Front-End Web Development

CSS Grid

GENERAL ASSEMBLY

Today's Learning Objectives

In this lesson, you will:

- Create responsive layouts using CSS Grid properties.
- Compare and contrast flexbox and Grid properties.
- Define fractional and percentage-based widths for elements

Front-End Web Development

Flexbox Review

Discussion:
Flexbox Questions

Think!

- In which direction does a flexbox natively go: horizontal or vertical?
  - What CSS property adjusts that?
- How do you reorder flexbox children?
  - What CSS property adjusts that?
- Can you nest a flexbox inside of a flexbox?
  - Why would we want to do that? Doesn't flexbox fix everything?

Guided Walk-Through:
Nested Flexboxes

Together, let's build a series of four sections aligned within a flex container, with each individual section also being a flex container! Remember, flex properties only affect **direct children**, often requiring flex children to be flex parents as well!

Starter code:

https://codepen.io/GAmarketing/pen/LYYgXgX

Solution code:

https://codepen.io/GAmarketing/pen/YzzJRRM

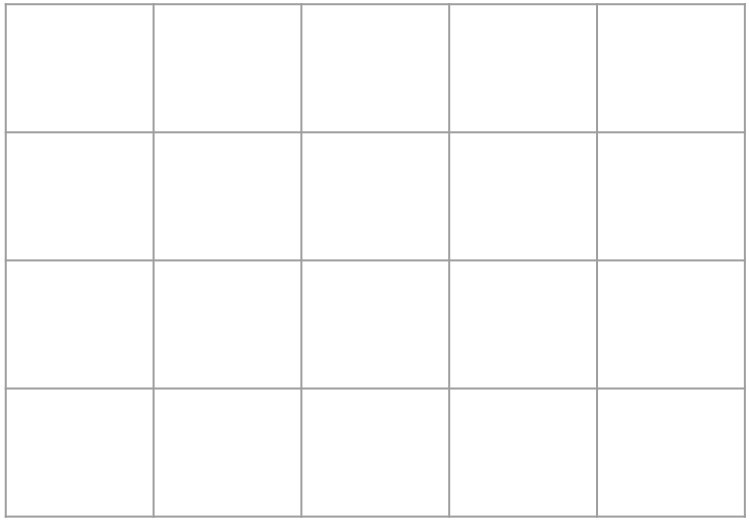
Front-End Web Development

CSS Grid

What Is a Grid?

Picture spreadsheets: Columns and rows into which you can lay content.

Grids ensure that elements will line up smoothly across the page.



Comparing Three Tools for Layouts

Floats

Floats help flow text around an image; they aren't great for layouts.

Interrupt regular document flow.

Flexbox

Flexbox only works in ONE dimension at a time, with a main axis and a cross axis.

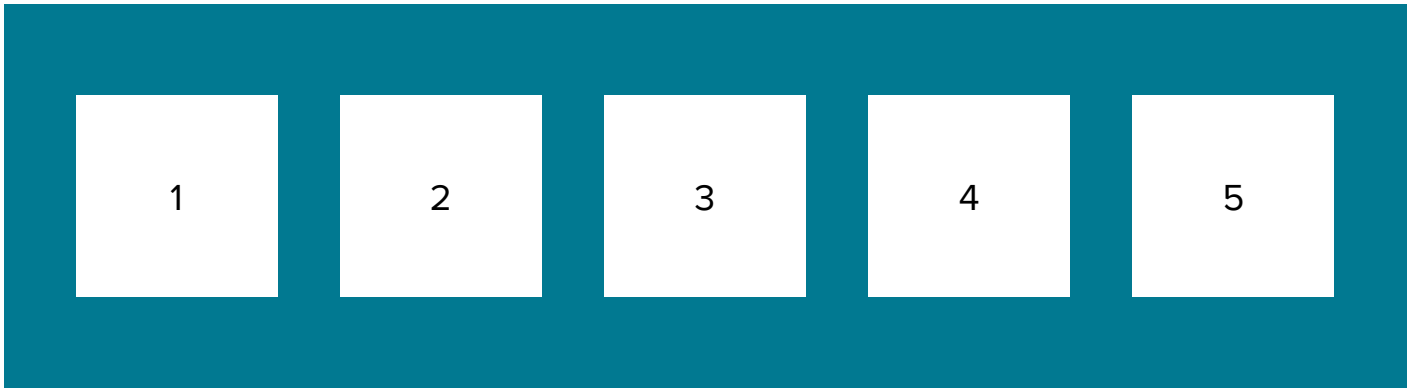
However, you can nest flex containers.

