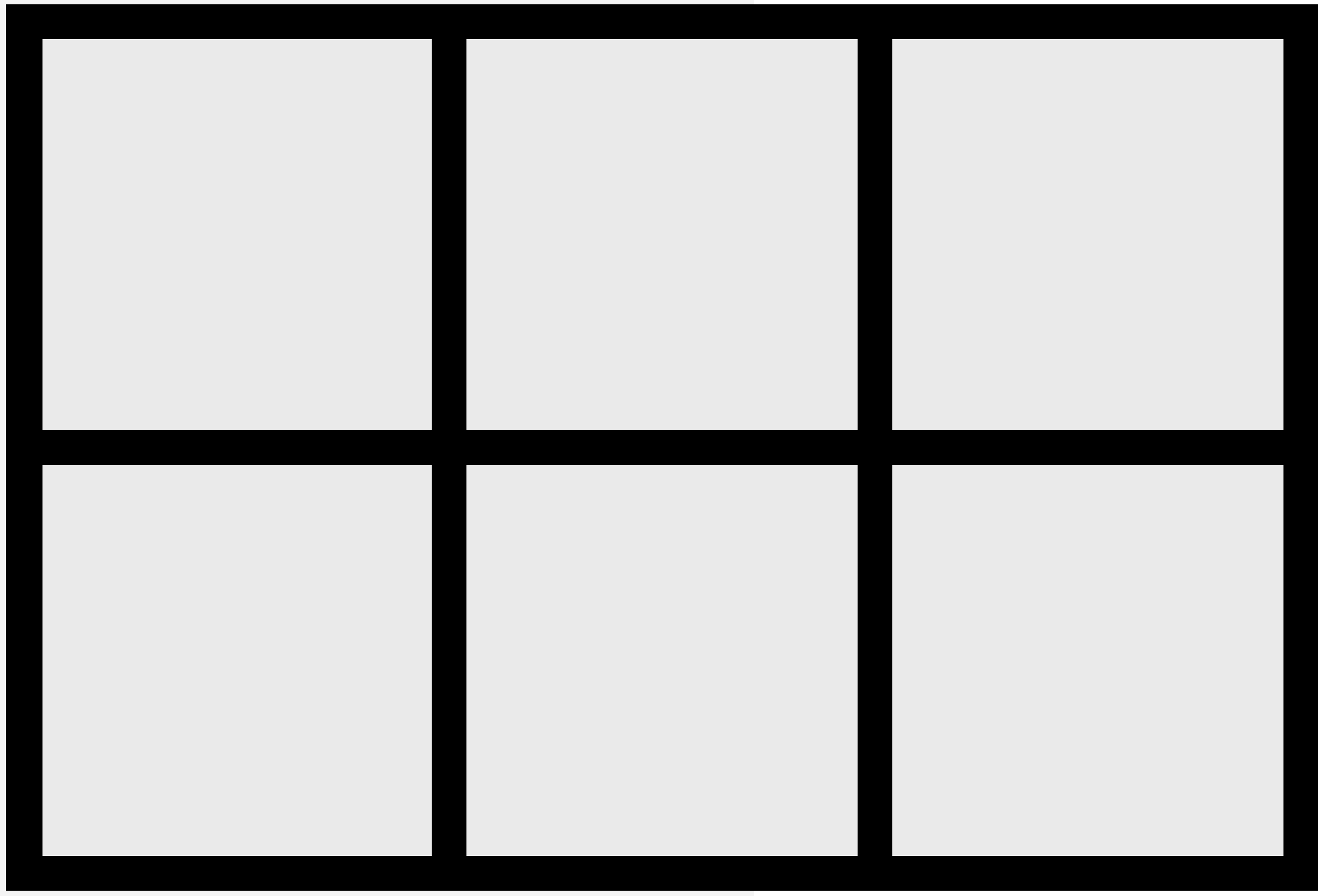
INTRODUCING CSS GRID

Jason Yingling (@jason_yingling)

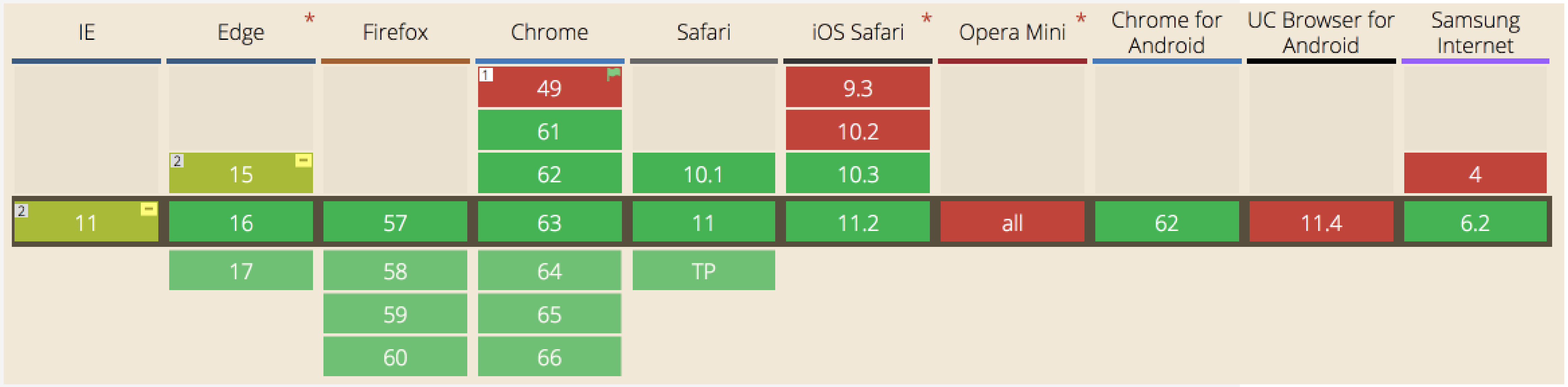
WHAT IS
CSS GRID

CSS Grid is a 2 dimensional layout system that can handle both columns and rows.



CAN I USE
CSS GRID

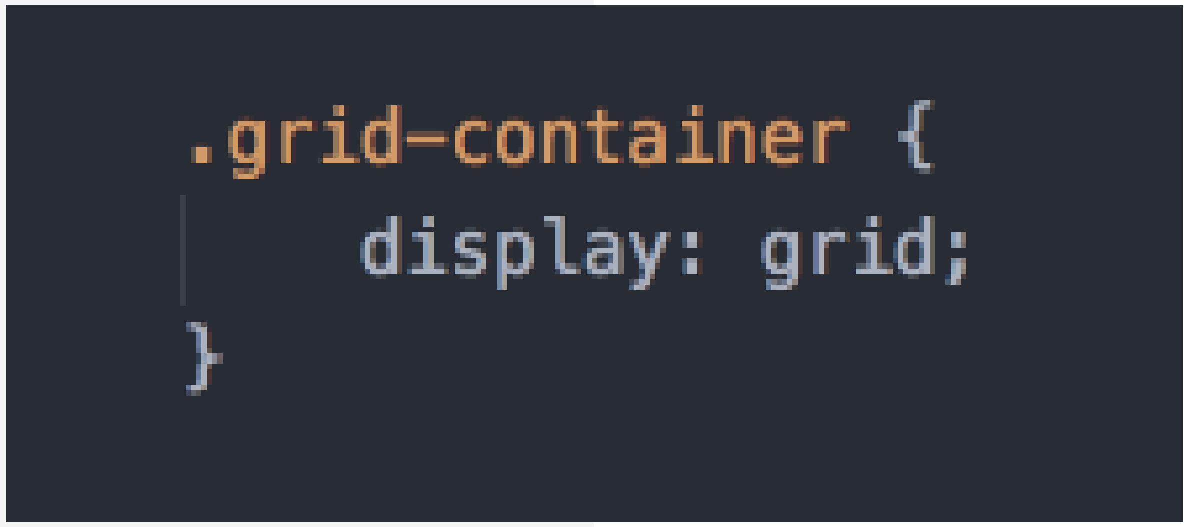
CSS Grid is supported in the latest version of all major browsers and can be safely used with a fallback.



IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			🟥 [1] 49		🟥 9.3				
			🟩 61		🟥 10.2				
	🟨 [2] 15		🟩 62	🟩 10.1	🟩 10.3				🟥 4
🟨 [2] 11	🟩 16	🟩 57	🟩 63	🟩 11	🟩 11.2	🟥 all	🟩 62	🟥 11.4	🟩 6.2
	🟩 17	🟩 58	🟩 64	🟩 TP					
		🟩 59	🟩 65						
		🟩 60	🟩 66						

GRID CONTAINER

The grid container is the parent of all grid items. It is declared with the display: grid; rule.

