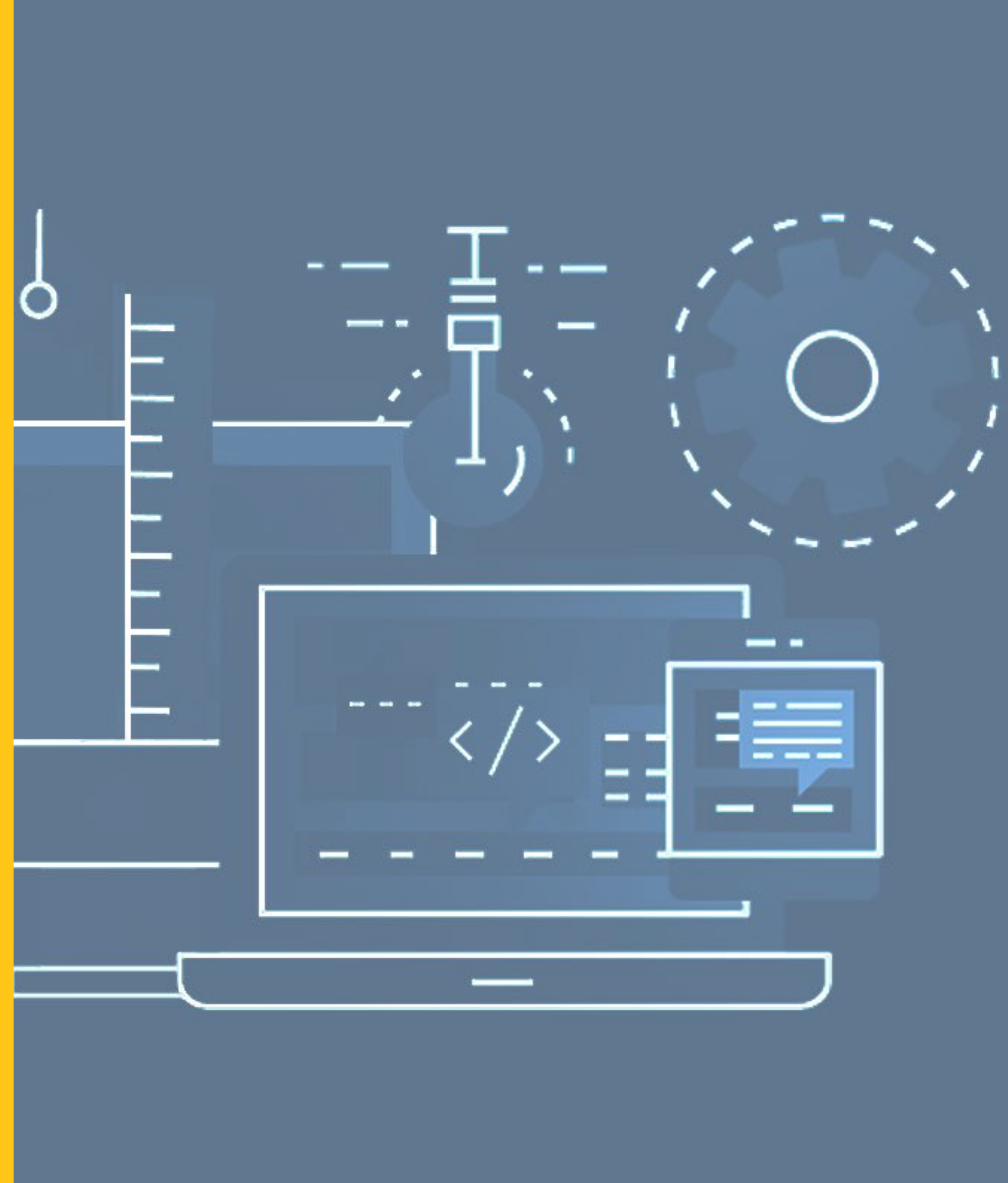


SHELLY GRAHAM, 02/17/2022

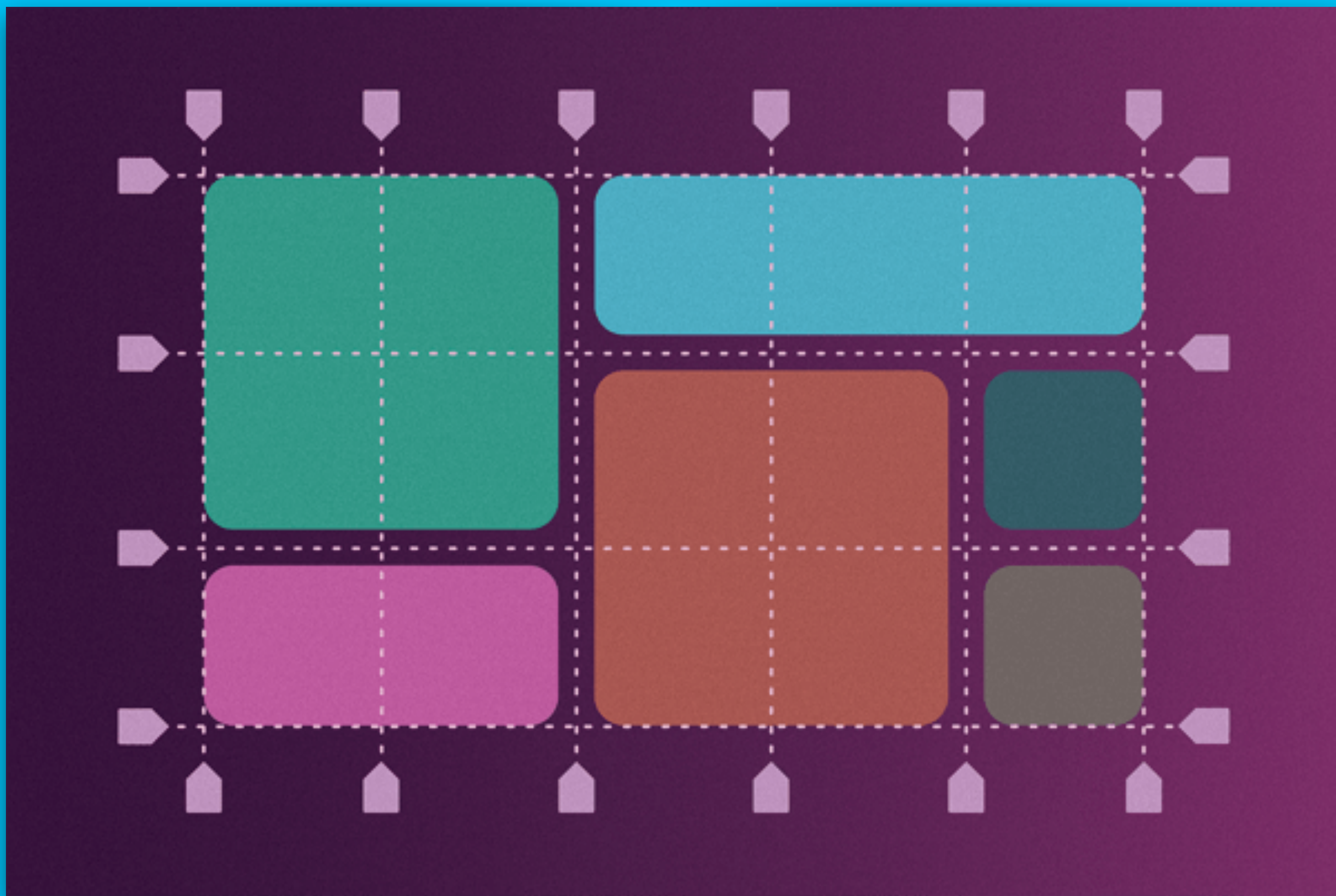
