

CSS GRID + WRITING- MODE =

By [Chen Hui Jing](#) / [@hj_chen](#)

WHAT IS CSS GRID?

Grid Layout is a new layout model for CSS that has powerful abilities to control the sizing and positioning of boxes and their contents.

Unlike Flexible Box Layout, which is single-axis-oriented, Grid Layout is optimized for 2-dimensional layouts.

Source: [CSS Grid Layout Module Level 1](#)

WHEN IS CSS GRID SUPPORTED?

Gecko: intent to ship in Firefox 52 (7 March, 2017)

Blink: intent to ship in Chrome 57 (14 March, 2017)

Webkit: ships in Safari 10.1 (release date March/April-ish)

Edge: on the backlog with High priority 🙋

ENABLE CSS GRID LAYOUT



CHROME

Enable the flag called **Enable experimental Web Platform features** going to:
`chrome://flags/#enable-experimental-web-platform-features`



FIREFOX

Enable the flag called **layout.css.grid.enabled** going to:
`about:config`



IE / EDGE

Enabled by default since IE10 (old syntax).
Prefixed -ms



OPERA

Enable the flag called **Enable experimental Web Platform features** going to:
`opera://flags/#enable-experimental-web-platform-features`



WEBKIT

In Safari Technology Preview and WebKit Nightly Builds you can enable the experimental feature called **CSS Grid** using the menu: *Develop > Experimental Features > CSS Grid*



EPIPHANY

Set the environment variable
WEBKITGTK_EXPERIMENTAL_FEATURES:

```
$ export WEBKITGTK_EXPERIMENTAL_FEATURES
```

Source: [Igalia](#)

SO I BUILT SOMETHING THE OTHER DAY...



WHAT IS CSS WRITING-MODE?

is fun!
Vertical text



Or make it read from
right-to-left.



right.
on its side from left-to-
browser literally turn
You can make your

WHAT IS CSS WRITING-MODE?