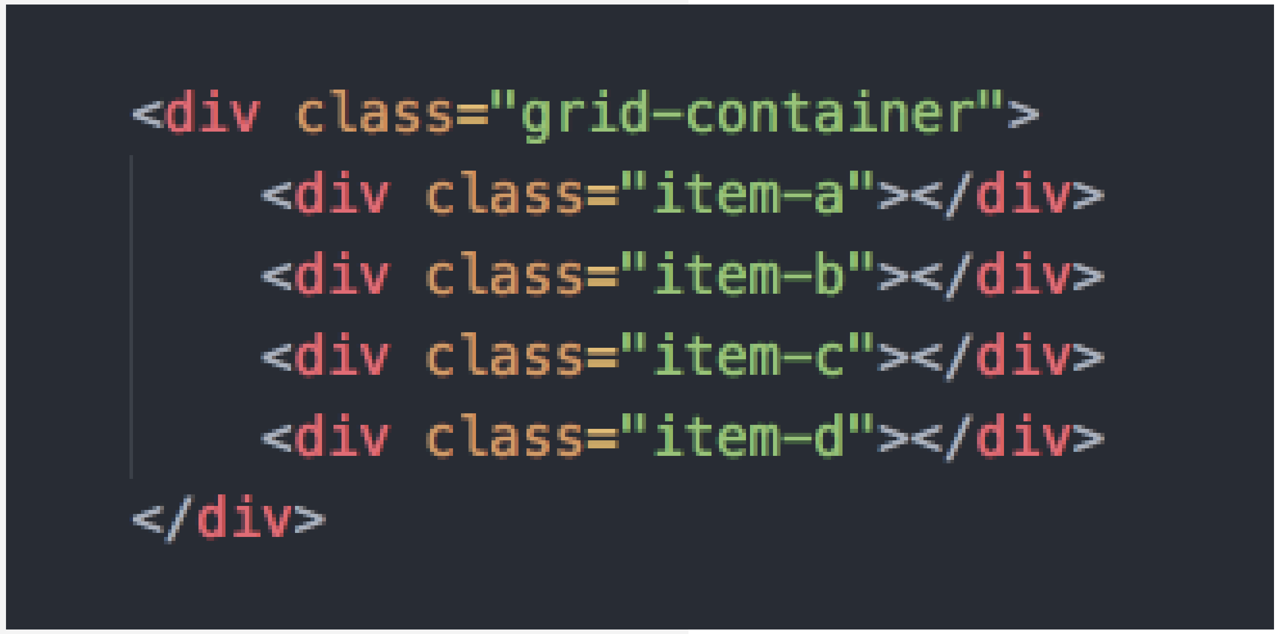


```
.grid-container {
    display: grid;
}
```

GRID ITEM

The items that are direct children of the grid container.

These are the elements that will be placed on the grid.

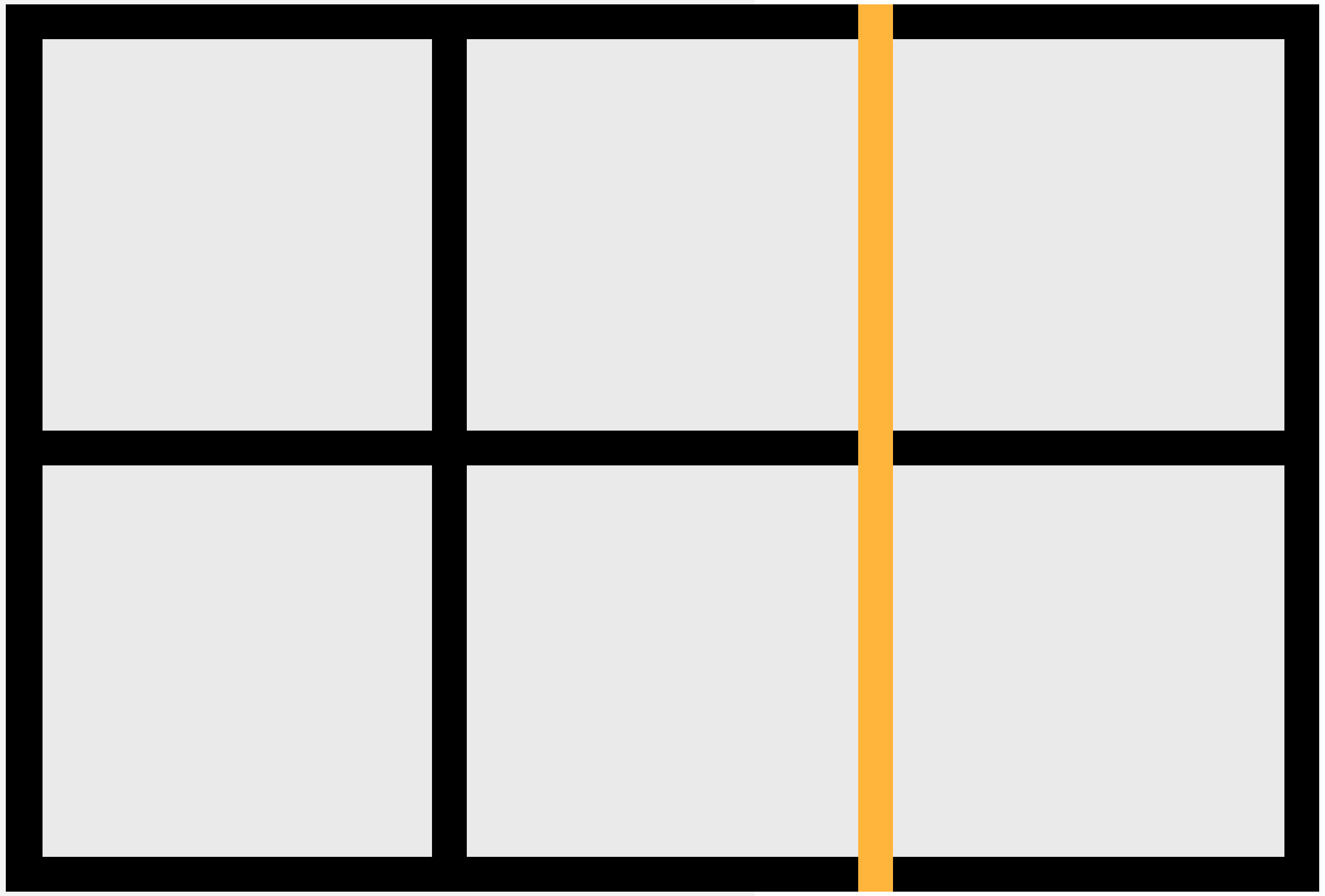


```
<div class="grid-container">
    <div class="item-a"></div>
    <div class="item-b"></div>
    <div class="item-c"></div>
    <div class="item-d"></div>
</div>
```

GRID LINES

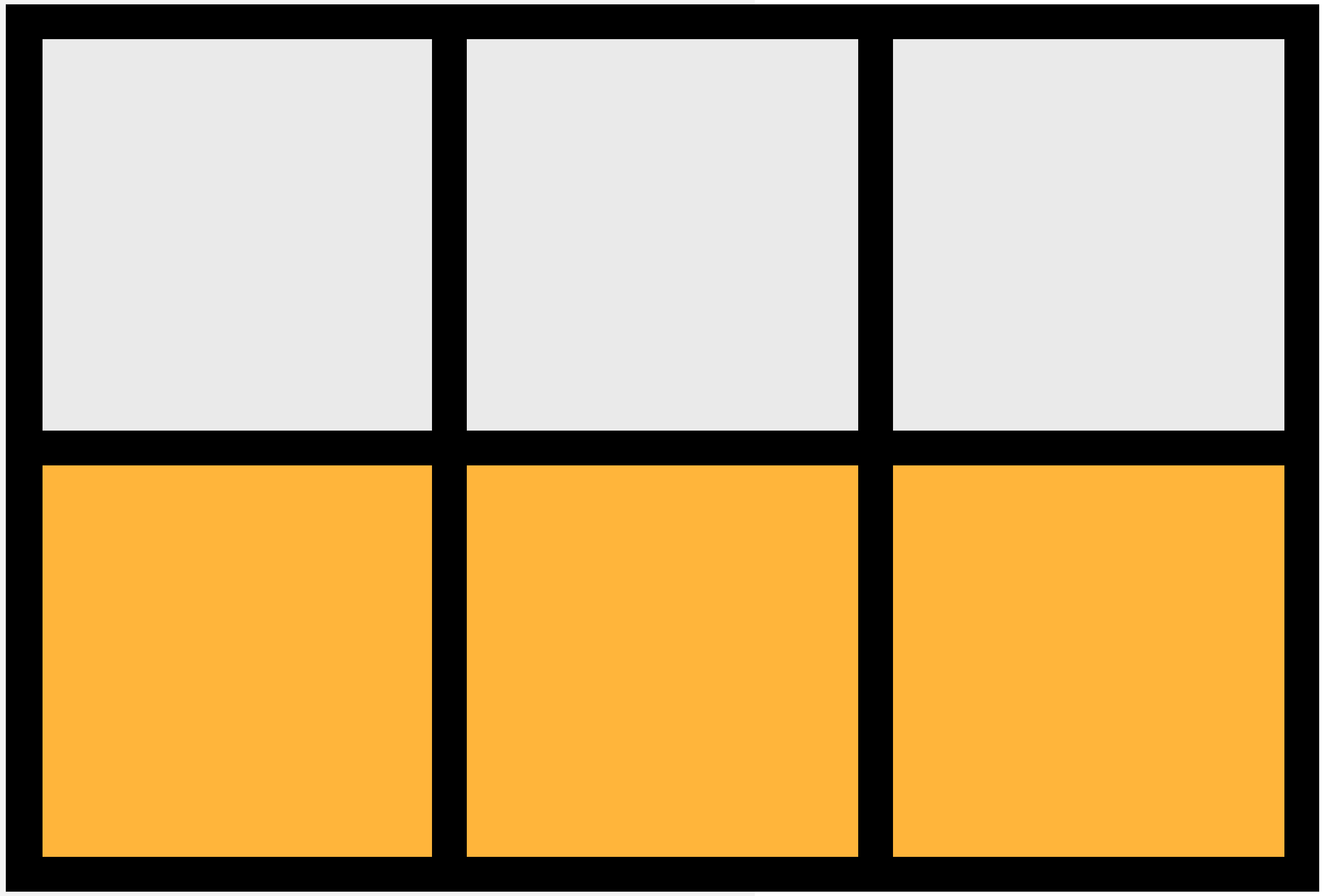
The lines that divide the columns and rows of the grid.

These are numbered starting with 1.



