

LEARNING OBJECTIVES

- › Describe responsive design.
- › Know the difference between fluid, fixed and responsive layouts
- › Apply media queries to achieve a responsive layout.

AGENDA

Review

Responsive — Layout Design

Responsive — Media Queries

Responsive — REM/EM

RESPONSIVE BASICS

REVIEW

FEWD

FLEX CONTAINER

PARENT WRAPPER (FLEX CONTAINER)

When you declare `display: flex;` on a container, it becomes a **flex container**.

parent wrapper:
flex container

```
<section>
  <article>
    ...
  </article>
  <article>
    ...
  </article>
  <article>
    ...
  </article>
</section>
```

```
section {
  display: flex;
}
```

ACTIVITY



EXERCISE

KEY OBJECTIVE

- ▶ Review Flexbox / Preview Media Queries

TYPE OF EXERCISE

- ▶ Individual/Partner

TIMING

15 min

1. Look at this [Codepen](#) with a partner and talk through steps 1 - 7.
2. Look up any flex properties you're unfamiliar with in this [guide](#).

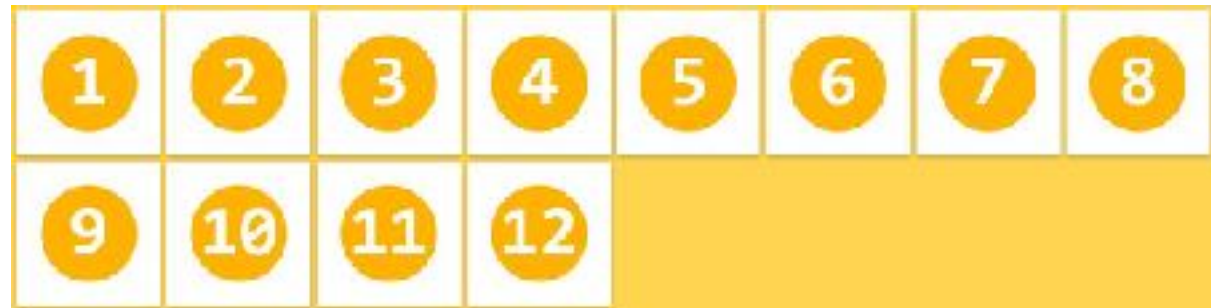
HOW DOES FLEXBOX WORK?

The initial flexbox concept is the container to set its items in one single line. The `flex-wrap` property controls if the flex container lay out its items in single or multiple lines, and the direction the new lines are stacked in.

```
section {  
  flex-wrap: nowrap;  
}
```



```
section {  
  flex-wrap: wrap;  
}
```



HOW DOES FLEXBOX WORK?

The justify-content property aligns flex items along the main axis of the current line of the flex container. It helps distribute left free space when either all the flex items on a line are inflexible, or are flexible but have reached their maximum size.

```
section {  
  display: flex;  
  justify-content: space-between;  
}
```



FEWD

MULTI-COLUMN LAYOUT

STEPS TO ACHIEVE A MULTI-COLUMN LAYOUT

1. MAKE SURE ALL THE CONTENT FOR EACH COLUMN HAS ONE WRAPPER AROUND IT IN YOUR HTML

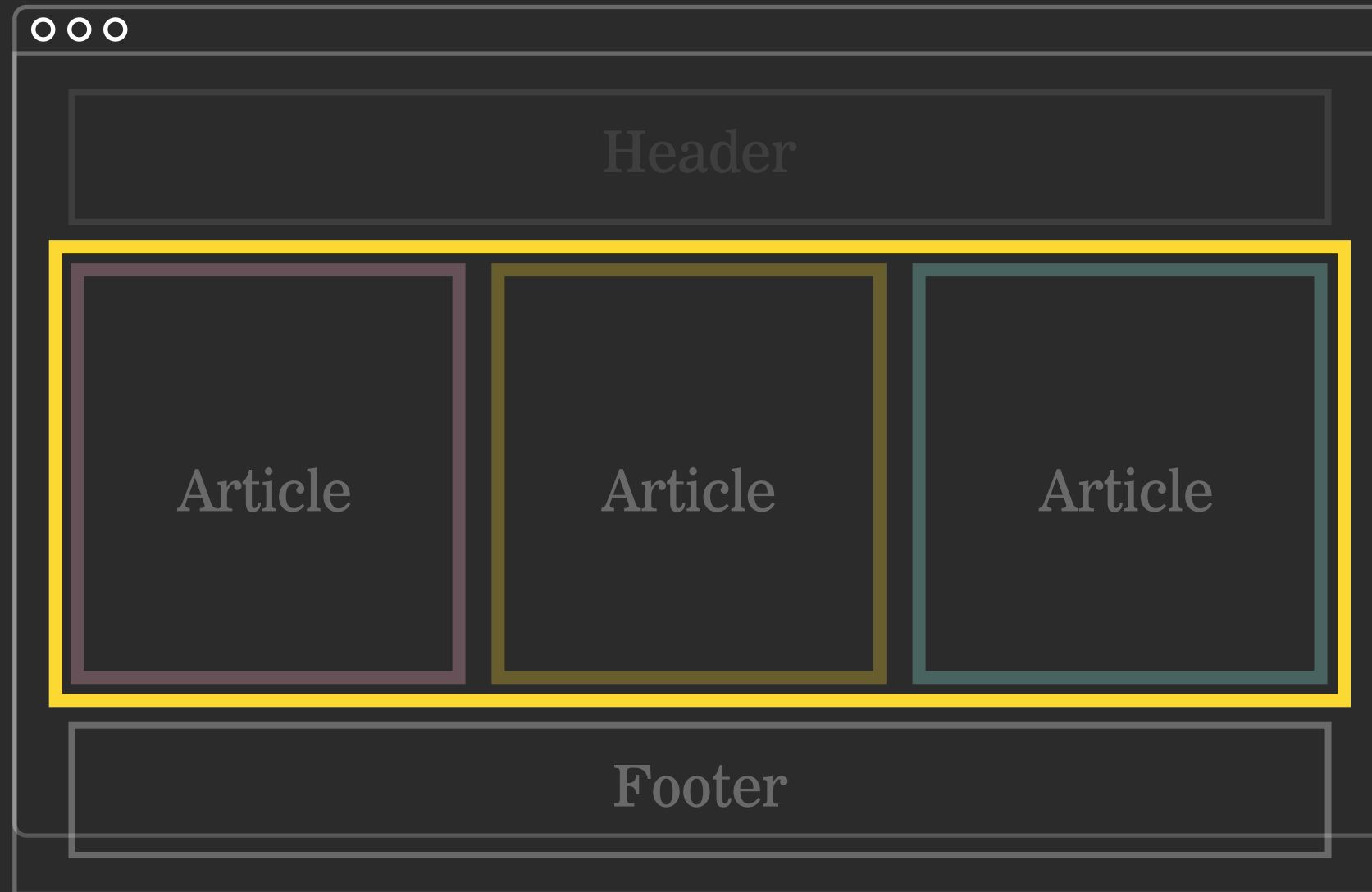
2. WRAP THE COLUMNS WITH A PARENT CONTAINER, THE "FLEX CONTAINER"