WEB DEV 3

WINTER 2022

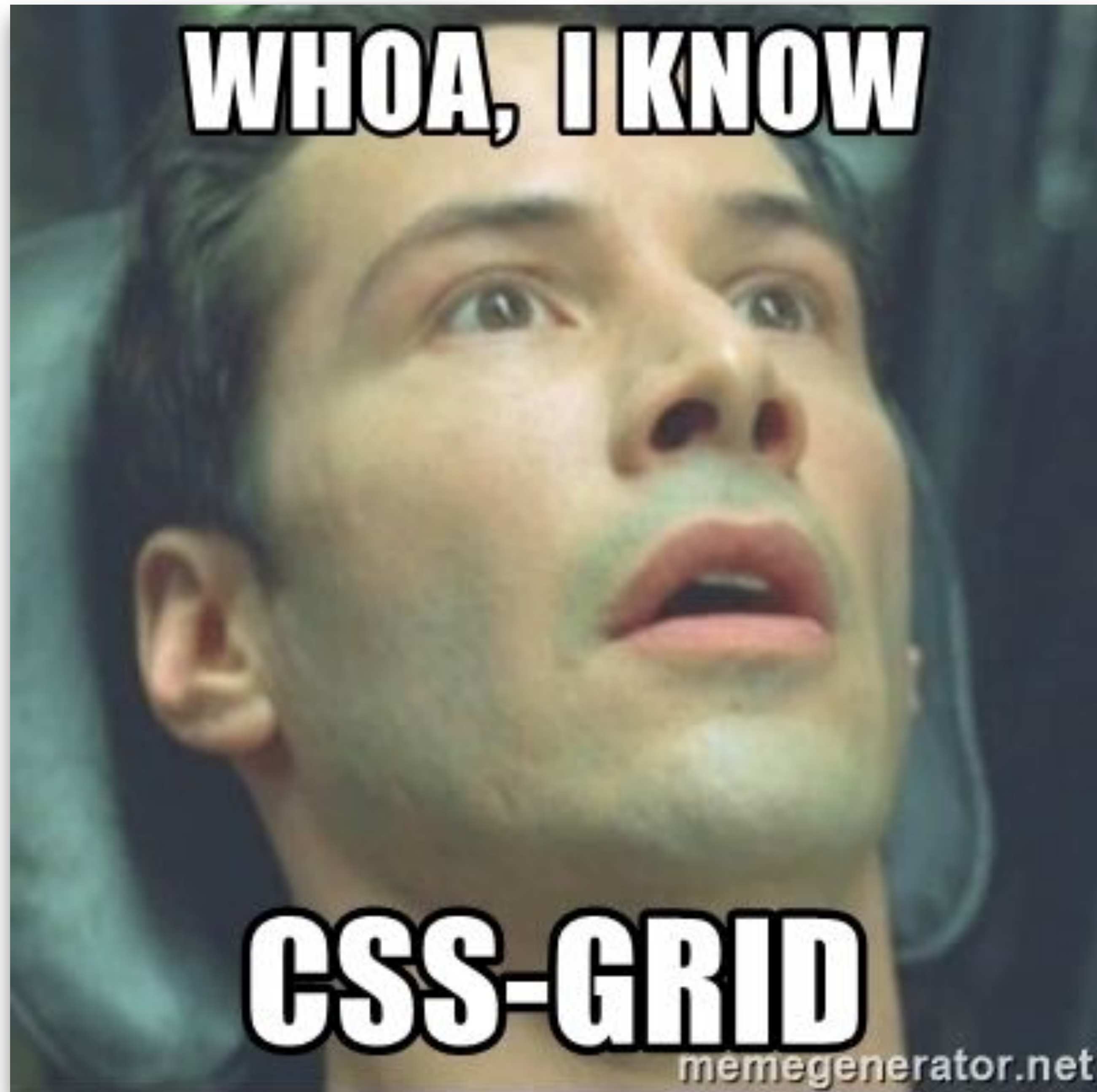
