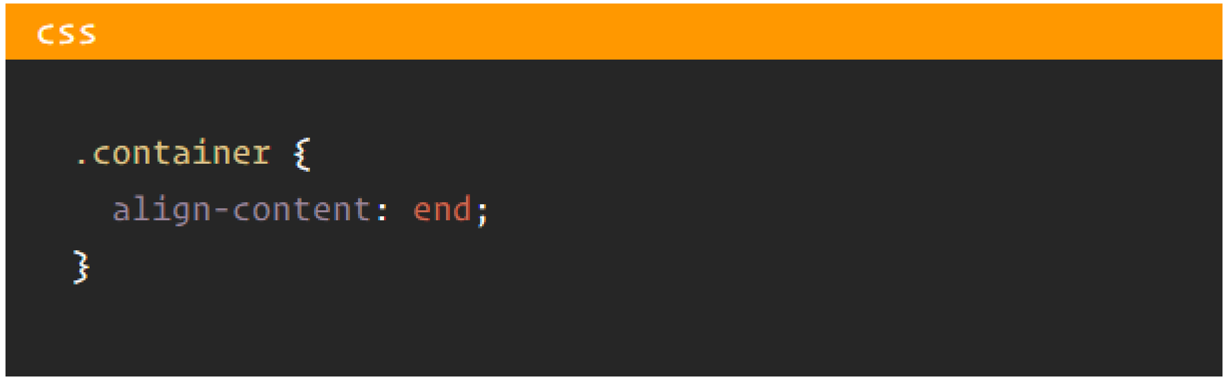
css

```css
.container {
  align-content: start;
}
```



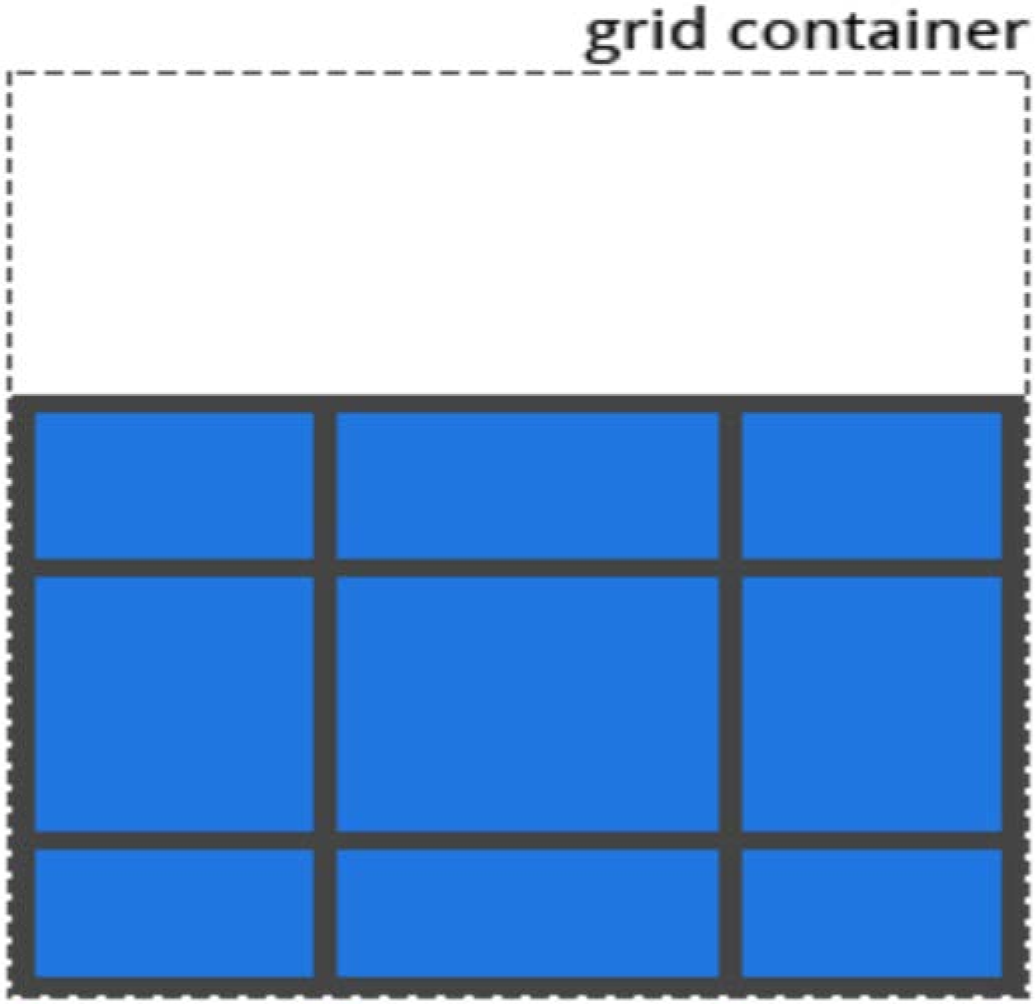
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Content - end



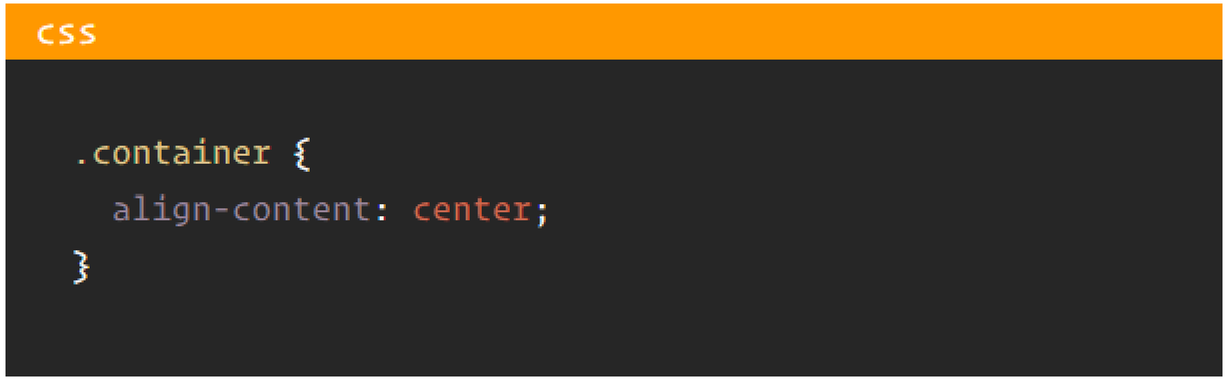
css

```
.container {
  align-content: end;
}
```