3. ADD DISPLAY: FLEX; TO FLEX-CONTAINER

4. GIVE EACH COLUMN A WIDTH IN PERCENTAGES

5. OPTIONAL: USE JUSTIFY-CONTENT ON FLEX CONTAINER TO DETERMINE COLUMN SPACING

CSS — MULTI-COLUMN LAYOUT



ACTIVITY



EXERCISE

KEY OBJECTIVE

- ▶ Review HTML/CSS Layouts

TYPE OF EXERCISE

- ▶ Individual/Partner

TIMING

15 min

1. Use HTML and CSS to recreate albums_1.png

RESPONSIVE BASICS

RESPONSIVE — LAYOUT DESIGN

RESPONSIVE DESIGN

“Day by day, the number of devices, platforms, and browsers that need to work with your site grows. Responsive web design represents a fundamental shift in how we’ll build websites for the decade to come.”

- Jeffrey Veen

RESPONSIVE DESIGN



RESPONSIVE DESIGN



ACTIVITY



EXERCISE

KEY OBJECTIVE

- ▶ Use HTML/CSS to create a mobile layout

TYPE OF EXERCISE

- ▶ Individual/Partner

TIMING

10 min

1. Open the main.css file from the first exercise and place a comment at the **bottom** (something like, /*overwriting CSS for new layout goes here*/).
2. Below this line, add CSS that will make the original page look like the albums_2.png.

RESPONSIVE BASICS

RESPONSIVE — MEDIA QUERIES

RESPONSIVE — TYPES OF LAYOUTS

FIXED VS. RESPONSIVE

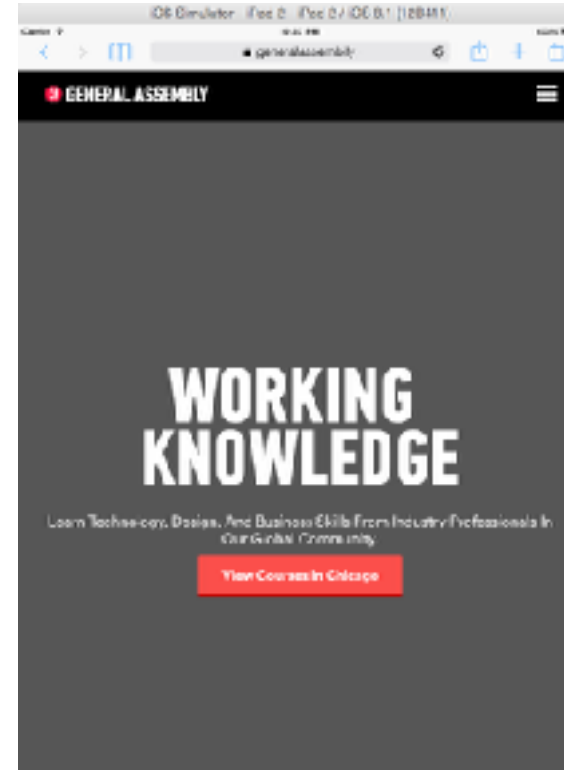
CHECK OUT THESE FIXED SITES:

- ups.com
- colourpixel.com



CHECK OUT THESE RESPONSIVE SITES:

- GeneralAssembly.ly
- KinHR.com



FIXED LAYOUT

- Relies on a container of a fixed width (uses static units)
- Resizing the browser/viewing it on a different device won't have an effect on the page

