

Discover Grid

Jacqueline Quintanilla

Twin Cities Code Camp

April 13, 2019

Discover Grid

Jacqueline Quintanilla
qjac.net

- Don't know what CSS Grid is?
- Curious about CSS Grid?
- Need to boost your grid skills?
- Not sure if grid is the right solution?

You're in the right room!

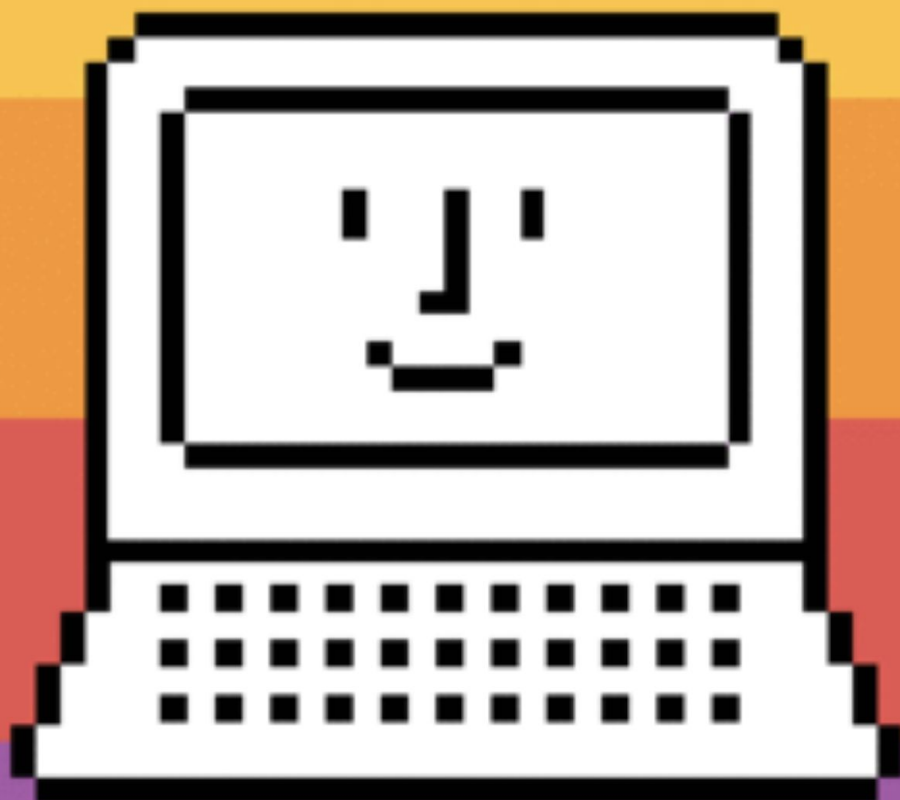
Some CSS knowledge is helpful, but not necessary.

Hello!

 qjac.net  [@_qjac](https://twitter.com/_qjac)    [@qjac](https://github.com/qjac)

Slides: tinyurl.com/discover-grid

- Frontend Developer
- Cat Lady
- Podcast Enthusiast
- Homebrewer
- Design Fanatic



CSS Grid to the rescue!

- Simple semantic markup!!
- Maintainability!!!
- 2-D and inherently flexible!!!!
- Center all the things!!!!

→ New Design
Possibilities

"Almost immediately, word spread of a Welshman who had built this incredible sounding sub bass system, and soon enough audiophiles began searching out these legendary subs that were built like no other."



Gibraltar

Introducing 212/SE, our most powerful and agile subwoofer, designed exclusively for larger systems and rooms to allow superior state of the art speakers to spring to full voice.

[View Product Details →](#)

THE EXPERIMENTAL LAYOUT LAB

OF JEN SIMMONS

FIREFOX GRID INSPECTOR
FIREFOX SHAPE PATH EDITOR
FIREFOX NIGHTLY

LAYOUT LAND VIDEOS
LAYOUT LAND WEBSITE
LEARN HOW TO USE CSS GRID
REVOLUTIONIZE YOUR PAGE: REAL
ART DIRECTION ON THE WEB
PROGRESSING OUR LAYOUTS
MODERN LAYOUTS: GETTING OUT
OF OUR RUTS
JENSIMMONS.COM
SHOP EXAMPLES

Follow @JENSIMMONS on Twitter for more as it happens.
Sign up for LAYOUT LAND, a place for us to show off new ideas for layout.

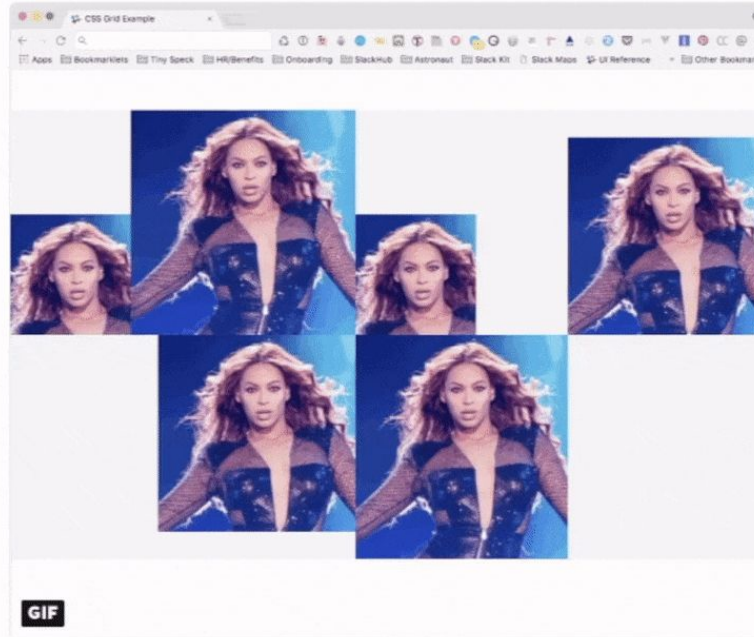
labs.jensimmons.com



Mina Markham
@MinaMarkham

Following

This is what happens when you ask me to talk about CSS Grid.
Just a lil' something for my @TheMixinSF demo tonight :)



6:37 PM - 11 Jul 2018

twitter.com/MinaMarkham/status/1017191293099728896

→ It's ready!!!!

CSS Grid Layout (level 1) 📄 - CR

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.

Usage % of all users ?
Global 89.42% + 2.56% = 91.98%
unprefixed: 89.42%

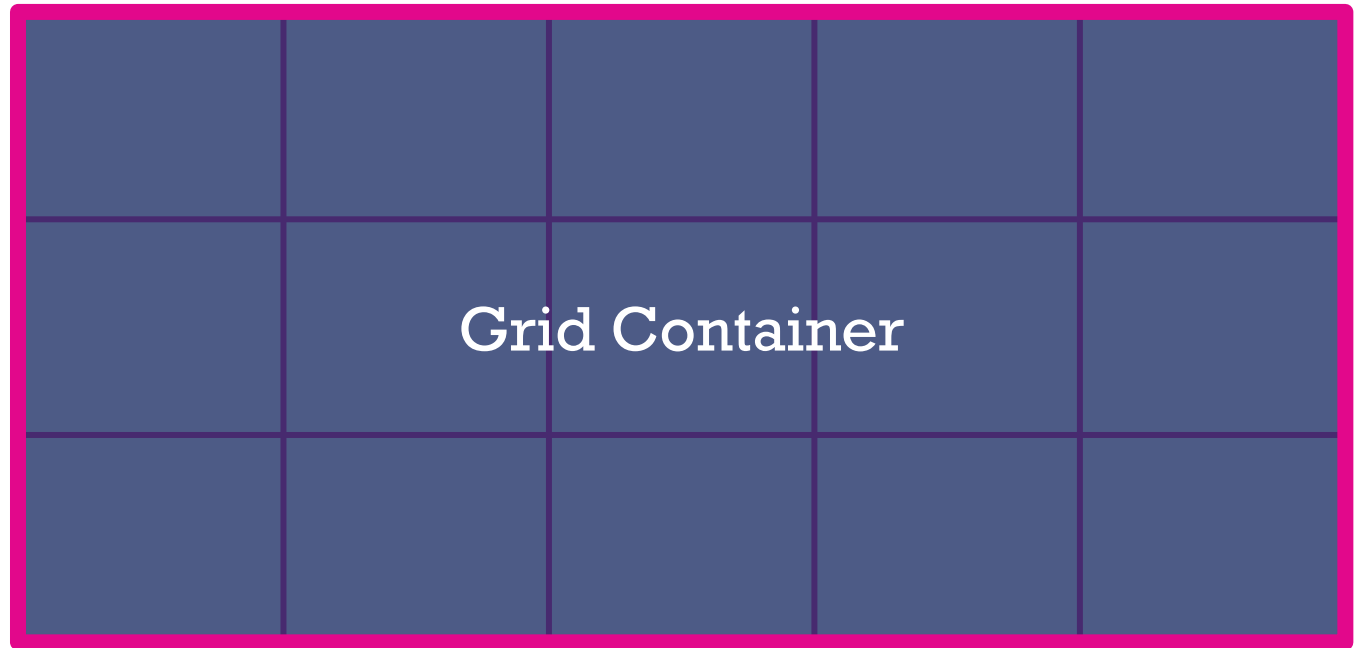
Current aligned	Usage relative	Date relative	Apply filters	Show all	?							
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile*	Chrome for Android	Firefox for Android
		2-39	4-28									
		³ 40-51	¹ 29-56		10-27							
6-9	² 12-15	⁴ 52-53	⁴ 57	3.1-10	¹ 28-43	3.2-10.2						
² 10	16-17	54-65	58-72	10.1-12	44-57	10.3-11.4		2.1-4.4.4	7	12-12.1		
² 11	18	66	73	12.1	58	12.1	all	67	10	46	71	64
		67-68	74-76	TP		12.2						