Grid

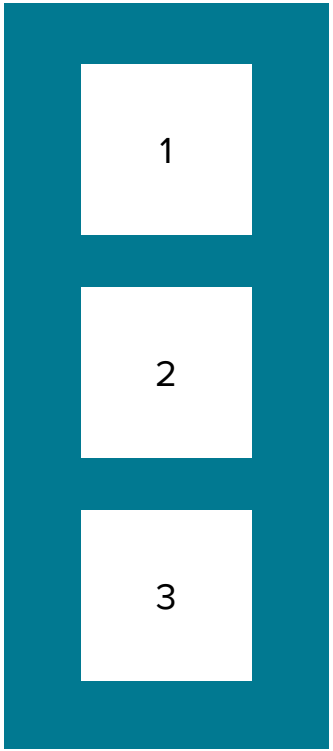
Grid works in **two dimensions** all of the time, defining rows and columns.

A clearer declaration of complicated patterns.

Flexbox Can Be a Row (`flex-direction: row;`)



Or a Column... (`flex-direction: column;`)



But flexbox can't be both!

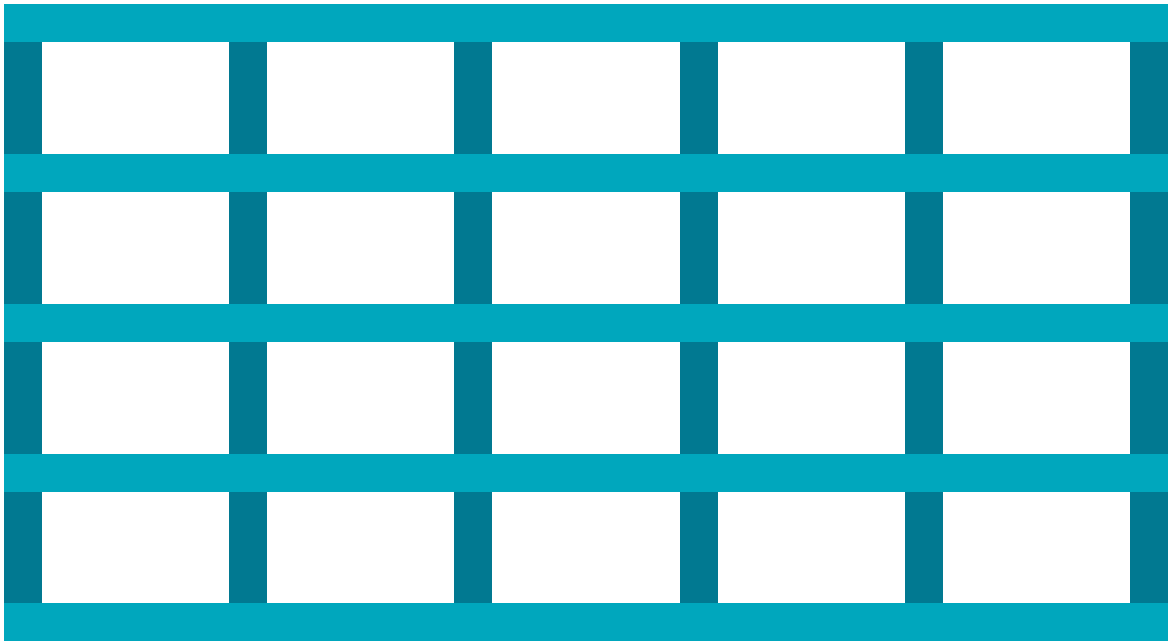
CSS Grid: A Cousin of Flexbox

- Built with the modern web in mind.
- Natively responsive.
- Parent container ➔ child Grid item relationship.
- Much like flexbox, a parent (or “container”) element will define the layout for its children.

Front-End Web Development

Using CSS Grid

Grid Tracks



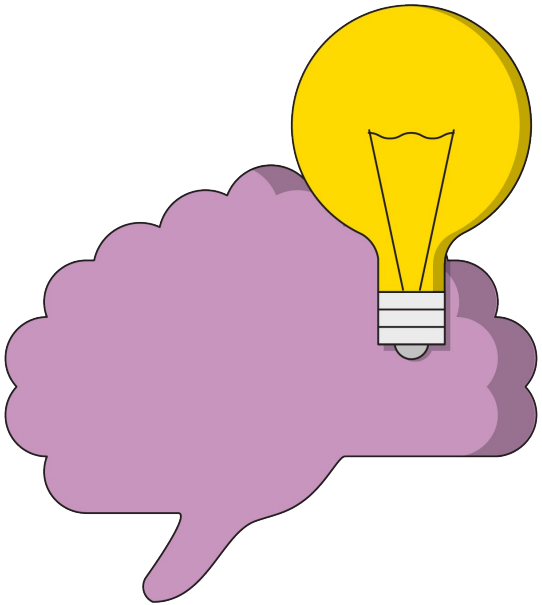
(That's what the rows and columns are called.)