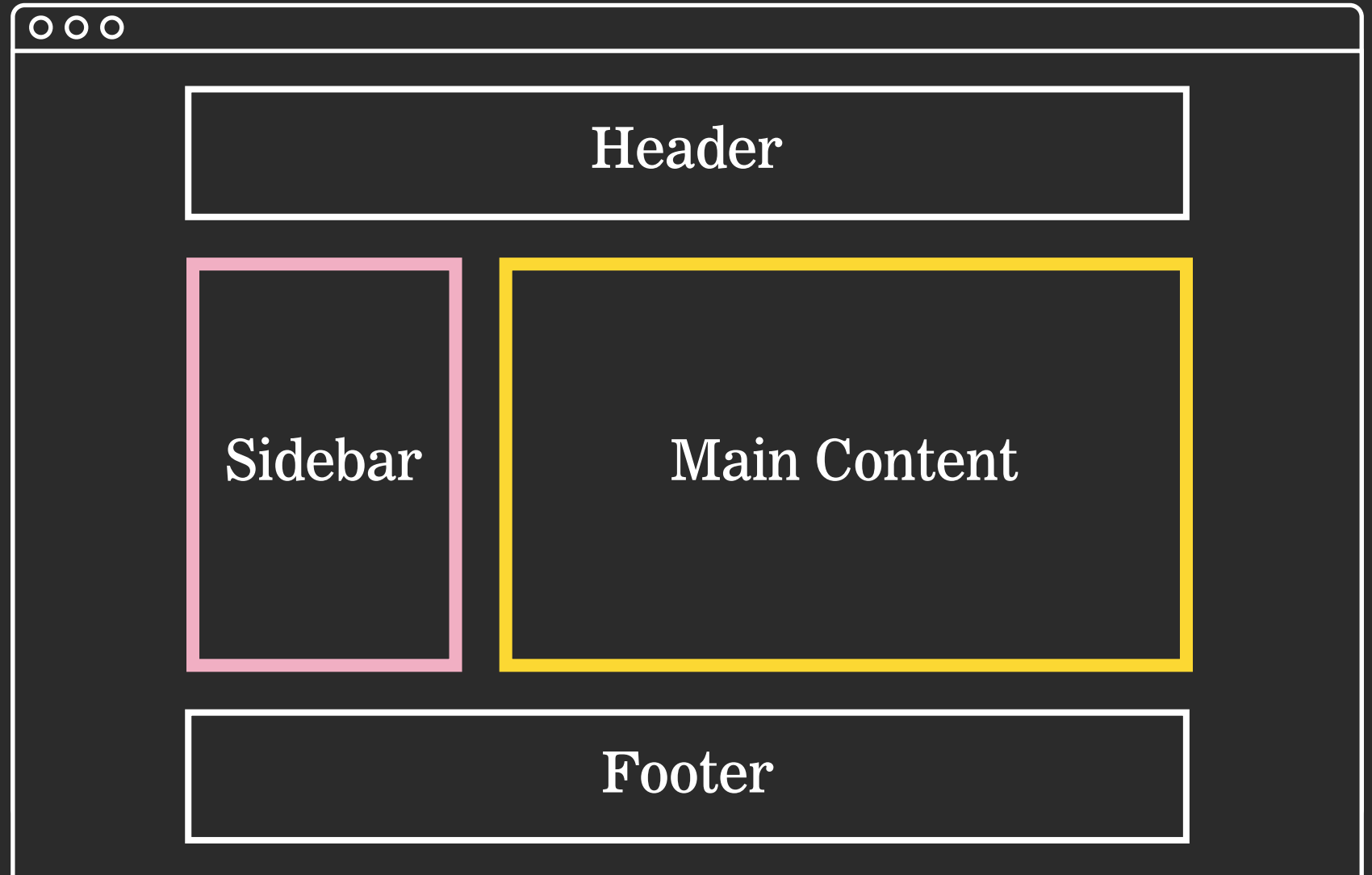


FIXED WIDTH LAYOUT

Fixed width layouts do not change size as the user increases/decreases width of browser window

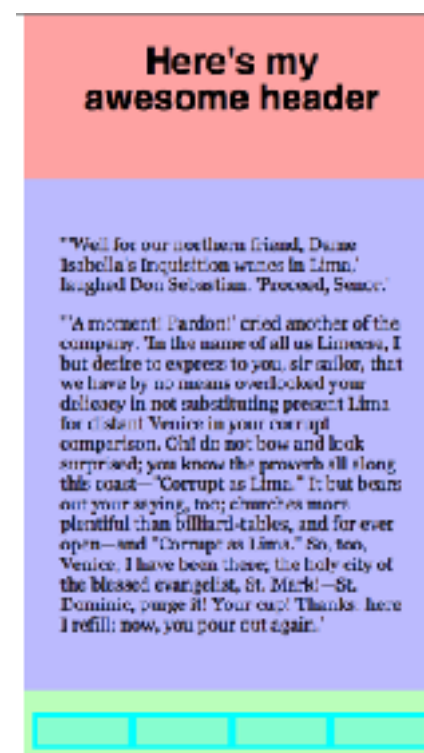
To create:

- Width of any main boxes is set in pixels
- Layout can be centered by setting the value of the left and right margins to auto