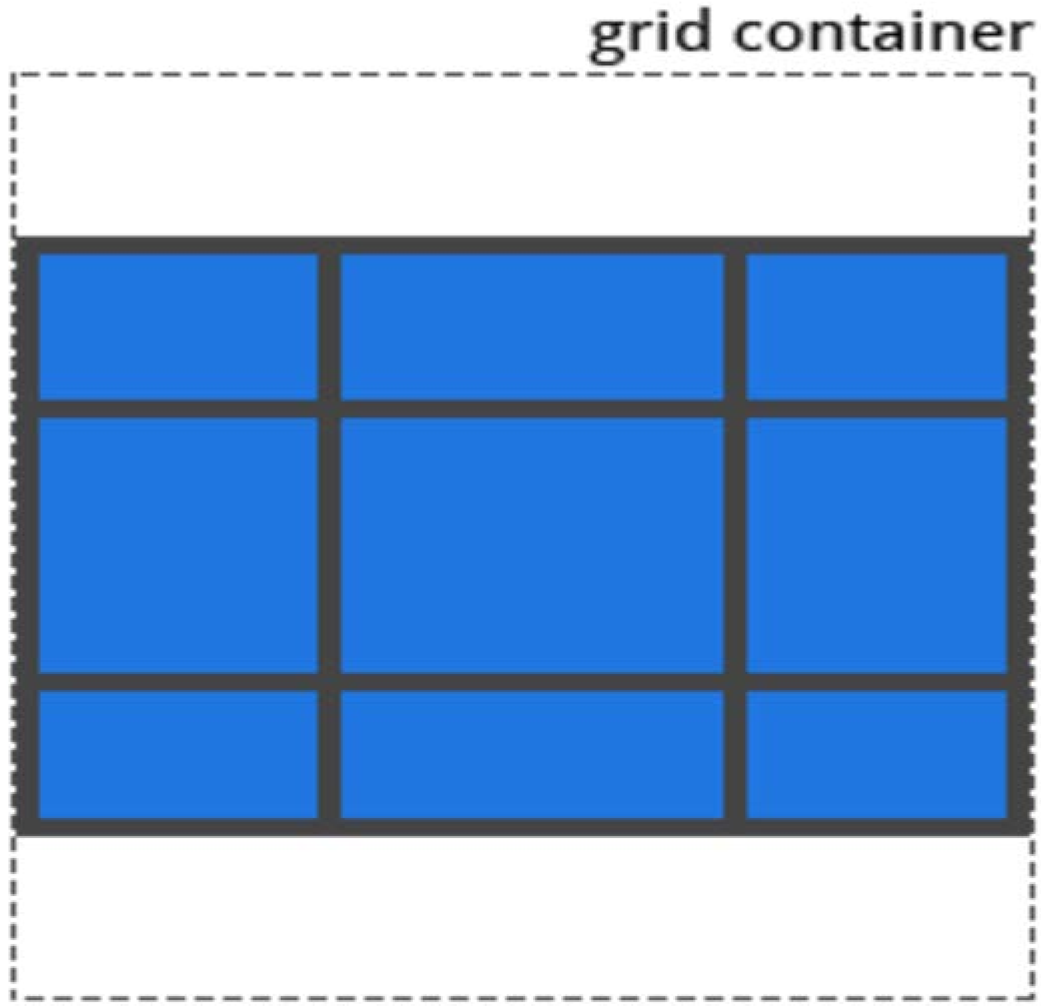


1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Content - center

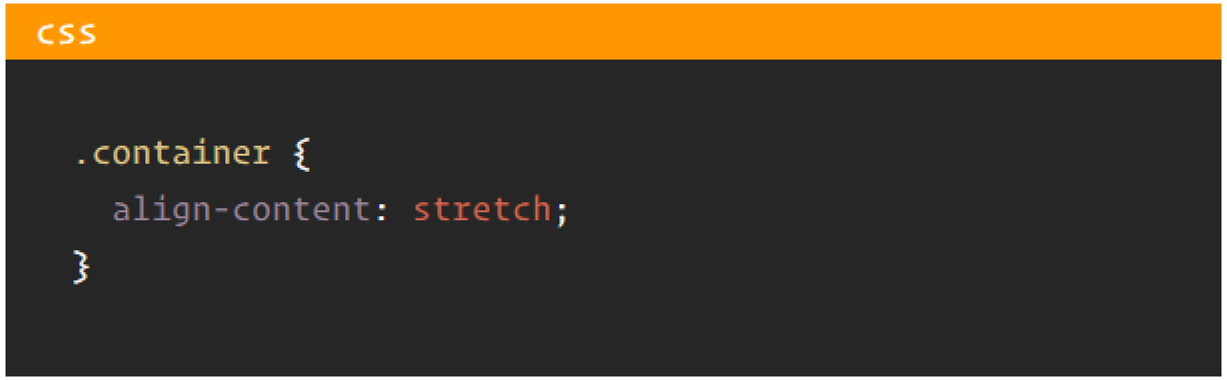


```css
.container {
  align-content: center;
}
```



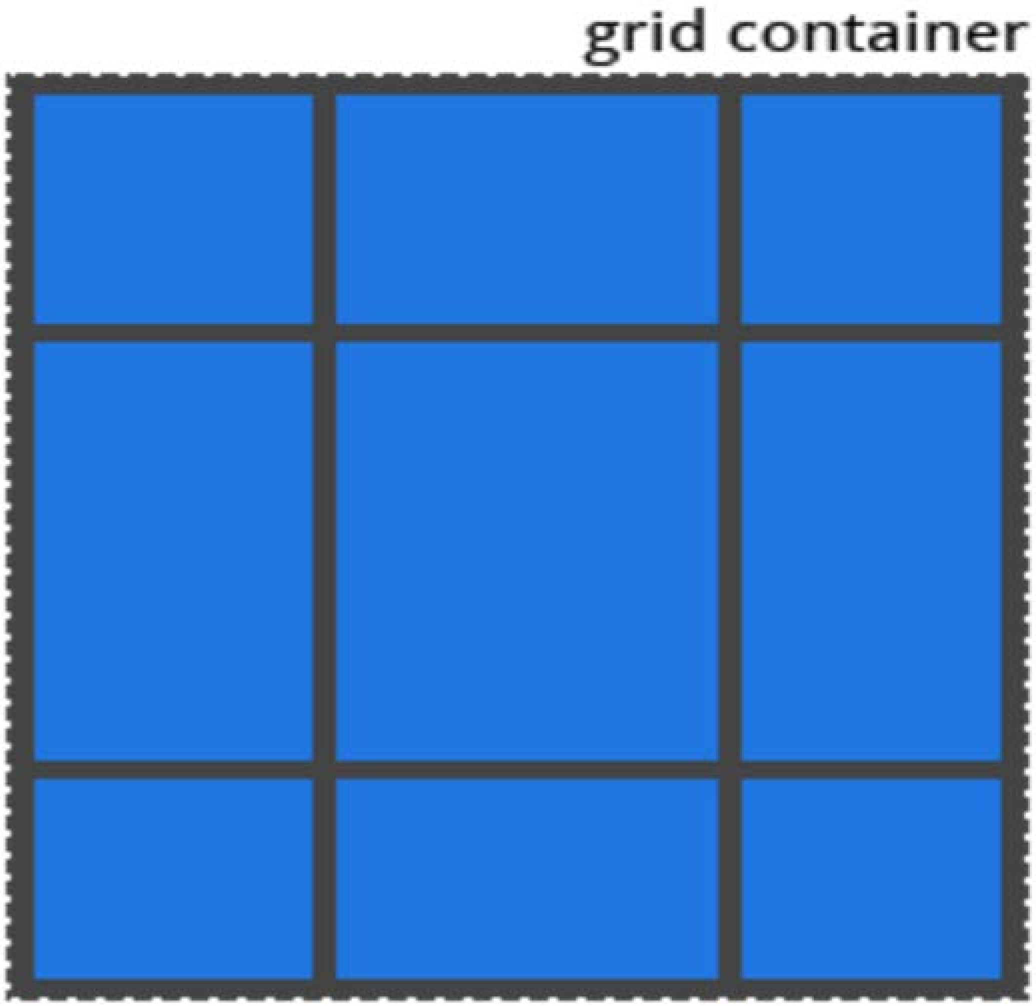
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Content - stretch



