

Intro to Responsive Web Design with Flexbox

Overview

Flexbox (CSS Flexible Box Layout) is a one-dimensional layout model that makes it simple to align, distribute, and reorder elements in a container—even when their size is unknown. This guide shows how to build a responsive card grid that adapts from desktop to mobile using only Flexbox.

Table of Contents

1. What Is Flexbox?
 2. Key Flexbox Properties
 3. Setting Up the Demo Project
 4. Building a Responsive Card Grid
 5. Testing & Debugging Flex Layouts
 6. Best Practices & Resources
 7. Credits & Contact
-

1. What Is Flexbox?

Flexbox lets items “flex” to fill available space or shrink to prevent overflow, eliminating the need for complex floats or media-query math.

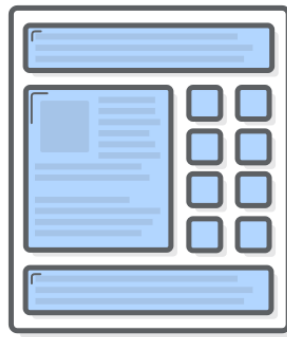
Live demo: <https://codepen.io/yourpen/FlexboxIntro>

(open link, play with the viewport slider)



FLOATS

(MAGAZINE-STYLE LAYOUTS)



FLEXBOX

(OVERALL PAGE STRUCTURE)

Property	Values	Purpose
display	flex, inline-flex	Enables flex context
flex-direction	row, column	Main axis orientation
flex-wrap	wrap, nowrap	Allow items to wrap
justify-content	center, space-between	Main-axis alignment
align-items	center, stretch	Cross-axis alignment
flex	<grow> <shrink> <basis>	Shorthand sizing

3. Setting Up the Demo Project

1. **Create project folders:** index.html, styles.css.
2. Add CDN link to **Google Fonts (Inter)** in <head>.
3. Paste boilerplate HTML:

```
<section class="cards">

  <article class="card">...</article>

  <!-- repeat 6× -->

</section>
```

4. Link styles.css.

Download starter zip: <https://github.com/yourusername/flexbox-starter/archive/main.zip>

4. Building a Responsive Card Grid

4.1 Desktop Layout

```
.cards {
  display: flex;
  flex-wrap: wrap;
  gap: 2rem;
}

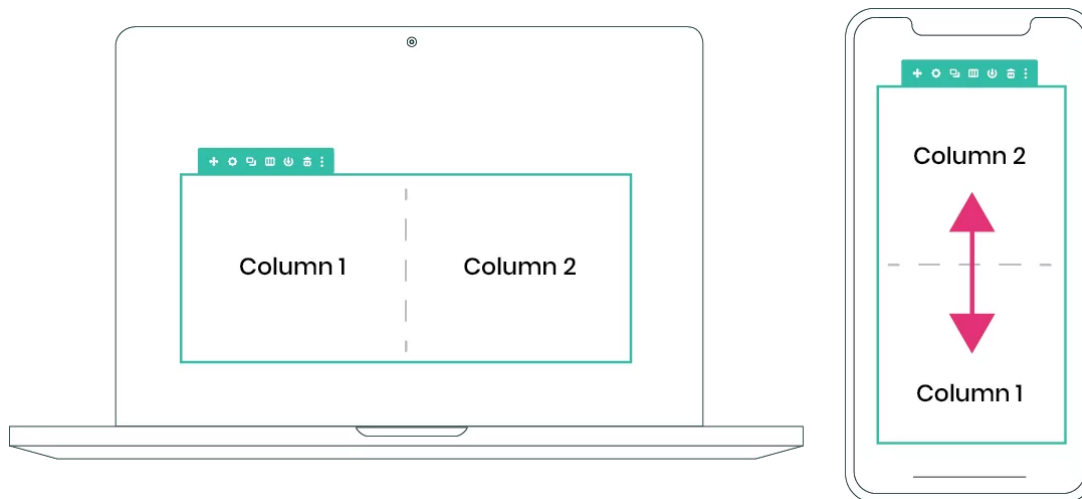
.card {
  flex: 1 1 calc(33.333% - 2rem);
}
```

Responsive 3-Column Layout



4.2 Mobile Layout

```
@media (max-width: 600px) {  
  .card {  
    flex: 1 1 100%;  
  }  
}
```

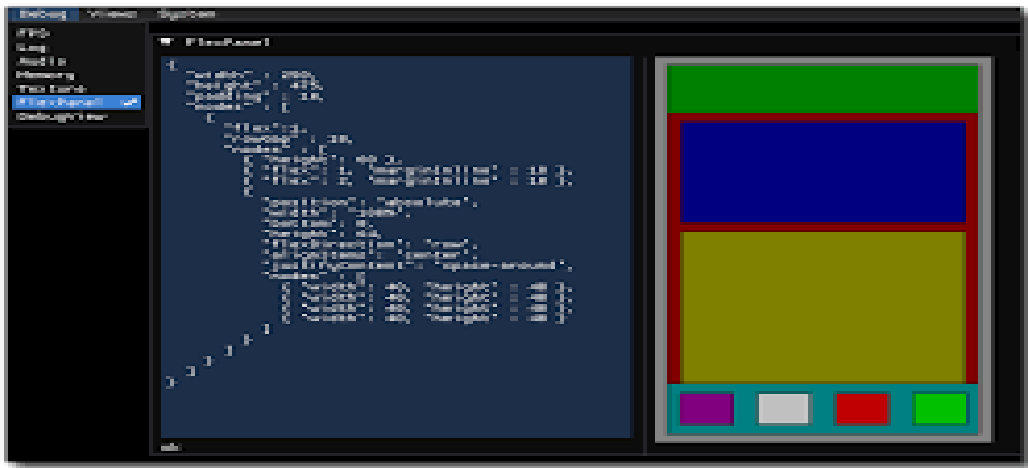


Interactive CodeSandbox: <https://codesandbox.io/s/flexbox-grid-demo>

5. Testing & Debugging Flex Layouts

Quick tips

- Toggle **Chrome DevTools** → **Layout** → **Flex** to visualize lines.
- Use `outline:1px dashed tomato;` on `.card` while debugging.
- Check for **flex-basis + min-width clashes** when items refuse to shrink.



6. Best Practices & Resources

Tip	Why
Prefer gap over margins	Cleaner spacing in modern browsers
Avoid fixed widths inside flex	Breaks responsiveness
Combine Flexbox + CSS Grid	Grid = 2-D layout; Flex = 1-D alignment
Reference	MDN Flexbox Guide, CSS-Tricks Flexbox Cheat sheet

7. Credits & Contact

Author Biruk Tilahun

GitHub <https://github.com/BirukHQ>

Feel free to share or adapt this guide—credit appreciated!