Week 7: CSS Grid - continued



WEEK 7: CSS GRID

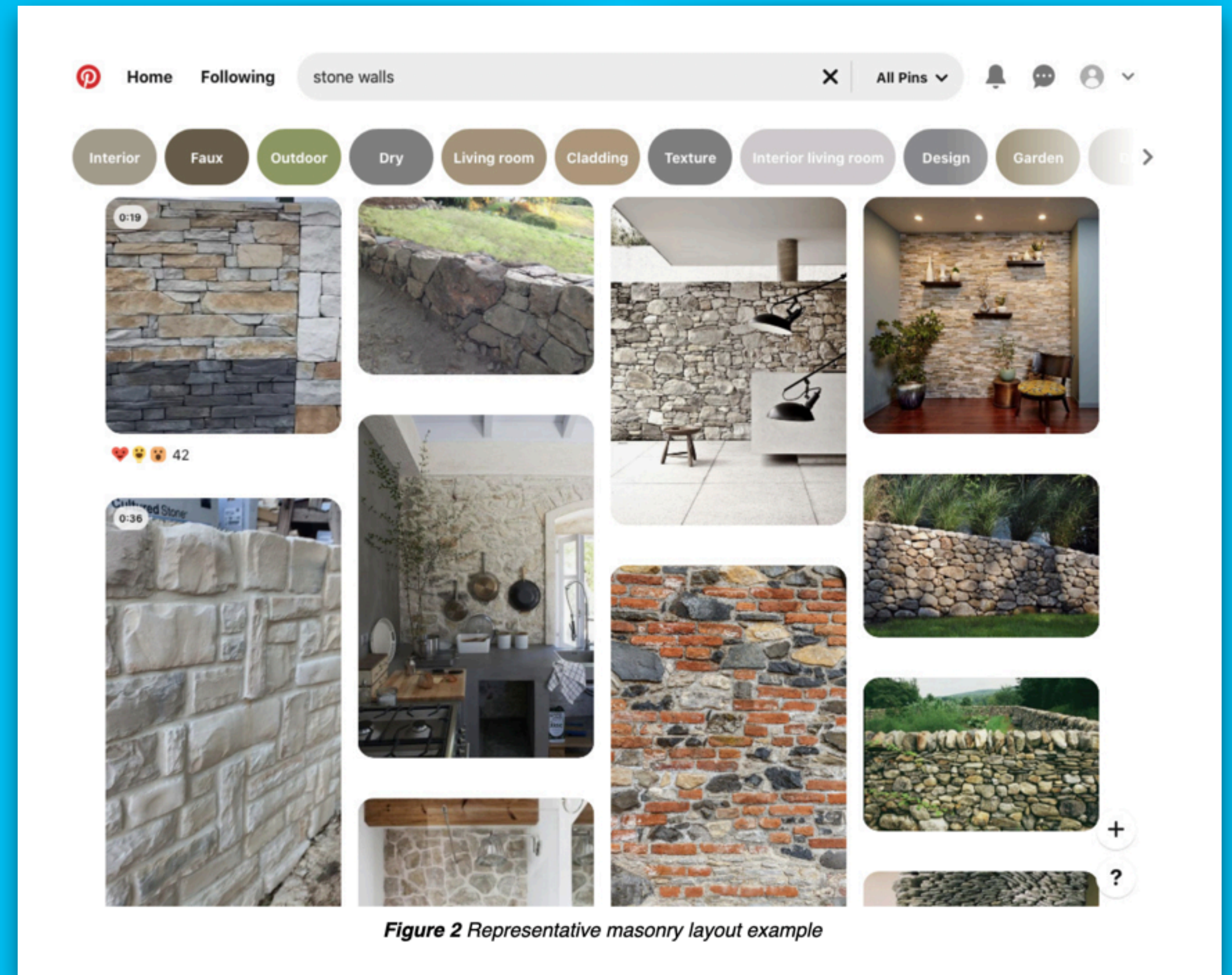


RECAP:



MASONRY-GRID

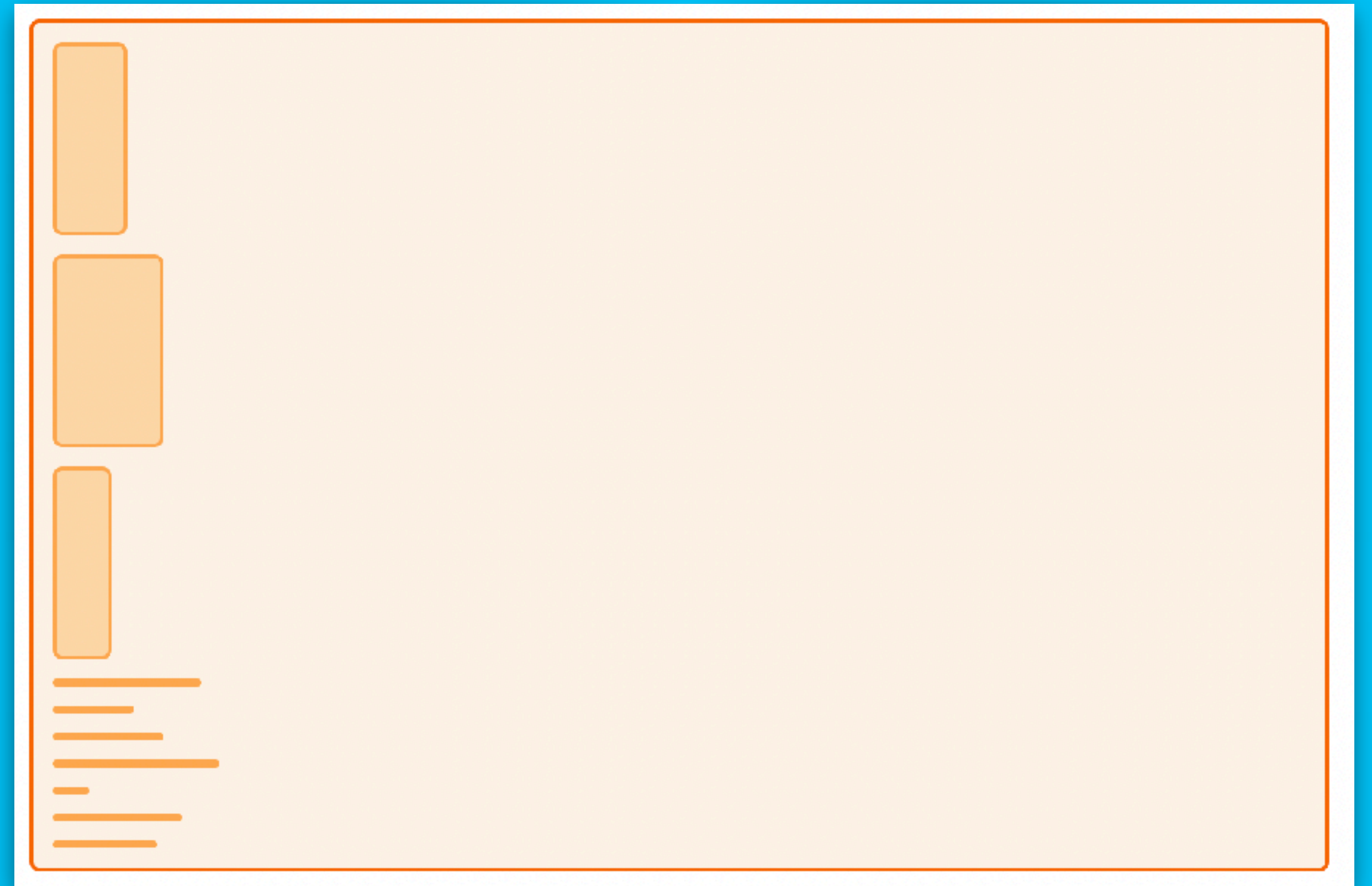
- An experimental feature of CSS Grid is masonry layout
- Items of an uneven size are laid out such that there aren't uneven gaps
- Note that there are [lots of approaches to CSS masonry](#), but mostly of them are trickery and either have major downsides or aren't what you quite expect