Yes!

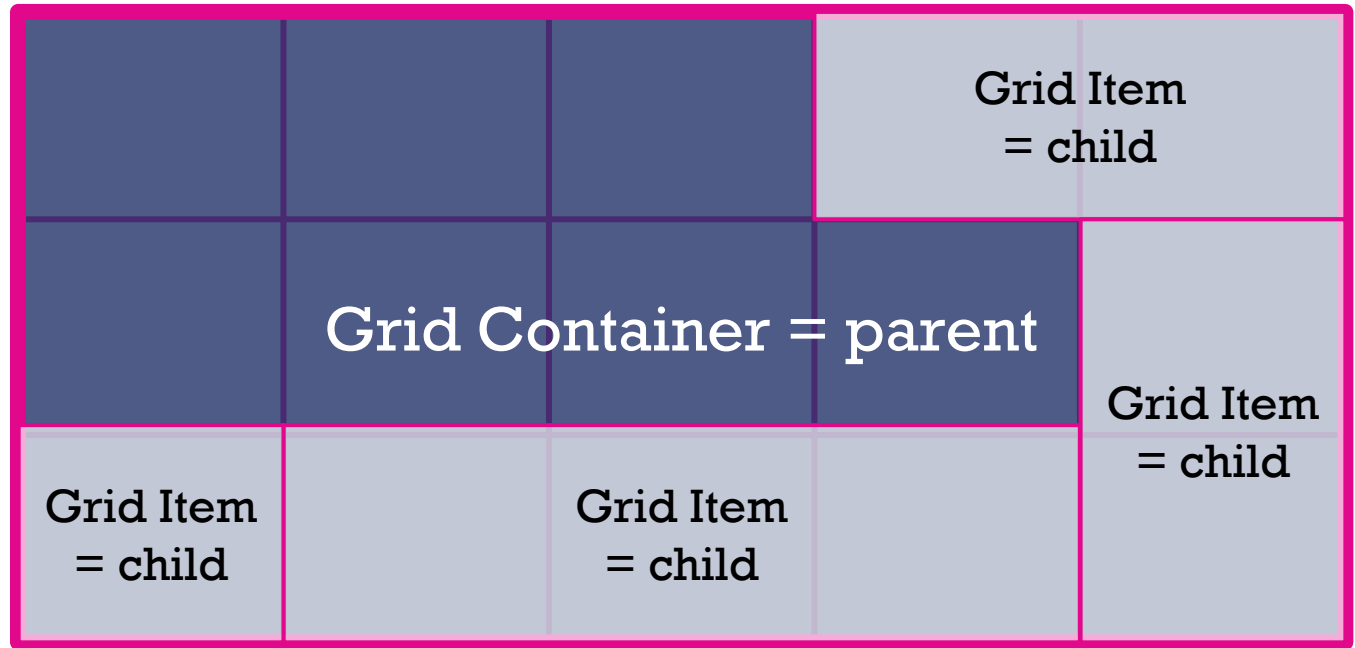
CSS Grid:

An overly brief primer

The Vocabulary



The Vocabulary



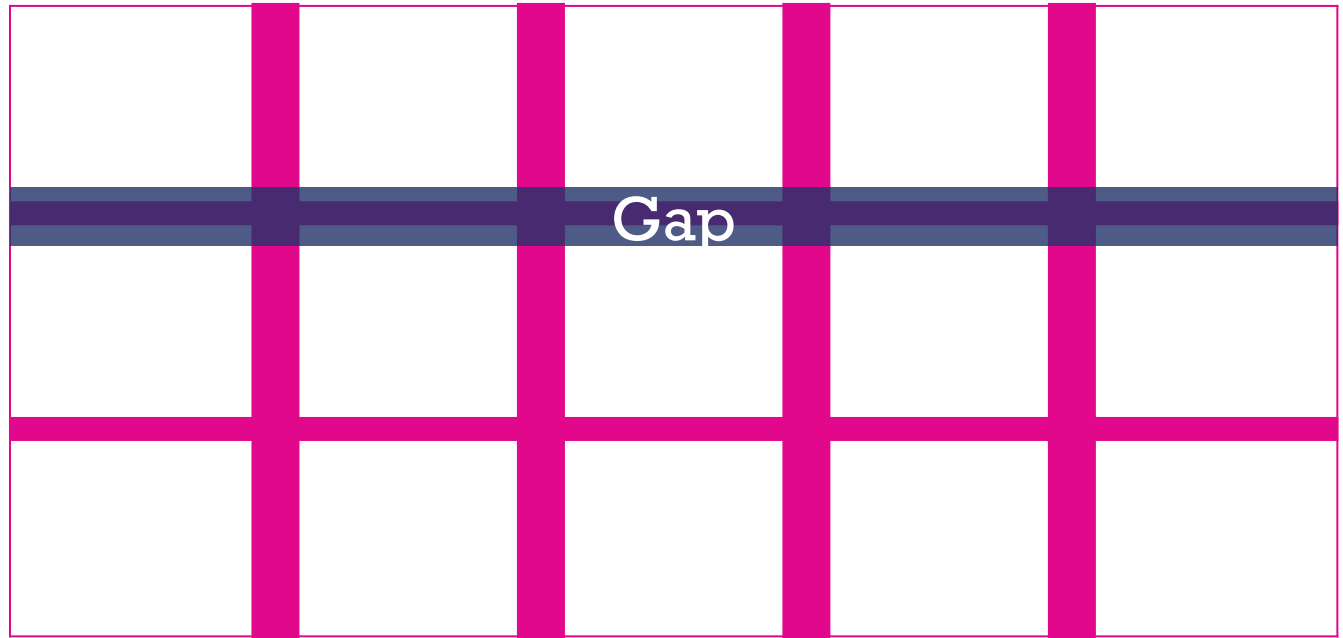
The Vocabulary

	Grid Track			
				Grid Track

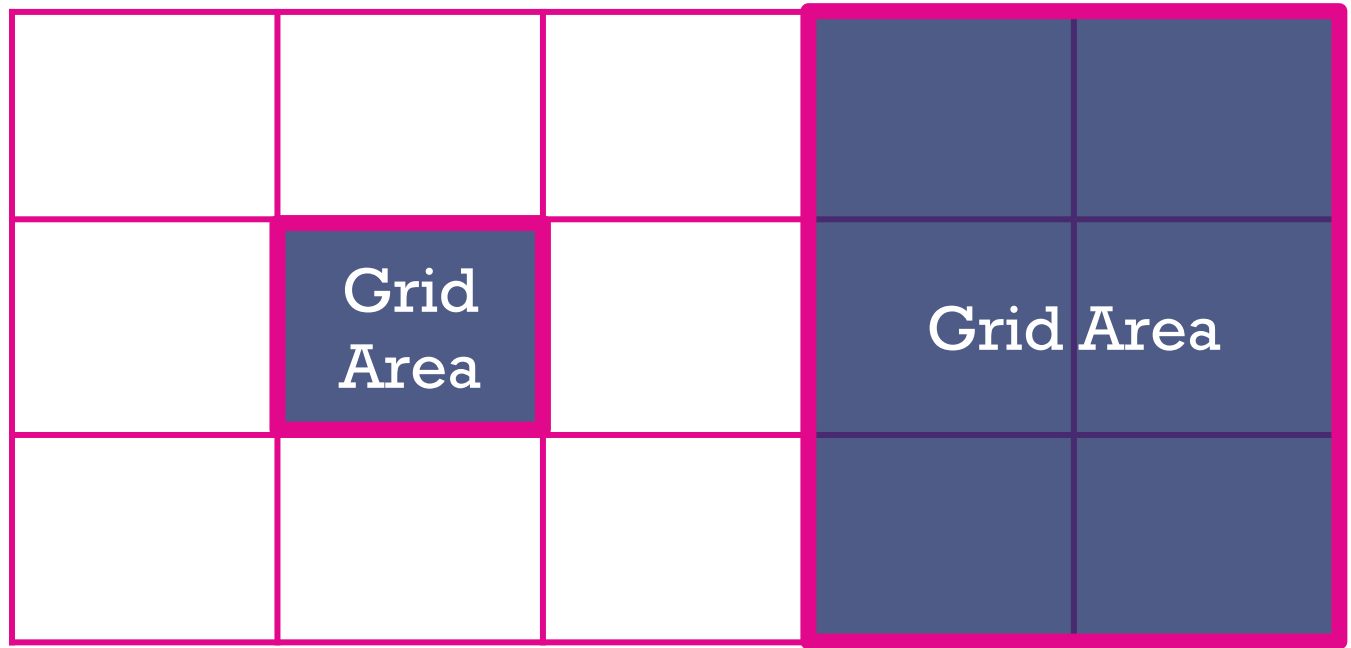
The Vocabulary

	Grid Cell			
			Grid Cell	

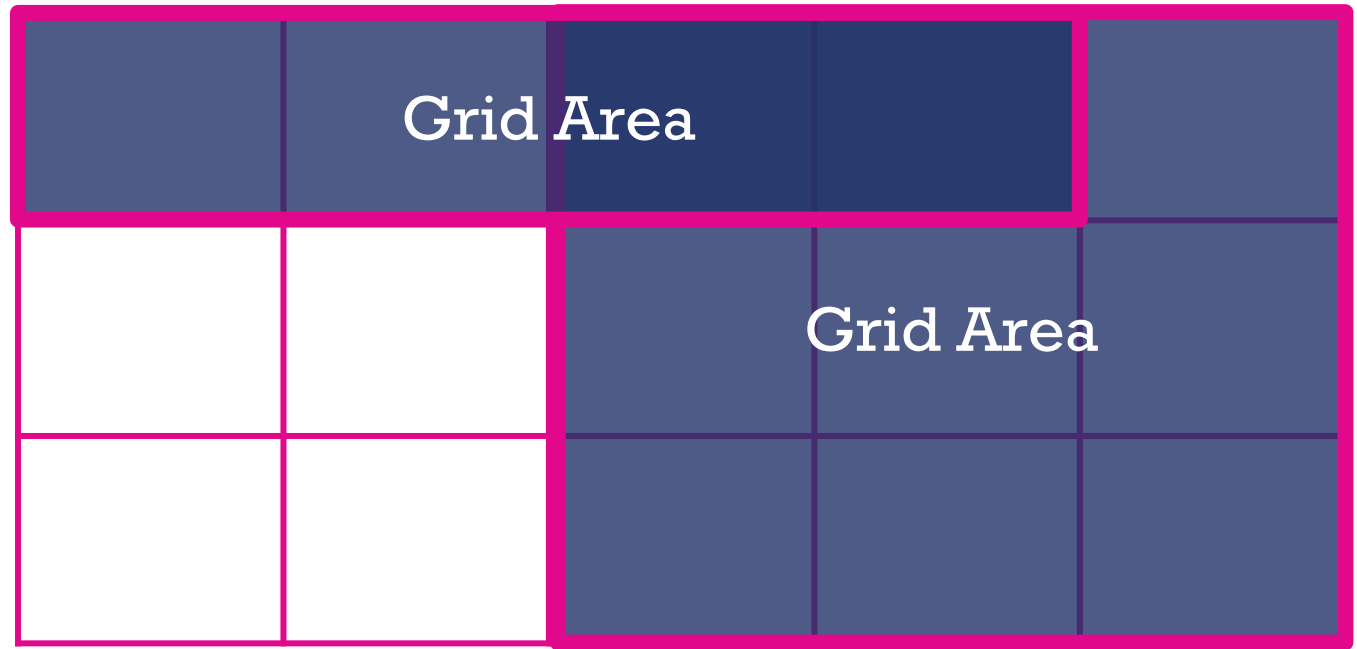
The Vocabulary



The Vocabulary



The Vocabulary



We have some HTML.

Now what?

```
<div class="wrapper">  
  <div class="child one">1</div>  
  <div class="child two">2</div>  
  <div class="child three">3</div>  
  <div class="child four">4</div>  
  <div class="child five">5</div>  
  <div class="child six">6</div>  
</div>
```

Make The Grid

```
.wrapper {  
    display: grid;  
}
```

Define your grid.

Grid Container

Define The Grid

```
.wrapper {  
    display: grid;  
    grid-template-columns: ???;  
    grid-template-rows: ???;  
    row-gap/column-gap/gap: ???;  
}
```

grid-template-columns: 300px 200px 100px;



What the fr?!

grid-template-columns: 3fr 2fr 1fr;

3fr

2fr

1fr

3fr

2fr

1fr

grid-template-columns: 300px 2fr 1fr;

300px

2fr

1fr

300px

2fr

1fr

grid-template-columns:

200px minmax(100px, 1fr) minmax(200px, 400px);

1

2

3

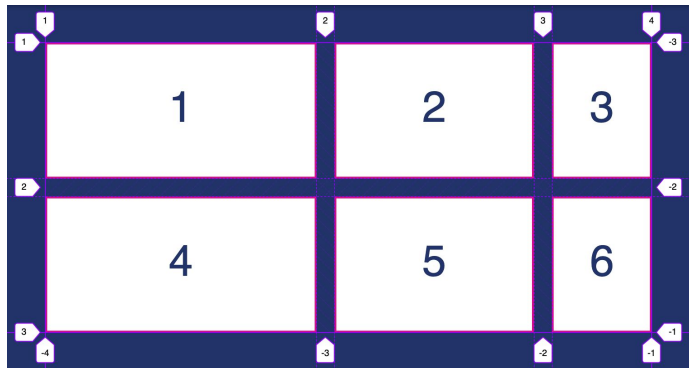
4

5

6

***So you have a
super sweet flexible grid...
Now what?***

Auto Placement



```
.wrapper {
```

```
  display: grid;
```

```
  grid-template-columns: 300px 2fr 1fr;
```

```
  grid-template-rows: 150px 150px;
```

