



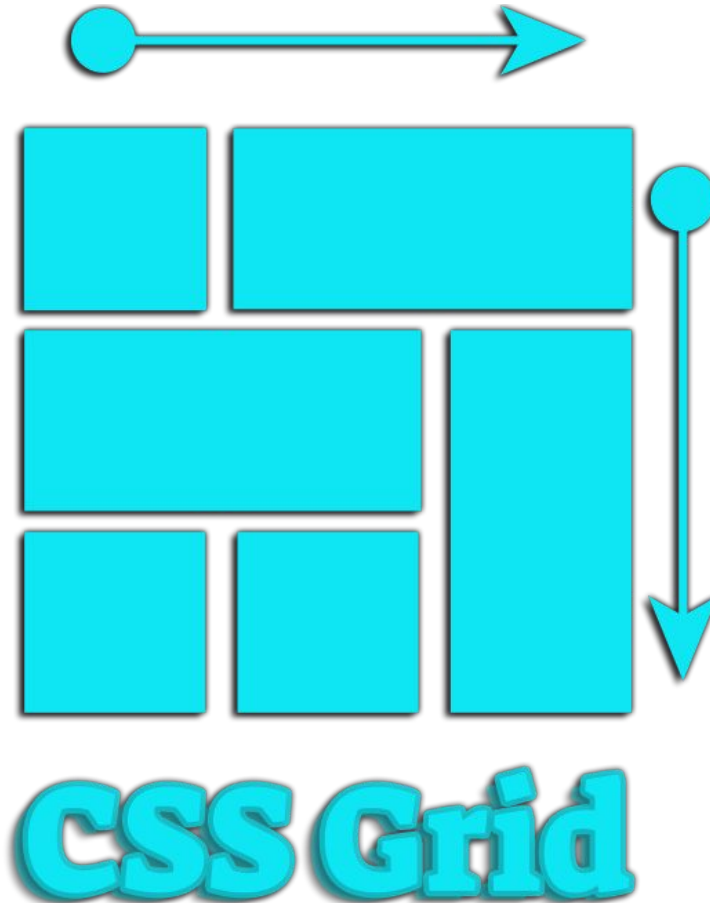
altexsoft
software r&d engineering

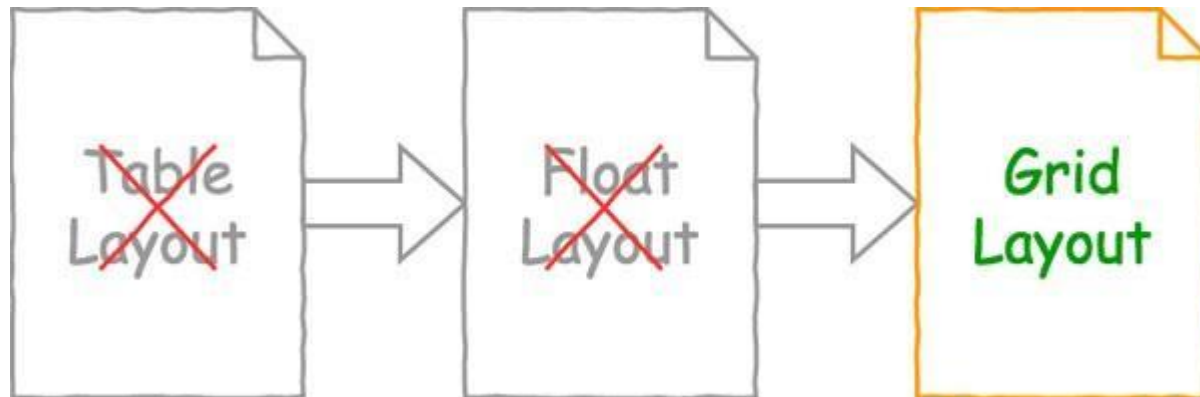
GRID'ОВЫЕ ИДЕИ

Maksym Bondar
UI Engineer at AltexSoft



CSS Grid Layout (aka "Grid")







Browser support

Can I use ? Settings

3 results found

CSS Grid Layout CR

Usage % of all users
Global 84.16% + 3.09% = 87.25%
unprefixed: 84.16%

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors. Includes support for all `grid-*` properties and the `fr` unit.

Current aligned | Usage relative | Date relative | Show all | ?

