MONUMENT LAB

WEEKLY OVERVIEW

WEEK 3	Layout Lab / Layout Lab & Wireframing
WEEK 4	Responsive Design / CSS Positioning & Animation

LEARNING OBJECTIVES

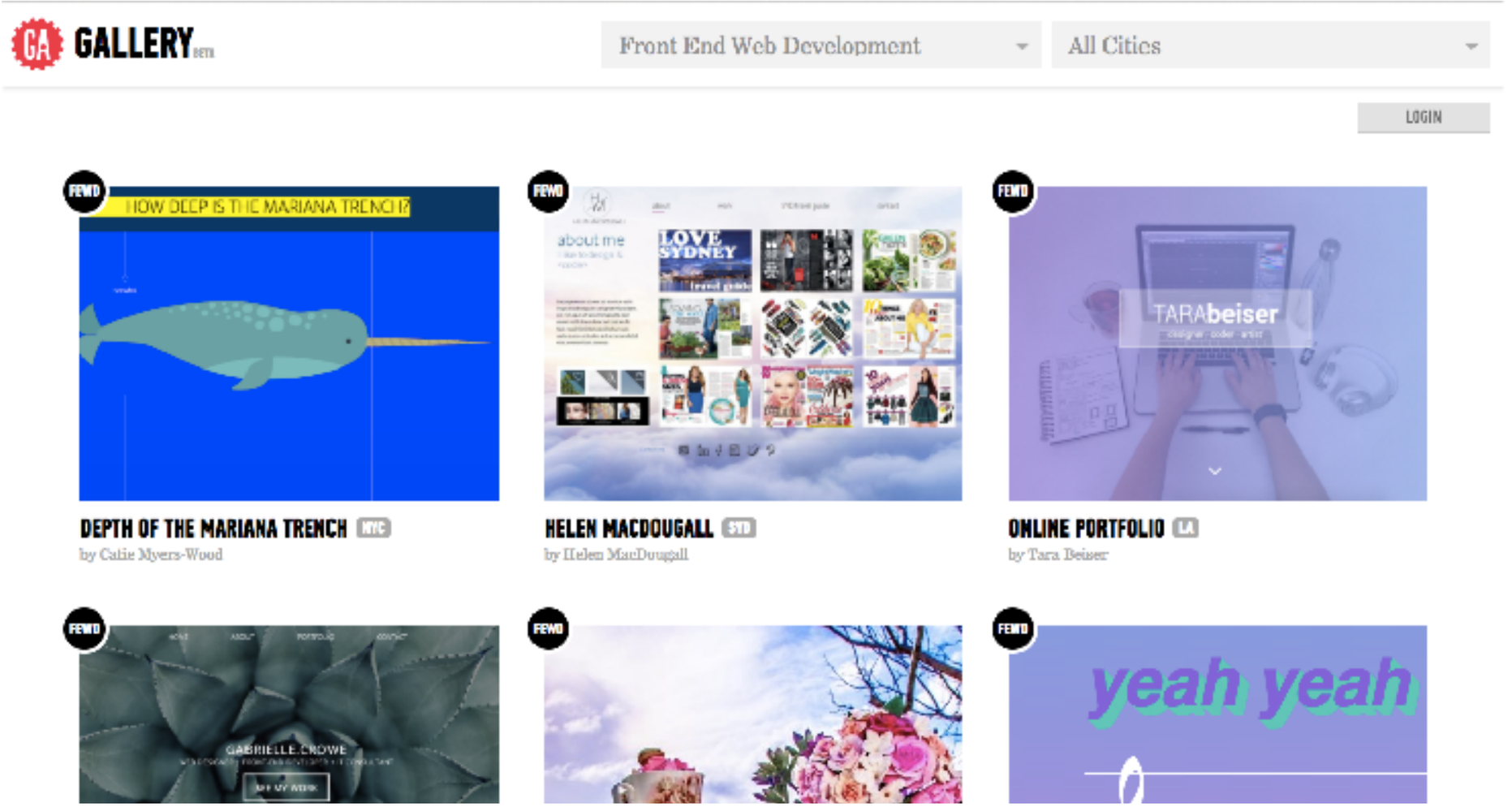
- Practice "debugging" using Chrome Dev Tools
- Practice web development by transforming a design comp into a webpage.

AGENDA

- Review
- Display
- CSS Grids Pt. 2
- Chrome Dev Tools & Debugging
- Monument Landing Page Lab