Use `display:grid` on the Container Element

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

Unlike **`display:flex`**, Grid doesn't give us much by default. We'll have to define the template for our grid according to the rows and columns we want.

Think! What Kind of Layout Do You Want?



2 rows, 3 columns?

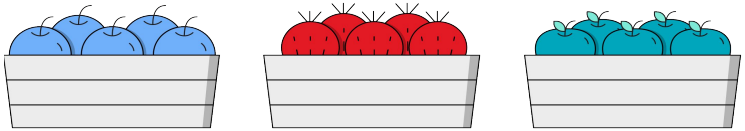
20 rows, 1 column?

9001 rows, 42 columns?

Define Your Rows and Columns

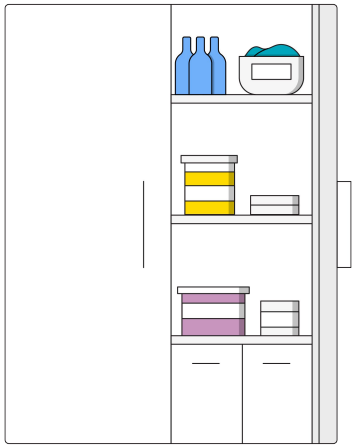
Rows

Use the CSS property **`grid-template-rows`**.



Columns

Use the CSS property **`grid-template-columns`**.



Basics in <html>

- Let's create a set of ** **s inside of **<section>** tags.

```
<section class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
</section>
```


Basics in CSS

- You'll need to use **`grid-template-columns`** and **`grid-template-rows`** on the parent to create a simple grid.

```
/* 2 column, 1 row grid */
.container {
    display: grid;
    grid-template-columns: 100px 100px;
    grid-template-rows: 100px;
}
```

All Together Now!

```
/* 2 column, 1 row grid */
.container {
  display: grid;
  grid-template-columns: 100px 100px;
  grid-template-rows: 100px;
}

.item {
 background-color: orange;
}
```

Guided Walk-Through:
A Basic Grid Example

Check out the grid in this example as a reference for using .

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18
19	20	

Reference code:

https://codepen.io/GAmarketing/pen/GRRNgzJ

Guided Walk-Through:
Flex-To-Grid Code-Along

Let's replicate grid functionality using flexbox rules!

Starter code:

https://codepen.io/GAmarketing/pen/yLLRQga

Solution code:

https://codepen.io/GAmarketing/pen/jOOeQLq

Front-End Web Development

Implicit Tracks

Implicit Tracks

You can leave off **`grid-template-rows`** and CSS Grid will automatically make the rows (conceptually, this is called the “implicit grid”).

If we don’t know how many rows our content will take, can we still define their height?

Turns out there’s a property called **`grid-auto-rows`** for just that purpose!

This Also Works

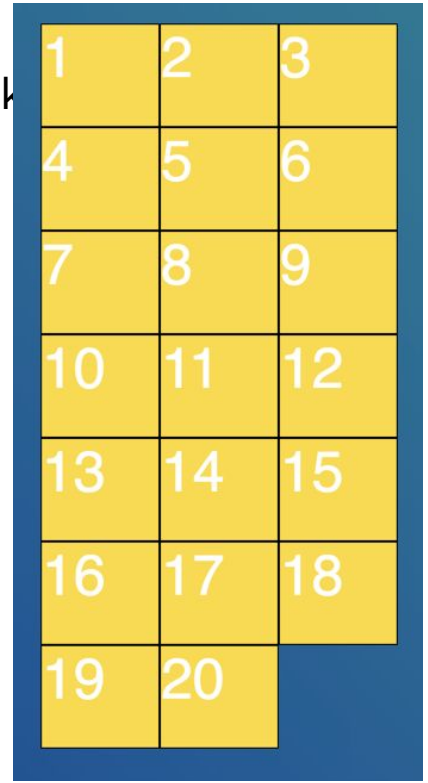
```
/* 2 column, ?? row grid */
.container {
  display: grid;
  grid-template-columns: 100px 100px;
}

.item {
 background-color: orange;
}
```