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49 ¹						
			63						
			67		10.3				
		61	68		11.2				4
11 ²	17	62	69	11.1	11.4	all	67	11.8	7.2
	18	63	70	12	12				
		64	71	TP					
			72						

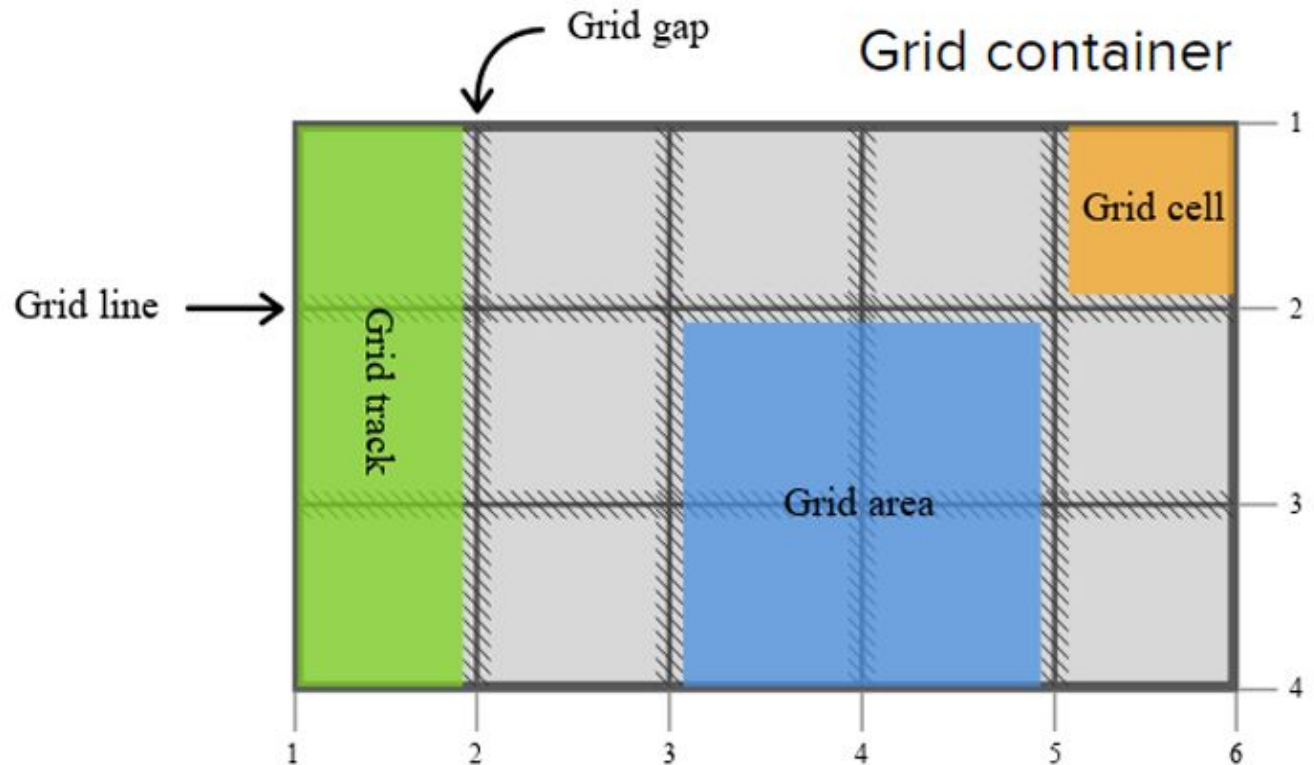
Notes | Known issues (2) | Resources (13) | Feedback

¹ Enabled in Chrome through the "experimental Web Platform features" flag in `chrome://flags`

² Partial support in IE refers to supporting an *older version* of the specification.