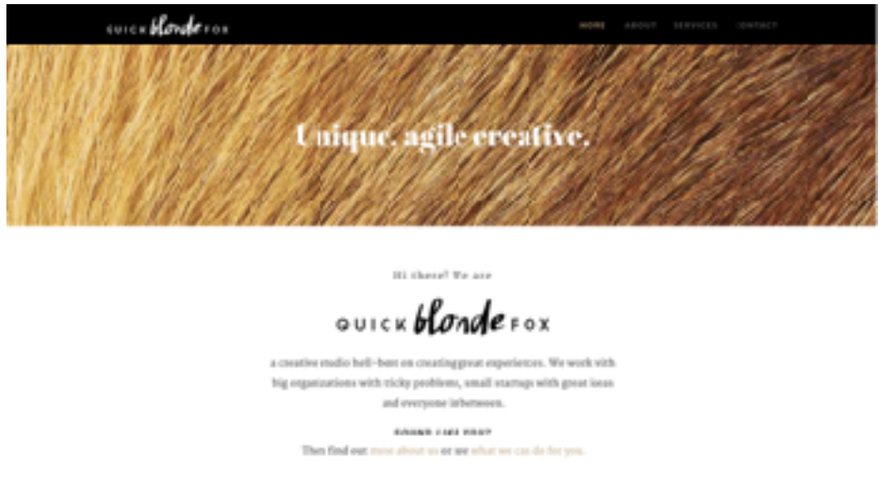
FINAL PROJECT OVERVIEW

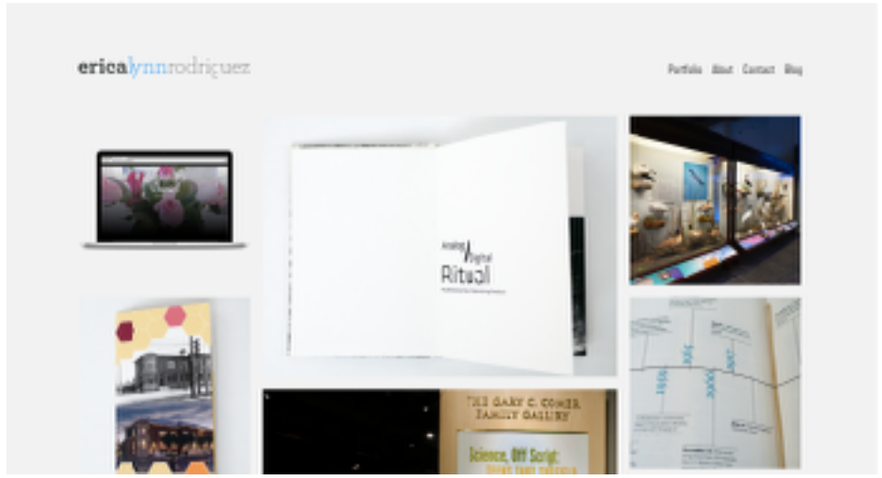
WHERE CAN I FIND PAST EXAMPLES?



WHAT KINDS OF PROJECTS TRADITIONALLY MAKE FOR GOOD FINAL PROJECTS?

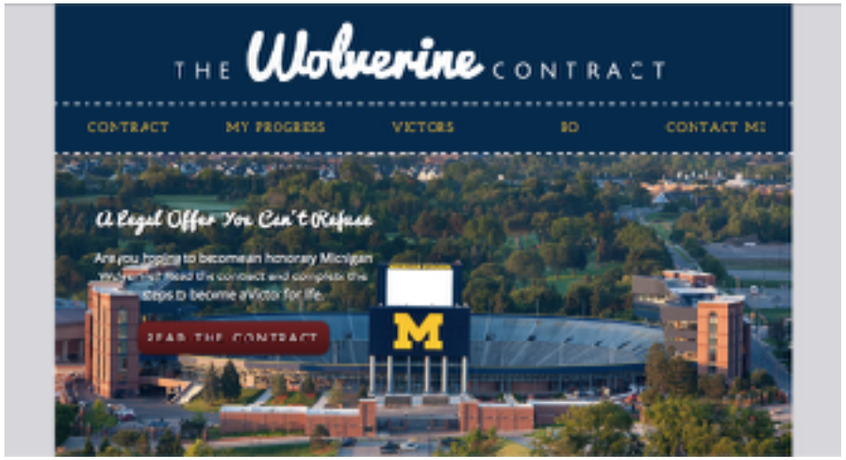
Portfolio Sites





WHAT KINDS OF PROJECTS TRADITIONALLY MAKE FOR GOOD FINAL PROJECTS?

Sites about hobbies or sites just for fun/practice.

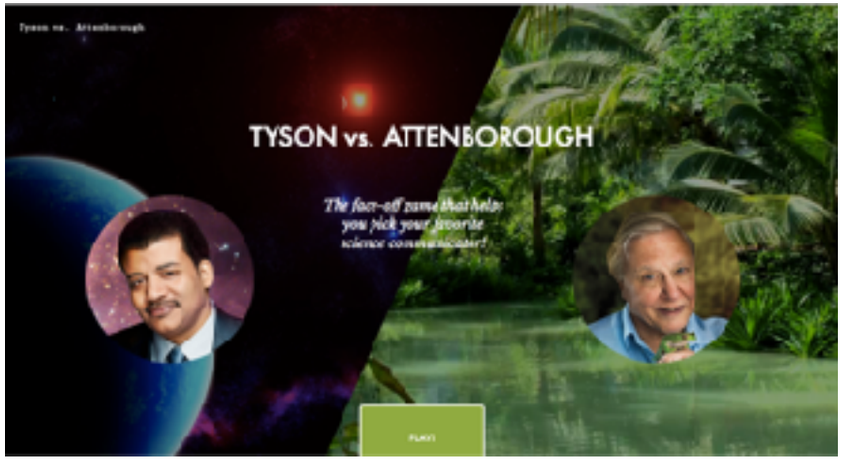


WHAT KINDS OF PROJECTS TRADITIONALLY MAKE FOR GOOD FINAL PROJECTS?

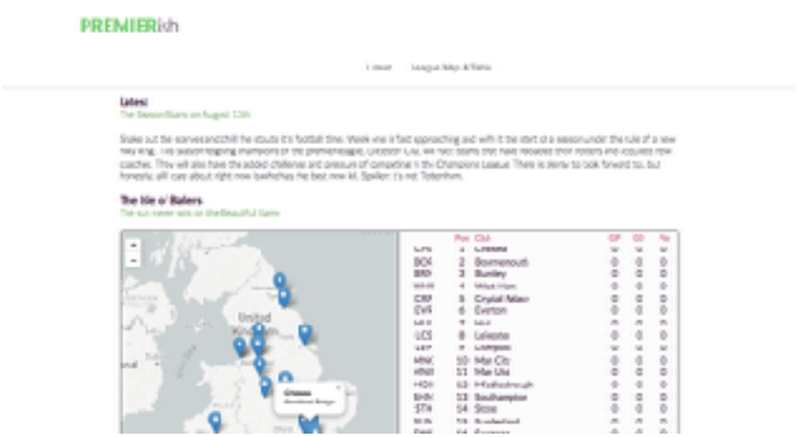
Redesign an existing site for family/friends or just any business with an ugly site :)

WHAT KINDS OF PROJECTS TRADITIONALLY MAKE FOR GOOD FINAL PROJECTS?