FLUID LAYOUT

- Uses relative widths (percentages)
- No media queries

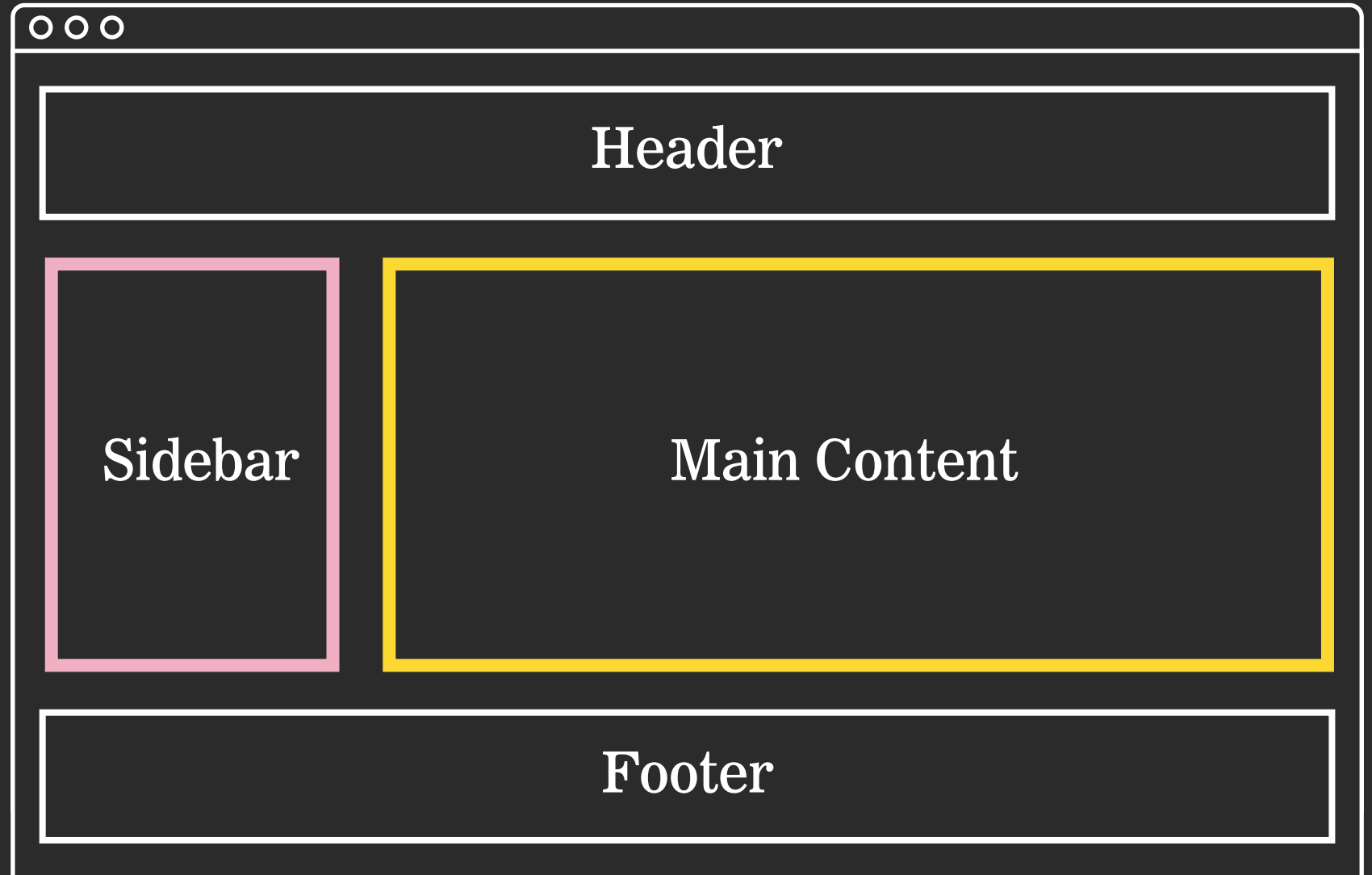


FLUID LAYOUT

Fluid layouts stretch and contract as the user increases/decreases the size of their browser window

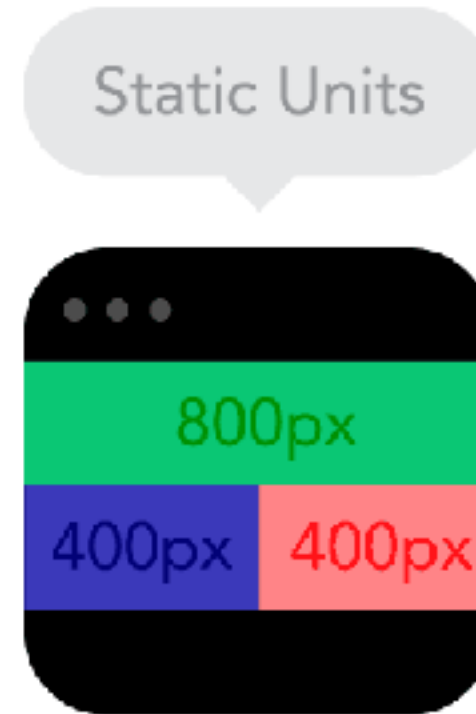
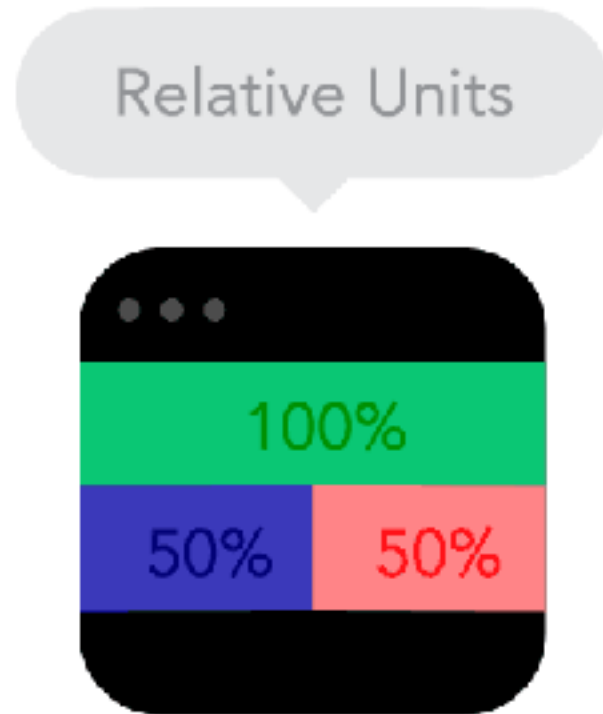
To create:

- Uses percentages to set the width of each box so that the design will stretch to fit the size of the screen