Concepts and Terminology

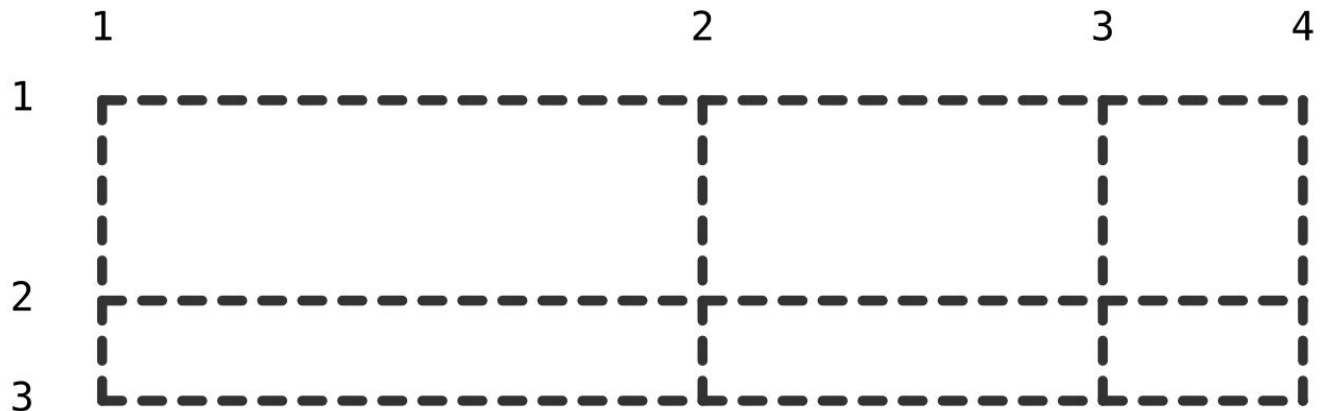




Grid Lines

```
<div class="grid"></div>
```

```
.grid{  
  display: grid;  
  
  grid-template-columns: 300px 200px 100px;  
  grid-template-rows: 100px 50px;  
}
```





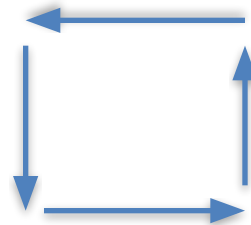
Grid Lines

Properties for the Grid Items:

- *Grid-column-start*
- *Grid-column-end*
- *Grid-row-start*
- *Grid-row-end*

Shorthands:

- *Grid-column: <grid-column-start> <grid-column-end>*
- *Grid-row: <grid-row-start> <grid-row-end>*
- *Grid-area: <row-start> <column-start> <row-end> <column-end>*

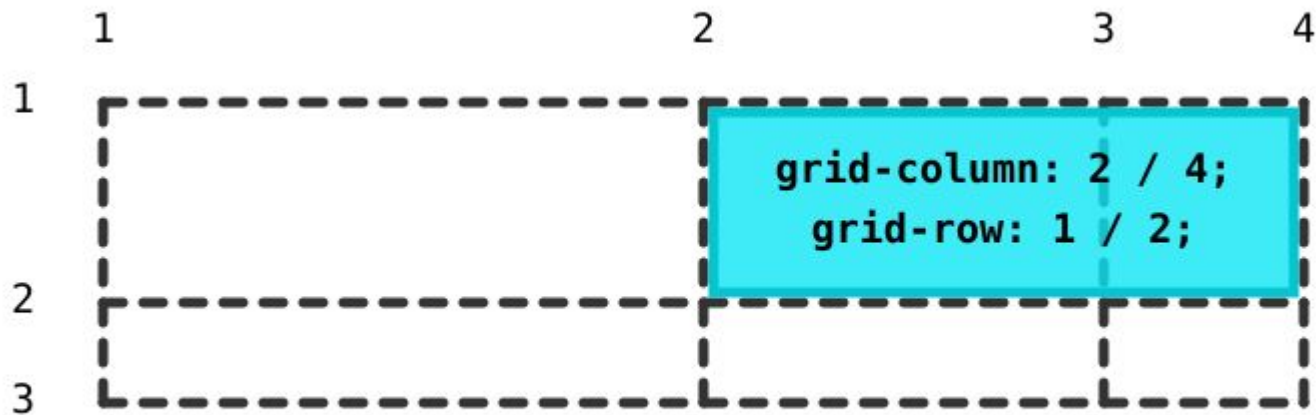




Grid Lines

```
<div class="grid">  
  <div class="grid-item">Grid Item</div>  
</div>
```