Quiz Site / Game



PROJECTS TYPES IF YOU WANT A CHALLENGE



WHERE YOU SHOULD BE

- You should have your final project idea finalized by the end of the week
- Take a look at the final project gallery for some inspiration
- Look at some sites on Site Inspire for inspiration
- Start planning — come to class next week with a list of the different pieces of content you plan to include in your final project

REVIEW

HOMEWORK — GROUP DISCUSSION

EXERCISE

TYPE OF EXERCISE

- Groups of 3 - 4

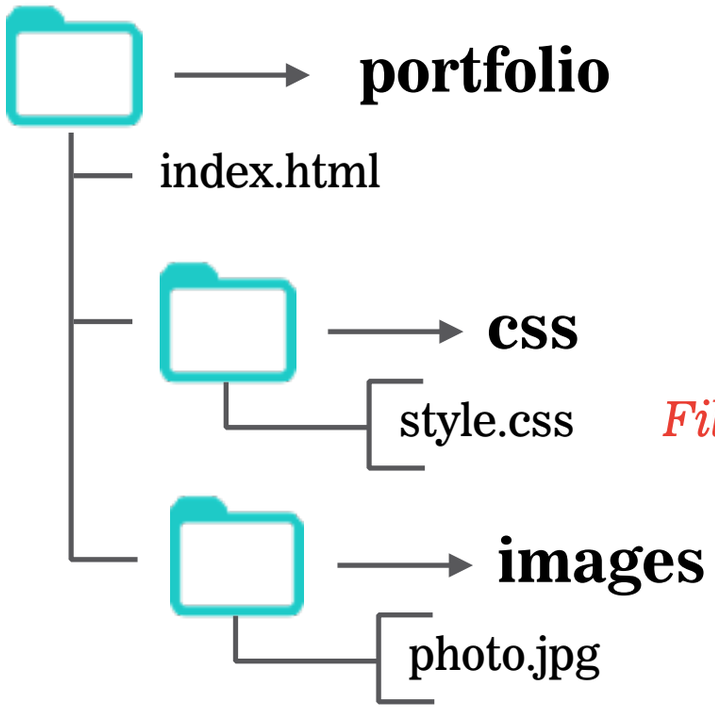
TIMING

10 min

1. Pick someone to take notes for your group.
2. Share 1 thing you're excited about being able to accomplish. Focus on the positives!
3. Have each person in the group note 1 thing they found challenging for the assignment and make note. Discuss as a group how you think you could solve that problem.
4. Discuss the bonus tasks (background image, font awesome icons). If anyone in your group was able to tackle one of these, share!

RELATIVE URLS — PART 2

RELATIVE URLS — CHILD FOLDER



File we want to add a background image to

If file you're working out of is in a **child folder**, include `../` folder in the relative url.

```
background-image: url("../images/header-bg.jpg");
```

BACKGROUND IMAGES

To add a background image:

```
background-image: url("../images/header-bg.jpg");
```

To make the background image scale with the container:

```
background-size: cover;
```

To specify a position:

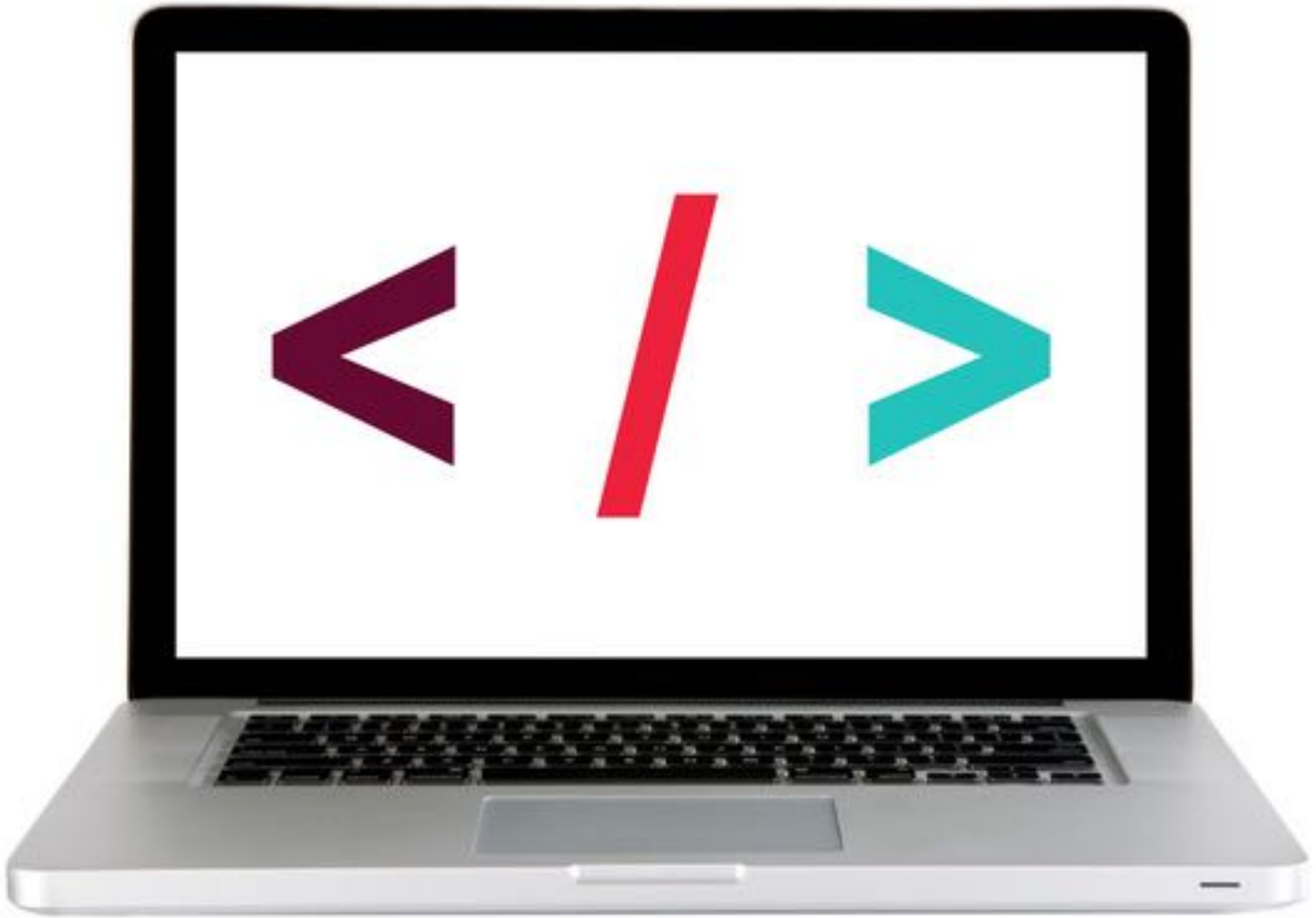
```
background-position: right top;
background-position: bottom;
```