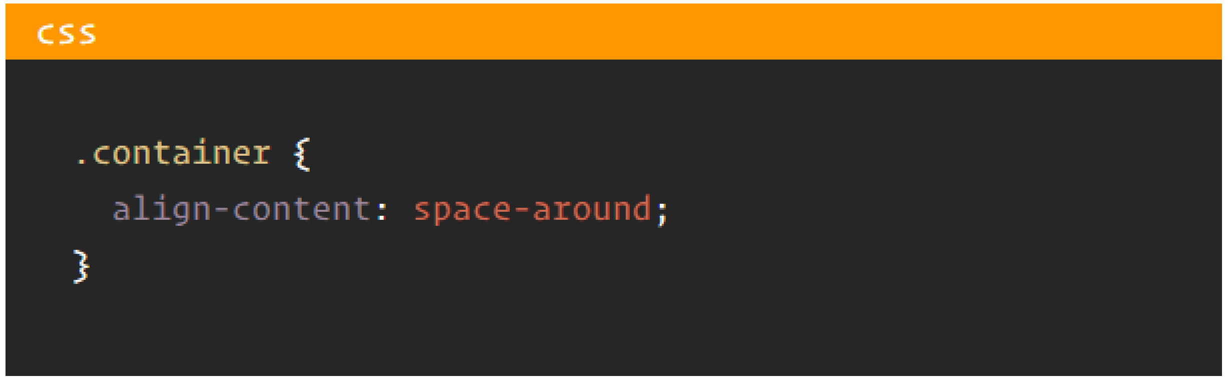
css

```css
.container {
  align-content: stretch;
}
```



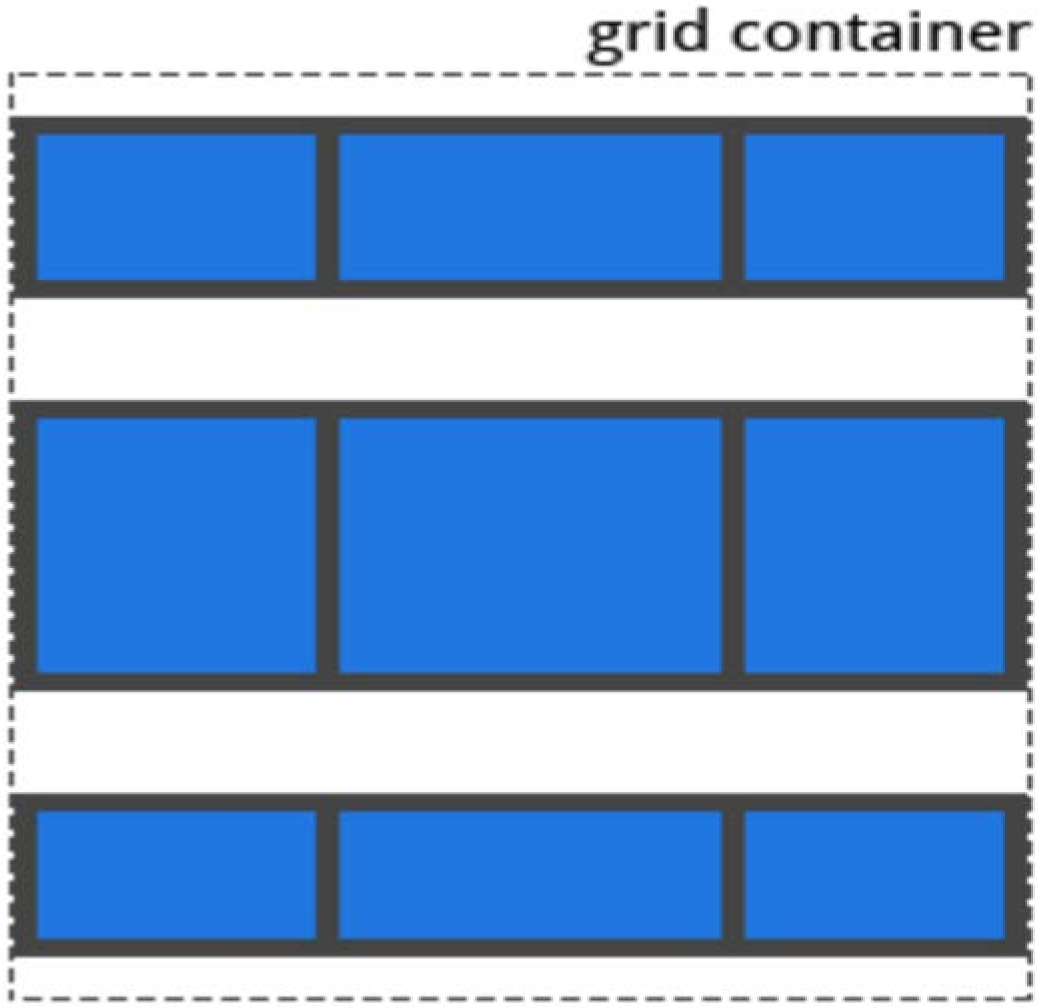
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Content – space-around



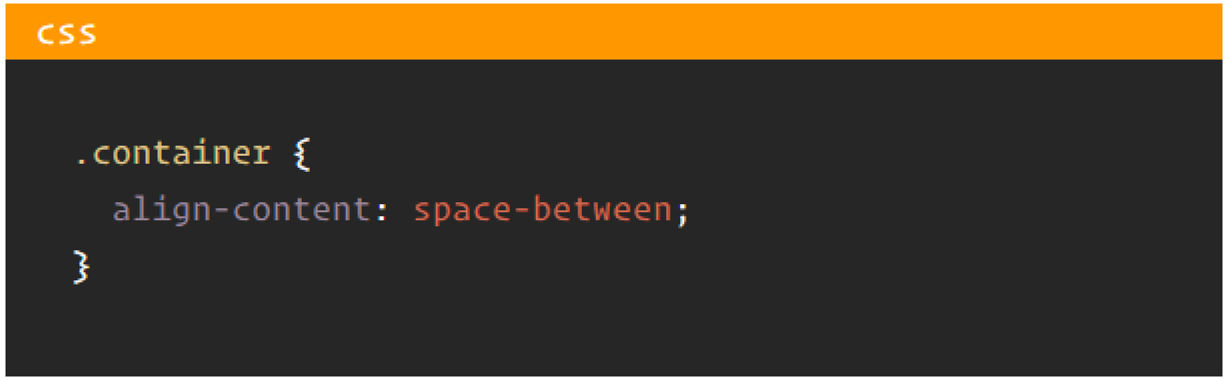
css

```
.container {
  align-content: space-around;
}
```



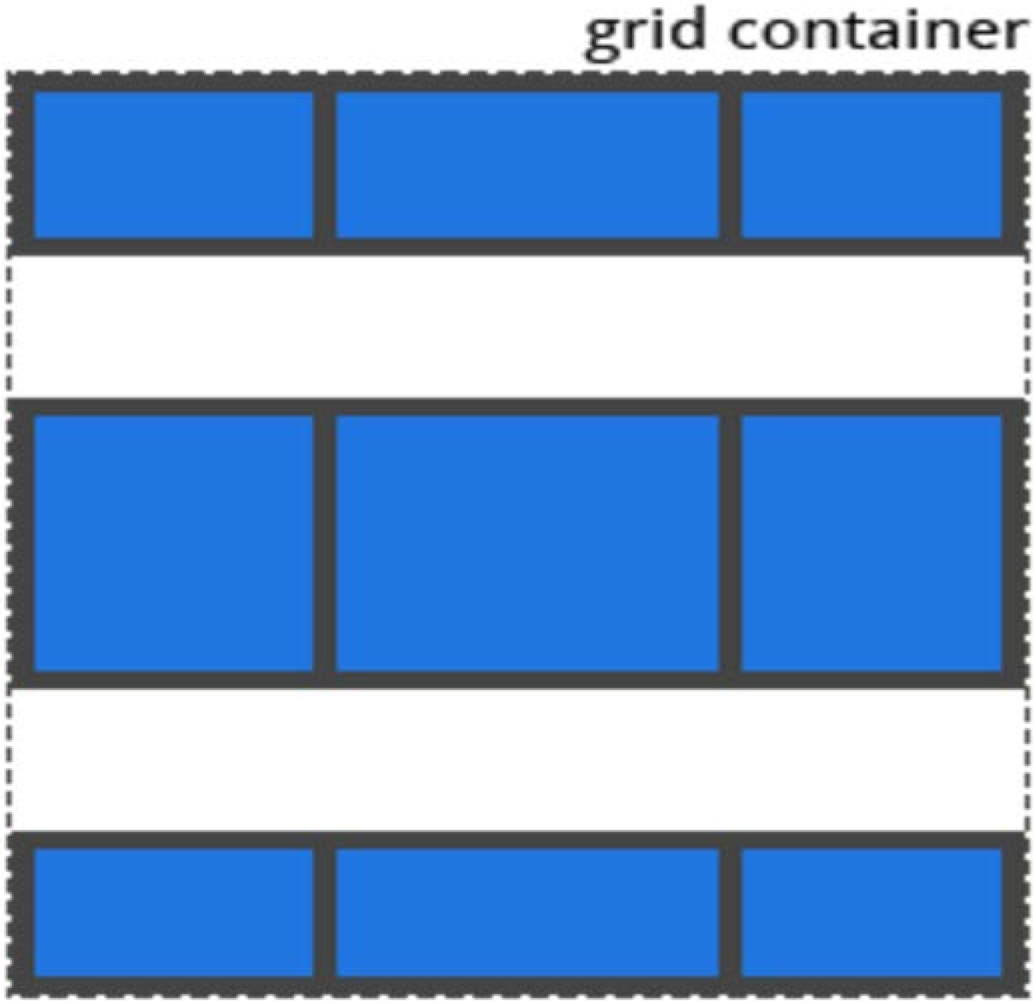
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Content – space-between