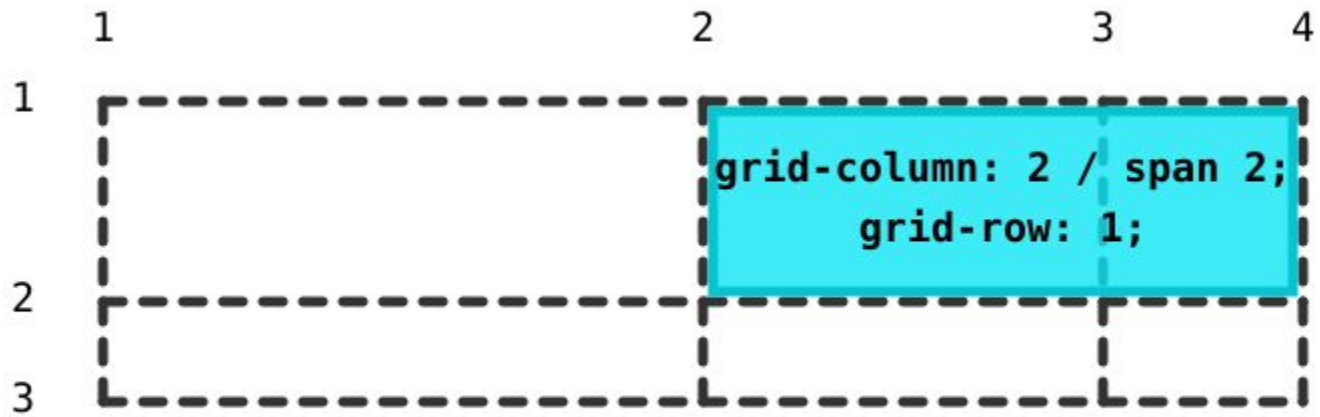
```
.grid-item{  
  grid-column: 2 / 4;  
  grid-row: 1 / 2;  
}  
  
.grid-item{  
  grid-area: 1 / 2 / 2 / 4;  
}
```





Grid Lines

*Cell Coverage ...key word *

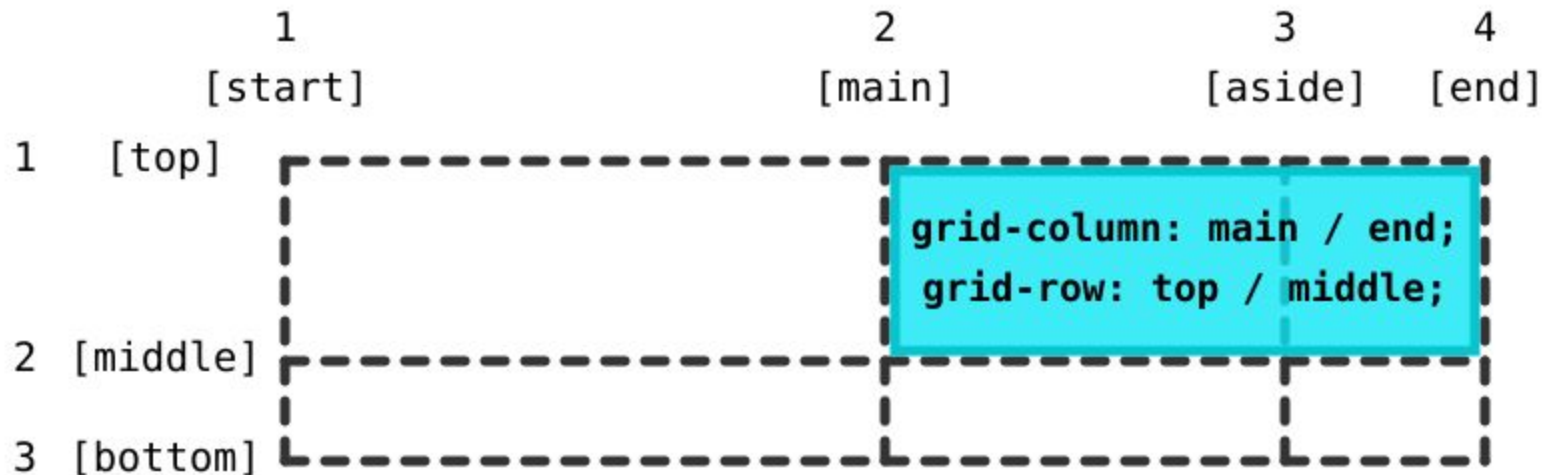


Grid Lines

Naming lines when defining a grid

```
.grid{
  display: grid;

  grid-template-columns: [start] 300px [main] 200px [aside] 100px [end];
  grid-template-rows: [top] 100px [middle] 50px [bottom];">
}
.grid-item{
  grid-column: main / end;
  grid-row: top / middle;
}
```