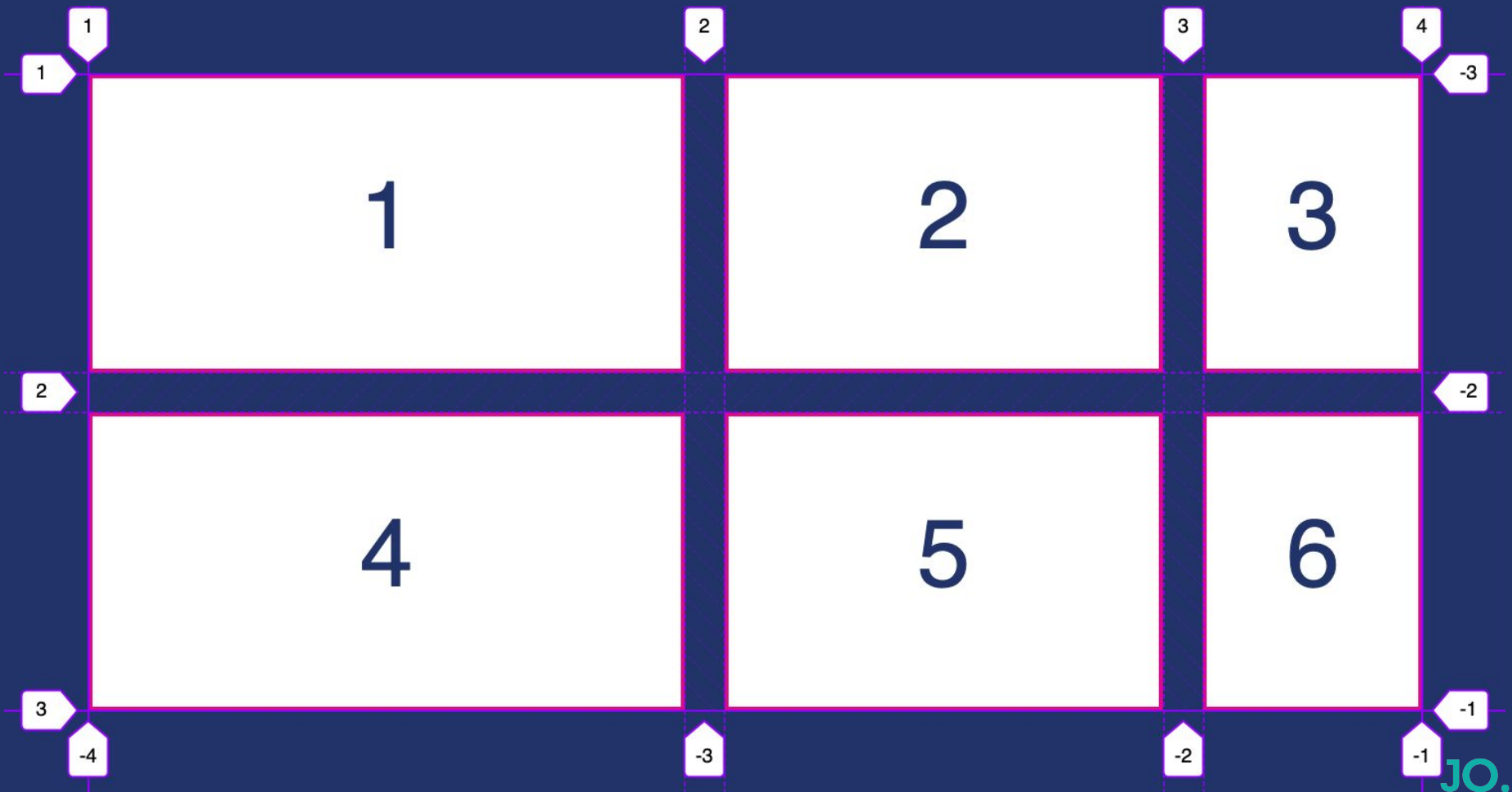
```
  gap: 20px;
```

```
}
```

1	1	2	3	4
1	1	2	3	-3
2	4	5	6	-2
3	???	???	???	-1
4	???			
5				
-4	-3	-2	-1	

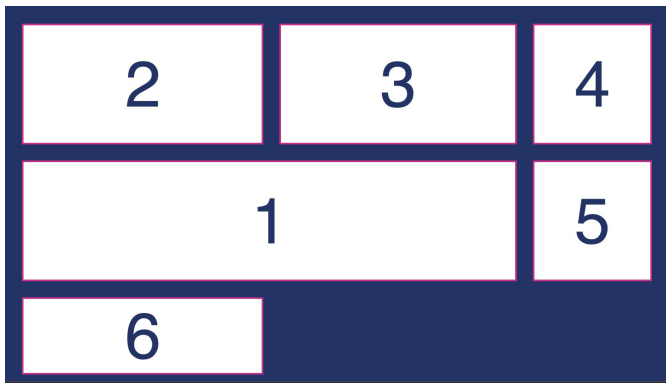
A close-up shot of a woman with dark, curly hair and a man wearing a blue baseball cap. The woman is on the left, looking towards the man on the right. They appear to be in an outdoor setting with a blurred background of trees and a building. The lighting is warm, suggesting late afternoon or early morning.

I can be flexible. As long as everything is exactly the way I want it, I'm totally flexible.



1	1	2	3	4
1	1	2	3	-3
2	4	5	6	-2
3	???	???	???	-1
4	???			
5				
-4	-3	-2	-1	

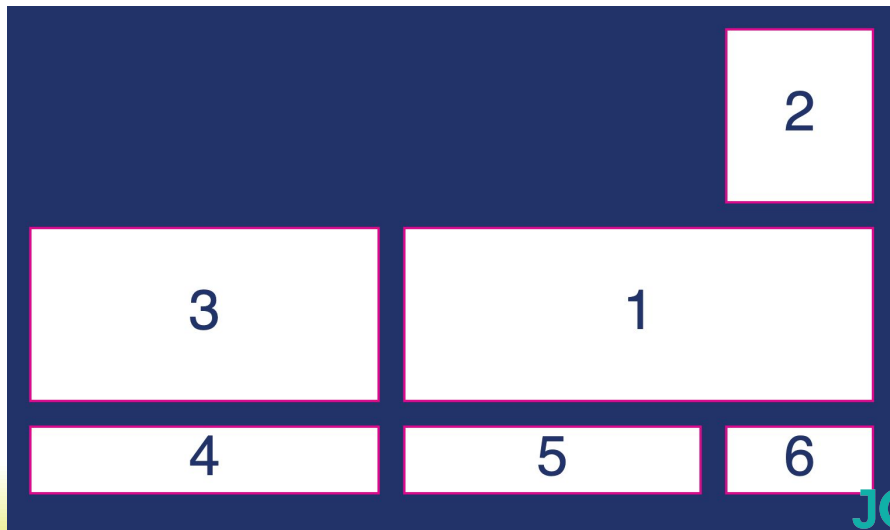
Line-based Placement



```
.one {  
  grid-column: 1 / 3;  
  grid-row: 2 / 3;  
}
```

Named Lines

```
.wrapper {  
  display: grid;  
  grid-template-columns: [start] 300px [main-start] 2fr [bob] 1fr [end];  
  grid-template-rows: [nav] 150px [row2] 150px;  
}  
  
.one {  
  grid-column: main-start/end;  
  grid-row: row2;  
}  
  
.two {  
  grid-column-start: bob;  
}
```



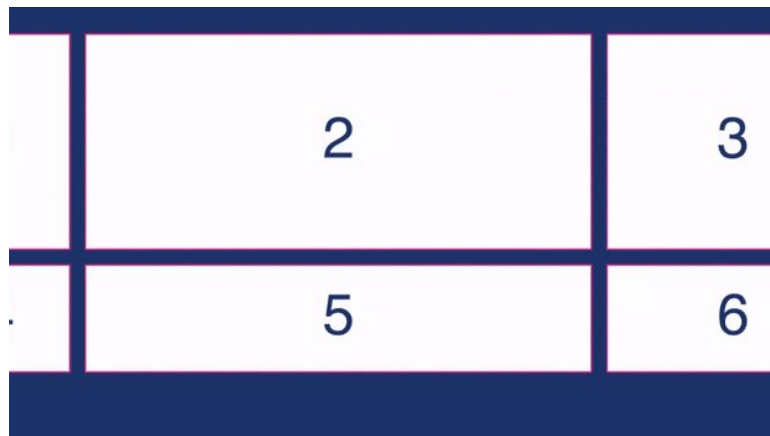


Area Placement

```
.wrapper {  
  grid-template-columns: 200px 2fr 1fr;  
  grid-template-areas:  
    "nav nav nav"  
    "aside main main"  
    "aside info1 info2"  
    "footer footer footer";  
}
```

	2	3
	5	6

Area Placement



```
.wrapper {  
  grid-template-columns: 200px 2fr 1fr;  
  grid-template-areas:  
    "nav nav nav"  
    "aside main main"  
    "aside info1 info2"  
    "footer footer footer";  
}
```

Area Placement

```
.wrapper {  
  grid-template-columns: 200px 2fr 1fr;  
  grid-template-areas:  
    "nav nav nav"  
    "aside main main"  
    "aside info1 info2"  
    "footer footer footer";  
}
```

```
.one {  
  grid-area: nav;  
}
```