FIXED VS. FLUID

Fluid layout

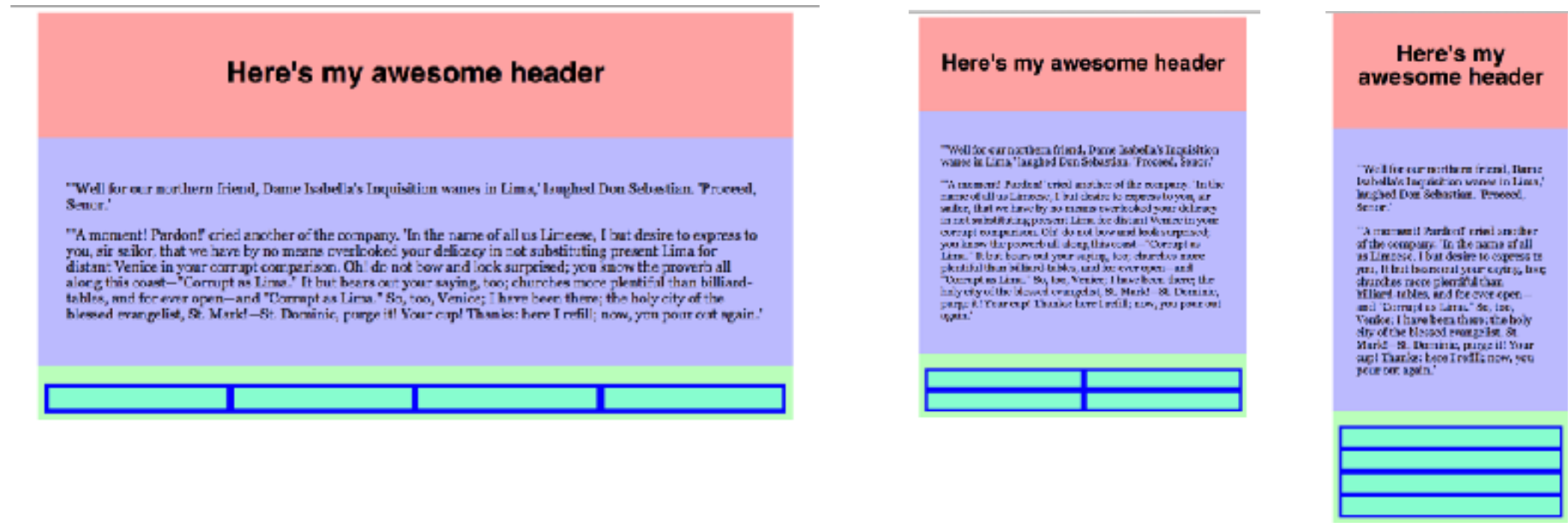


Fixed

Gif credit: [Fast Company](#)

RESPONSIVE LAYOUT

- ▶ Uses relative widths (built on a fluid grid)
- ▶ Use media queries to control design and content as it scales down or up with the browser or device



WITH BREAKPOINTS VS. WITHOUT BREAKPOINTS

With Breakpoints

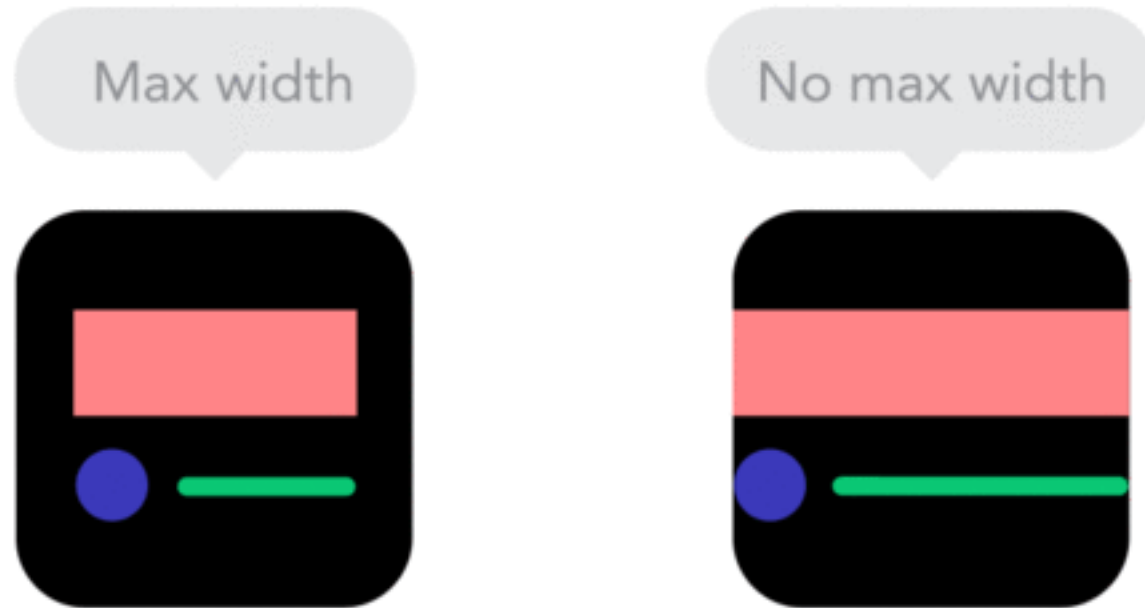


Without Breakpoints



Gif credit: [Fast Company](#)

MAX-WIDTH — A HELPFUL TOOL FOR LAYOUT



Gif credit: [Fast Company](#)

ACTIVITY



EXERCISE

KEY OBJECTIVE

Describe the difference between fixed, fluid and responsive layouts.