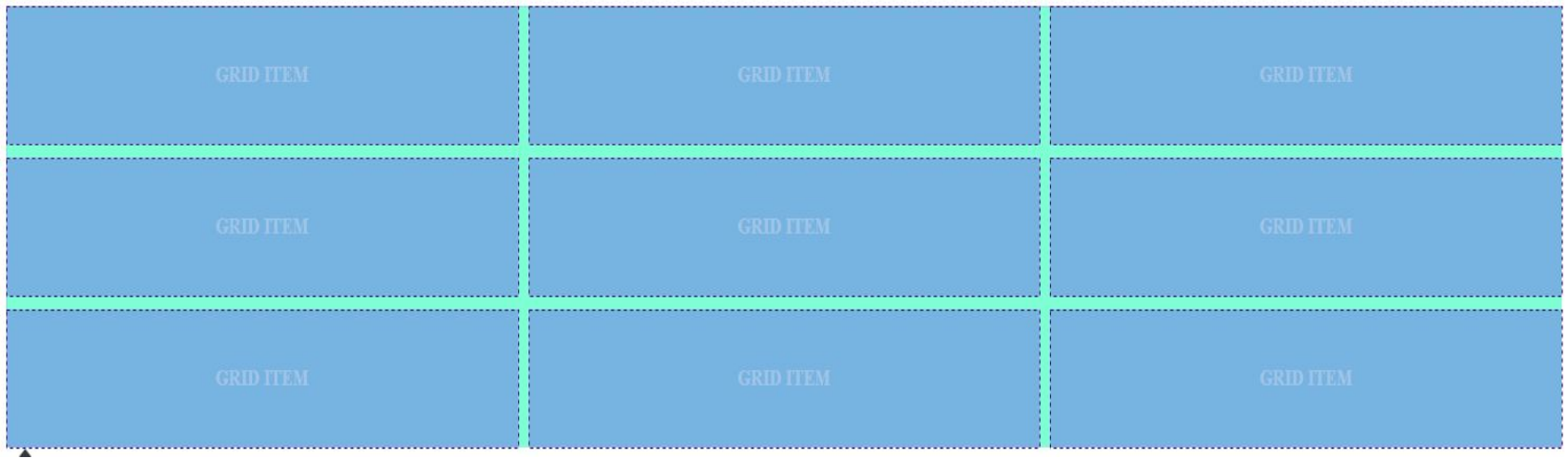




Grid Track

New Fr CSS length (Fraction)

```
grid-template-columns: 1fr 1fr 1fr;  
grid-template-rows: 100px 100px 100px;  
Repeat();  
grid-template-columns: repeat(3, 1fr);
```





Grid Track

Function minmax();

minmax(min,max);

- **Static length values**

```
grid-template-columns: minmax(100px, 200px) 200px 200px;
```

- **Dynamic length values**

```
grid-template-columns: minmax(50%, 80%) 1fr 1fr;
```

- **The min-content, max-content**

```
grid-template-columns: minmax(min-content, max-content) 1fr 1fr;
```



Grid Track

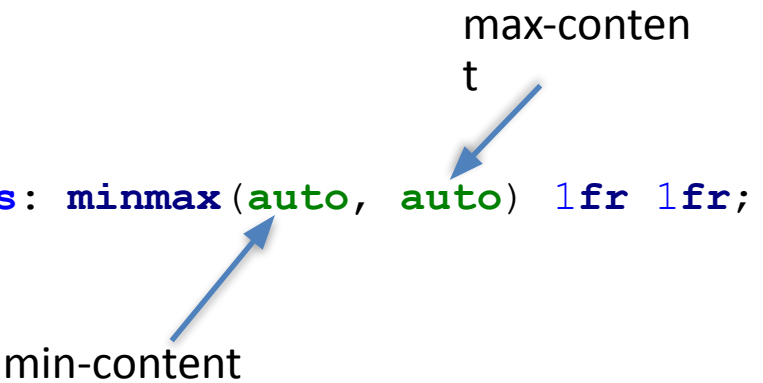
Function minmax();

- **Auto keyword**

```
grid-template-columns: minmax(auto, auto) 1fr 1fr;
```

min-content

max-content



[DEMO](#)



Responsive GRID without media queries

- *Auto-fill*

```
grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
```

- *Auto-fit*

```
grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
```

[DEMO](#)