- **Usage:**
grid-template-rows: masonry;



THE PARENT - TEMPLATE

JUSTIFY-TRACKS

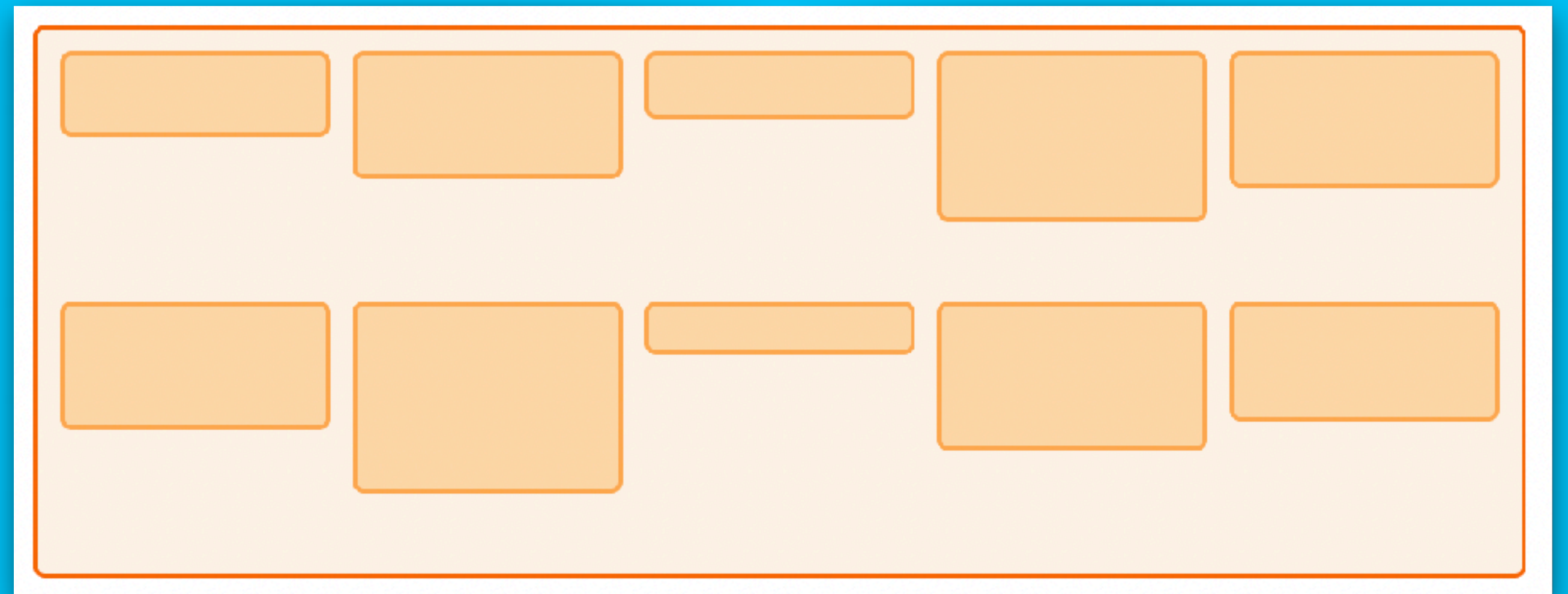
- Sets the alignment in the masonry axis for grid containers that have masonry in their inline axis
- Values:
 - start
 - end
 - center
 - stretch (default)
 - space-around
 - space-between
 - space-evenly