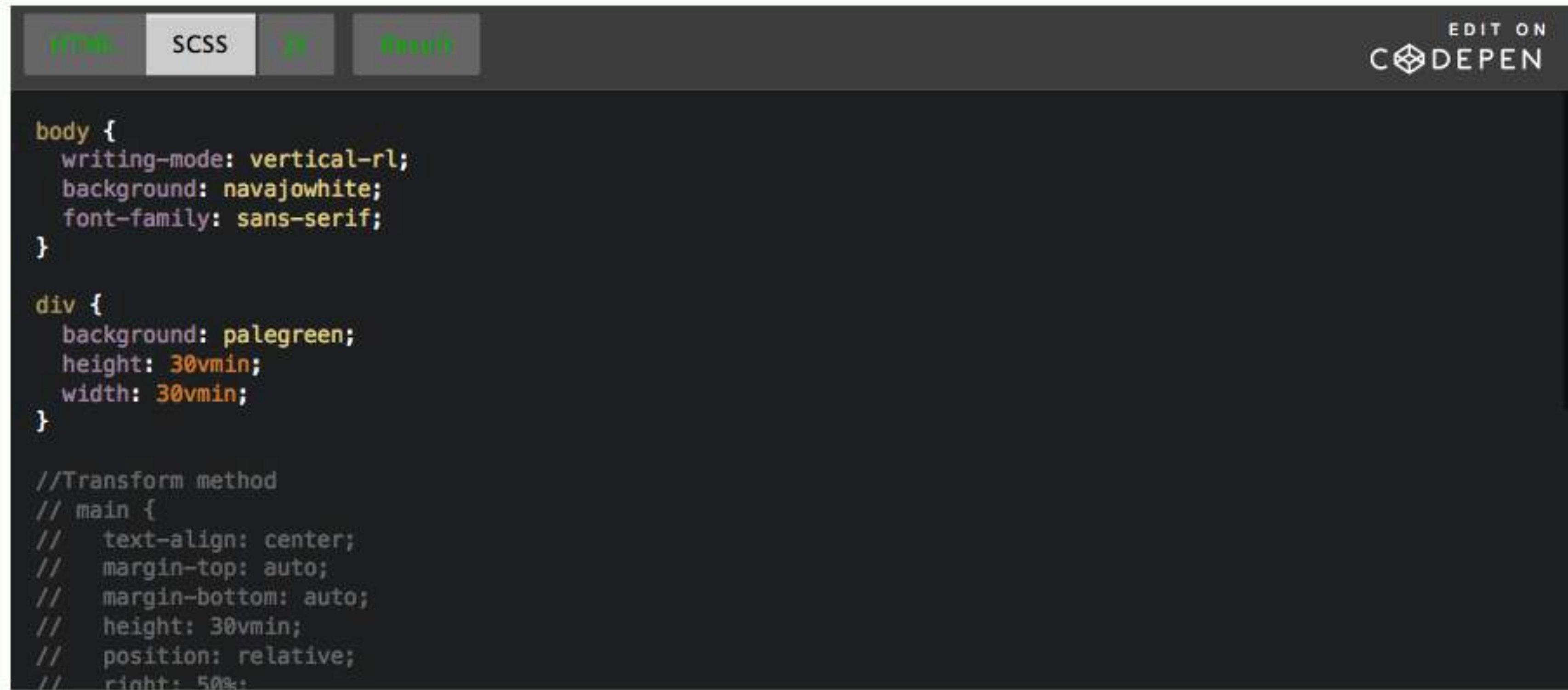


WHAT IS CSS WRITING-MODE?

CSS Writing Modes Level 3 defines CSS features to support for various international writing modes, such as left-to-right (e.g. Latin or Indic), right-to-left (e.g. Hebrew or Arabic), bidirectional (e.g. mixed Latin and Arabic) and vertical (e.g. Asian scripts).

DIFFERENT BUT SAME

Centring things is no fun



The image shows a CodePen editor interface. At the top, there are tabs for 'HTML', 'SCSS', 'JS', and 'Result'. The 'SCSS' tab is selected. In the top right corner, there is a link that says 'EDIT ON CODEPEN'. The main area contains the following SCSS code:

```
body {  
  writing-mode: vertical-rl;  
  background: navajowhite;  
  font-family: sans-serif;  
}  
  
div {  
  background: palegreen;  
  height: 30vmin;  
  width: 30vmin;  
}  
  
//Transform method  
// main {  
//   text-align: center;  
//   margin-top: auto;  
//   margin-bottom: auto;  
//   height: 30vmin;  
//   position: relative;  
//   right: 50%;
```

TODAY WE LEARNT

- Centre along vertical axis using:

```
margin-top: auto;  
margin-bottom: auto;
```

- Sideways equivalent of vertical centring is 🍌
- Flexbox no like writing-mode on Firefox (for now)
- Browsers are very inconsistent at the moment 😞

CAN I USE? CAN.

Browser support: CSS Writing Mode

CSS writing-mode property [↗](#)

Property to define whether lines of text are laid out horizontally or vertically and the direction in which blocks progress.

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
9	12	50	55	9	41	9.2		4.4	
10	13	51	56	9.1	42	9.3		4.4.4	
11	14	52	57	10	43	10.2	all	53	56
	15	53	58	10.1	44				

✓ ✗ Partial Support Prefixed

Global: 89.31% + 4.95% = 94.26%

Data from caniuse.com | Embed from caniuse.bitsofco.de

CSS LOGICAL PROPERTIES LEVEL 1

Introduces logical properties and values that provide the author with the ability to control layout through logical, rather than physical, direction and dimension mappings.

Source: [CSSWG](#)




















start and end vs. left and right




CAN I USE? CANNOT.

Browser support: CSS Logical Properties

CSS Logical Properties

Use start/end properties that depend on LTR or RTL writing direction instead of left/right

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
9	12	50	55 	9 	41 	9.2 		4.4 	
10	13	51	56 	9.1 	42 	9.3 		4.4.4 	
11	14	52	57 	10 	43 	10.2 	all	53 	56 
	15	53	58 	10.1 	44 				

  Partial Support  Prefixed

Global: 6.11% + 82.16% = 88.27%

Data from caniuse.com | Embed from caniuse.bitsofco.de

Disclaimer: If you're reading this from the future, maybe can.