BACKGROUND IMAGES

Don't forget `../` if your CSS file is in a CSS folder!!


```css
background-image: url("../images/header-bg.jpg");
```

LET'S TAKE A CLOSER LOOK



USING CHROME DEV TOOLS

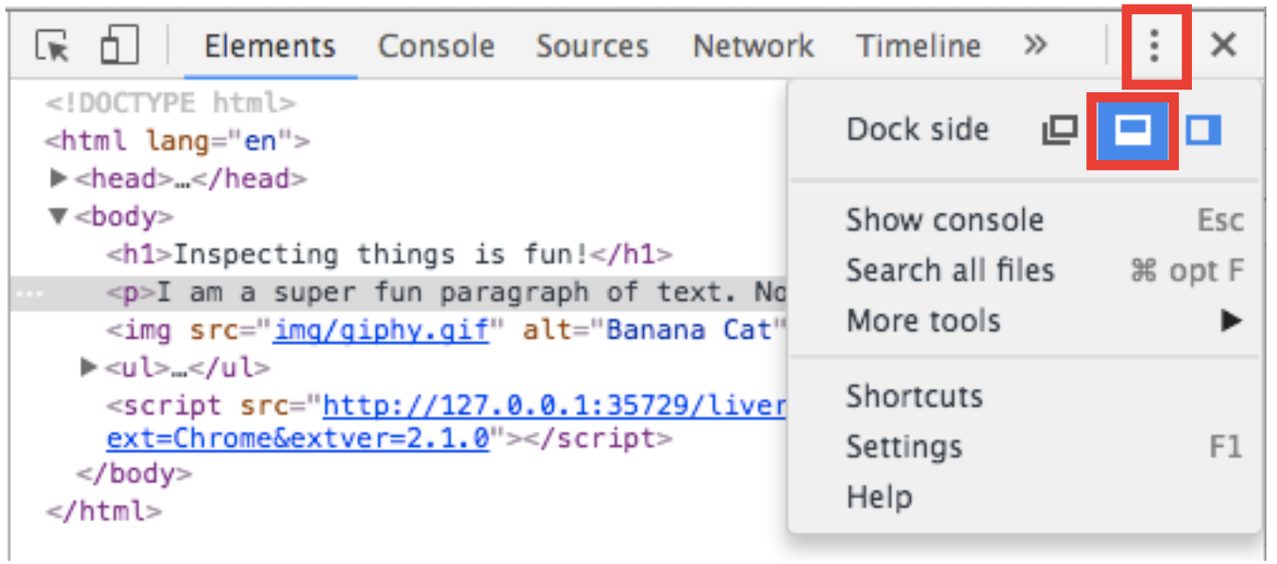
LET'S USE THE INSPECTOR!

There are several ways to open Chrome dev tools:

- Right click on an element and click "inspect"
- In Chrome, go to view > developer > Developer Tools
- Keyboard shortcut: Mac: *Cmd + Opt + I* Windows: *F12*, *Ctrl + Shift + I*

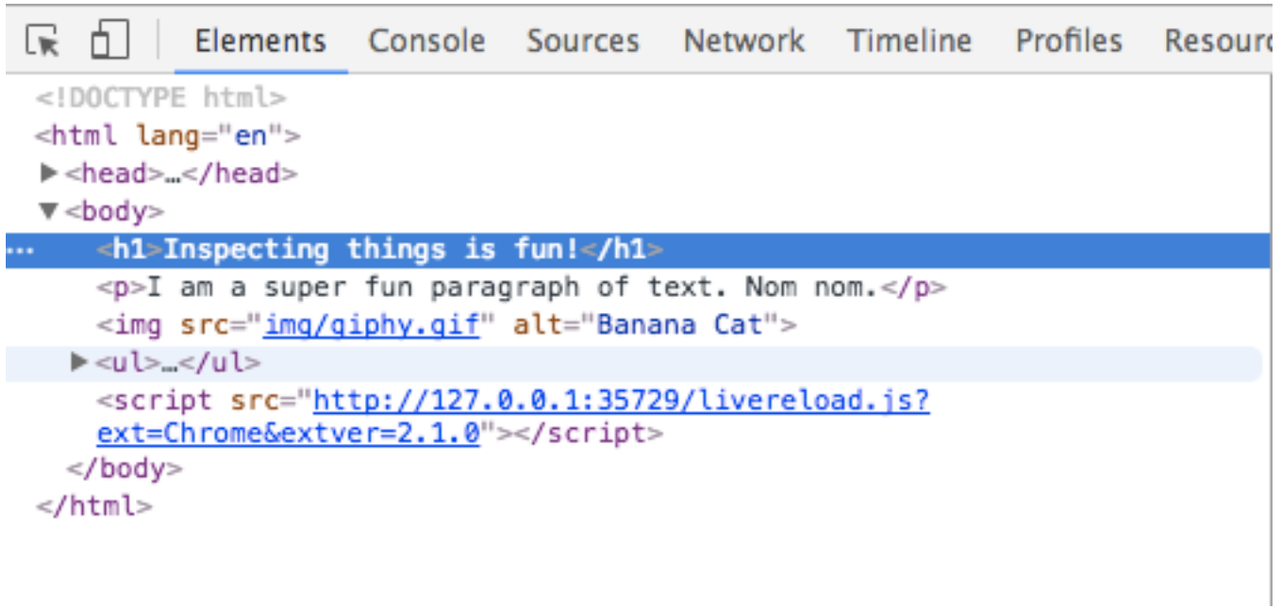
DOCK LOCATION

I find it easiest to move the dock to the bottom of the window.