THE PARENT - TEMPLATE

ALIGN-TRACKS

- Sets the alignment in the masonry axis for grid containers that have masonry in their block axis
- Values:
 - start
 - end
 - center
 - stretch (default)
 - space-around
 - space-between
 - space-evenly



Can I use

masonry

? ⚙ Settings

2 results found

CSS property: masonry

Usage
Global

Current aligned

Usage relative

Date relative

Filtered

All



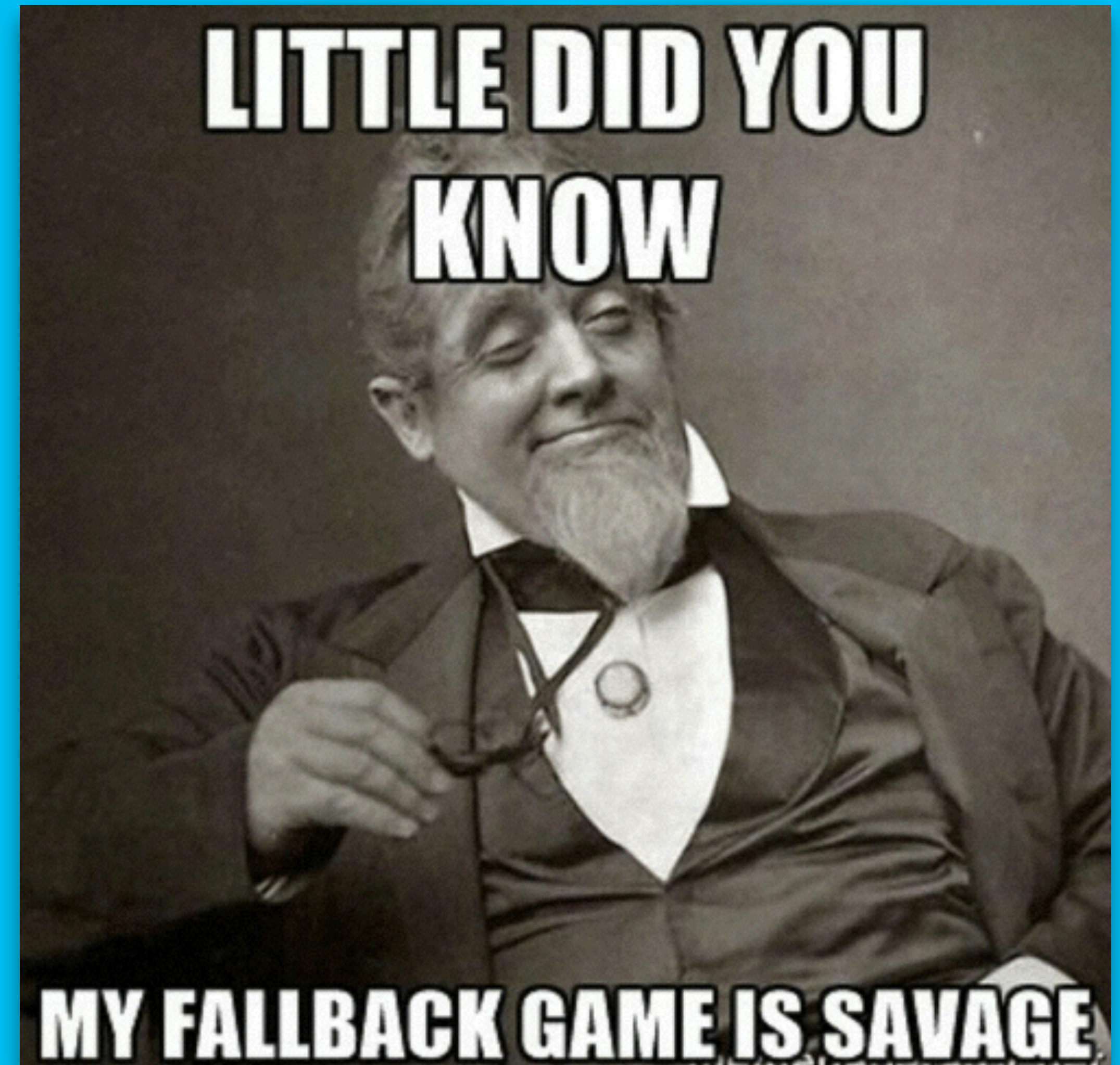
IE	Edge [*]	Firefox	Chrome	Safari	Opera	iOS Safari [*]	Opera Mini [*]	Android Browser [*]	Opera Mobile [*]	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet
		2-76											
6-10	12-87	¹ 77-84	4-87	3.1-13.1	10-71	3.2-13.7		2.1-4.4.4	12-12.1				4-12.0
11	88	¹ 85	88	14	72	14.4	all	81	59	88	83	12.12	13.0
		¹ 86-87	89-91	TP									