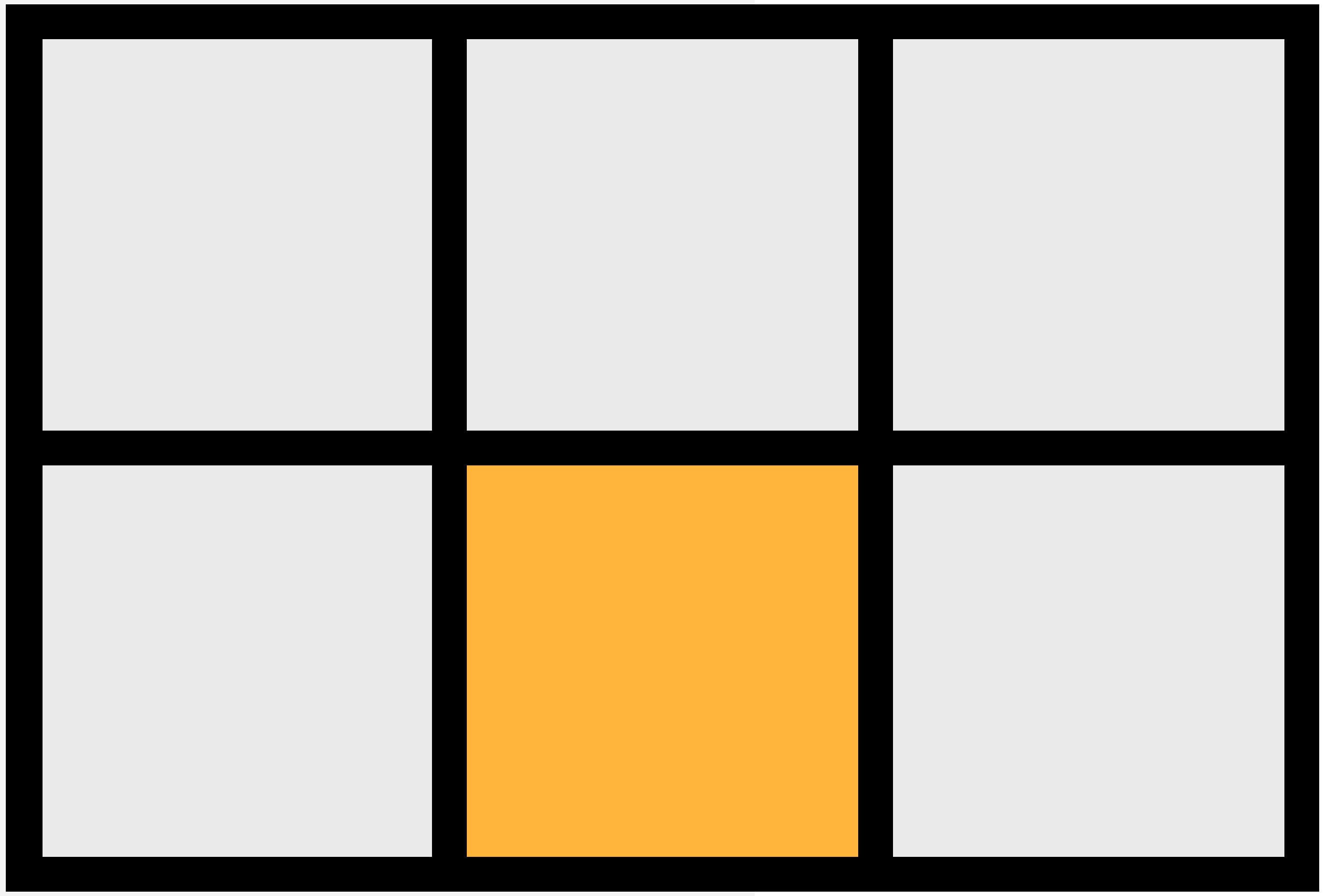
GRID
TRACK

The space between 2 adjacent grid lines. Essentially your columns and rows.



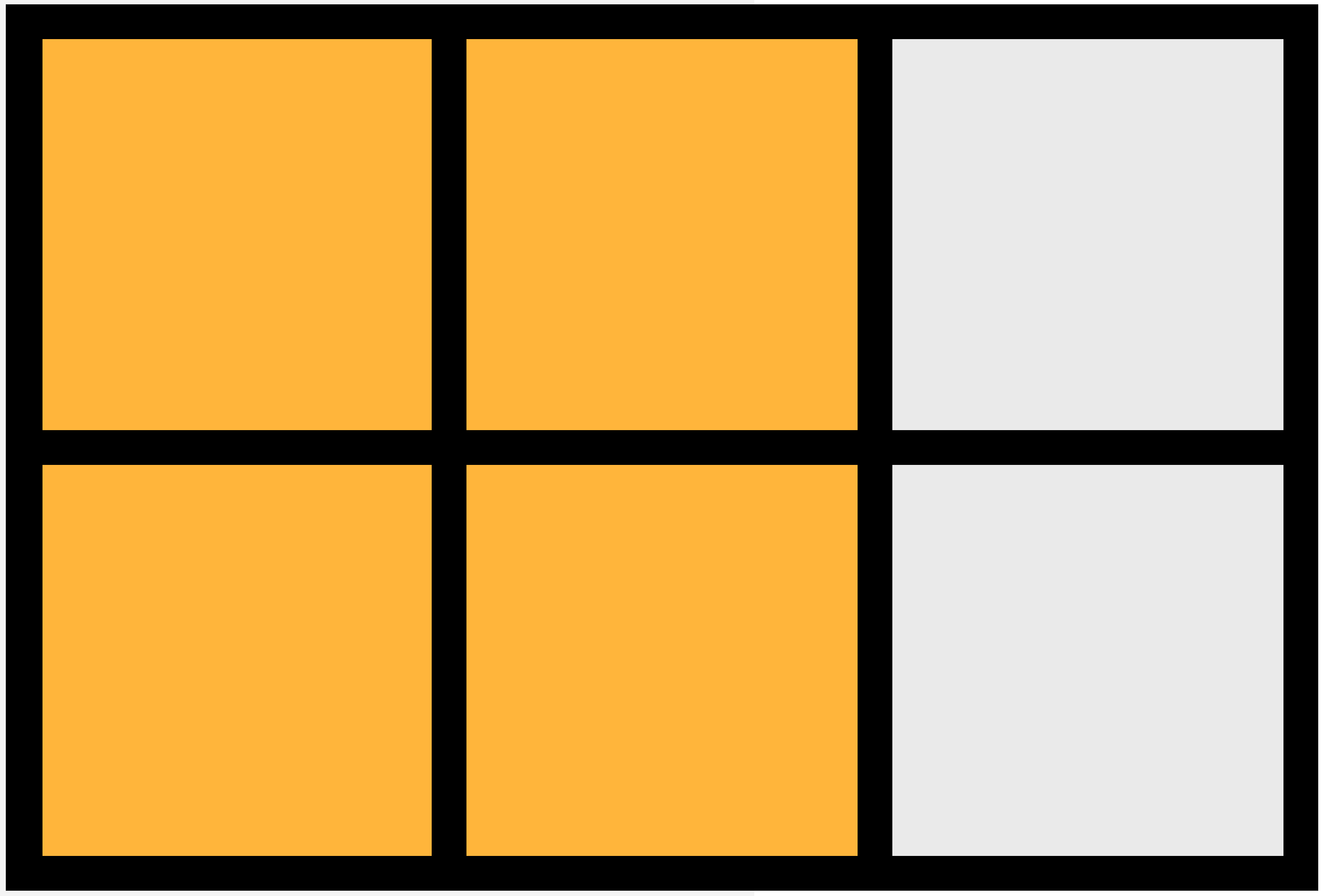
GRID
CELL

The space between 2 adjacent grid column lines and 2 adjacent grid row lines.



GRID AREA

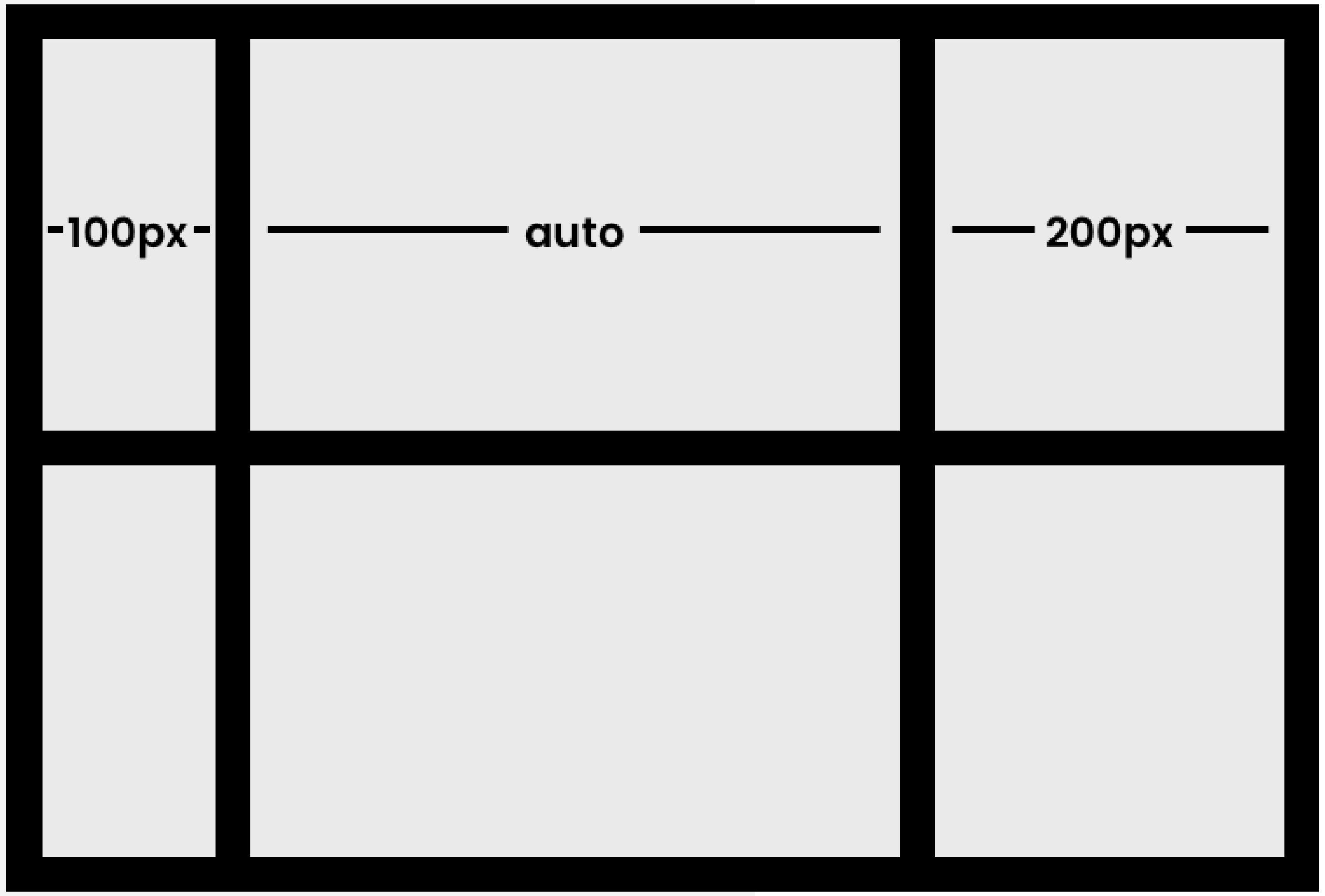
The space between any 4 grid lines. It can cover any number of rows and columns.



