



# Responsive Web Design with Flexbox and CSS Grid

## Day 1: Flexbox

Jen Kramer @jen4web

- Wifi: Connect to MJGWorkshop  
Password: PlayTeachDeliver
- Download workshop files:  
<https://github.com/jen4web/fem-layout-2>
- Have a way to edit HTML and CSS files available.  
Make sure you have Firefox and/or Chrome installed.
- Hipster donuts up front, coffee on the way, help yourself to the fridge!
- Restrooms around corner to left



# Responsive Design

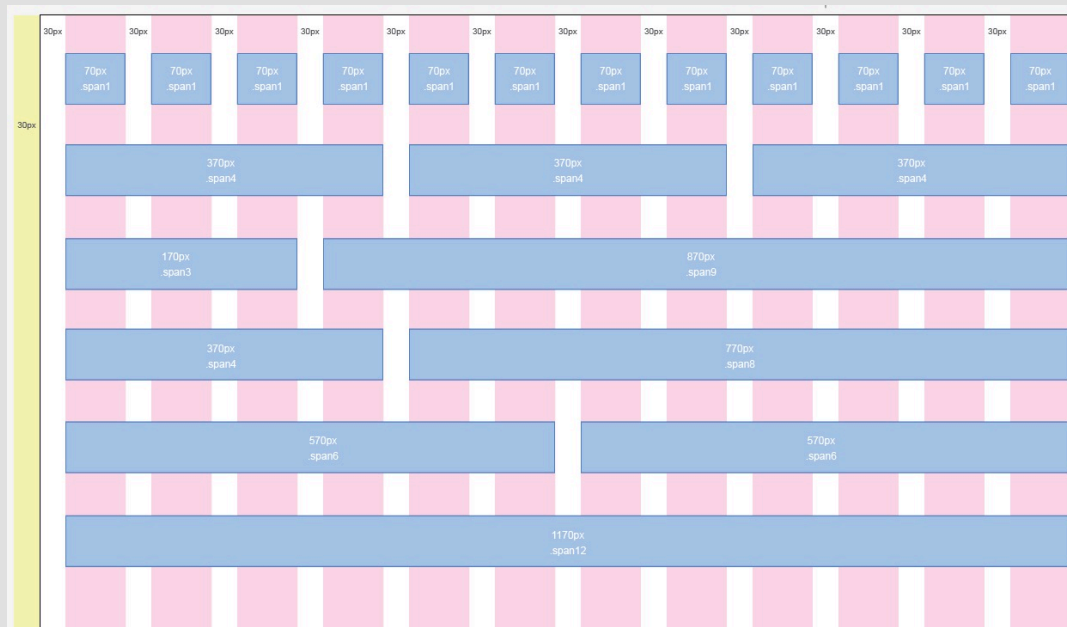
- Defined by three characteristics
  - Flexible grid-based layout
  - Media queries (CSS3)
  - Images that resize

[www.alistapart.com/articles/responsive-web-design/](http://www.alistapart.com/articles/responsive-web-design/)

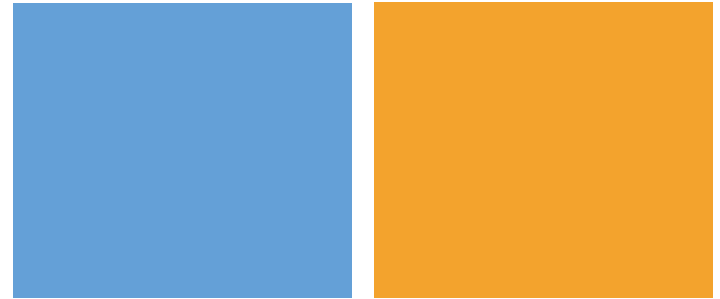
<https://ethanmarcotte.com/wrote/responsive-design-at-11-container-queries/>