BROWSER SUPPORT?

—

BROWSER SUPPORT?

- What do we do if certain properties aren't supported across all browser?
- Not use them at all?
- Use older tools to be “safe”?
- NO! How would anything ever change?!



BASIC CODE AS FALLBACK

- Instead of writing complex code that might only be accessible for a handful of browsers (basically never IE and Edge -.-) - lets write basic code that is accessible for most browsers!
- CSS Grid is widely accepted across all modern browsers but certain properties within CSS Grid are not - for example masonry, align-tracks or justify-tracks
- Turn on Masonry in Firefox:
 1. Type **about:config** in the URL bar
 2. Search for “masonry”
 3. Set *layout.css.grid-template-masonry-value.enabled* to **true**
- Let's write a fallback for CSS Grid Masonry!

COLUMN-COUNT

- **column-count:** breaks an element's content into the specified number of columns
- **Values:**
 - any integer
 - auto (only works if paired with column-width)

```
.column {  
    column-count: 2;  
    column-count: 3;  
    column-count: 4;  
    column-count: auto;  
}
```


COLUMN-WIDTH

- `column-width`: sets the ideal column width in a multi-column layout
- Similar to `fr` unit
- Values:
 - any length
 - `auto` (only works if paired with `column-count`)

```
.column {  
    column-width: 6rem;  
    column-width: 120px;  
    column-width: auto;  
}
```

COLUMNS

- **Shorthand** for:
column-count
column-width

```
.column {  
    columns: 6rem auto;  
    columns: 2 250px;  
    columns: 3;  
}
```


COLUMN-RULE

- column-rule sets the width, style, and color of the line drawn between columns in a multi-column layout
- **Shorthand** for:
 - column-rule-width
 - column-rule-style
 - column-rule-color
- Similar to border shorthand

```
.column {  
  column-rule: solid 6px;  
  column-rule: solid blue;  
  column-rule: thick inset red;  
  column-rule: 4px dotted rgb(79, 185, 227);  
}
```

COLUMN-GAP

- column-rule sets the amount of space between each column
- If combined with column-rule, will center border automatically between each column

```
.column {  
    column-gap: 20px;  
    column-gap: 1em;  
    column-gap: 5ch;  
}
```


COLUMN-SPAN

- column-span makes it possible for an element to span across all columns when its value is set to **column-span: all**
- Default: column-span: none;

column-span: none;

London.	Implacable	would not be
Michaelmas	November	wonderful to
term lately	weather. As	meet a
over, and the	much mud in	Megalosaurus,
Lord	the streets as if	forty feet long
Chancellor	the waters had	or so, waddling
sitting in	but newly	like an
Lincoln's Inn	retired from the	elephantine
Hall.	face of the	lizard up
Spanner?	earth, and it	Holborn Hill.

column-span: all;

London.	over, and the	sitting in
Michaelmas	Lord	Lincoln's Inn
term lately	Chancellor	Hall.
Spanner?		
Implacable	retired from the	forty feet long
November	face of the	or so, waddling
weather. As	earth, and it	like an
much mud in	would not be	elephantine
the streets as if	wonderful to	lizard up
the waters had	meet a	Holborn Hill.
but newly	Megalosaurus,	

LETS CODE!