GRID TEMPLATE COLUMNS

Defines the width of the columns in the grid.

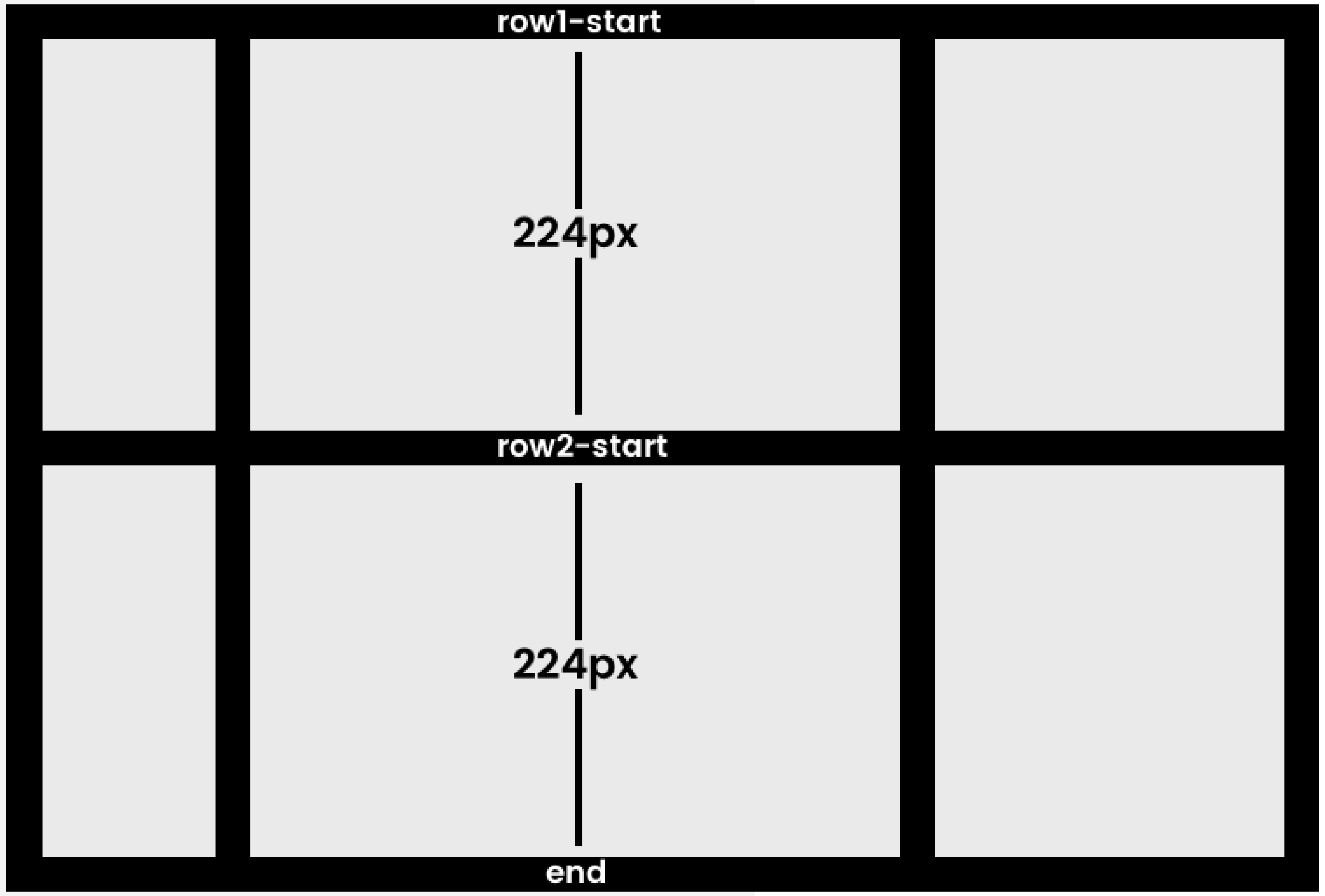
```
grid-template-columns: 100px auto 200px;
```



GRID TEMPLATE ROWS

Defines the height of the grid rows.

grid-template-rows: [row1-start] 224px [row2-start] 224px [end];



GRID COLUMN GAP

Define the size of the grid lines between columns.

```
grid-column-gap: 30px;
```

GRID ROW GAP

Defines the size of the grid lines between rows.

```
grid-row-gap: 40px;
```

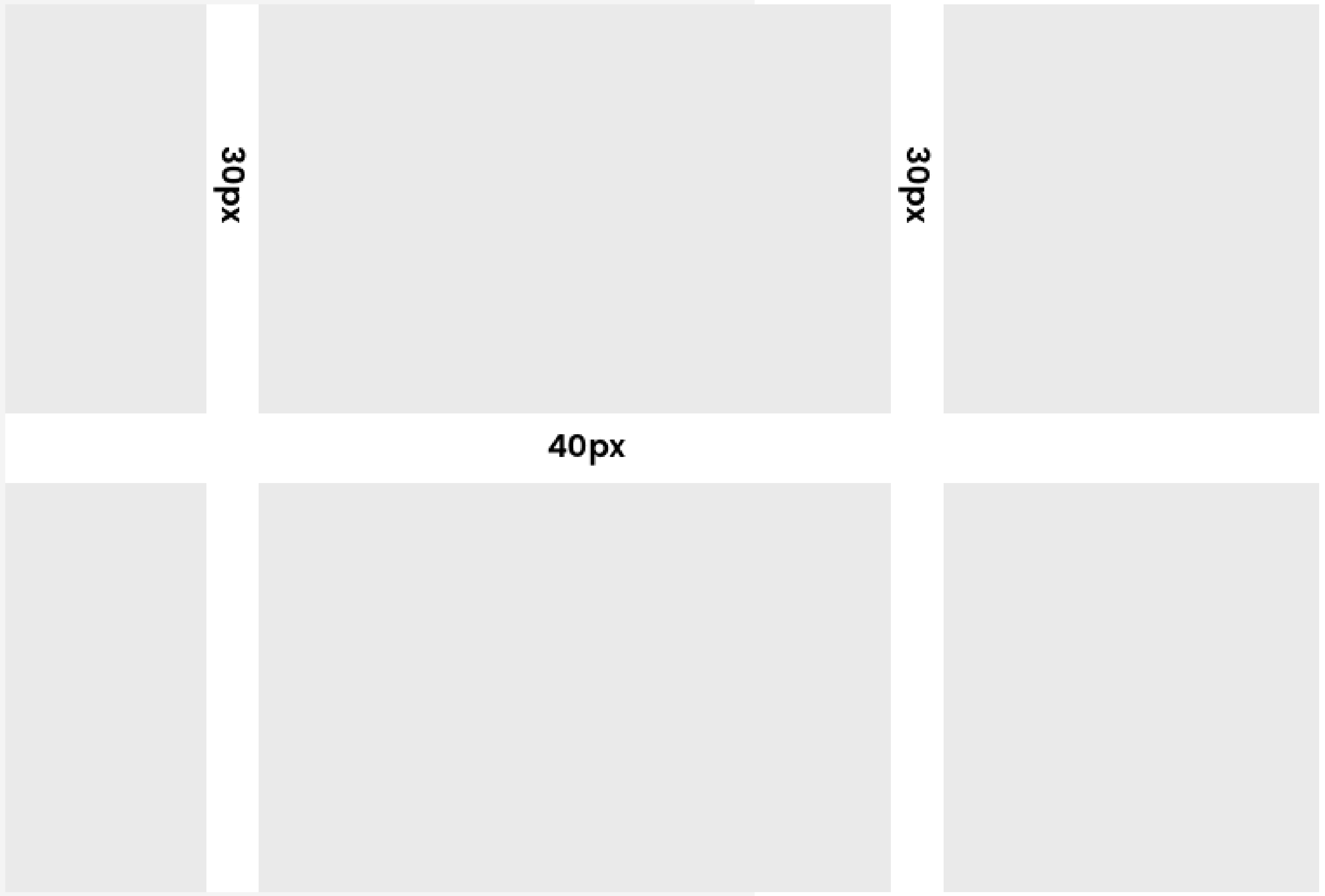
GRID GAP

Shorthand for row and column gap.

grid-gap: <grid-row-gap> <grid-column-gap>;

grid-gap: 40px 30px;