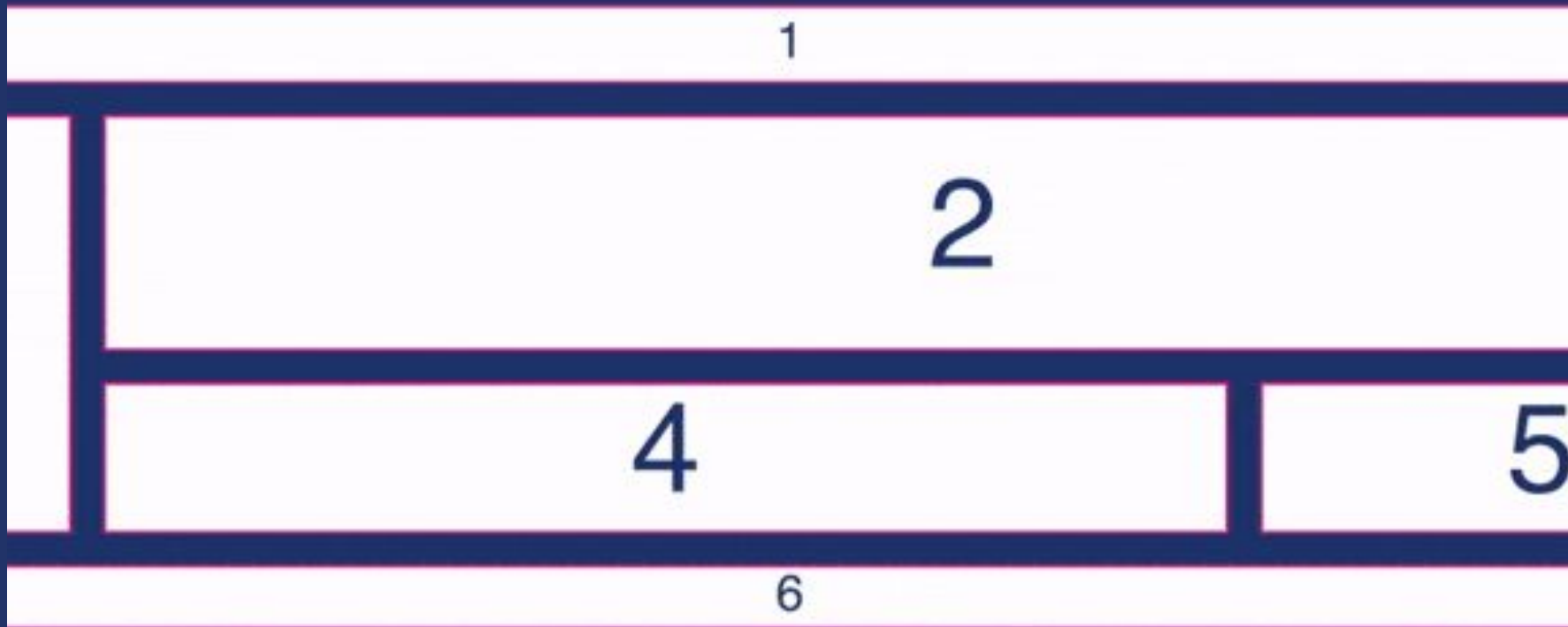
```
.two {  
  grid-area: main;  
}
```

```
.three {  
  grid-area: aside;  
}
```

```
.four {  
  grid-area: info1;  
}
```

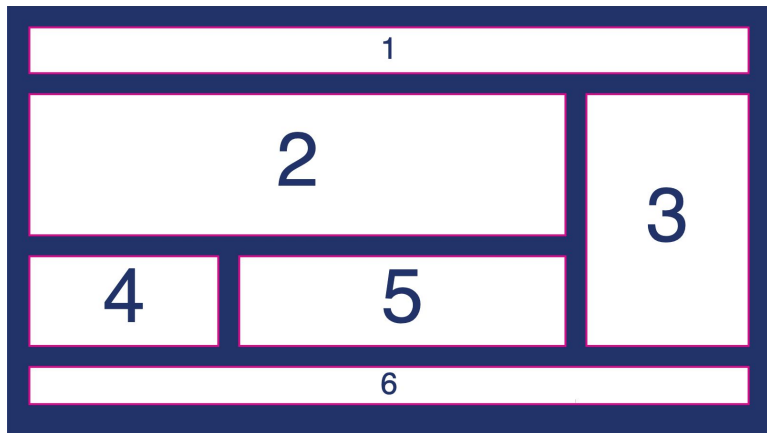
```
.five {  
  grid-area: info2;  
}
```

```
.six {  
  grid-area: footer;  
}
```



Wanna see a trick?

Area Placement



```
.wrapper {  
  grid-template-columns: 200px 2fr 1fr;  
  grid-template-areas:  
    "nav nav nav"  
    "aside main aside"  
    "aside info2 aside"  
    "footer footer footer";  
}
```

A close-up shot of a man with a beard and short brown hair, wearing a grey suit, white shirt, and a patterned tie. He is looking down and slightly to his right with a contemplative or emotional expression. The background is dark with warm, out-of-focus lights, suggesting a Christmas tree or festive decorations.

So many emotions come at you at once.

Not enough for you yet?

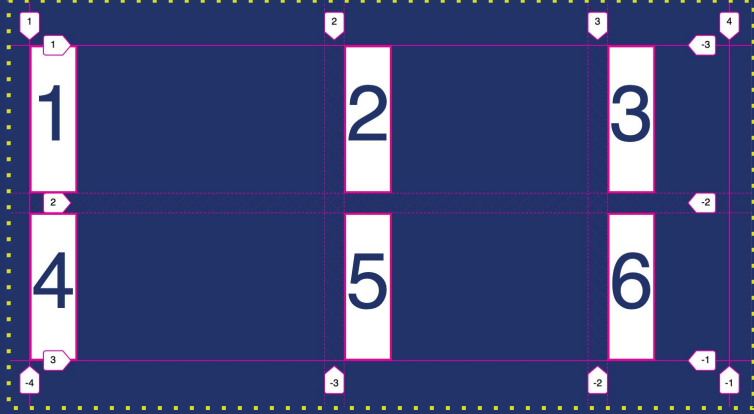
Alignment

Justify-items? Align-items?

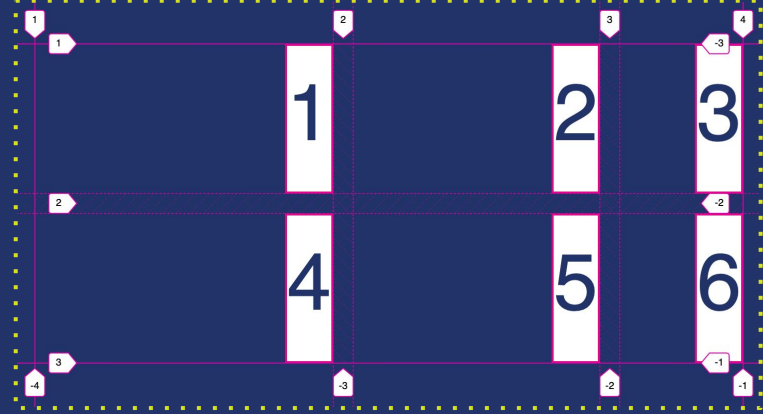
Justify-content? Align-content?

- Justify = along the inline (row) axis
- Align = along the block (column) axis
- Items = all grid items in container
- Content = the entire grid in the container

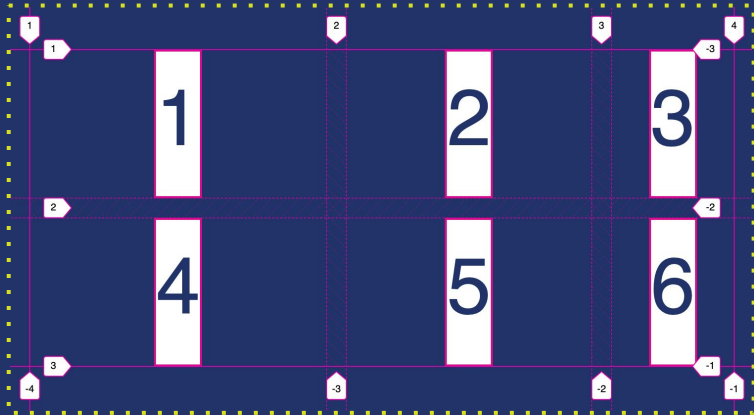
justify-items: start;