TIMING

1 min

1. Turn to partner and discuss

2 min

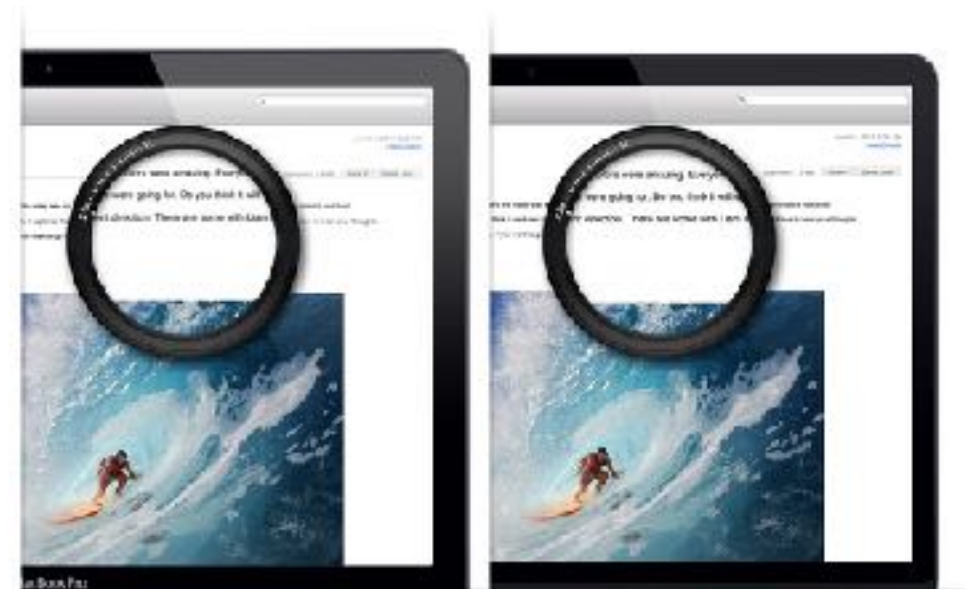
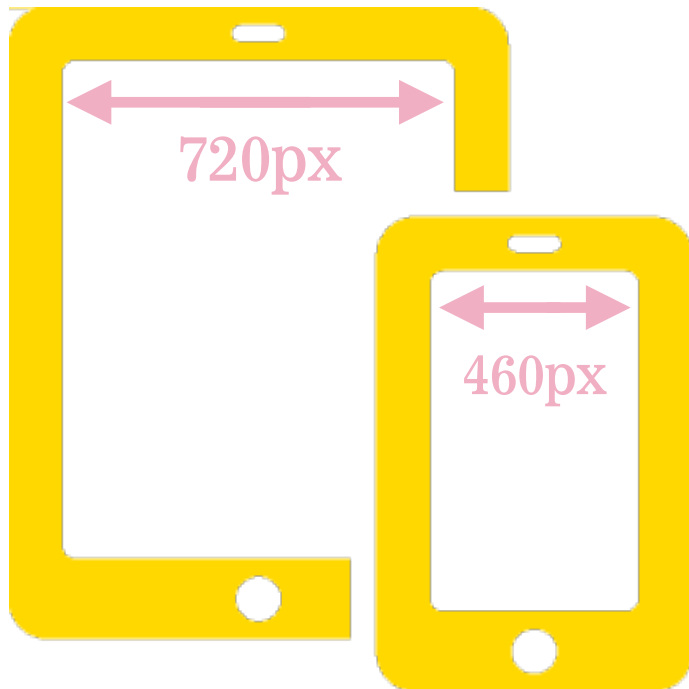
2. Share with rest of class

RESPONSIVE BASICS

RESPONSIVE — MEDIA QUERIES

MEDIA QUERIES

- Media queries allow us to target CSS rules based on screen size, device orientation, display density, etc.



MEDIA QUERIES

- ▶ We can use media queries to allow certain rules to apply for an iPad or iPhone, to add styles for a printer, or to create a responsive site.
- ▶ With media queries, we can allow most of our styles to remain the same, while we make **small tweaks for specific formats**.




ADDING MEDIA QUERIES

```
@media screen and (max-width: 600px) {  
  section {  
    width: 50%;  
  }  
}
```

MEDIA QUERIES — SYNTAX

STYLES FOR COMPUTER SCREENS

```
@media  screen and (max-width: 600px) {  
  section {  
    width: 50%;  
  }  
}
```

STYLES FOR PRINT

```
@media  print {  
  /* All your print styles go here */  
  header, footer, nav {  
    display: none;  
  }  
}
```

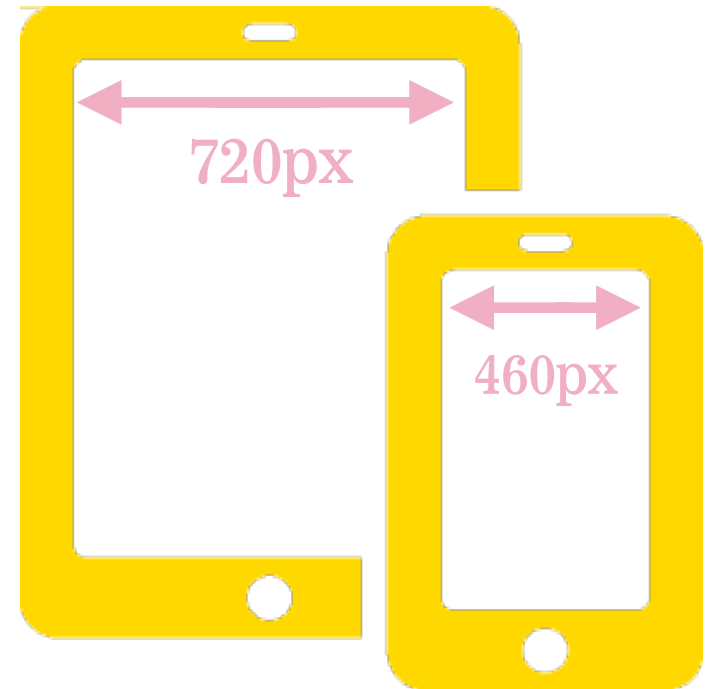
MEDIA QUERIES — SYNTAX

MAX-WIDTH: SCREENS 600PX OR LESS

```
@media screen and (max-width: 600px) {  
  section {  
    width: 50%;  
  }  
}
```