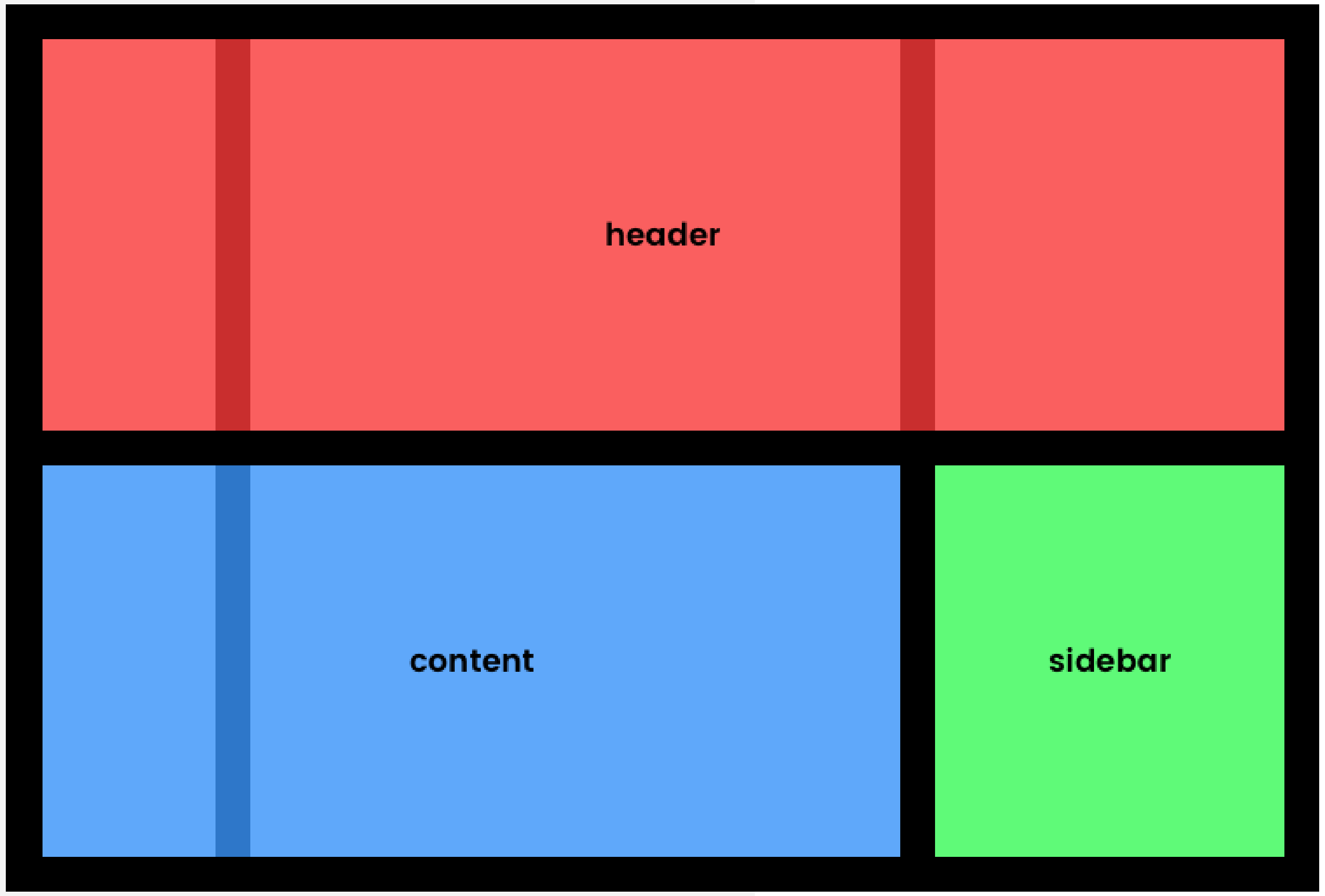
Note: The gap is only between columns and rows. Not outside.



GRID TEMPLATE AREA

Allows you to name sections of the grid.

```
grid-template-columns: 100px auto 200px;

grid-template-rows: [row1-start] 224px [row2-start] 224px [end];

grid-template-areas:

  "header header header"

  "content content sidebar";
```



FRACTIONAL (FR) UNITS

Grid comes with the fr unit. Which uses up a fraction of the remaining free space in the grid.

```
grid-template-columns: 1fr 50px 1fr 1fr;
```

REPEAT() NOTATION

Allows you to easily repeat settings.

```
grid-template-columns: repeat(12, 1fr);
```

GRID COLUMN
START

Grid Column Start / Grid Column End | Grid Column

```
grid-column-start: 2;
grid-column-end: 4;
```

Shorthand

```
grid-column: 2 / 4;
```

GRID ROW
START

Grid Row Start / Grid Row End | Grid Row

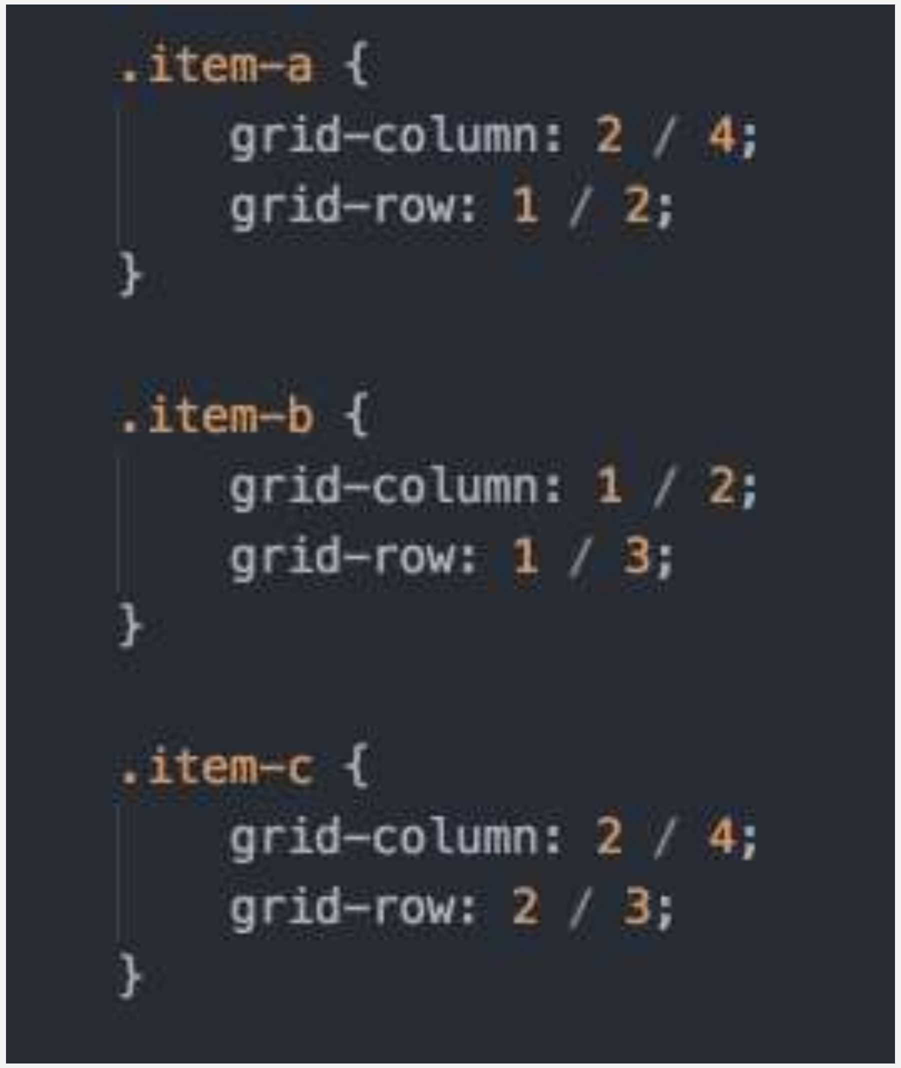
grid-row-start: 1;

grid-row-end: 2;