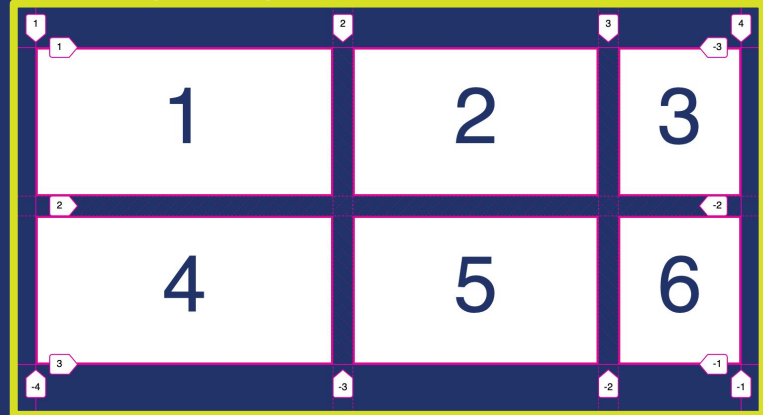
justify-items: end;



justify-items: center;

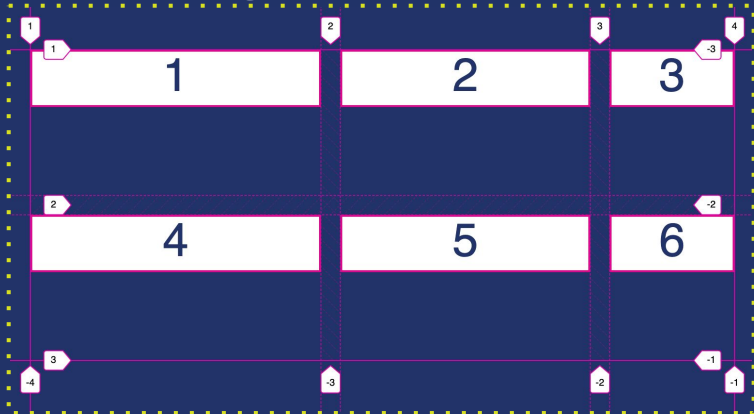


justify-items: stretch;

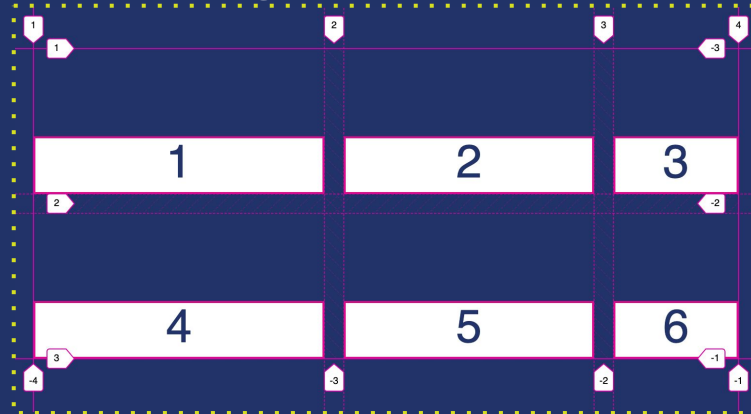


<https://codepen.io/qjac/pen/XQXeoV>

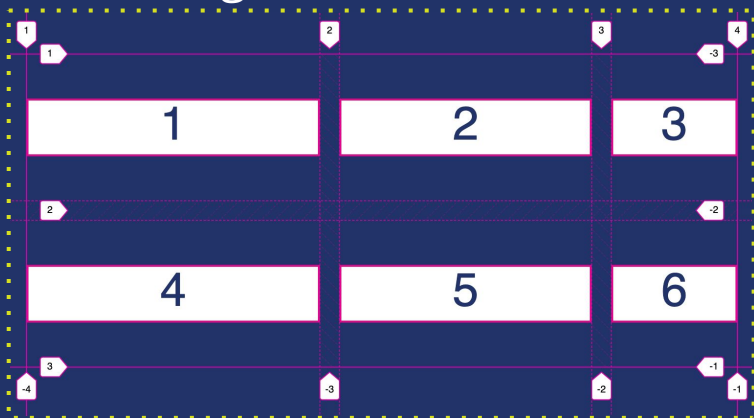
align-items: start;



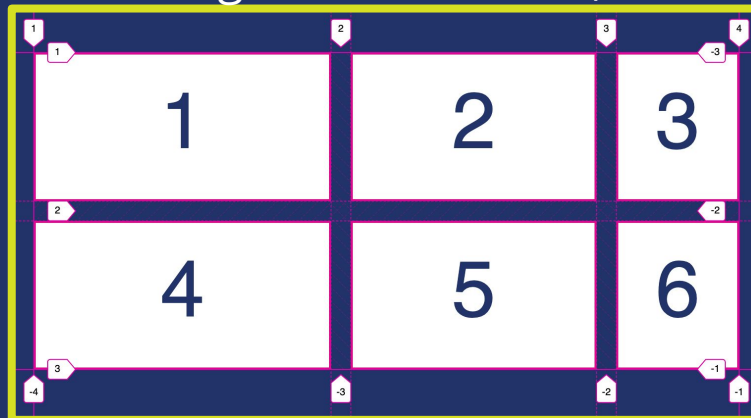
align-items: end;



align-items: center;

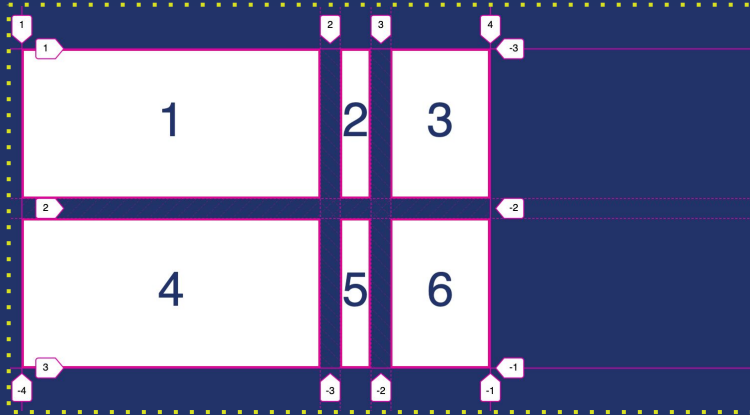


align-items: stretch;

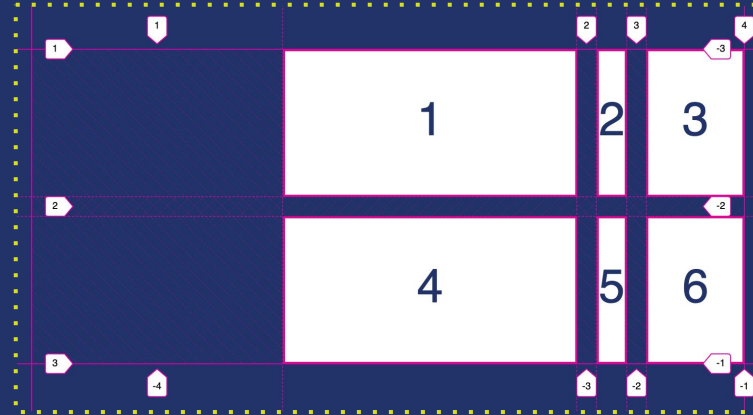


<https://codepen.io/qjac/pen/pBgWBe>

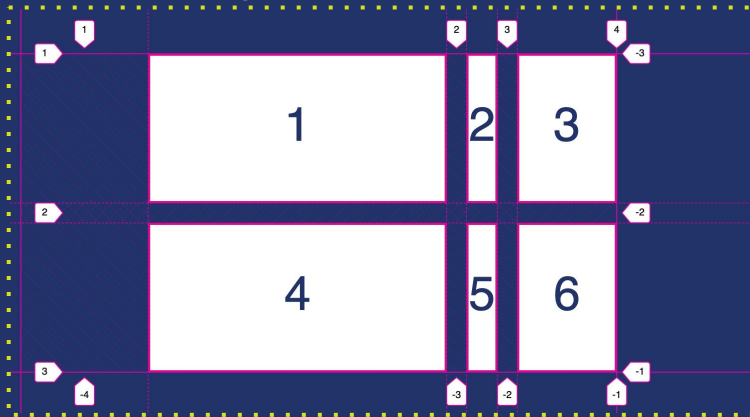
justify-content: start;