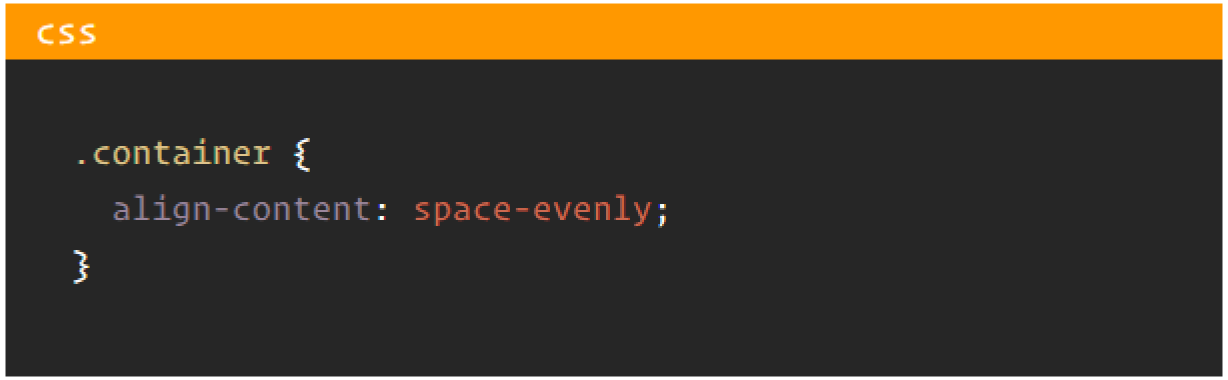
css

```
.container {
  align-content: space-between;
}
```

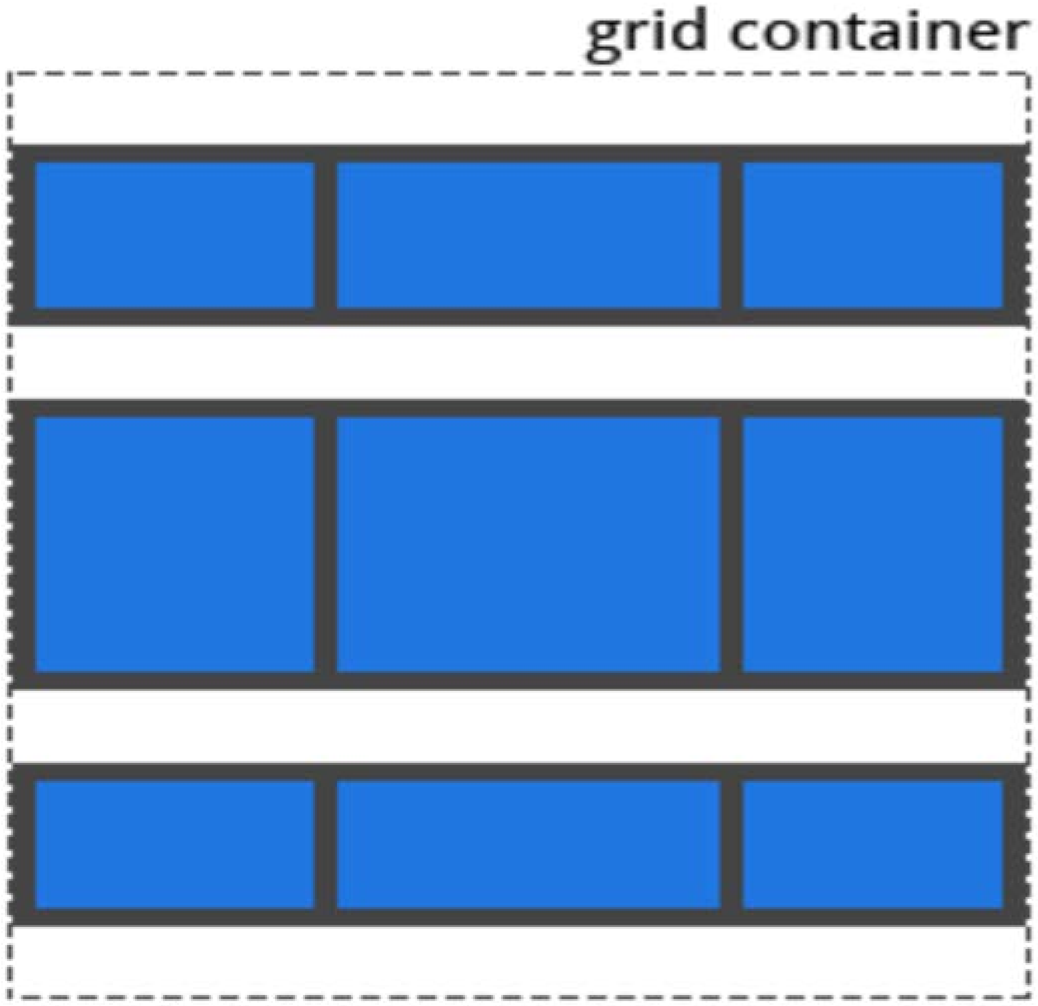


1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Align Content – space-evenly


css

```css
.container {
  align-content: space-evenly;
}
```



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

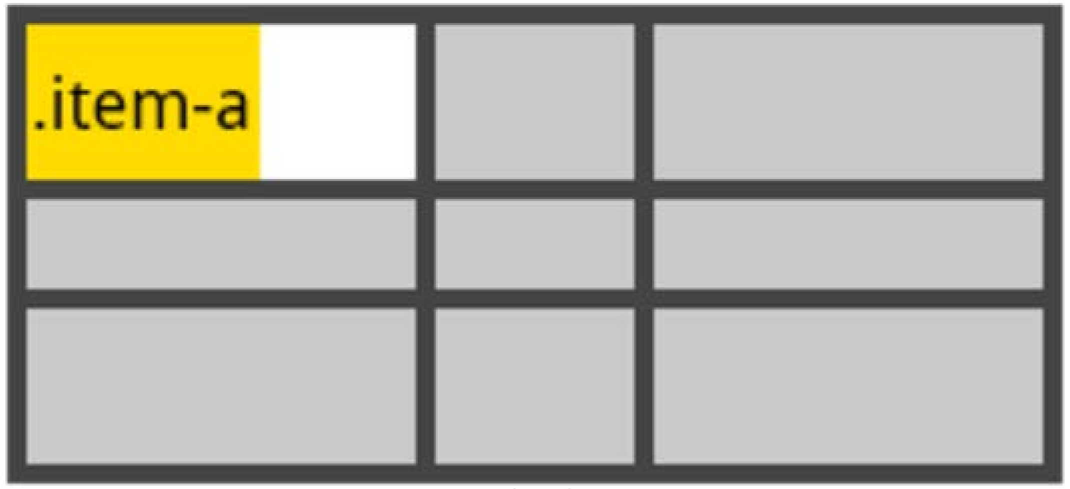
Justify Self

- Works similarly to justify-items except it is applied to a single grid item
- Aligns the content inside a grid item along the row axis (as opposed to align-self which aligns along the column axis)
- This value applies to the content inside a single grid item