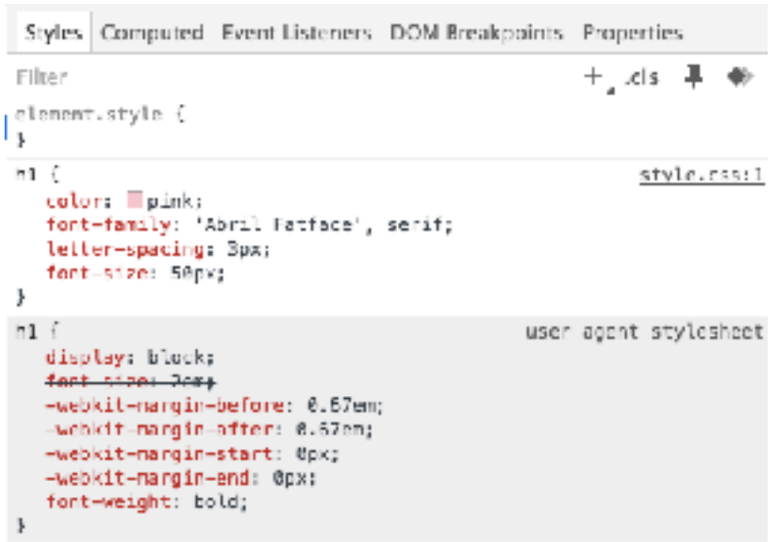
PICKING AN ELEMENT

Choose an element to inspect by clicking on the element in the "Elements" tab

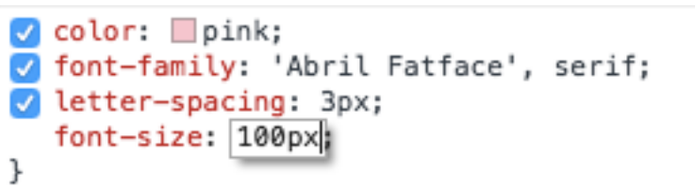


SEEING STYLES AND EXPERIMENTING

You can see what styles you've added (and default styles added by the browser) to the element in the "styles" panel.

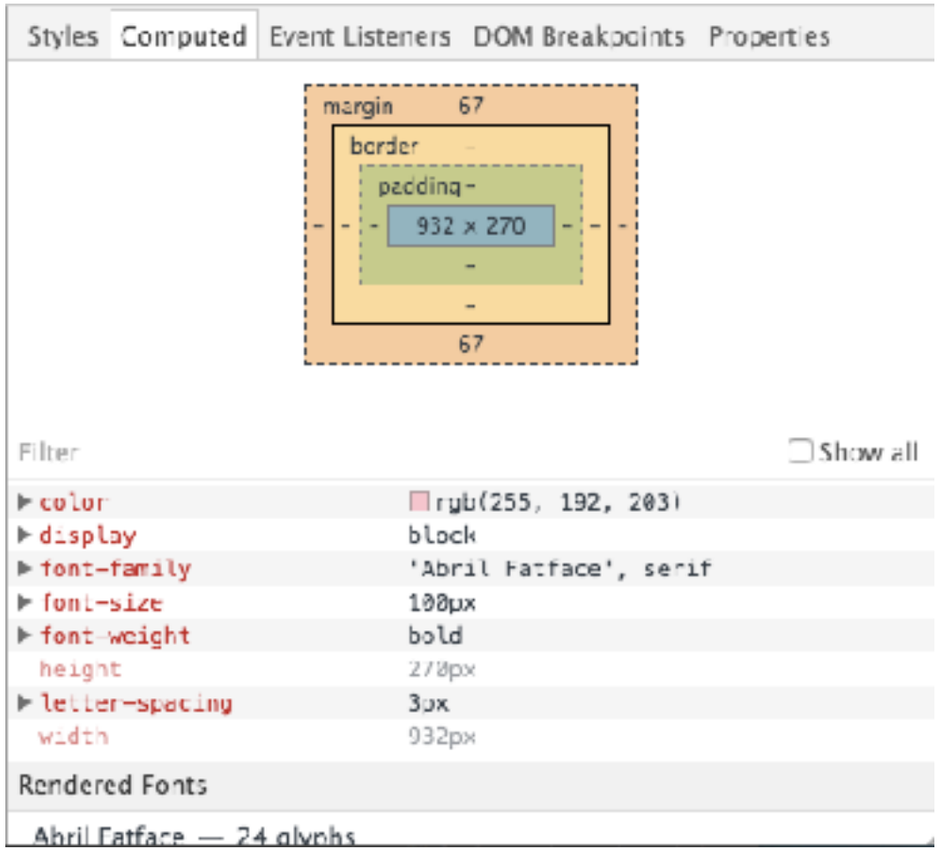


You can also experiment here!



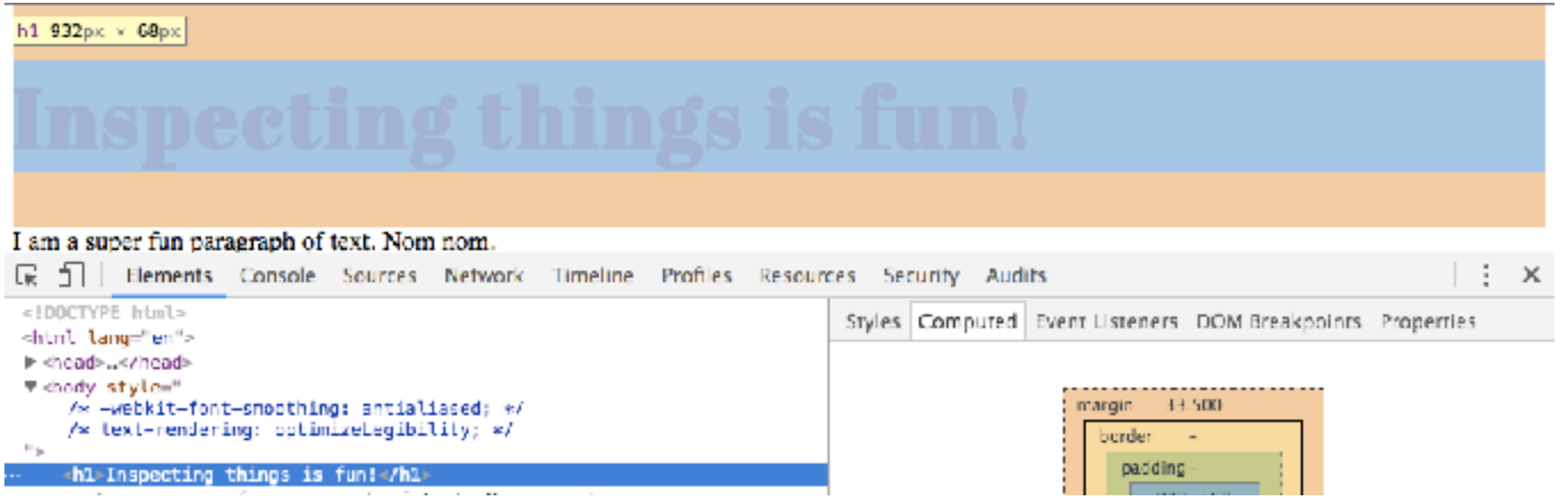
FINAL RENDERED STYLES AND DIMENSION

In the "computed" tab you can see all the styles that are being rendered on screen for an element. You can also see dimensions for an element including width, height, padding, margin and border.