BUT, WE HAVE CSS GRID

Disclaimer: you cannot learn CSS grid overnight.

bù shì yī fān hán chè gǔ zěn de méi huā pū bí xiāng
不是一番寒彻骨 怎得梅花扑鼻香

It'd be nice if you read [the specification](#) 🧐

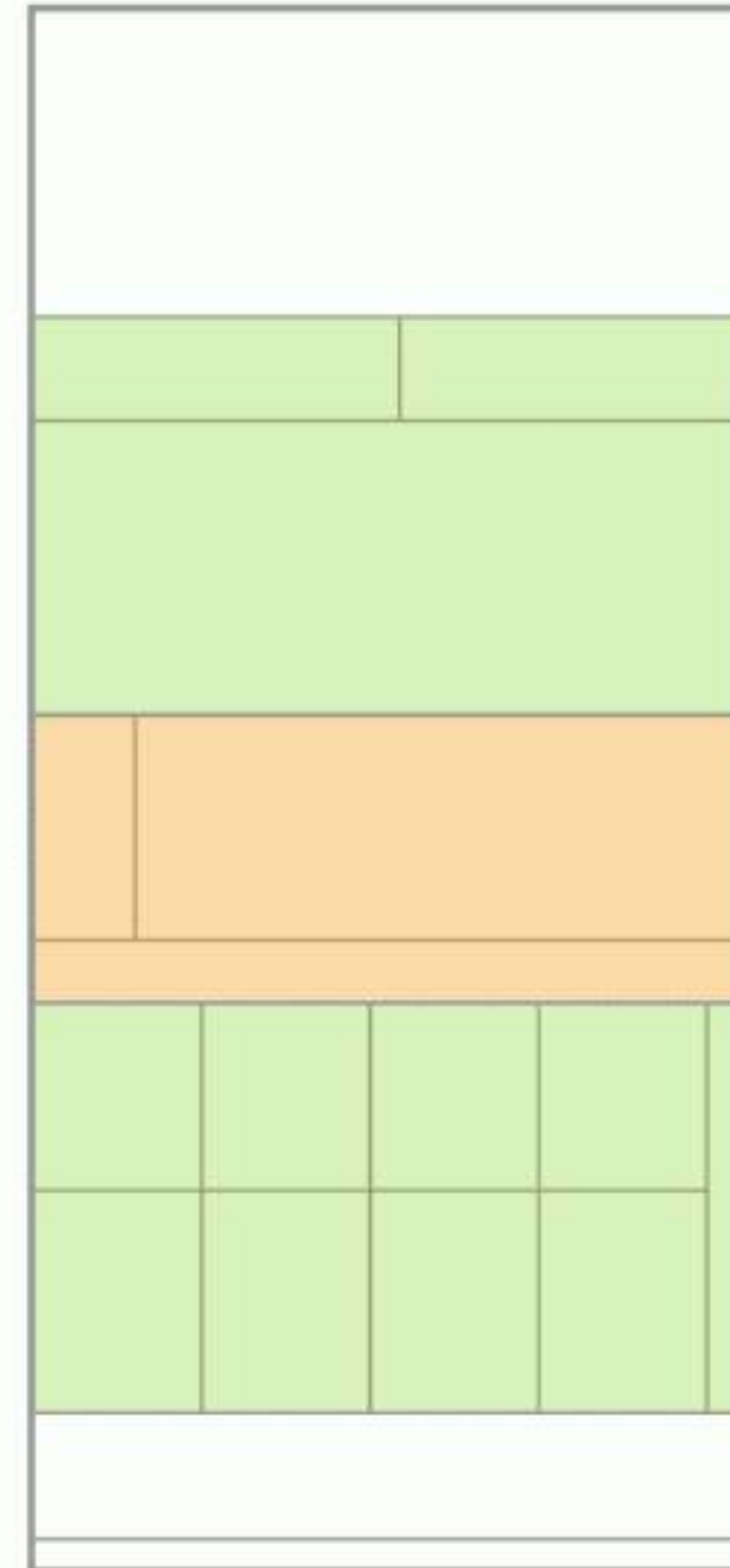
Grid container

- display: grid;
- grid-template-columns
- grid-template-rows
- grid-column-gap
- grid-row-gap