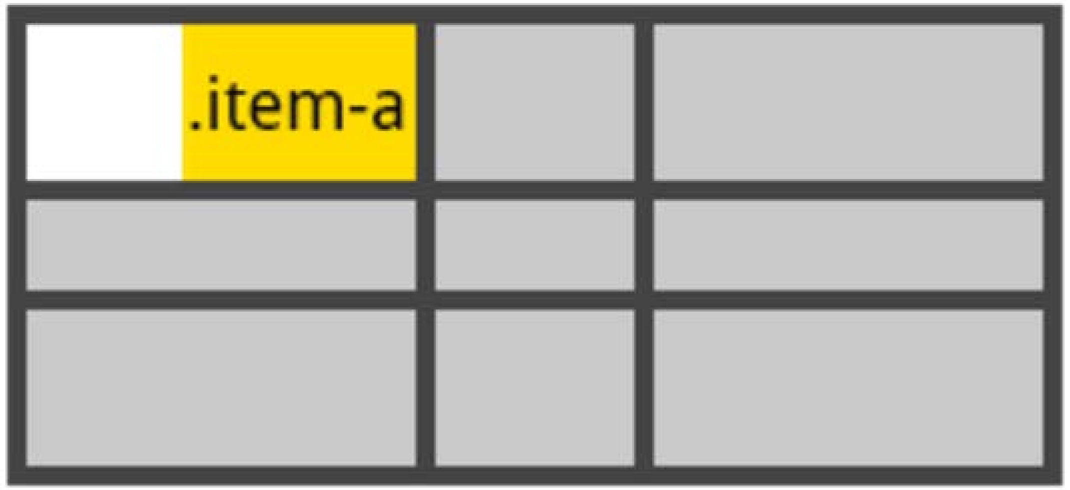
- Values:
  - **start** - aligns the content to the left end of the grid area
  - **end** - aligns the content to the right end of the grid area
  - **center** - aligns the content in the center of the grid area
  - **stretch** - fills the whole width of the grid area (this is the default)

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

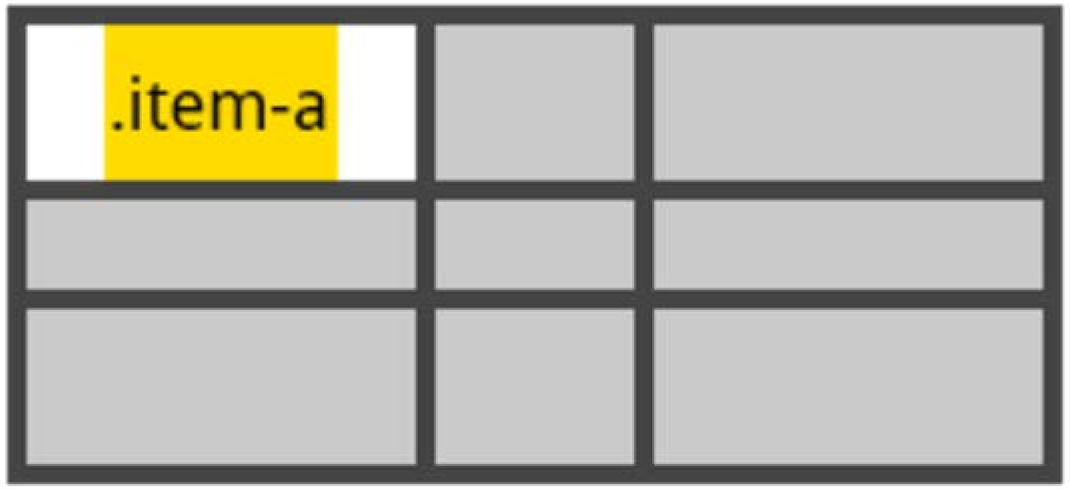
Justify Self



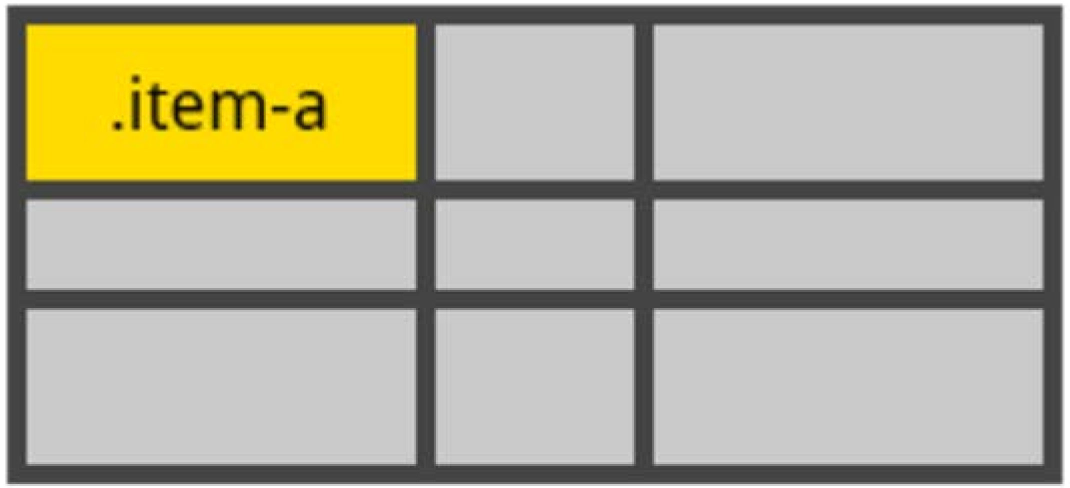
start



end



center



stretch

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

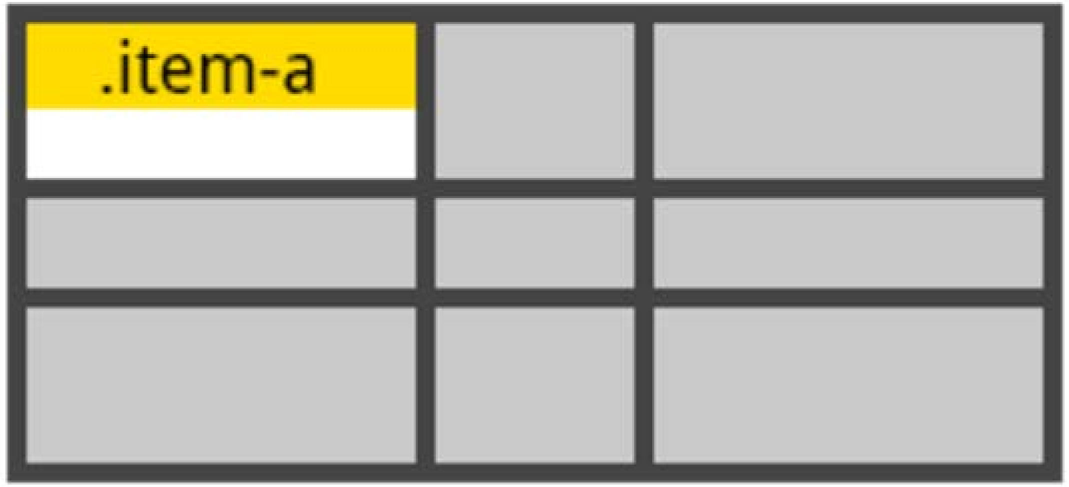
Align Self

- Works similarly to align-items except it is applied to a single grid item
- Aligns the content inside a grid item along the column axis (as opposed to justify-self which aligns along the row axis)
- This value applies to the content inside a single grid item