# Grid-based layout

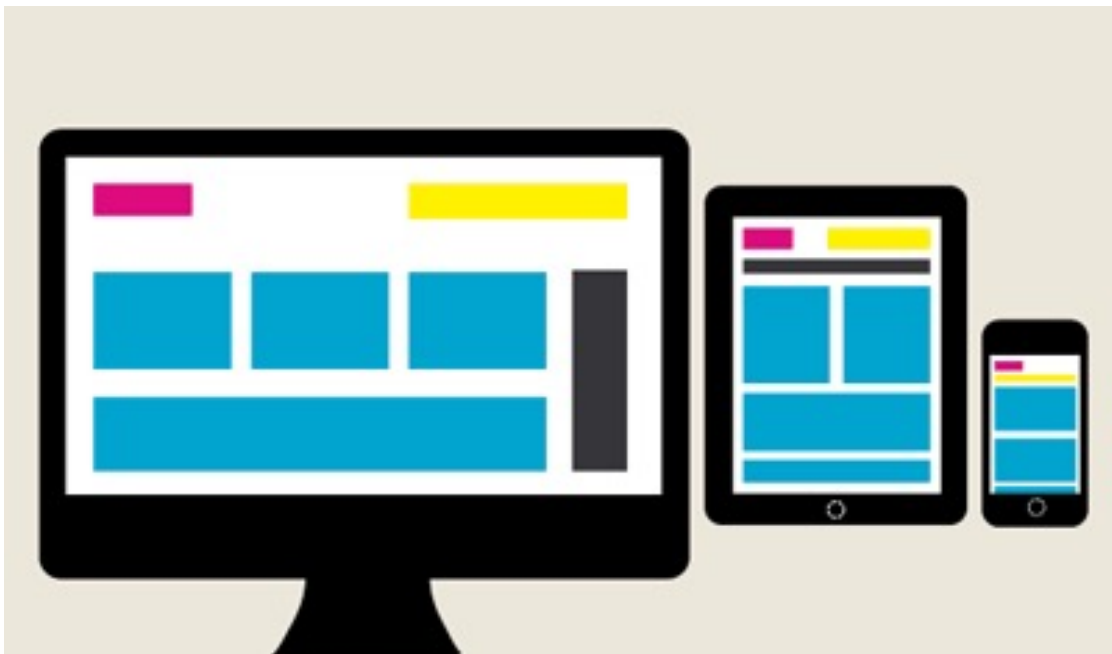


# Images that resize



- Images should change size, based on screen resolution
- Today's solutions include `<picture>`, *sizes*, and *srcset*, part of HTML spec





# CSS Media Queries

- Browser reports screen resolution
- Based on current width, serve a stylesheet with layout for that width
- No JavaScript involved
  
- PS – media queries go beyond screen size!
  - Aspect-ratio
  - orientation

How does  
this hold  
up today?

Grid-based layout associated with frameworks

Images that resize are often forgotten

Devs try to avoid media queries

CONTAINER QUERIES / SUBGRID  
WILL SOLVE ALL (ahem)

# Consider these guiding principles



**GRID-BASED  
LAYOUT**



**IMAGES THAT  
RESIZE**



**MEDIA QUERIES**



The word "Flexbox" is centered within a large white circle. The circle is set against an orange background. On the left side of the circle, there is a dashed yellow arc. On the bottom right edge of the circle, there is a small solid blue circle.

Flexbox

# Flexbox

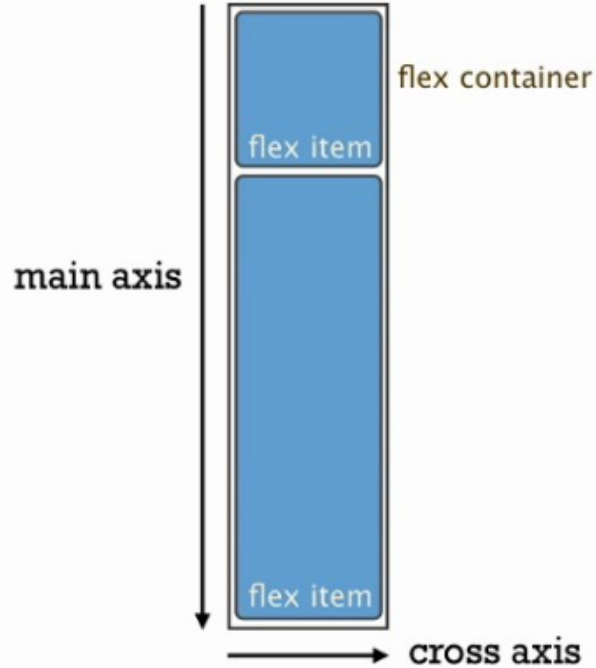
- The first layout elements – but not envisioned to lay out whole web pages
- Excels at vertical centering and equal heights
- Very easy to reorder boxes