MIN-WIDTH: SCREENS 600PX OR MORE

```
@media screen and (min-width: 600px) {  
  section {  
    width: 50%;  
  }  
}
```



MEDIA QUERIES — ORDER

1. Place your media queries **at the bottom of your stylesheet** so they can override code that comes before them.
2. If you have multiple media queries, they should go from largest to smallest (max-width: 600 would be placed above max-width: 400)

```
@media screen and (max-width: 600px) {  
  section {  
    width: 50%;  
  }  
}
```

```
@media screen and (max-width: 400px) {  
  section {  
    width: 100%;  
  }  
}
```

VIEWPORT META TAG — AN IMPORTANT NOTE!!

- The viewport meta tag controls how a webpage is displayed on a mobile device.
- Without the tag, mobile devices will assume you want the full desktop experience and will set the viewport width at 980px (iOS)

DEVICE-WIDTH

- This tells the browser “My Website adapts to your width”

INITIAL-SCALE

- Sets the initial zoom level and prevents default zooming

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

ACTIVITY



EXERCISE

KEY OBJECTIVE

- ▶ Apply media queries to achieve a responsive layout.

TYPE OF EXERCISE

Individual / Partner

TIMING

25 min

1. Demo the site
2. Add media queries to make albums_2.png exercise responsive.
3. Add a media query for a max-width of 450px
4. Challenge: create versions for max-width of 660px and a max-width of 960px (images in starter code folder)

RESPONSIVE BASICS

ASSIGNMENT

ACTIVITY



EXERCISE

KEY OBJECTIVE

- ▶ Practice applying media queries to achieve a responsive layout.

TYPE OF EXERCISE

- ▶ Individual/Partner

TIMING

5 min

1. Spend 5 minutes sketching out the responsive version of the Monument Landing page with a group

20 min

2. Write media queries at the bottom of your CSS file to create a responsive version of the Monument Landing Page.

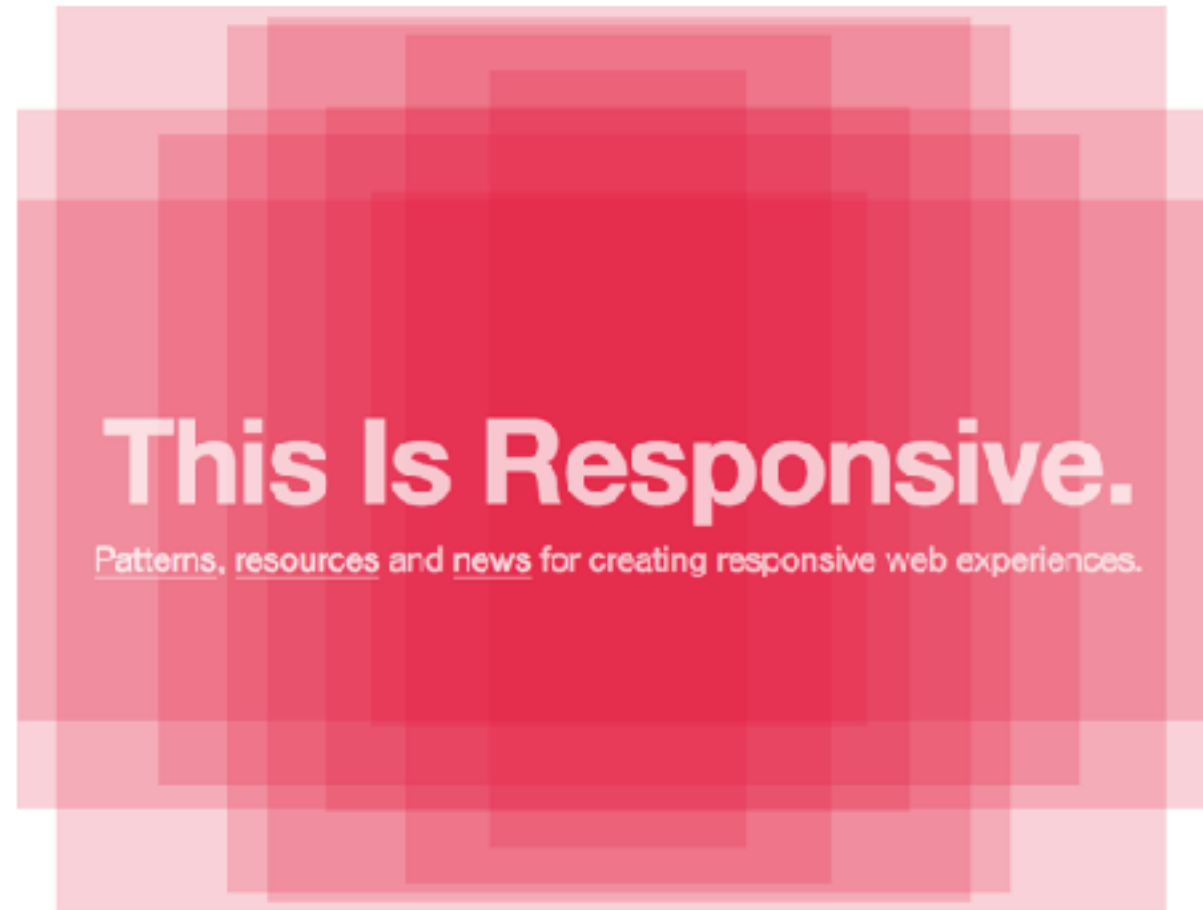
RESPONSIVE BASICS

MORE RESOURCES

MORE RESOURCES



MORE RESOURCES — THIS IS RESPONSIVE



MORE RESOURCES — MEDIA QUERIES



FROONT




International Polar Foundation



MORE RESOURCES — REMS/EMS

jeremy(hurch

[/index](#) [/about](#) [/contact](#) 

February 24, 2014

Confused About REM and EM?