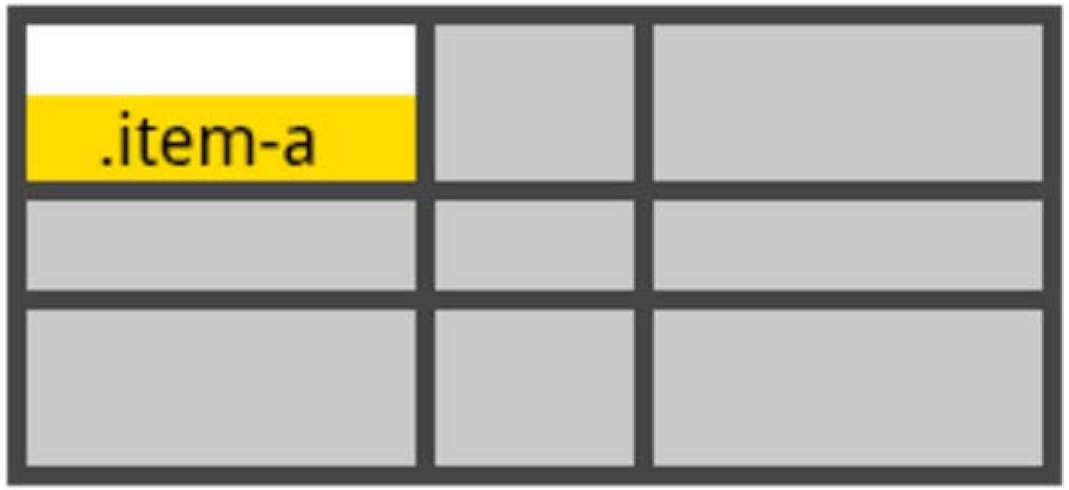
- Values:
  - **start** - aligns the content to the top of the grid area
  - **end** - aligns the content to the bottom of the grid area
  - **center** - aligns the content in the center of the grid area
  - **stretch** - fills the whole height of the grid area (this is the default)

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

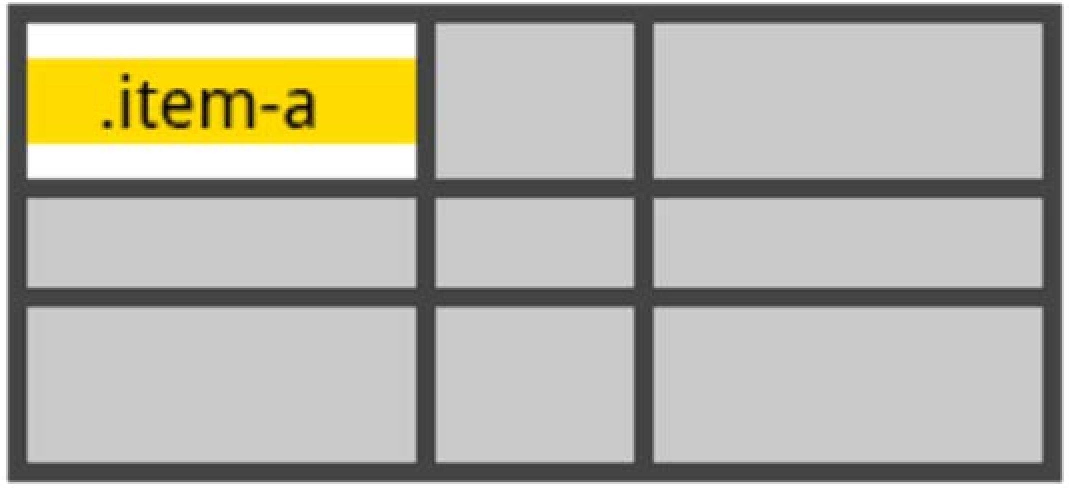
Align Self



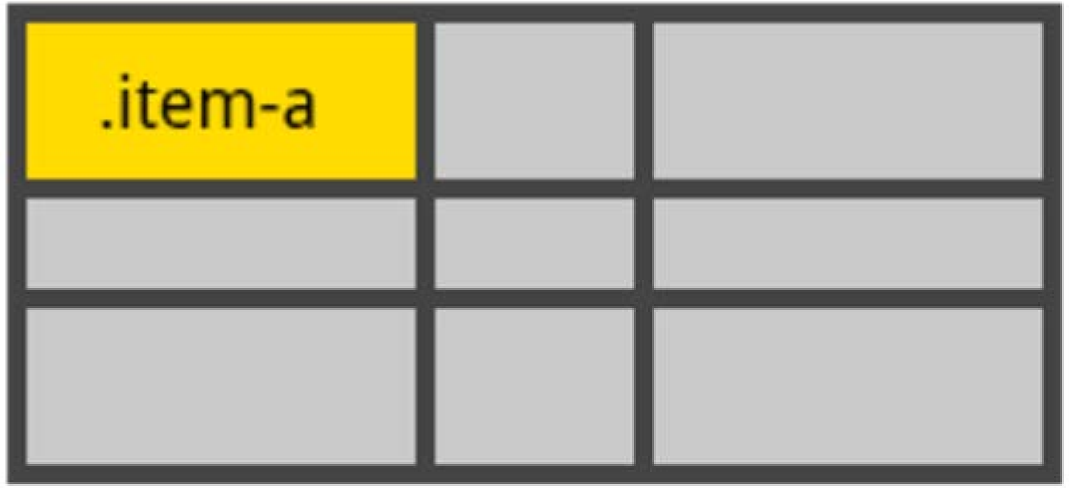
start



end



center



stretch

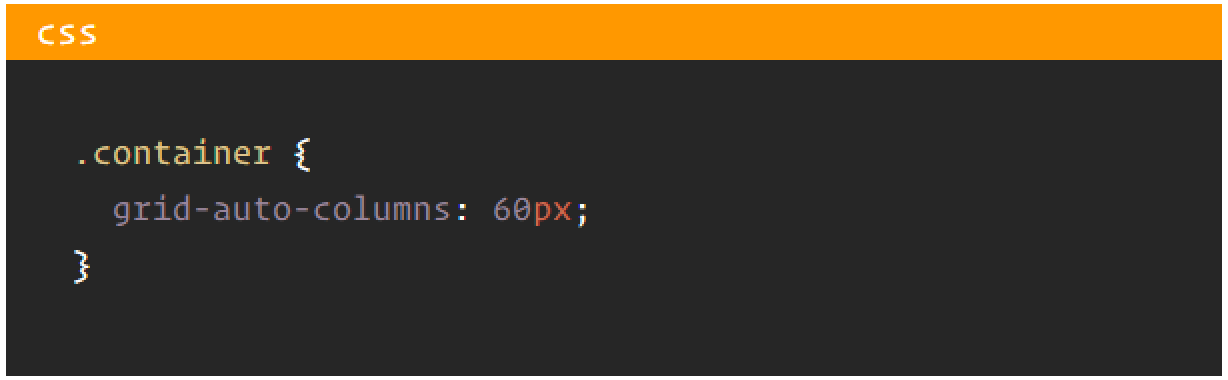
1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Grid Auto Columns and Grid Auto Rows