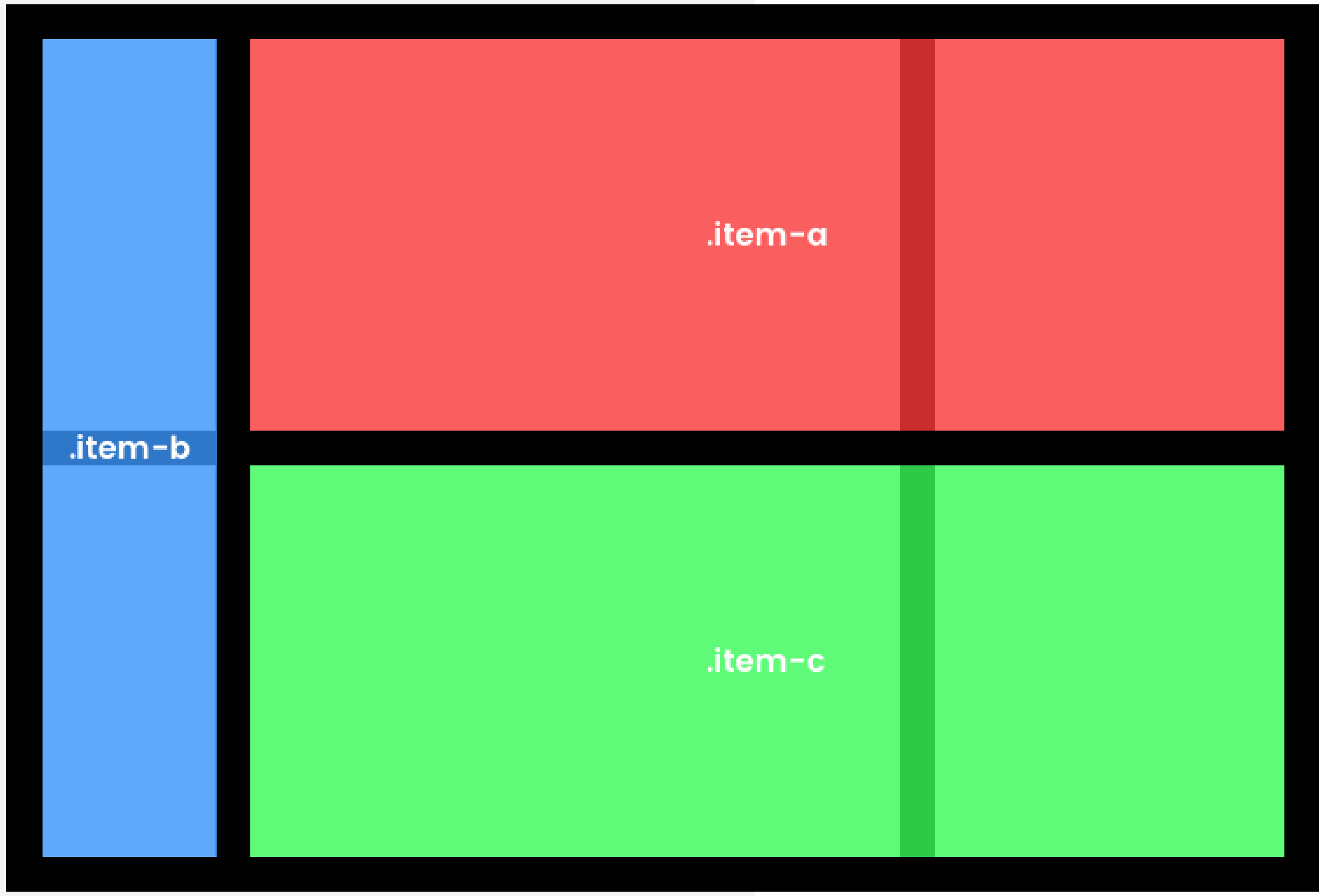
Shorthand

grid-row: 1 / 2;