VISUALIZING DIMENSIONS

One of the things I find most helpful when working through layout issues is to hover over different elements and see where everything is.



Orange: margin
Green: padding
Also notice width and height in a little yellow box by the element

LET'S USE THE INSPECTOR!

EXERCISE

LOCATION

- Starter Code > inspect_element

LAB SESSION

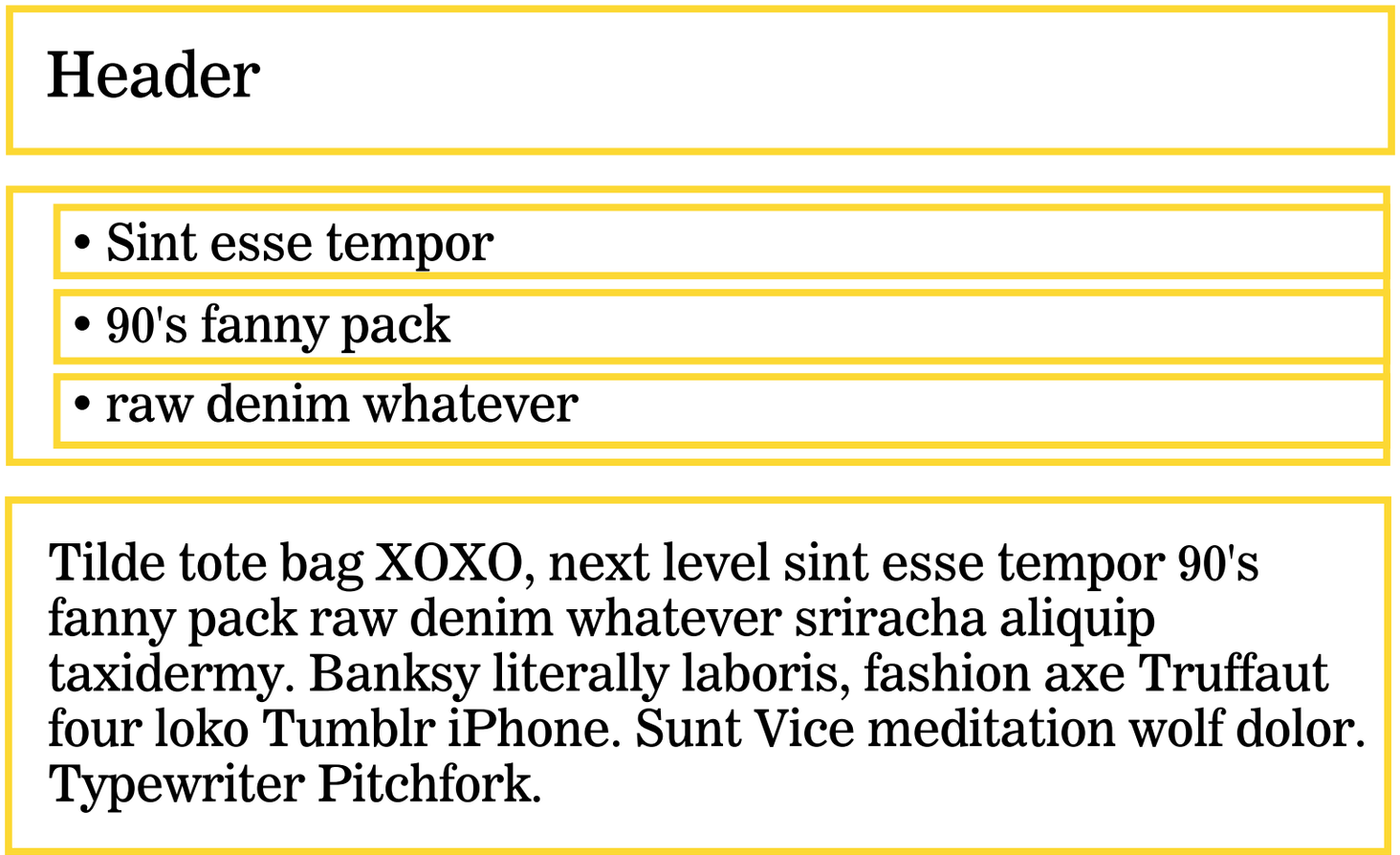
10 min

1. Inspect one of the **`h3`** elements (the dates for the articles). Hover over the `h3` in the elements panel to see if any margin (orange) or padding (green) are being applied.
2. What is the margin-top value in pixels for the `h3`?
3. Where is the margin-top for the `h3` coming from? Our stylesheet or the browser? How can you tell?
4. Inspect the **`header`** element. In the styles panel, uncheck and then re-check the checkboxes next to each style that's being applied to the header. This is a great way to see what the header would look like without those properties.
5. In the styles panel, update the `height` of the `header` to `400px` to see what that would look like.
6. We've tried to remove the underline for the anchors but it's not working. Right click and inspect any anchor on the page and look at the styles being applied. Why is the underline not being removed? See if you can fix it so it's working again by making a fix in the styles panel in the inspector.

DISPLAY

BLOCK ELEMENTS

An **block element** will always start on a new line.



