FEWD Week 2 • Class 4: Layouts

Review Time!

Objectives

- Learn the different techniques for CSS layouts
- Understand the advantages and challenges of each

Layouts in CSS

- Floats
- Flexbox
- Grid

Floats

Floats review

Floats allow us to float content either on the left or the right of its container. Elements after a floated element in the source order will wrap next to it if there is available space.

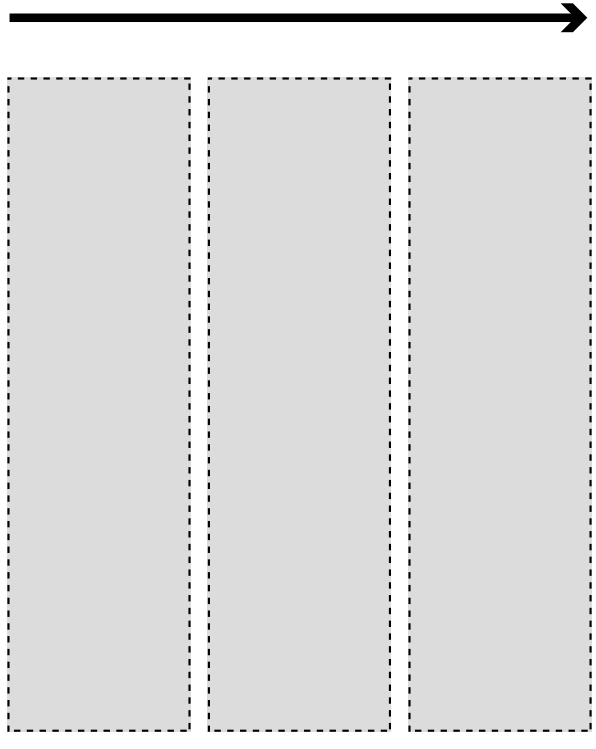
Working with Floats

Float Challenges

- ❌ Block elements still behave as block
- ❌ Margins and padding may produce undesired results
- ❌ Content cannot stretch to fill available height

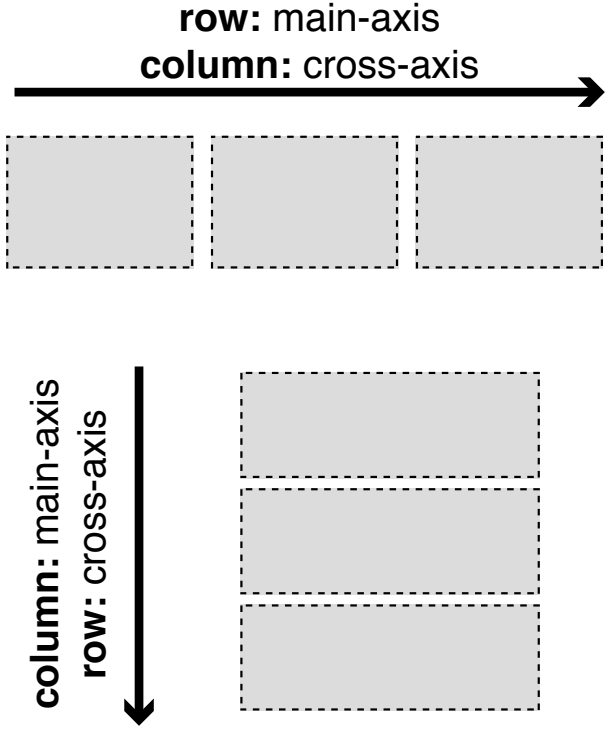
Flexbox

Introducing Flexbox



Flexbox is a newer standard. It allows us to more easily layout elements in 1 direction.