REM can be confusing, especially without a solid understanding of its partner EM and their archvillain, the PX.

Relative Units

Both rem and em are relative units. px is not. Before considering rem, it's important to understand the relationship between em, markup and inheritance.

Below the example demonstrates how each nested child assumes the parent is 3em (1.875em). Thus children inherit size by scaling in relation to the parent font size.

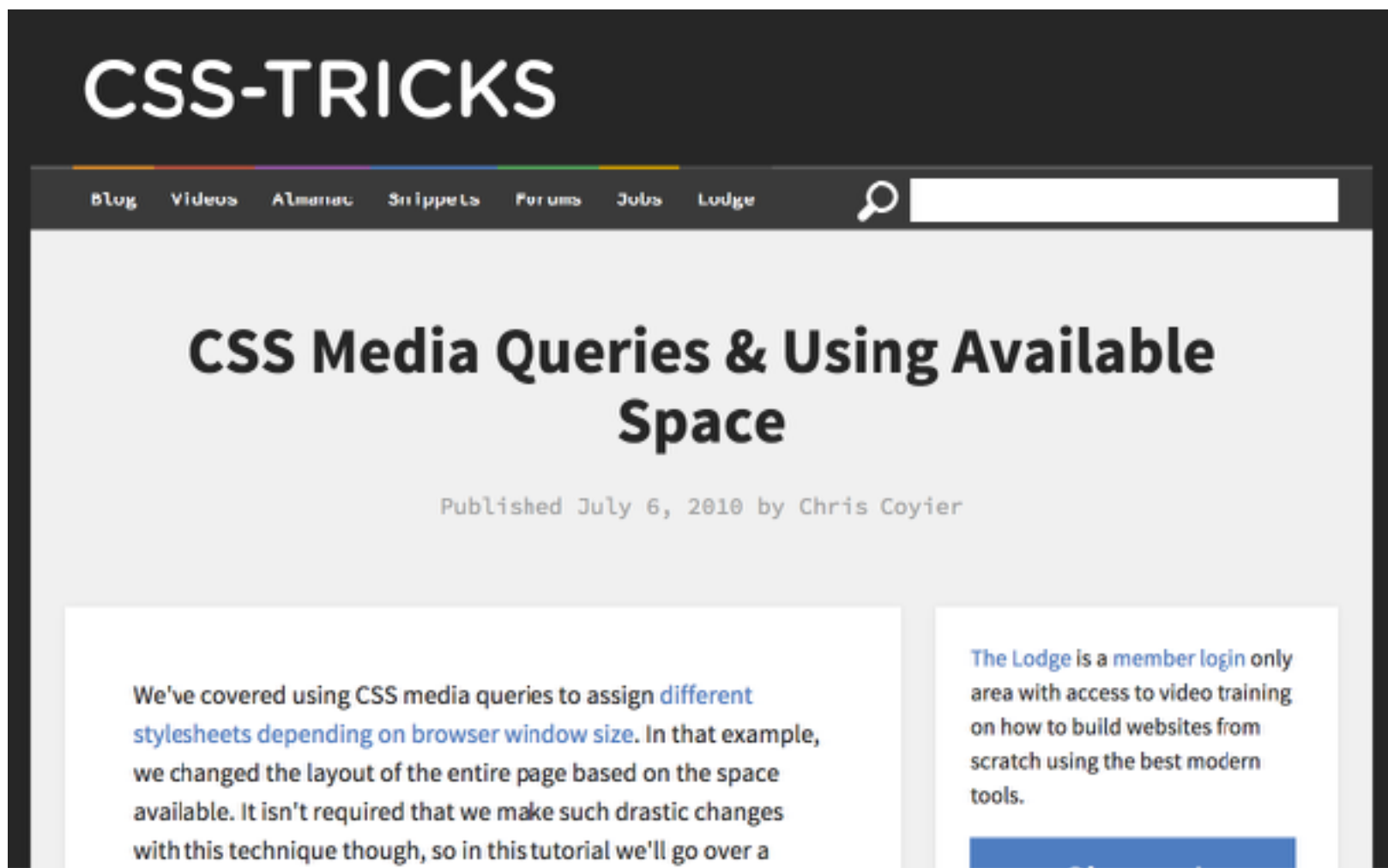
EM values inherit from their parent

HTML	Sass	Result	Edit
	html { font-size: 1.375em; }	100% (22px)	
	.font_small { font-size: 0.773611; }	77.3% (17px)	
	.font_small { font-size: 0.773611; }	77.3% (17px)	

PX values don't inherit

HTML	Sass	Result	Edit
	html { font-size: 22px; }		
	.font_small { font-size: 17px; }		
	.font_small { font-size: 17px; }		

MORE RESOURCES — MEDIA QUERIES



LEARNING OBJECTIVES

- › Describe responsive design.
- › Know the difference between fluid, fixed and responsive layouts
- › Apply media queries to achieve a responsive layout.

WEEKLY OVERVIEW

WEEK 8

Responsive Basics / Final Project Lab

WEEK 9

Student's Choice / Final Project Lab

WEEK 10

Final Project Lab / Presentations

HTML BASICS

EXIT TICKETS