Guided Walk-Through:
An Implicit Tracks Example

Check out the grid in this example, which uses implicit track

Reference code:

https://codepen.io/GAmarketing/pen/RwwoPgR



Front-End Web Development

When You Just Need Some Space: `grid-gap`

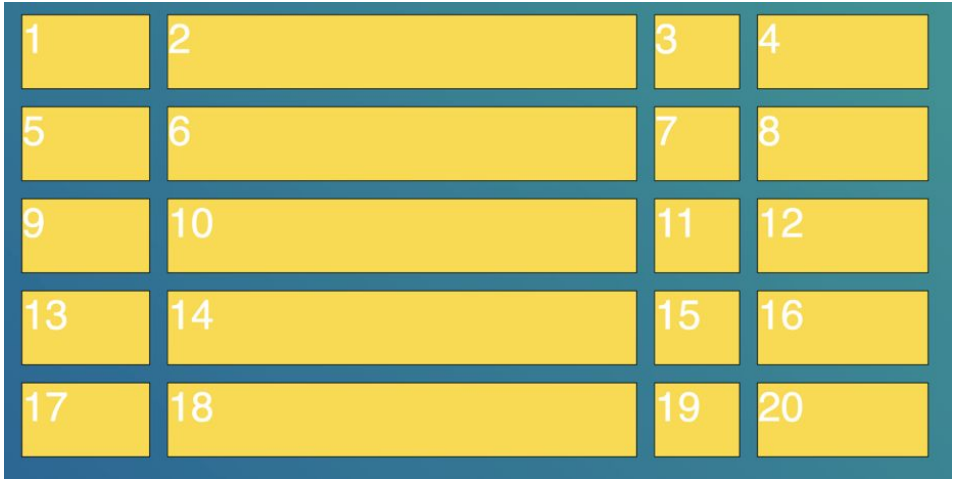
grid-gap

grid-gap is the margin/padding of CSS Grid. If you want space between your grid items, you apply **grid-gap** to the container. So easy!

```
.container {
  display: grid;
  grid-template-columns: 100px 100px;
  grid-gap: 20px;
}

.item {
 background-color: orange;
}
```


`grid-gap` (Cont.)

When **`grid-gap`** is given one value, it applies an equal amount of space between rows and columns.



Two-Value `grid-gap`

```
.grid01 {
    display: grid;
    grid-template-columns: 100px 100px;
    grid-gap: 25px 10px;
}
```

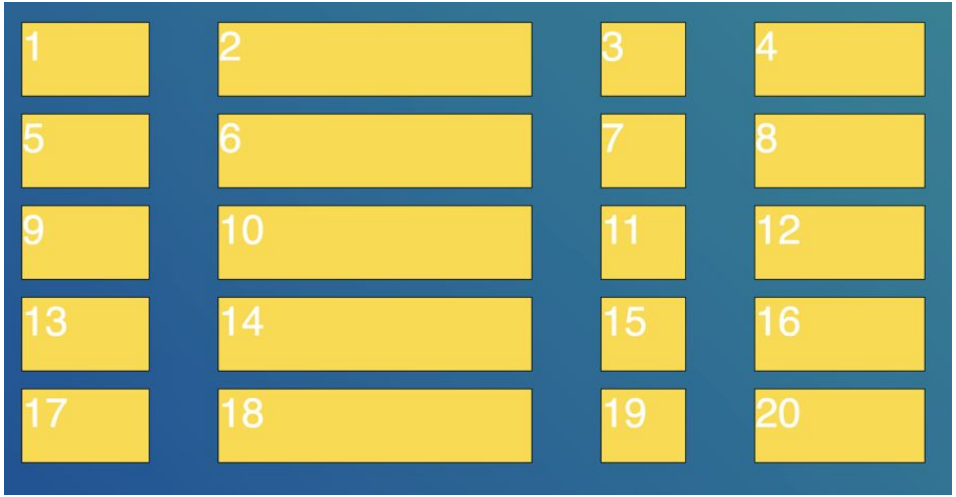
`grid-gap` (Cont.)

Keep in mind: **`grid-gap`** can take two values, which set column and row padding (respectively).

```
.container {
  display: grid;
  grid-template-columns: 100px 100px;
  grid-gap: 20px 80px;
}

.item {
 background-color: orange;
}
```


`grid-gap` (Cont.)

Note the difference when **`grid-gap`** is given two values: Different amounts of space between rows and between columns.



Guided Walk-Through:
A `grid-gap` Example

Use this CodePen as a reference for incorporating **`grid-gap`** properties.