@SUPPORTS

```
supports (display  
  .main {  
    display: gr  
  }
```

@SUPPORTS AKA FEATURE QUERY

- The @supports at-rule lets you specify declarations that depend on a browser's support for one or more specific CSS features.
- This is called a *feature query*
- *With @supports, you can write a small test in your CSS to see whether or not a particular CSS property or value is supported*
- *Personally, one website doesn't have to look the exact same in every browser because I'm lazy and I don't like to write elaborate fallbacks to force unity.*
- *I am for a situation where a natural failure of a property:value doesn't do anything drastic to destroy functionality*
- *NOT supported in IE (duh!) but so many people still use it - soooo we need a fall back!*

BE SMART ABOUT HOW YOU USE IT!

- If a browser understands border-radius, then it will put rounded corners on the box.
- If it doesn't, it will skip that line of code and move on, leaving the edges of the box to be square.
- There is no reason to run a test or use a Feature Query. This is just how CSS works.
- Browsers simply skip over code they don't understand, without throwing an error.
- Use a Feature Query when you want to apply a mix of old and new CSS, but only when the new CSS is supported!



@SUPPORTS USE CASES:

Does browser support @supports?	Does browser support property or value in question?	What happens?	Is This What We Want?
Supports Feature Queries	Supports the feature in question	CSS is applied	Yes
Supports Feature Queries	Does not support the feature	CSS is not applied	Yes
Does not support Feature Queries	Does not support the feature	CSS is not applied	Yes
Does not support Feature Queries	Supports the feature in question	CSS is not applied	Probably not!

@SUPPORTS

- Checks for browser support and only executes code if browser can display certain CSS properties **or** values
- We need a property - value pair because it checks for both
- Looks like media query - so treat it like media query and write it **AFTER** your default code
- Can be used with keywords: and/or/not

```
@supports (grid-template-rows: masonry) {  
  .grid {  
    max-width: 60rem;  
    margin: 0 auto;  
  
    display: grid;  
    gap: 1em;  
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
    grid-template-rows: masonry;  
    align-tracks: stretch;  
  
  }  
}
```

VENDOR PREFIXES

- Helps us ensure that certain (newer) CSS properties work across all browsers
- Tedious work, so much more code. Plus I'm lazy and so are 93% of developers
- **-webkit-** (Chrome, Safari, iOS Safari / iOS WebView, Android)
- **-moz-** (Firefox)
- **-ms-** (Edge, Internet Explorer)
- **-o-** (Opera, Opera Mini)
- Sometimes still needed but will face out hopefully sooner rather than later

```
.class {  
    -webkit-transition: opacity .4s ease-out;  
    -moz-transition: opacity .4s ease-out;  
    -ms-transition: opacity .4s ease-out;  
    -o-transition: opacity .4s ease-out;  
    transition: opacity .4s ease-out;  
}
```

QUESTIONS?



RESOURCES

- <https://caniuse.com/>
- <https://autoprefixer.github.io/>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/@supports>
- <https://css-tricks.com/how-supports-works/>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/column-count>
- <https://thenounproject.com/>

HOMEWORK - PROJECT 2

- Time to combine all your knowledge from this class and make one beautiful a** project!
- This is a two week project!
- **ATTENTION!** Please download resources **ASAP** since I had to send through a WeTransfer due to file size! WeTransfer will expire soon!
- Use the provided resources to recreate the Poke Bowl website. If done right, this will be a solid portfolio piece but at the very least an amazing panel piece to showcase what you've learned in this class!
- Use CSS Grid where it makes sense. This can be across the entire page or just during certain sections. I'd like to see a minimum of two grids that change as the browser resizes.
- Optimize the provided images to keep page load to a minimum. Use the techniques learned (lazy loading, srcset...) to achieve this.
- Add the transitions shown in the demo videos but also add at least 1 other transition of your own.
- As usual, the page must be fully responsive. Please use Sass in your CSS.
- Submit a link to your GitHub repo.

FIN