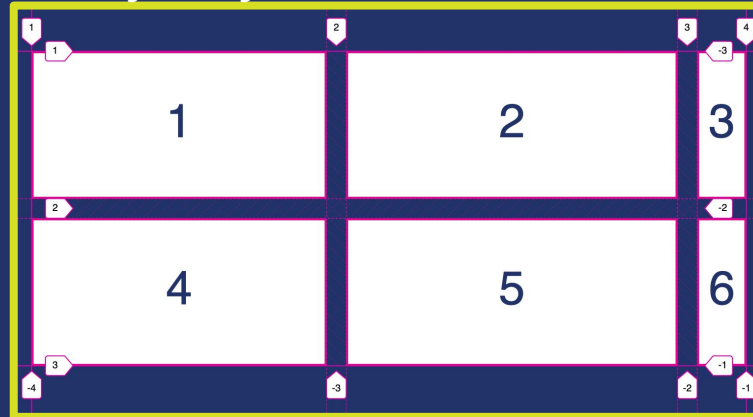
justify-content: end;



Justify-content: center;

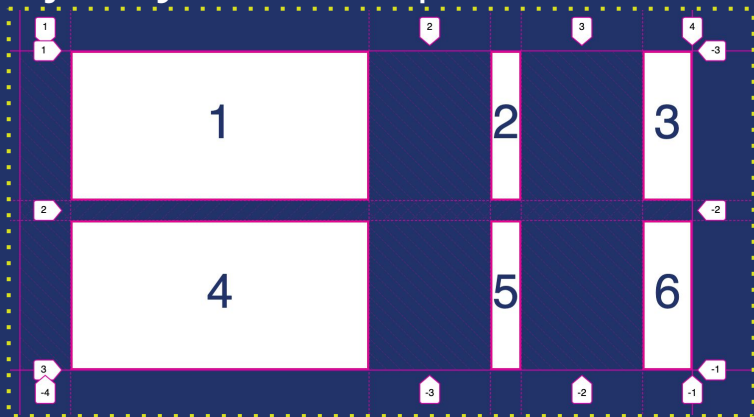


justify-content: stretch;

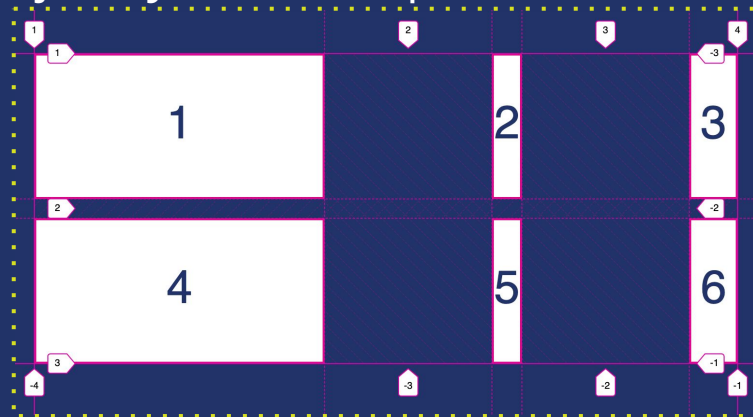


<https://codepen.io/qjac/pen/axNBjL>

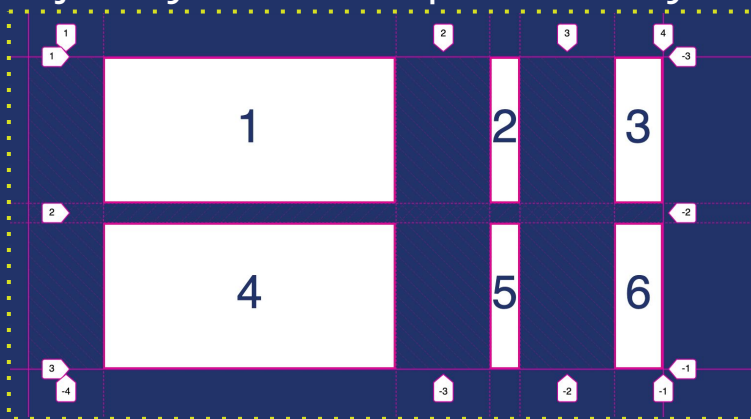
justify-content: space-around;



justify-content: space-between;

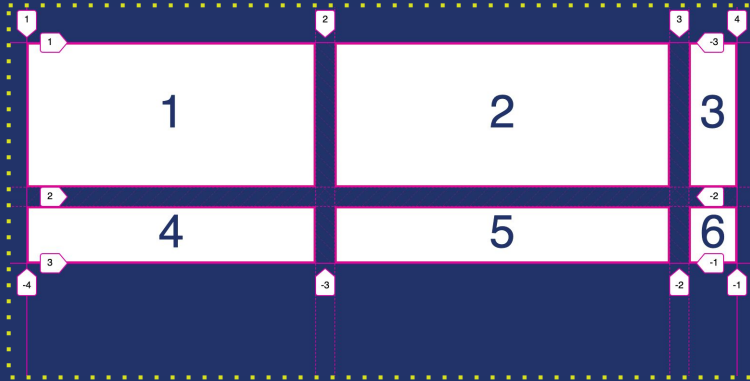


justify-content: space-evenly;

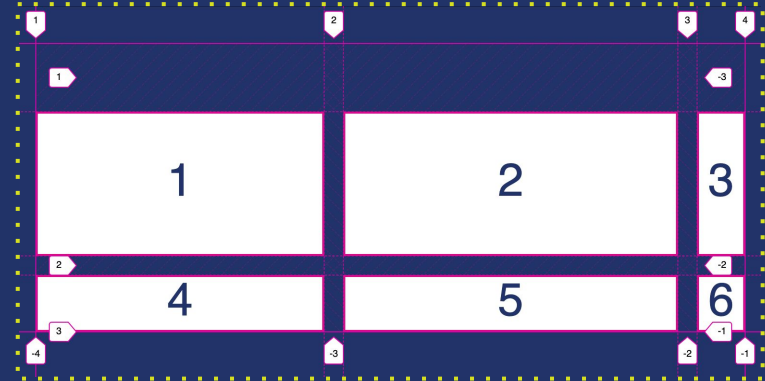


<https://codepen.io/qjac/pen/axNBjL>

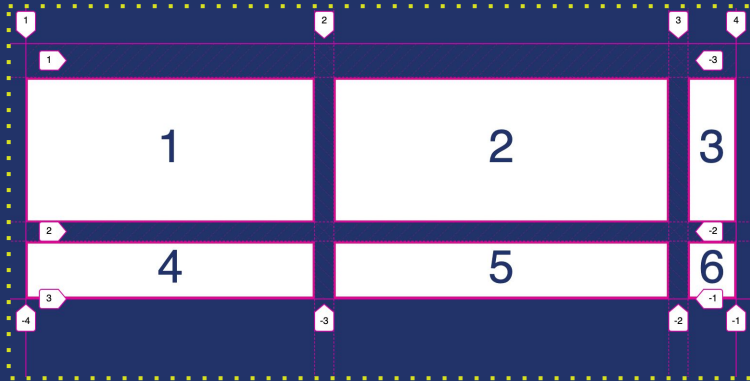
align-content: start;



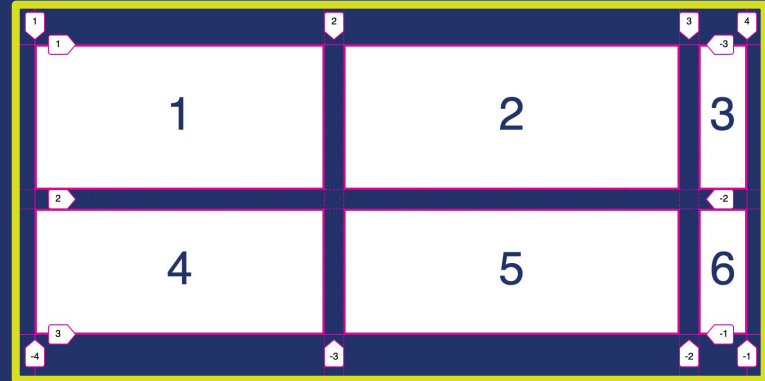
align-content: end;