Reference code:

https://codepen.io/GAmarketing/pen/vYYyOMx

Front-End Web Development

Fractional Units With CSS Grid

Fractional Units

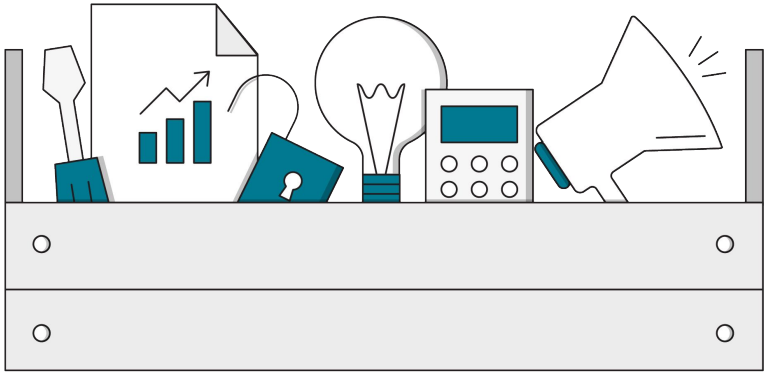
The **fractional unit** (**`fr`**) is a newer unit to be used with CSS Grid (similar to percentages). It stands for fractions of a grid container, and it intelligently takes **`grid-gap`** into account.

```
.container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr 1fr;
  grid-gap: 20px;
}
```

Can I Use Different Measurement Values Together?

We can feel free to mix fixed-width and variable-width units such as pixels, `fr`, and percentages.

They can be used together. When used together with fixed-width units, `fr` is saying: ‘Use the remainder of the space left over after the fixed-width units.’

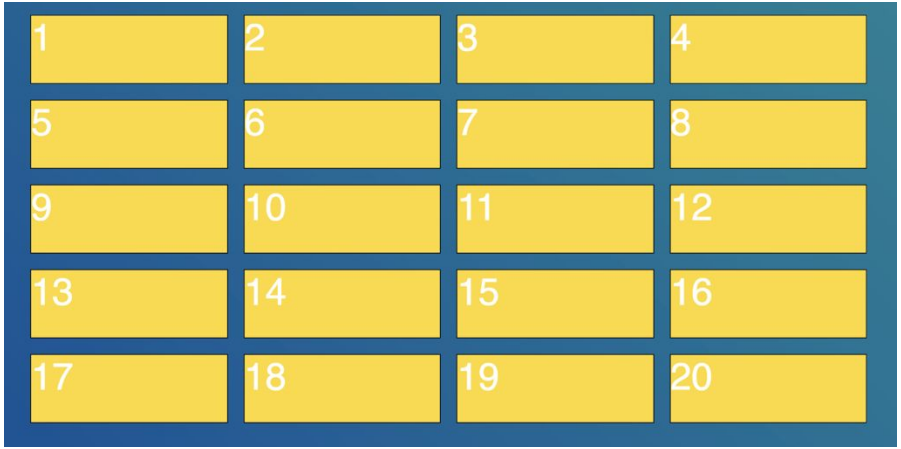


Guided Walk-Through:
A `fr` Unit Example

The following CodePen demonstrates how to use fractional units to generate this grid.

Reference code:

https://codepen.io/GAmarketing/pen/bGGBdPK



Solo Exercise:

Make a Grid With Images

20-30 minutes

Build a grid of the images provided in the starter code. The images are BIG! How can you solve for that?

Starter code:

https://codepen.io/GAmarketing/pen/zYYMZqL

Solution code:

https://codepen.io/GAmarketing/pen/WNNYppj

Front-End Web Development

Mixing Layout Methods

Combining Layout Tools

You can use CSS Grid with flexbox and even floats. They all have a role to play and a purpose in your toolkit.

- **Float** an image when you want text to wrap around it.
- **Flexbox** works great for website navigation and big horizontal sections.
- **Grid** is a great tool for large, repetitive display patterns.

Experiment! There isn't just a single way to build layouts; experience will help you select the right tool.

Guided Walk-Through:
Grid + Flexbox

See how these tools work together in this CodePen.

Reference code:

https://codepen.io/GAmarketing/pen/eYYQWeO

Key Takeaways

Grids Use Two Dimensions

- Define rows and columns in either pixels or responsive units.
- Implicit tracks can be used to expand patterns indefinitely.

For Next Time

HTML + CSS Layouts Lab

- We've learned a lot so far. Time to put it to good use!


GA