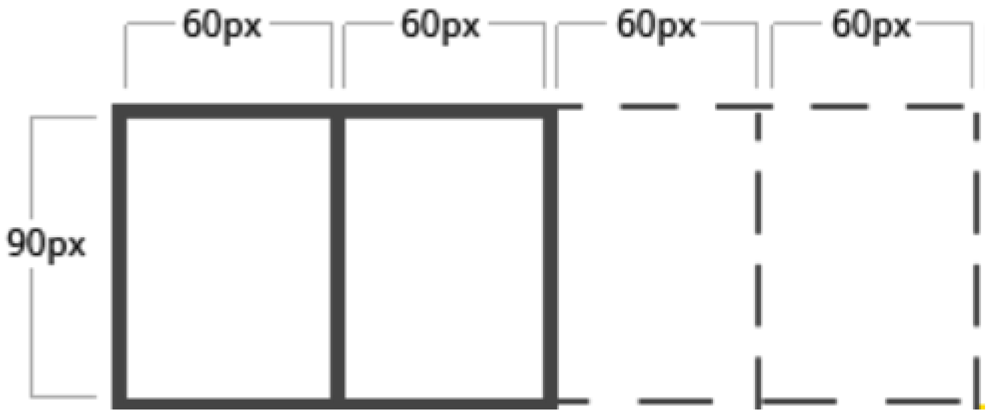
- Specifies the size of any auto-generated grid tracks (aka implicit grid tracks)
- Implicit grid tracks get created when you explicitly position rows or columns (via grid-template-rows/grid-template-columns) that are out of range of the defined grid
- We can use grid-auto-columns and grid-auto-rows to specify the widths of these implicit tracks

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Grid Auto Columns and Grid Auto Rows



```css
.container {
  grid-auto-columns: 60px;
}
```



1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Grid Auto Flow

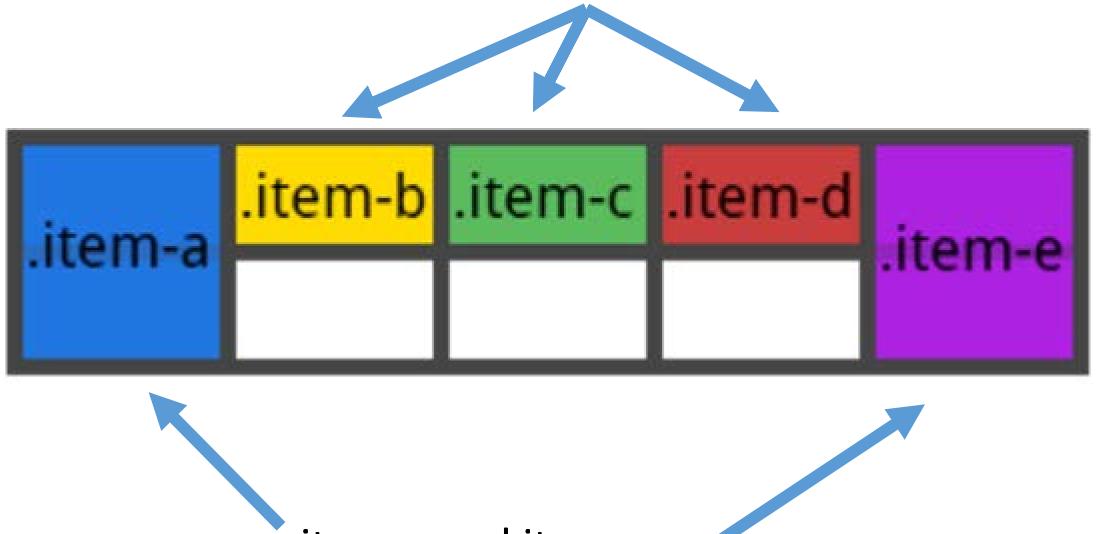
- If you have grid items that you don't explicitly place on the grid, the auto-placement algorithm kicks in to automatically place the items
- This property controls how the auto-placement algorithm works

- Values:
  - **row** - tells the auto-placement algorithm to fill in each row in turn, adding new rows as necessary
  - **column** - tells the auto-placement algorithm to fill in each column in turn, adding new columns as necessary
  - **dense** - tells the auto-placement algorithm to attempt to fill in holes earlier in the grid if smaller items come up later

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Grid Auto Flow - row

item-b, item-c and item-d are not set explicitly on the grid, so they are placed automatically along the first available row

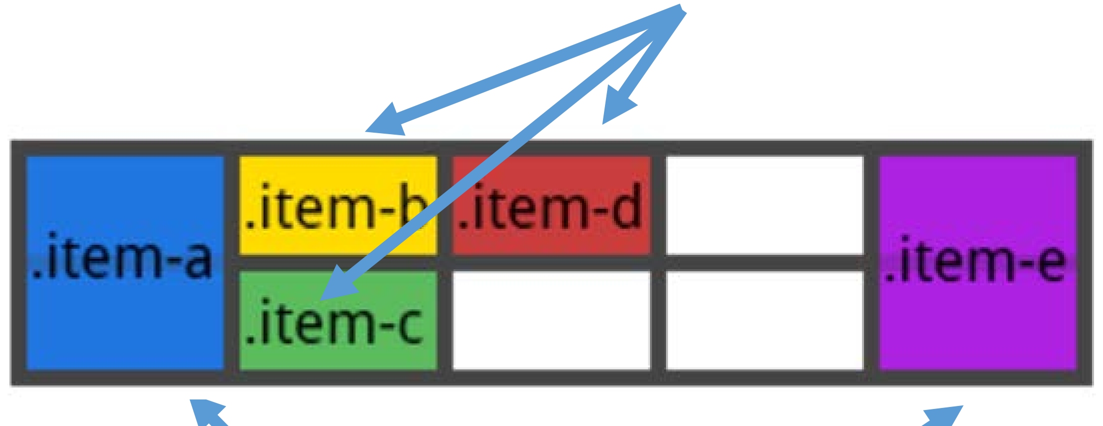


item-a and item-e are set explicitly on the grid

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Grid Auto Flow - column

item-b, item-c and item-d are not set explicitly on the grid, so they are placed automatically along the first available column



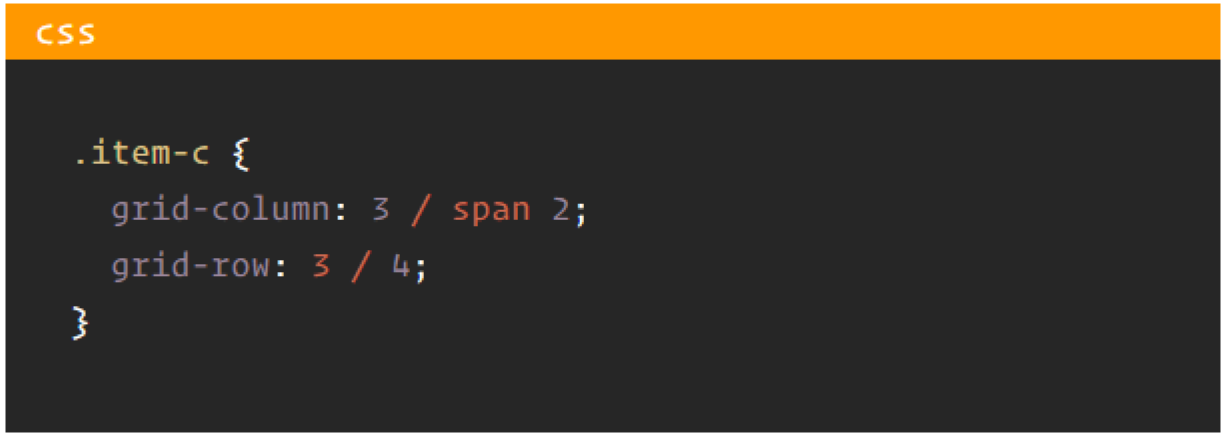
item-a and item-e are set explicitly on the grid