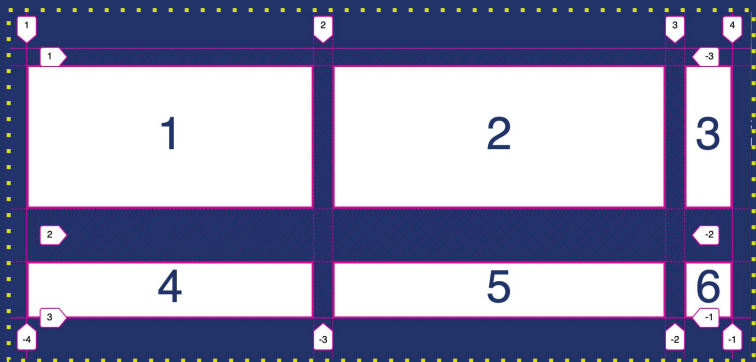
align-content center;



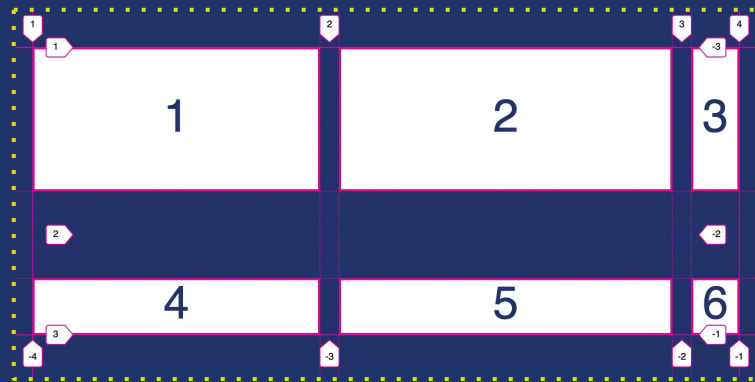
align-content: stretch;



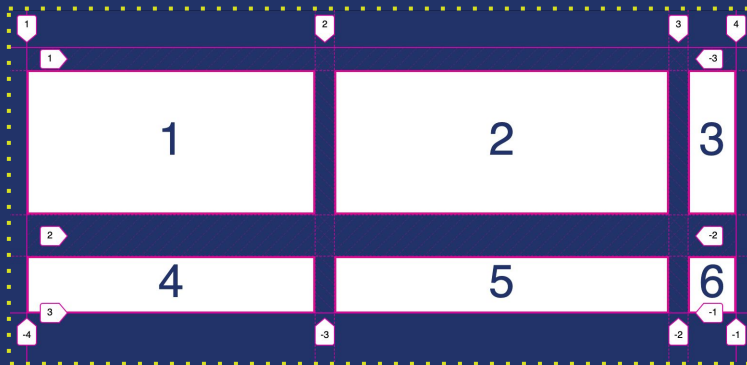
align-content: space-around;



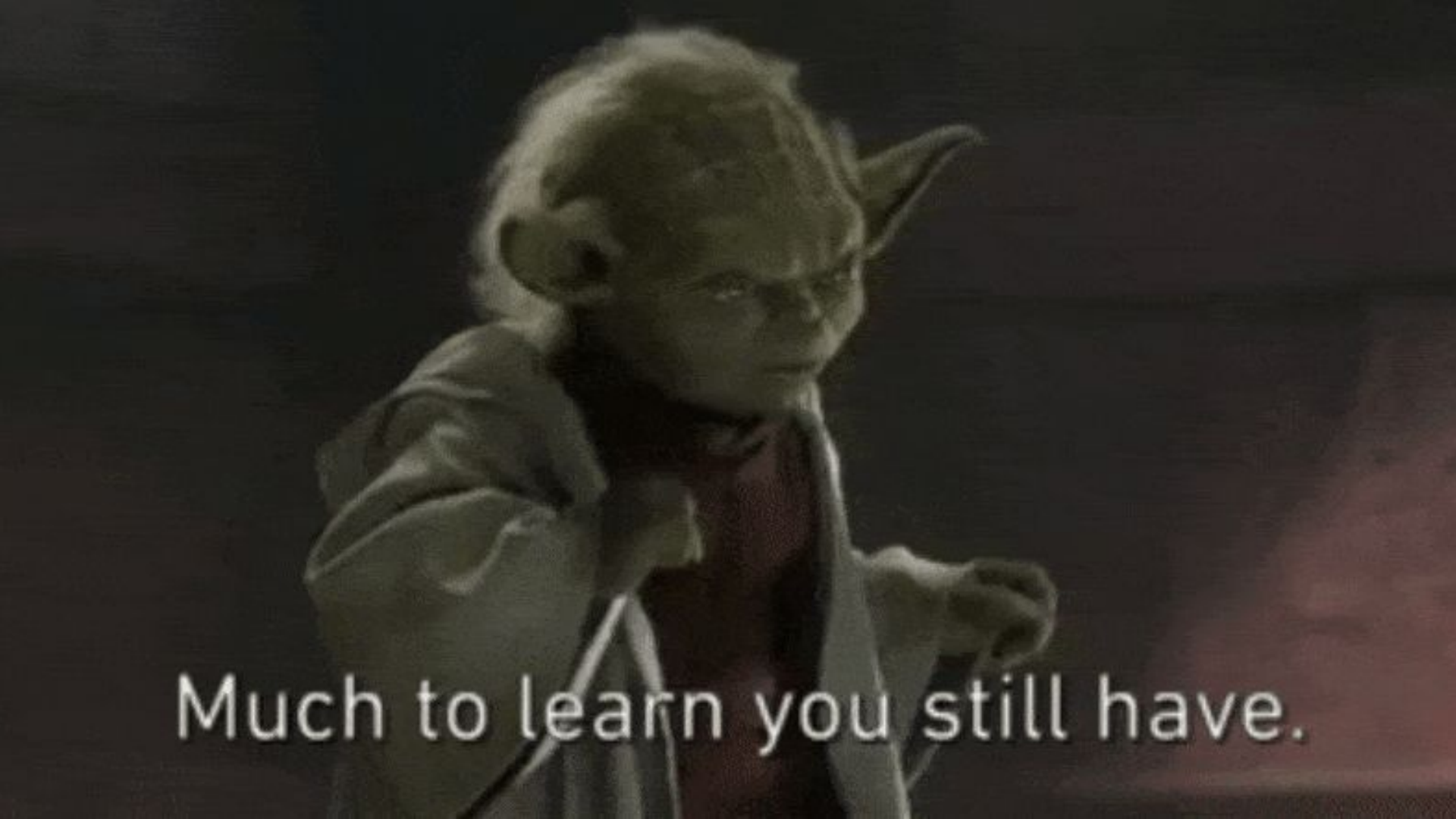
align-content: space-between;



align-content: space-evenly;



<https://codepen.io/qjac/pen/qwZmBr>

A close-up shot of Yoda from Star Wars, looking slightly to the right with a serious expression. He is holding a lightsaber with both hands, the hilt visible in the foreground. The background is dark and out of focus.

Much to learn you still have.

Get Schooled

Practice makes perfect.

Slides: tinyurl.com/discover-grid

- [Jen Simmons' Layout Lab](#)
- [Grid by Example](#)
- [Learn CSS Grid with Wes Bos](#)
- [Learn CSS Grid for Free](#)
- [Grid Garden](#)

Read Up

Tips + tricks from
the experts.

Slides: tinyurl.com/discover-grid

- [The New Layout Standard For The Web](#)
by Rachel Andrew
- [CSS Grid Gotchas And Stumbling Blocks](#)
by Rachel Andrew
- [Best Practices With CSS Grid Layout](#)
by Rachel Andrew
- [Building Production-Ready CSS Grid Layouts Today](#)
by Morten Rand-Hendriksen
- [3 CSS Grid Features That Make My Heart Flutter](#) by Una Kravets

Keep Handy

Essential references.

Slides: tinyurl.com/discover-grid

- [Firefox Grid Inspector](#)
- [The MDN Docs](#)
- [CSS Tricks Grid Guide](#)
- [GridBugs](#)
- [W3C Docs](#)
- [Demos from this talk](#)

Tools made for the job

- Static is still the default
- Floats for floating
- Tables for tabular data
- Columns for text blocks
- Absolute for absolute positioning
- Fixed for fixing to the viewport
- Relative for positioning relative to the default
- Flex for one-dimensional flexible layout
- Grid for two-dimensional flexible layout



Thank you!

 qjac.net  [@_qjac](https://twitter.com/_qjac)    [@qjac](https://github.com/qjac)

Slides: tinyurl.com/discover-grid

Thank you!

 qjac.net  [@_qjac](https://twitter.com/_qjac)    [@qjac](https://github.com/qjac)

Slides: tinyurl.com/discover-grid