Flexbox Container




```
.parent {
  display: flex;
  /* row or column */
  flex-direction: row;
  /* main axis */
  justify-content: space-between;
  /* cross axis */
  align-items: stretch;
  /* wrap items or not*/
  flex-wrap: wrap;
}
```

Justify-content Values

flex-start

flex-end

center

space-between

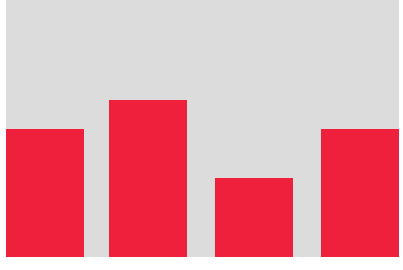
space-around

space-evenly

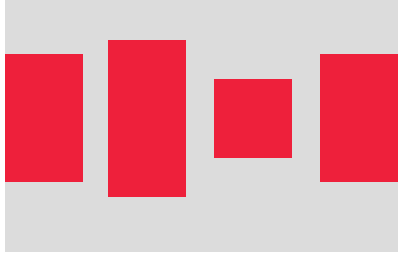
Align-items Values

flex-start

flex-end

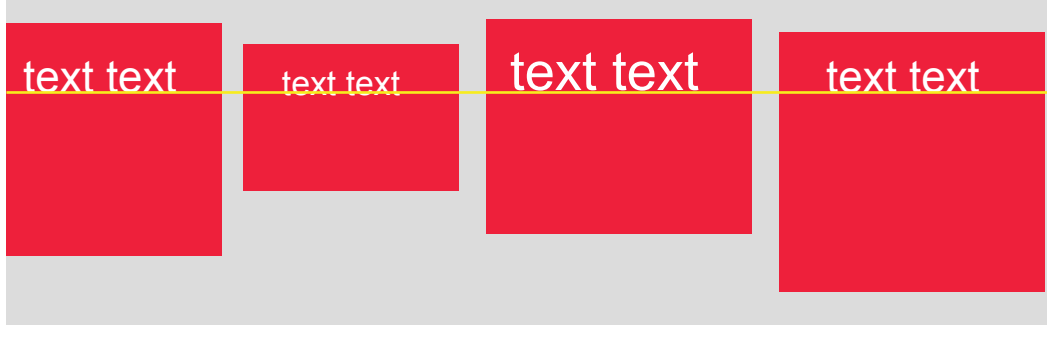


center



stretch

baseline



Flexbox Items

Alignments can be separately set for the individual items.

```
#item-2 {
  align-self: center;
}
```

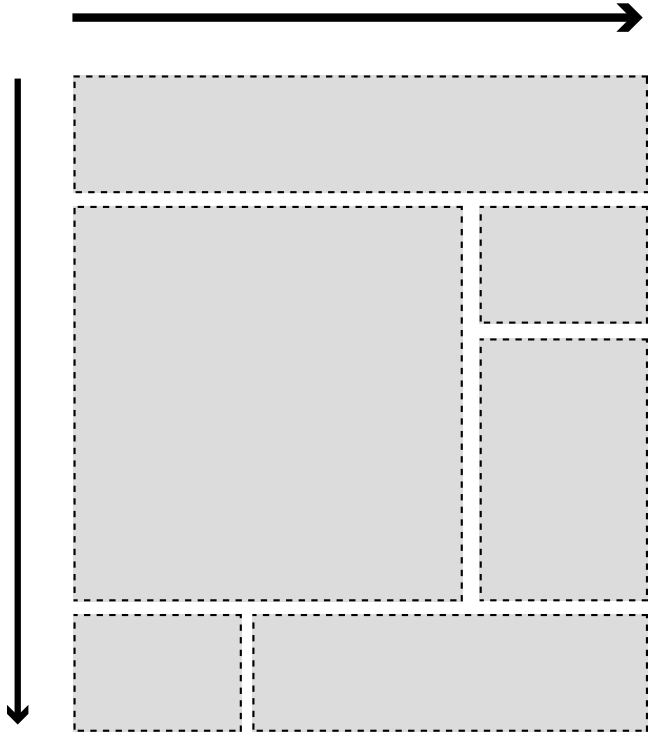
Working with Flexbox

Flexbox Challenges

- ❌ Only works in one direction
- ❌ No flex box line break
- ❌ Elements with fixed heights/width can break layouts

CSS Grid

Introducing CSS Grid



CSS Grid is a newer standard. It allows us to more easily layout elements in 2 directions!