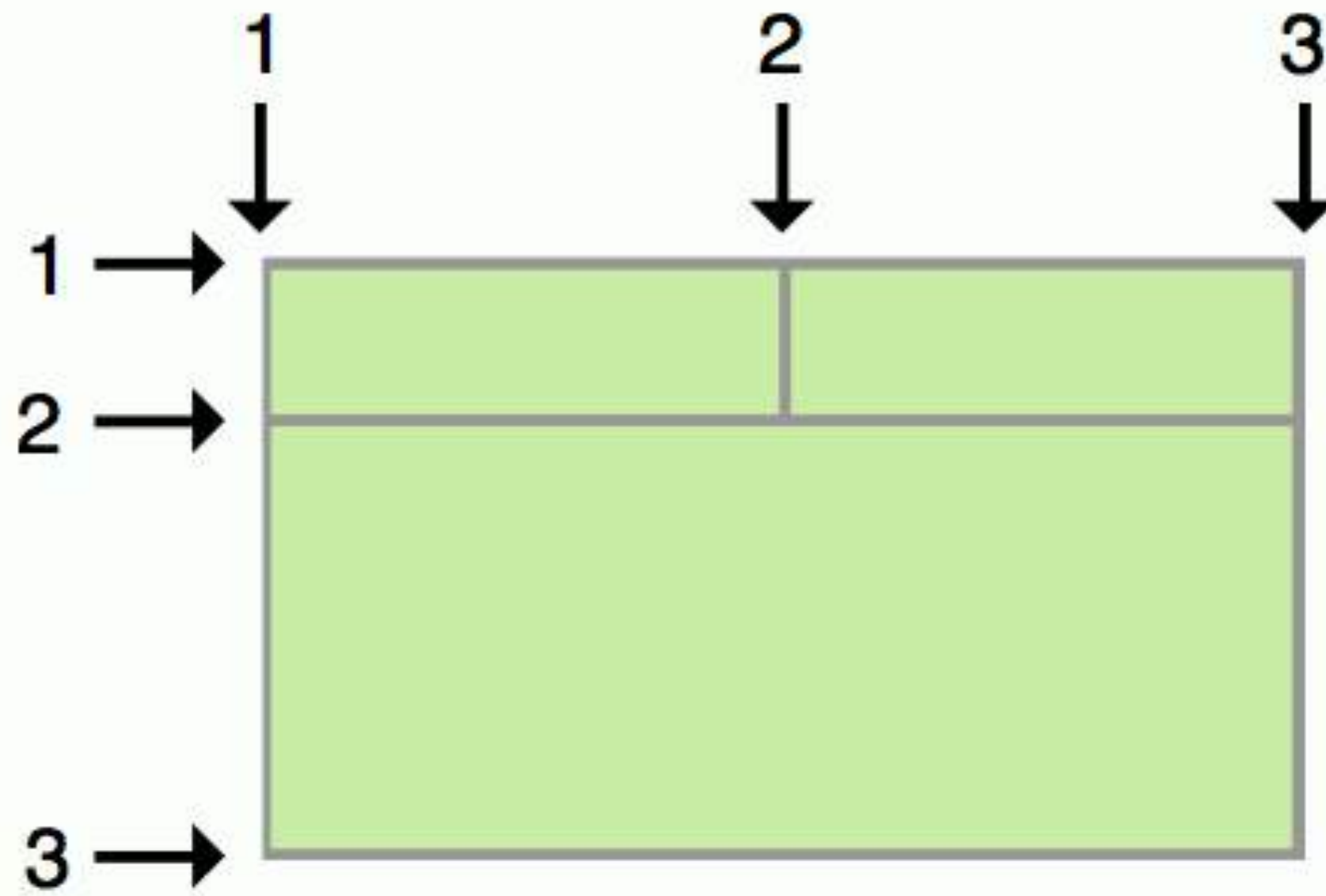
Grid item

- grid-row
- grid-column
- align-self
- justify-self

SWITCHING TO GRID



SIMPLE GRID



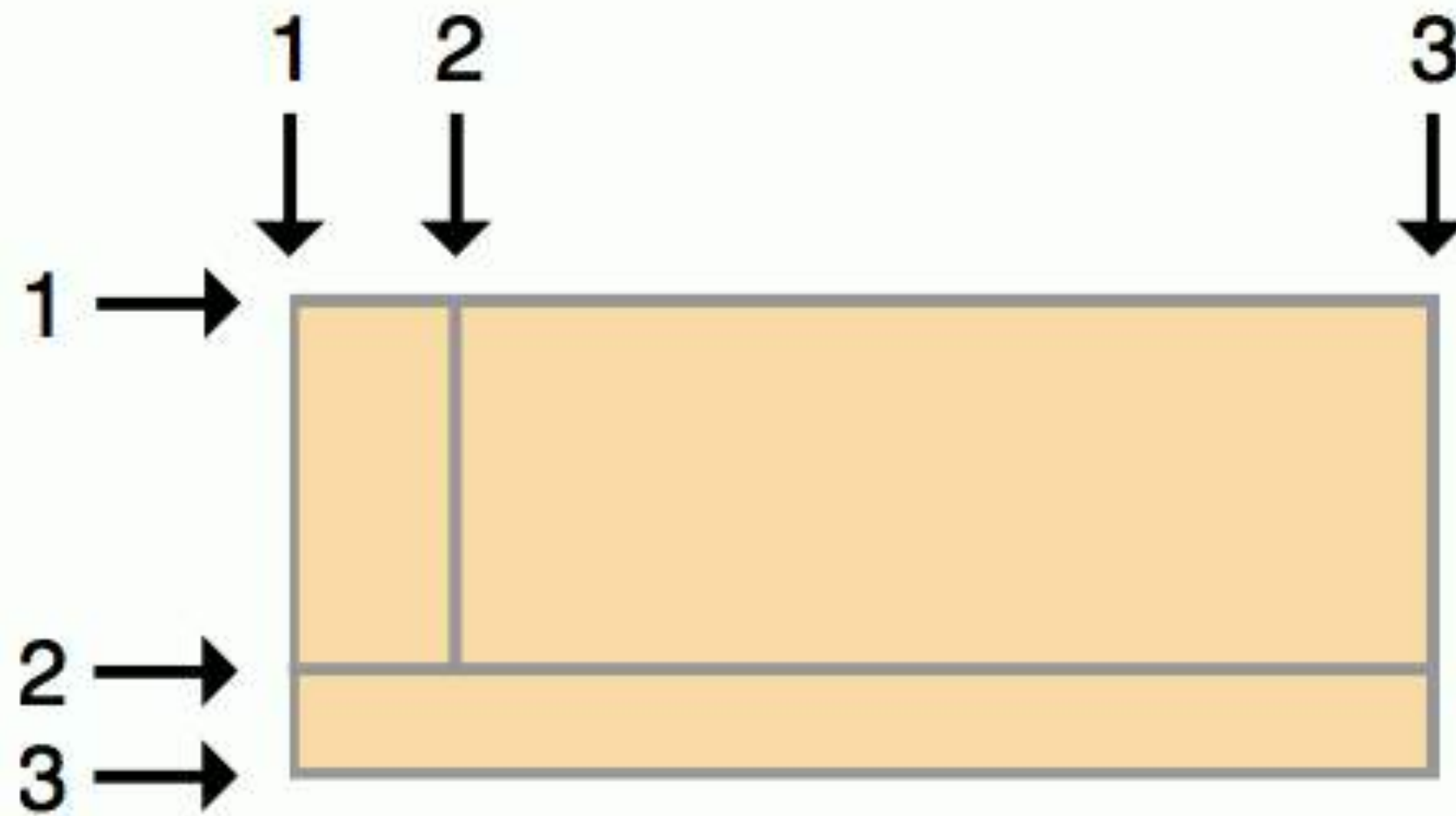
```
.container {  
  display: grid;  
  grid-column-gap: 1em;  
  grid-row-gap: 1em;  
}
```

```
.top-left {  
  grid-row: 1 / 2;  
  grid-column: 1 / 2;  
  align-self: center;  
  justify-self: end;  
}
```

```
.top-right {  
  grid-row: 1 / 2;  
  grid-column: 2 / 3;  
  align-self: center;  
  justify-self: start;  
}
```