**Disadvantage:**

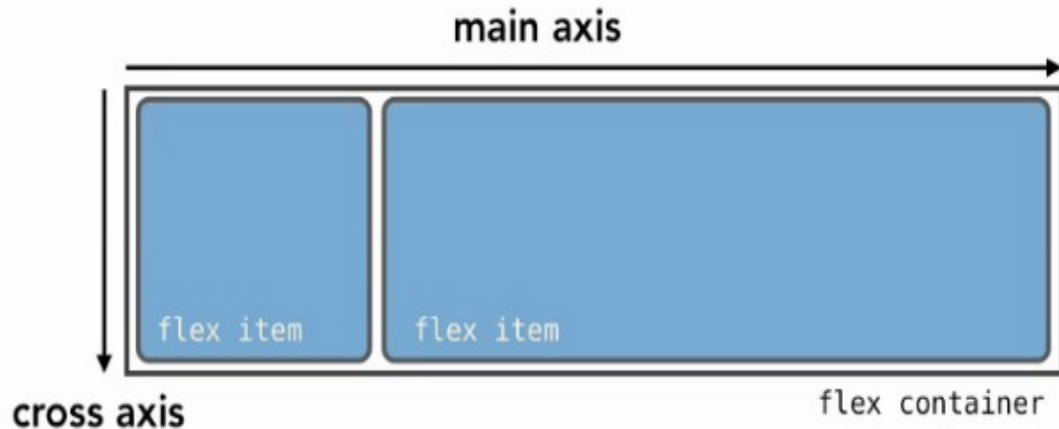
**Wasn't designed to be locked down for layouts!**  
**Works in 1 dimension only.**



# Flex container set to column



# Flex container set to row



# Three versions of Flexbox

- 2009: display: box;
- 2011: display: flexbox; (“tweener” syntax)
- 2016: display: flex;
- Generally good cross-browser compatibility except for IE



Games to practice Flexbox syntax:

- Flexbox Froggy:  
<http://flexboxfroggy.com/>
- Flexbox Defense:  
<http://www.flexboxdefense.com/>

# Flexbox properties

## Parent (Flex Container)

`display: flex | inline-flex;`

`flex-direction: row | row-reverse | column | column-reverse;`

`flex-wrap: wrap | nowrap | wrap-reverse;`

`flex-flow` (shorthand for `flex-direction` and `flex-wrap`)

`justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;`

`align-items: flex-start | flex-end | center | baseline | stretch;`

`align-content` (cross axis - adjust to largest item): `flex-start | flex-end | center | stretch | space-between | space-around;`

## Children (Flex Items)

`order: <integer>;`

`flex-grow: <number>;`

`flex-shrink: <number>;`

`flex-basis: <length> | auto;`

`flex`: shorthand for `grow`, `shrink`, and `basis` (default: `0 1 auto`)

`align-self`: overrides alignment set on parent

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>



Flexbox grid



.ROW/CONTAINER

ITEM/  
.COL-1

ITEM/  
.COL-1

ITEM/  
.COL-1

ITEM/  
.COL-1

```
<div class="row">  
  <div class="col-1"></div>  
  <div class="col-1"></div>  
  <div class="col-1"></div>  
  <div class="col-1"></div>  
</div>
```

```
.row {  
  display: flex;  
  flex-flow: row wrap;  
  justify-content: center;  
  gap: 1%;  
}
```

```
.col-1 {  
  
  flex-basis: 24%;  
  /* desktop */  
  
  flex-basis: 48%;  
  /* tablet */  
  
  flex-basis: 98%;  
  /* phone */  
  
}
```

```
/* rearranging the columns */
```

```
.col-push-1 {  
    order: 2;  
}  
.col-pull-3 {  
    order: 1;  
}
```