Grid Container

Grids also start with a container that provides the basic grid definition.

```
.grid {
  display: grid;
  /* column definitions */
  grid-template-columns: 75% 25%;
  /* row definitions */
  grid-template-rows: 10vh auto auto
  /* named areas */
  grid-template-areas:
    "header header"
    "main aside"
    "footer footer";
}
```

Grid Templates

```
grid-template-columns: [line-1] 75% [line-2] 25% [line-3];
```

```
grid-template-rows: [line-1] 10vh [line-2] auto [line-3] auto [line-4];
```

Using Fractionals

The preferred unit of measurement for grids is `fr`, which stands for fractional. Fractionals are basically just a ratio of the overall grid.

```
grid-template-columns: 3fr 1fr; /* same as 75% 25% */
```

Using the Grid

We place the items in the grid with: `grid-row` and `grid-column` or `grid-area`

Grid Line Layouts

`grid-row` and `grid-column` accept the starting line number (or name) then a `/` followed by the ending line number (or name). Alternatively, you can give the start and use `span` followed by the number of rows/columns the element should span.

```
header {
  grid-column: 1 / 3; /* alt: 1 / span 2 */
}
```

Grid Area Layouts

The `grid-template-areas` accepts names as a string. Each row is surrounded by `" "`.

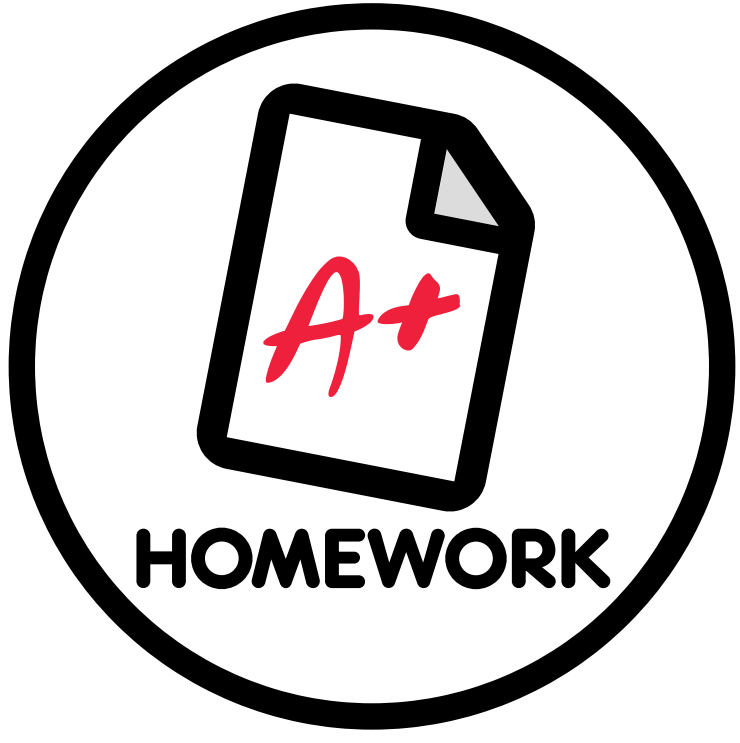
```
grid-template-areas:
  "header header" /* header will span 2 columns */
  "main aside"
  "footer footer"; /* footer will span 2 columns */
```

Using Grid Areas

To use the grid area name, we just pass it to the element!

```
header {
  grid-area: header;
}
main {
  grid-area: main;
}
```

Working with CSS Grid



Relaxr Blog

Go Do Awesome Things!