ANOTHER SIMPLE GRID

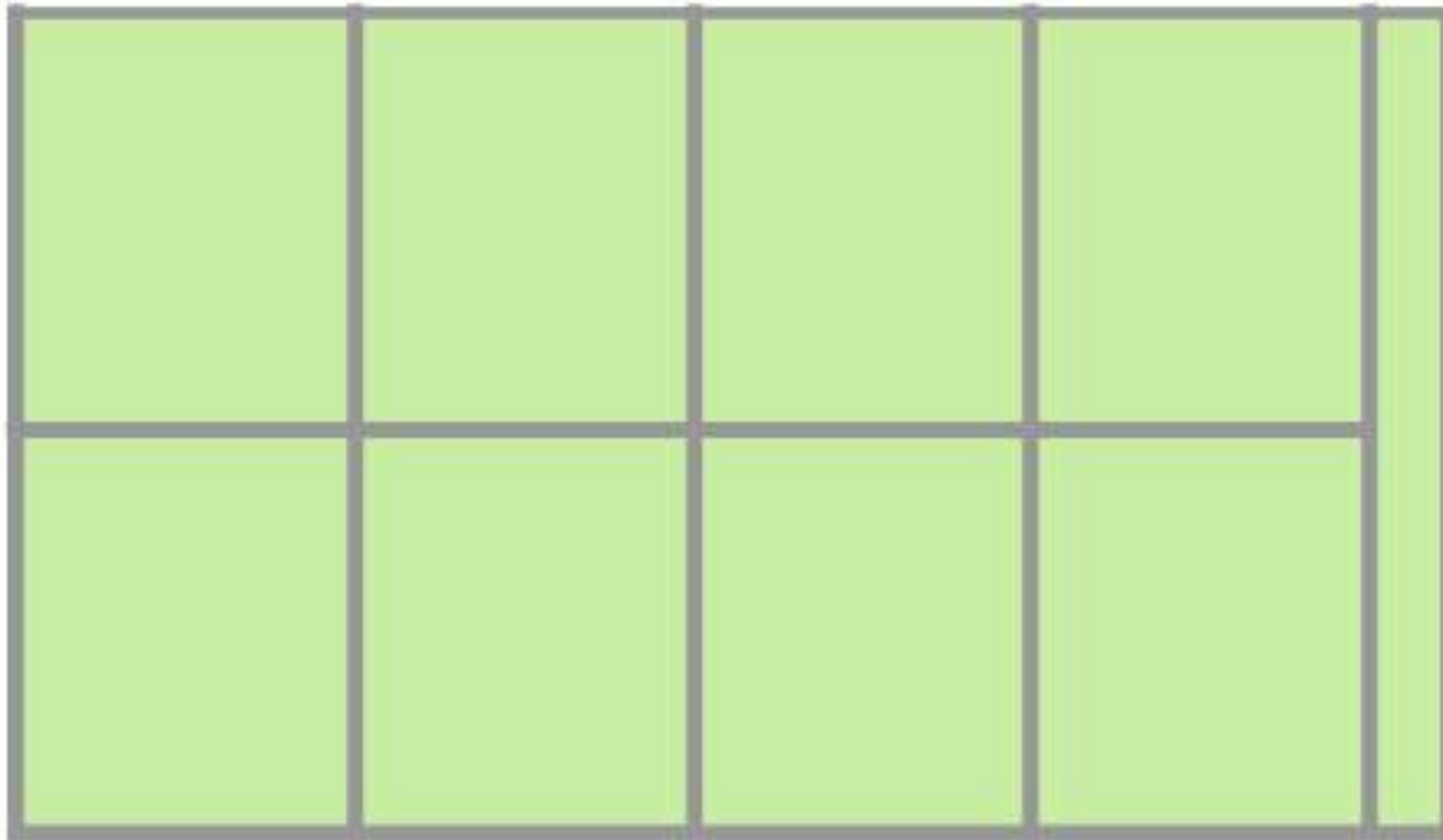
fr → represents a fraction of the free space in the grid container.



```
.container {  
  display: grid;  
  grid-template-columns: 3.6em 1fr;  
  grid-column-gap: 0.5em;  
  grid-row-gap: 1em;  
}  
  
.left {  
  grid-row: 1 / 2;  
  align-self: center;  
}  
  
.top-right {  
  grid-row: 1 / 2;  
  grid-column: 2 / 3;  
  align-self: center;  
  justify-self: start;  
}
```

MODERATE GRID

repeat() → represents a repeated fragment of the track list



```
.outer-container {  
  display: grid;  
  grid-template-columns: 1fr 2em;  
}  
  
.inner-container {  
  grid-column: 1 / 2;  
  grid-row: 1 / 2;  
  
  display: grid;  
  grid-column-gap: 0.5em;  
  grid-row-gap: 1em;  
  
  @media all and (max-width: 719px) {  
    grid-template-columns: repeat(2, 1fr);  
  }  
}
```


FURTHER READING

- [CSS Grid Layout Module Level 1](#)
- [Grid by Example](#) by Rachel Andrew
- [CSS Grid Layout Examples](#) by Igalia
- [A Complete Guide to Grid](#) by Chris House

THE END

 <http://www.chenhuijing.com>

 @hj_chen

 @hj_chen

 @huijing