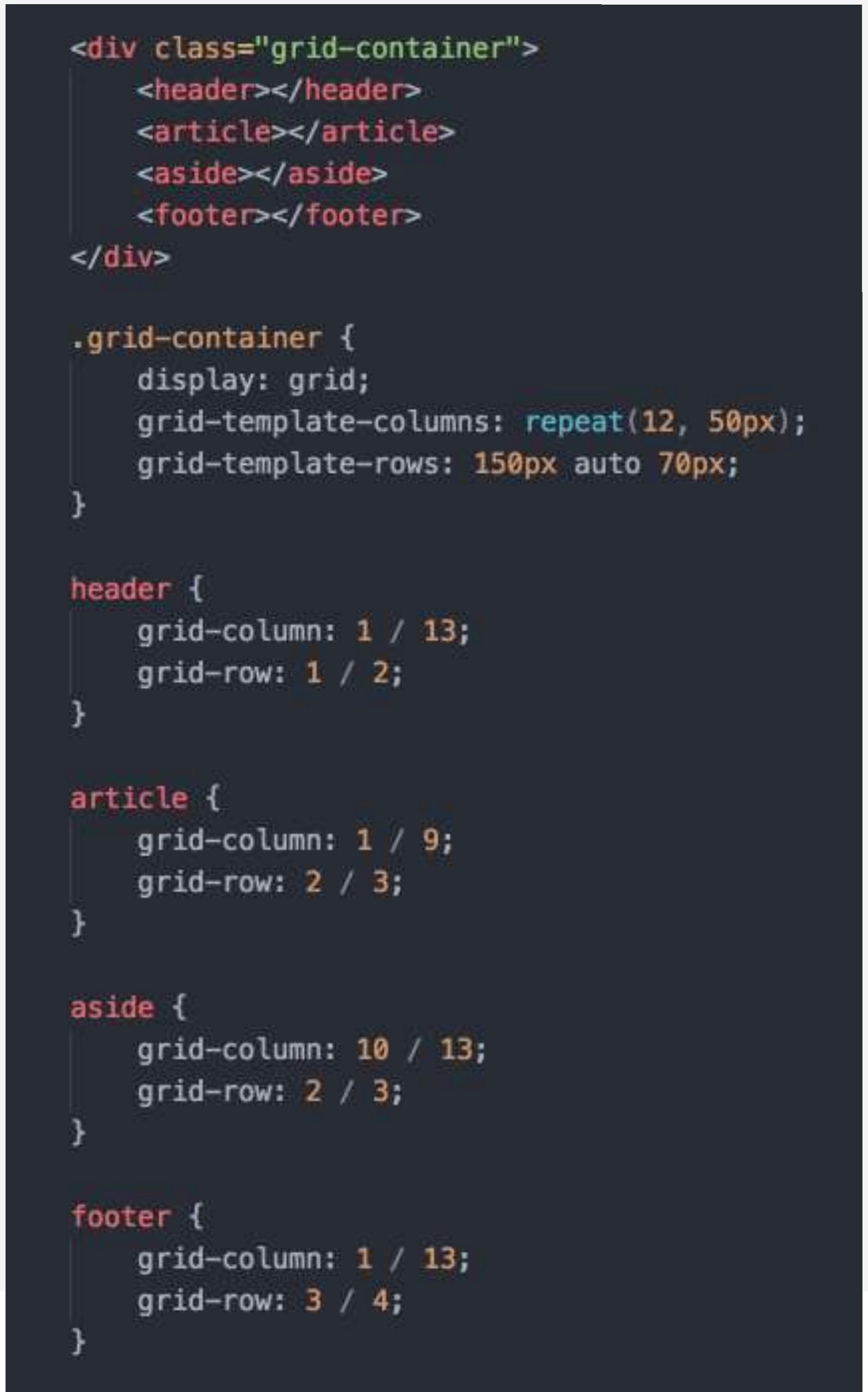
PLACING GRID ITEMS



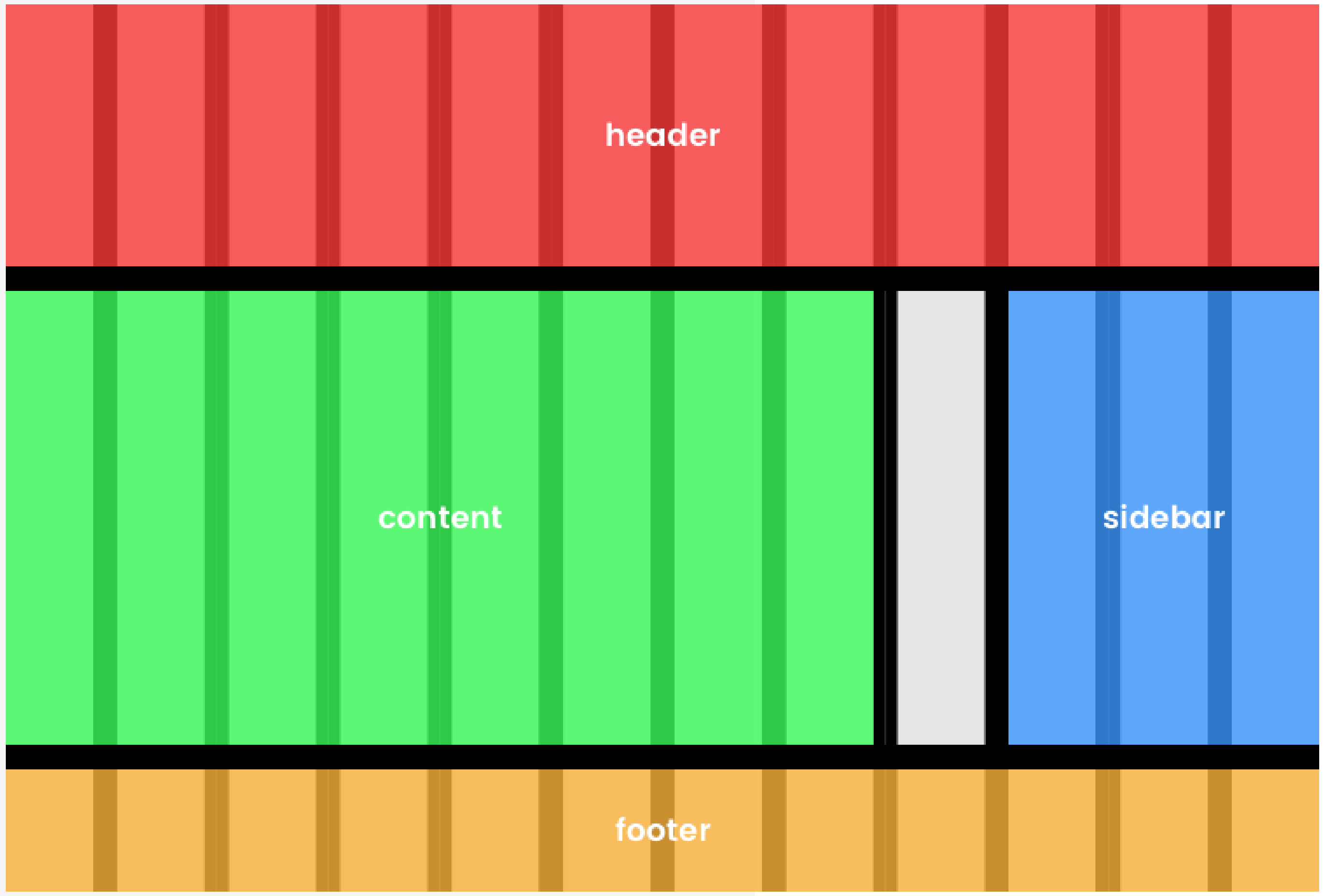
```css
.item-a {
    grid-column: 2 / 4;
    grid-row: 1 / 2;
}

.item-b {
    grid-column: 1 / 2;
    grid-row: 1 / 3;
}

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



BUILDING A LAYOUT

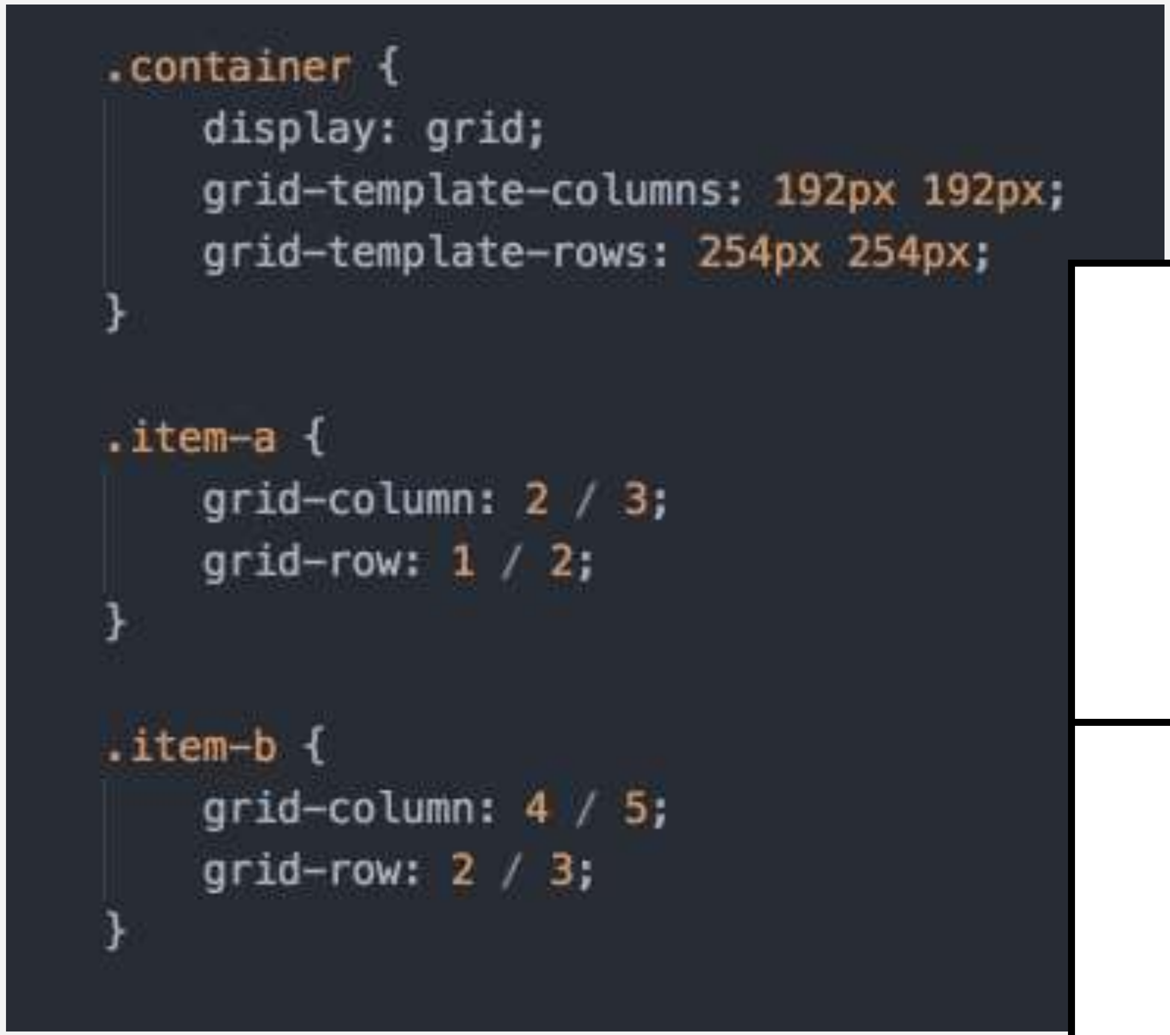


```
<div class="grid-container">
    <header></header>
    <article></article>
    <aside></aside>
    <footer></footer>
</div>

.grid-container {
    display: grid;
    grid-template-columns: repeat(12, 50px);
    grid-template-rows: 150px auto 70px;
}

header {
    grid-column: 1 / 13;
    grid-row: 1 / 2;
}

article {
    grid-column: 1 / 9;
    grid-row: 2 / 3;
}

aside {
    grid-column: 10 / 13;
    grid-row: 2 / 3;
}

footer {
    grid-column: 1 / 13;
    grid-row: 3 / 4;
}
```

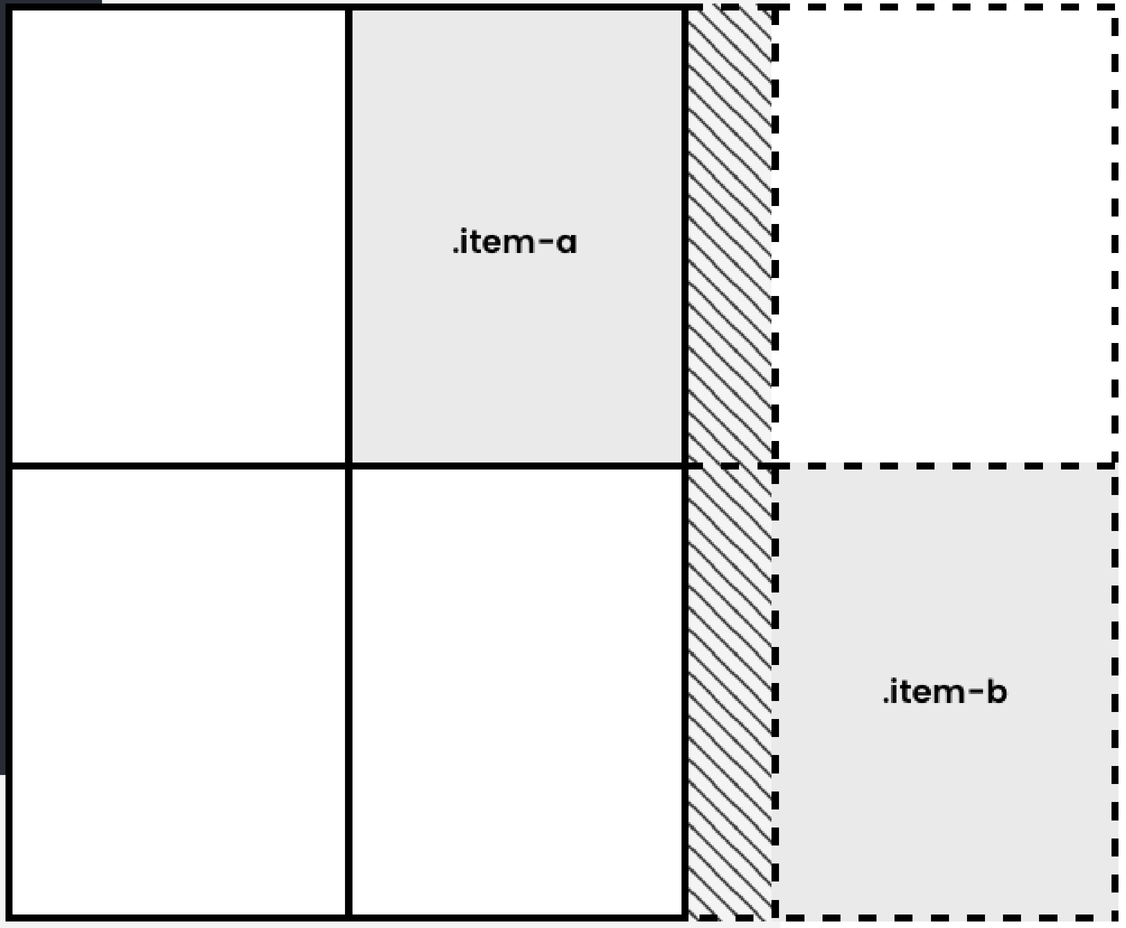


GRID AUTO ROWS & COLUMNS

Set row or column sizes for those not explicitly set.

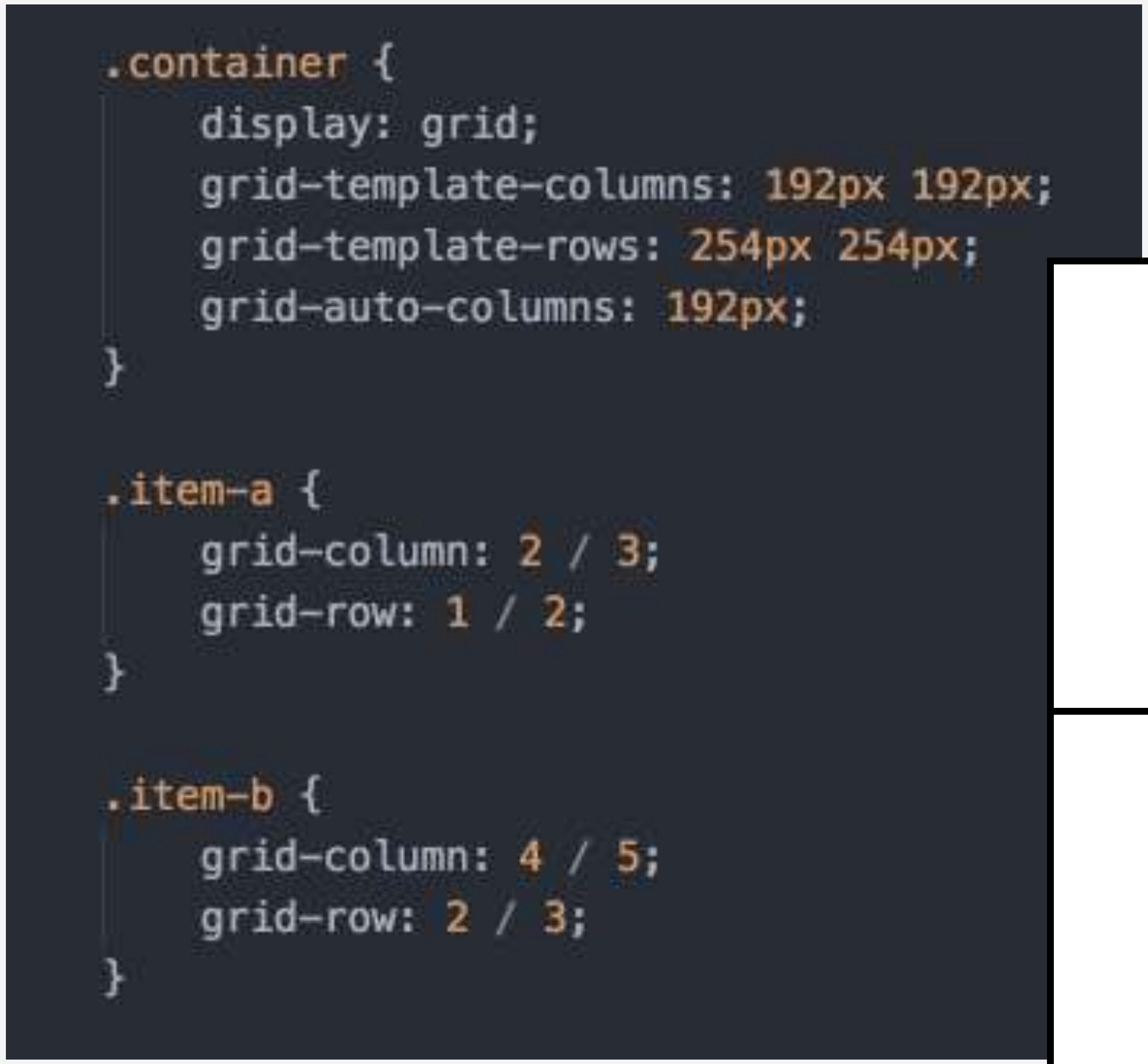