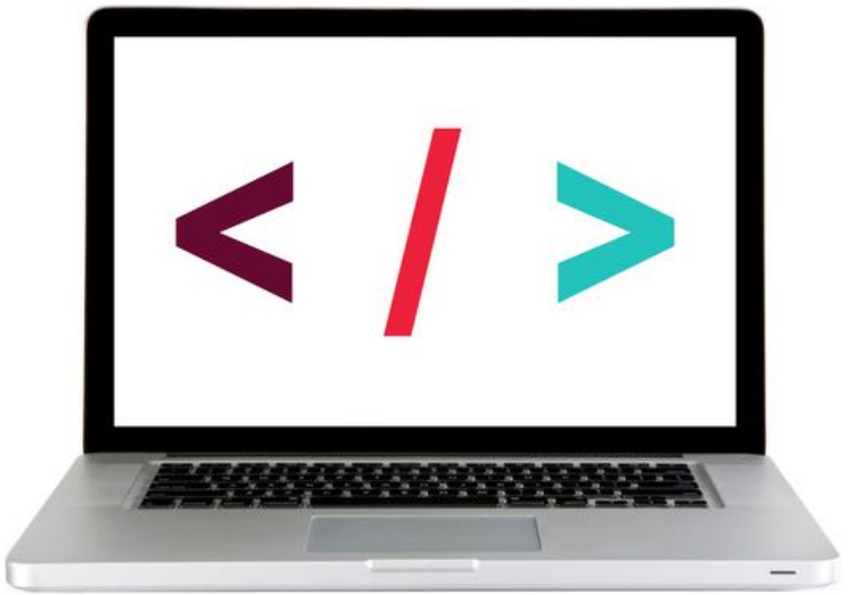
INLINE ELEMENTS

An **inline element** does not start on a new line and only takes up as much space as needed.

Tilde tote bag XOXO, next level sint esse tempor 90's fanny pack raw denim whatever sriracha aliquip taxidermy. Banksy **literally** laboris, fashion axe Truffaut four loko Tumblr iPhone. Sunt Vice meditation *wolf* dolor. Typewriter www.Pitchfork.com.

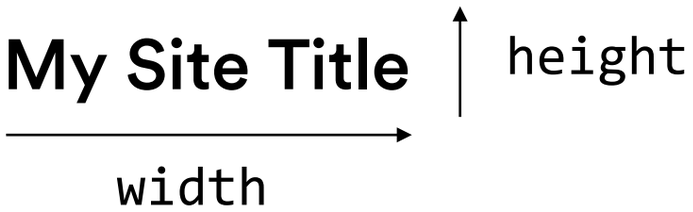
 Banksy literally laboris, fashion axe Truffaut four loko Tumblr iPhone. Sunt Vice meditation wolf dolor. Typewriter www.Pitchfork.com.

LET'S TAKE A CLOSER LOOK - PART 1



https://codepen.io/sarahholden/pen/qrLdJJ

CONTENT



BOX MODEL — BLOCK ELEMENTS

DIMENSION – A KEY DIFFERENCE BETWEEN INLINE AND BLOCK ELEMENTS

If you try to add dimension to an inline element:

- `Padding` and `margin` will only apply to the *left and right*
- `Width` and `height` will have *no effect*

DISPLAY

You can change whether elements are displayed as inline or block elements by using the **`display`** property.

```
li {
  display: inline-block;
}
```

DISPLAY — INLINE (RARE)

To make a block-level element act like an inline element — *only used rarely*:

BEFORE:

Monday

Tuesday

Wednesday

```
li {
   display: inline;
}
```

Monday Tuesday Wednesday

- Elements will sit next to each other
- **Still can't set a width, height, or margin and padding on top and bottom**

DISPLAY — INLINE-BLOCK

Make a **`block`** *or* **`inline`** element flow like an **`inline`** element,
while allowing us to set a `width`, `height`, `padding`, and `margin`:

BEFORE:

Monday

Tuesday

Wednesday

```
li {
   display: inline-block;
}
```

Monday Tuesday Wednesday

- Elements will sit on a line next to each other
- **We can now set a width, height, and margin & padding on top and bottom!!**

DISPLAY — BLOCK

Make an inline element act like a block-level element:

```
a {
   display: block;
}
```

BEFORE:

Link Link Link

Link

Link

Link

- Elements will stack on top of each other
- We can add all dimensions (width, height, padding, margin)

DISPLAY — NONE

Hide an element from the page:

BEFORE:

- Monday
- Tuesday
- Wednesday

```
li {
   display: none;
}
```

- Elements will be hidden from the page

TEXT-ALIGN

	TEXT-ALIGN
BLOCK	yes
INLINE / INLINE-BLOCK	no

ACTIVITY — DISPLAY LAB

EXERCISE

KEY OBJECTIVE

- Get practice using the `display` property

LOCATION OF FILES

- starter code > **display_lab** folder

TIMING

5 min 1. Follow the instructions in steps 1-4

CENTER ALL THE THINGS!

CENTERING THINGS — TEXT

```html
<p>
    <a href="">Home</a>
    <a href="">About</a>
    <a href="">Contact</a>
</p>
```

Option 1: Set `text-align: center` on parent. This will be inherited by all children.

```css
p {
    text-align: center;
}
```

Option 2: Update the display to block on the inline element you want to center.

```css
a {
    text-align: center;
    display: block;
}
```

CENTERING THINGS — IMAGES AND CONTAINERS

Centering an image:

```
img {
    display: block;
    margin: 0 auto;
}
```

Centering a layout:

```
main {
    max-width: 1000px;
    margin: 0 auto;
}
```

CSS GRID

ROWS

COLUMNS

GRID — BASIC TERMINOLOGY

GRID CONTAINER

parent wrapper:
grid container

```
<div class="container">
  <div class="item"></div>
  <div class="item">
    <p class="sub-item"></p>
  </div>
  <div class="item"></div>
</div>
```

GRID ITEMS

direct children:
grid items

```
<div class="container">
  <div class="item"></div>
  <div class="item">
    <p class="sub-item"></p>
  </div>
  <div class="item"></div>
</div>
```

GRID — CONTAINER PROPERTIES

GRID — CONTAINER PROPERTIES

CSS Grid
A PEN BY Sarah Holden

Save | Fork | Settings | Change View

HTML

```
<section class="grid-wrapper">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</section>
```