1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

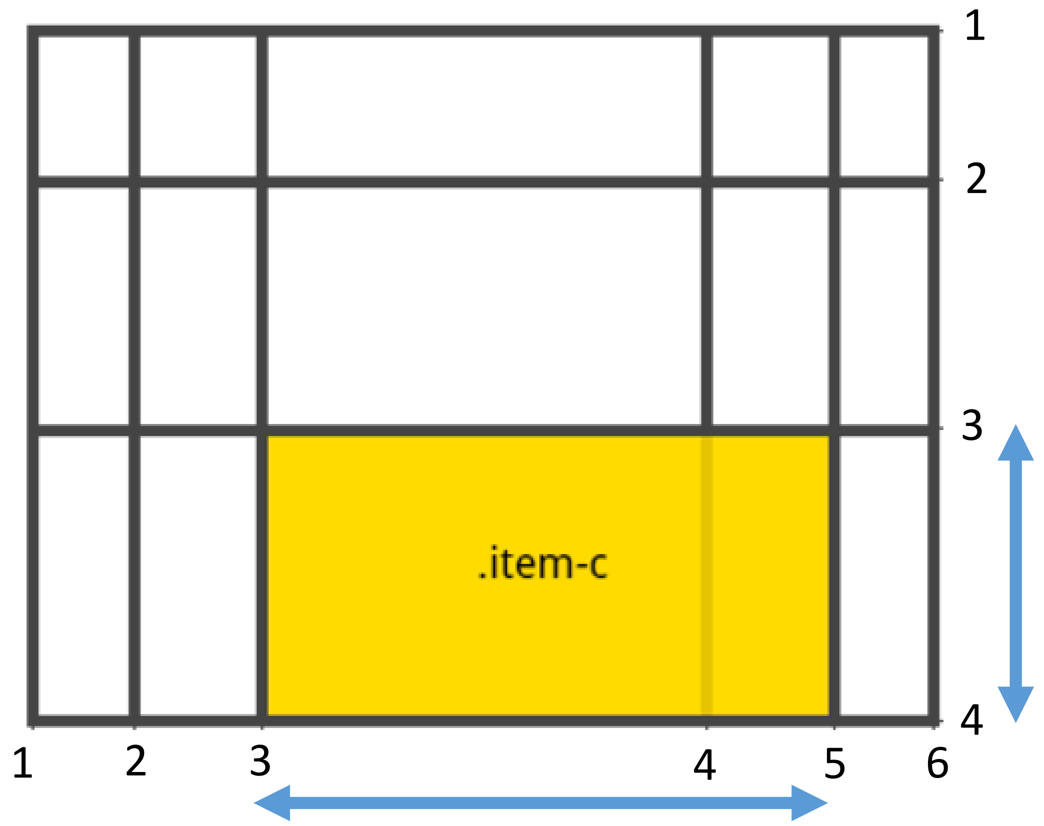
Grid Column and Grid Row

- The "grid-column" and "grid-row" property are set on a grid item
- They determine where on the grid a grid item will go
- The first value is the starting line followed by a " / " (the space is important)" and then a second value which is the ending line
- You can tell a grid item to span a number of lines by using the "span" keyword

Grid Column and Grid Row



```css
.item-c {
  grid-column: 3 / span 2;
  grid-row: 3 / 4;
}
```

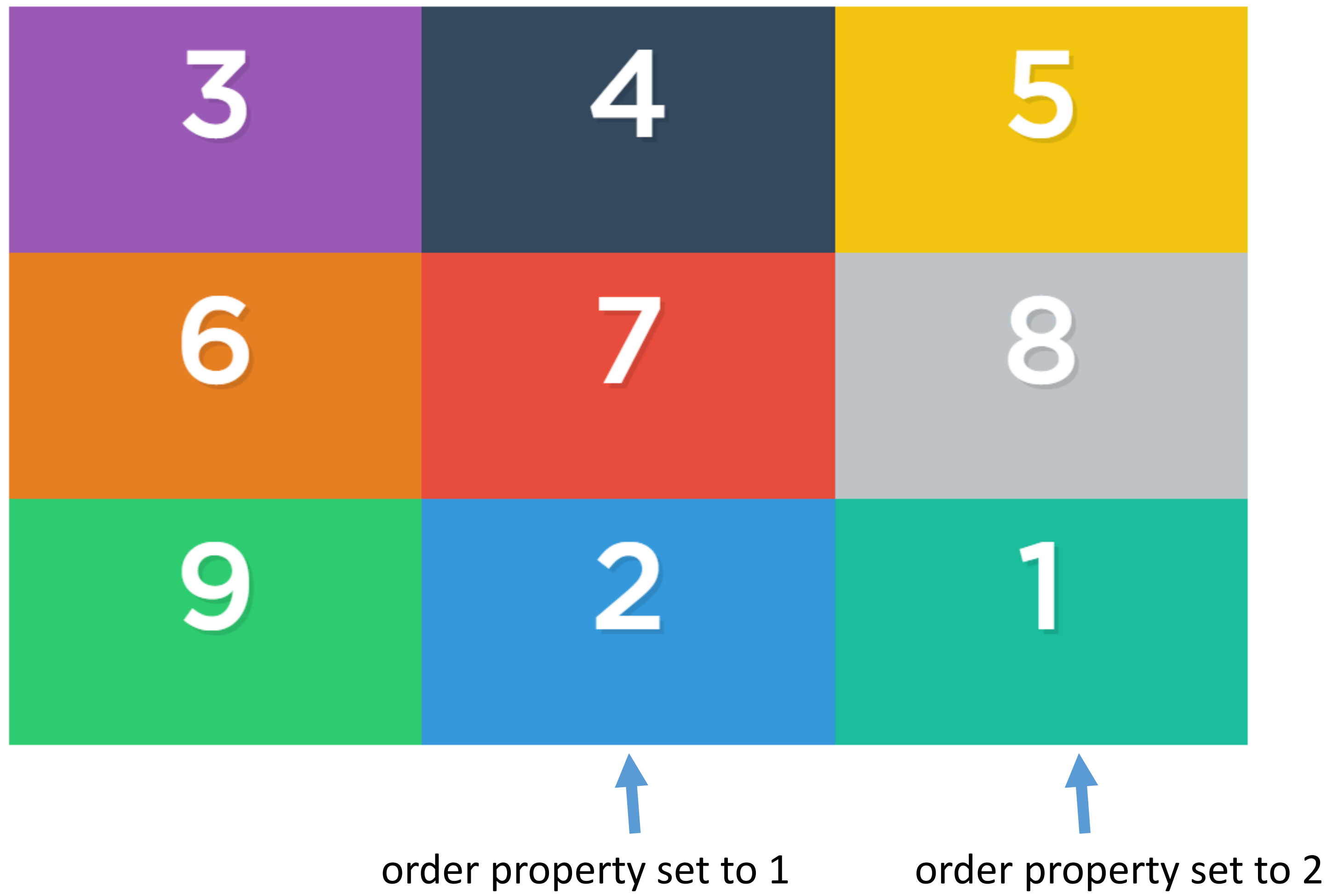


1. Information on this slide from: https://css-tricks.com/snippets/css/complete-guide-grid/

Grid Order

- Similar to flex box's order property. This is set on the grid items
- Any numerical value is valid. Grid items order values with larger numbers go to the end of the grid

Grid Order


3
4
5
6
7
8
9
2
1
order property set to 1
order property set to 2