```
.container {
    display: grid;
    grid-template-columns: 192px 192px;
    grid-template-rows: 254px 254px;
}

.item-a {
    grid-column: 2 / 3;
    grid-row: 1 / 2;
}

.item-b {
    grid-column: 4 / 5;
    grid-row: 2 / 3;
}
```

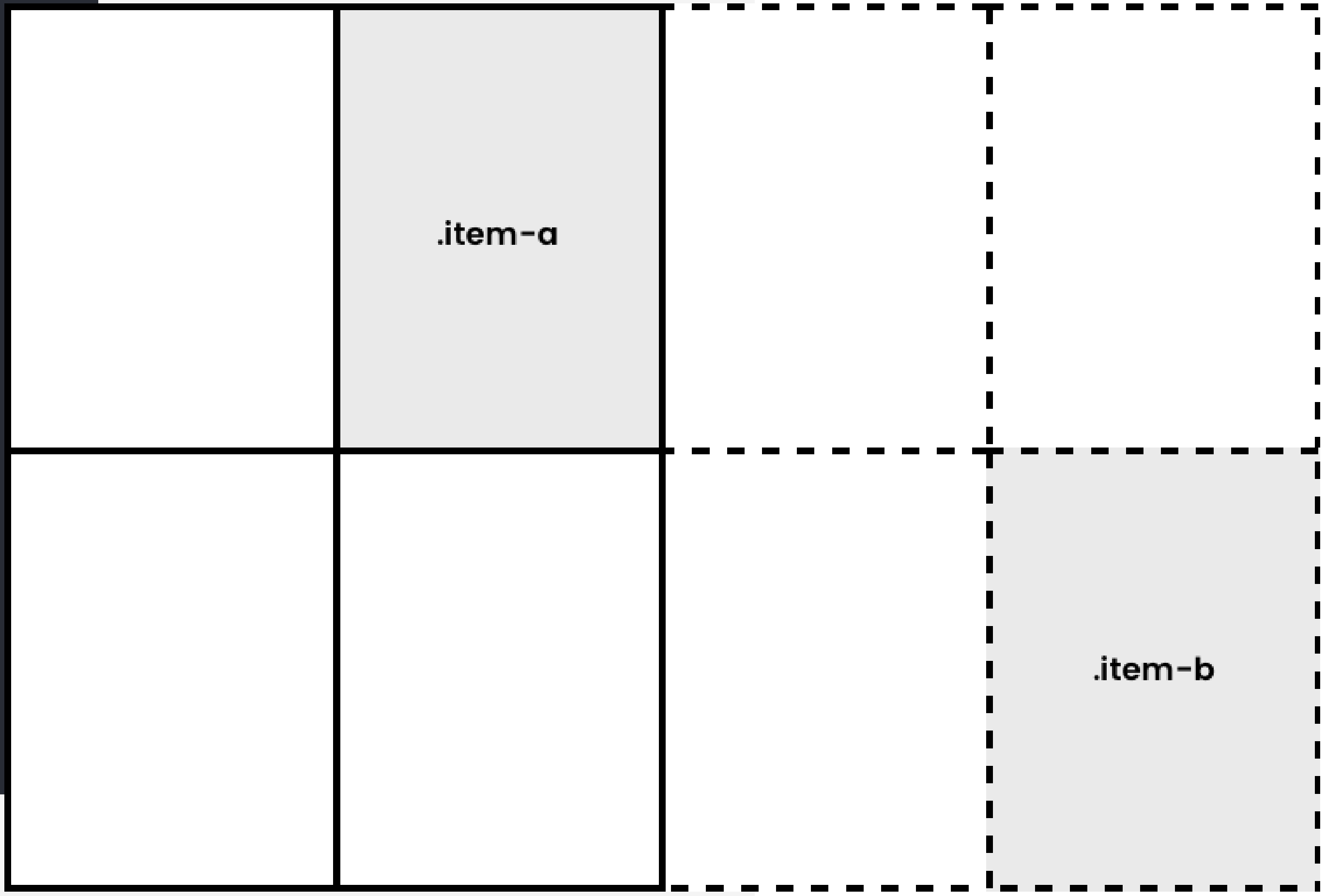


GRID AUTO
ROWS & COLUMNS

Set row or column sizes for those not explicitly set.



```
.container {
    display: grid;
    grid-template-columns: 192px 192px;
    grid-template-rows: 254px 254px;
    grid-auto-columns: 192px;
}

.item-a {
    grid-column: 2 / 3;
    grid-row: 1 / 2;
}

.item-b {
    grid-column: 4 / 5;
    grid-row: 2 / 3;
}
```



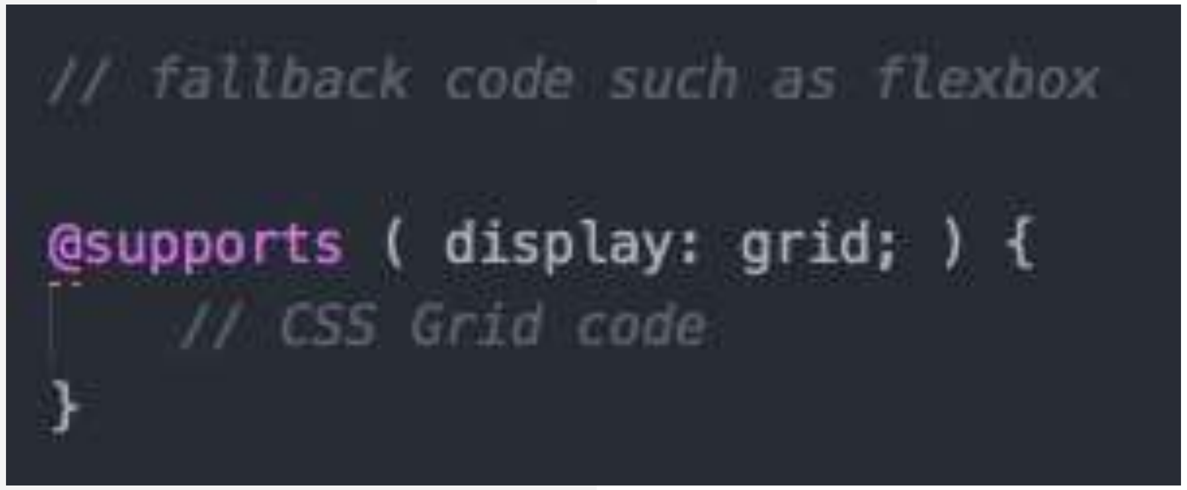
HOW TO FALLBACK

CSS Grid is well supported by most recent browsers.

Of course we still have to support the likes of IE 11.

Rather than polyfill the best bet is to degrade gracefully.

Example: https://preview.themes.pizza/zuul/blog/



```
// fallback code such as flexbox

@supports ( display: grid; ) {
    // CSS Grid code
}
```

LEARNING MORE

LET'S GET GRIDDY WITH IT
CSS GRID.
LEARN CSS GRID WITH WES BOS IN 25 PRETTY GOOD VIDEOS
FROM FUNDAMENTALS TO REAL WORLD IN ABOUT 4 HOURS.

CSSGrid.io by Wes Bos

CSS GRID RESOURCES

CSSGrid.io - Wes Bos

A Complete Guide to Grid - CSS Tricks

Get Ready for CSS Grid Layout - A Book Apart

11 Things I Learned Reading The CSS Grid Specification - Ohans Emmanuel

How to recreate Medium’s article layout with CSS Grid - Per Harald Borgen

CSS Grid Spec - W3C

Firefox Developer Edition

CSS GRID
QUESTIONS

Twitter: @jason_yingling

Email: jason@jasonyingling.me