Grid Area

- *Grid-area*

```
.grid-item{  
    grid-area: <name>;  
}
```

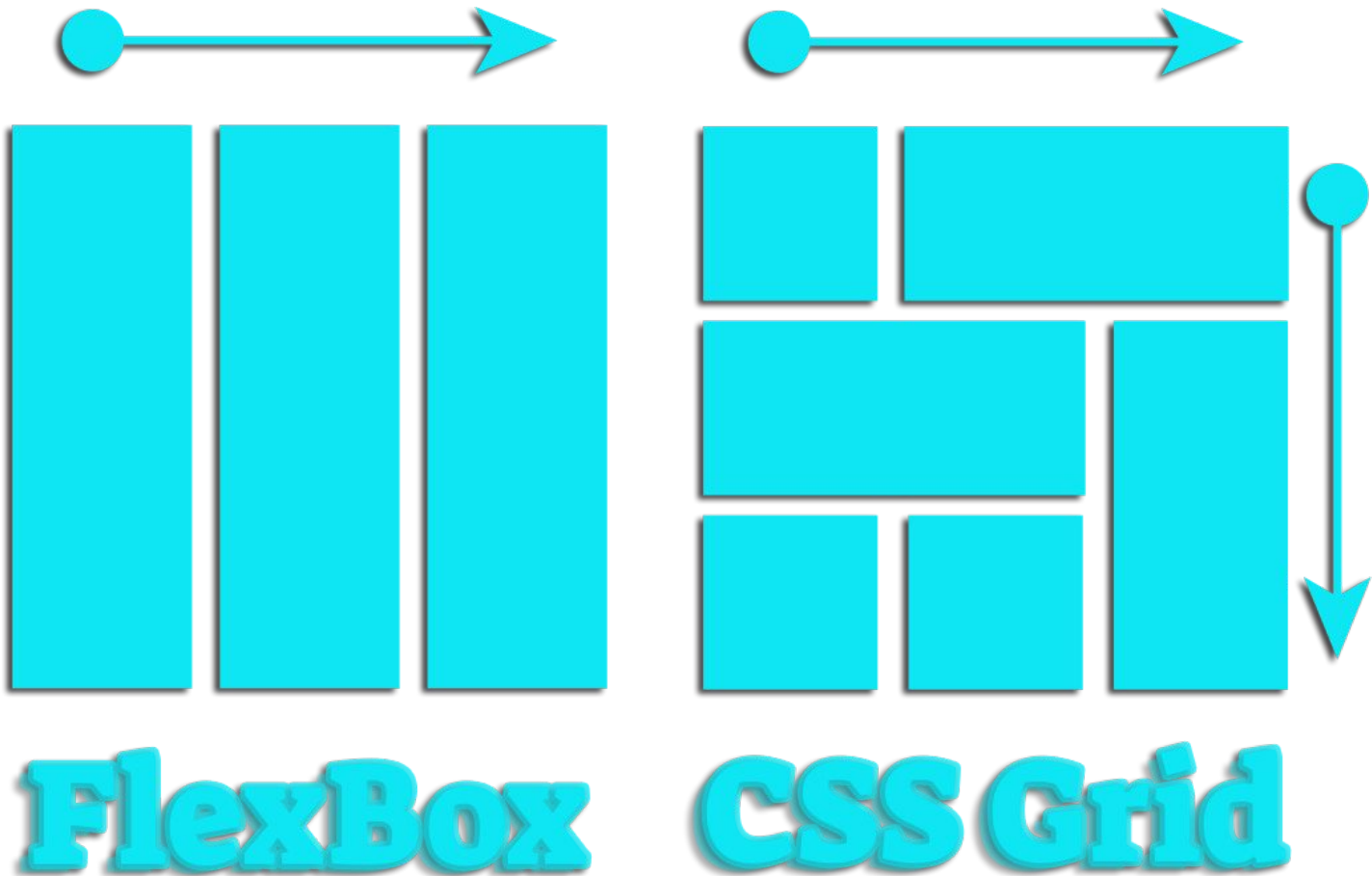
- *Grid-template-areas*

```
grid-template-areas: "header header header header header"  
                    ". . . . ."  
                    "menu . content . sidebar"  
                    ". . . . ."  
                    "footer footer footer footer footer";
```

[DEMO](#)



One-dimensional vs Two-dimensional



[DEMO](#)



Summary

Advantages:

- *Provide a convenient mechanism for placing content on a virtual grid*
- *Clear semantic structure*
- *Code support*

Disadvantages

- *Subgrid doesn't work properly*



Thank you for attention



Questions?