CSS

```
/* Grid Container!! */
.grid-wrapper {
  height: 500px;

}

div {
  font-size: 30px;
  font-family: sans-serif;
  text-align: center;
```

1

2

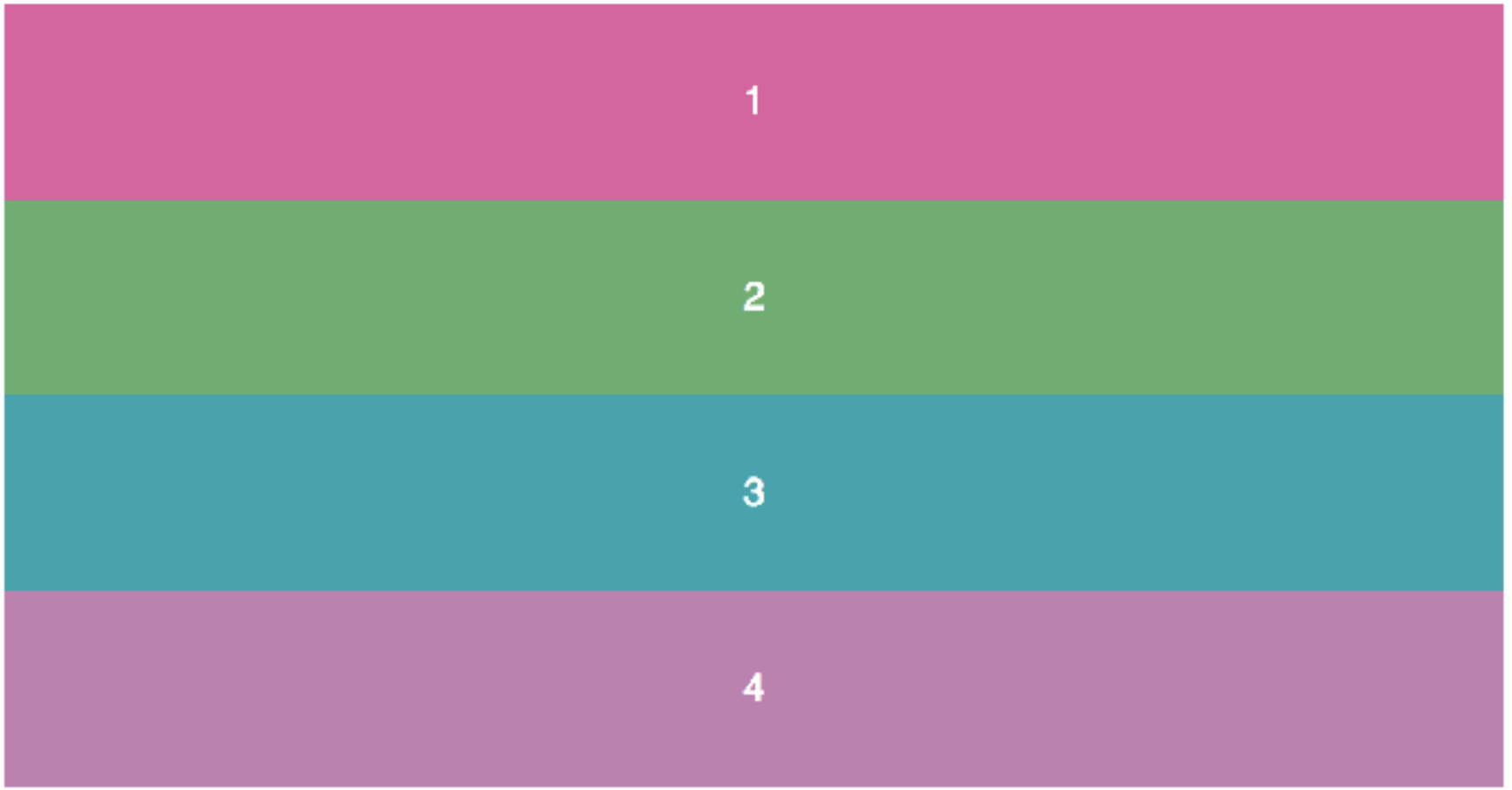
DISPLAY

```
<div class="container">
    <section>
        <p>Content here</p>
    </section>
    <aside>
        <p>Content here</p>
    </aside>
</div>
```

DISPLAY

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

DISPLAY — RESULT

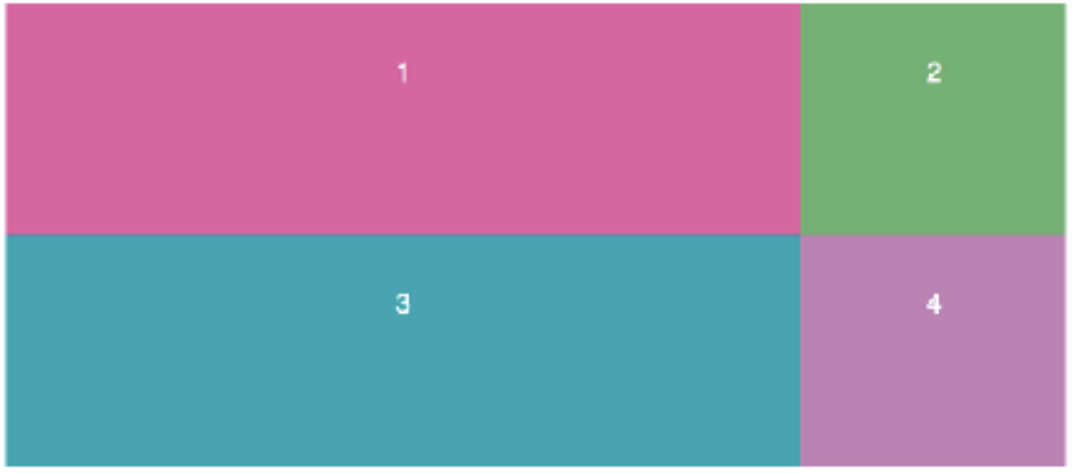


GRID-TEMPLATE-COLUMNS

```
.container {
  display: grid;
  grid-template-columns: 50% 50%;
}
```

GRID-TEMPLATE-COLUMNS — CODE ALONG RESULT

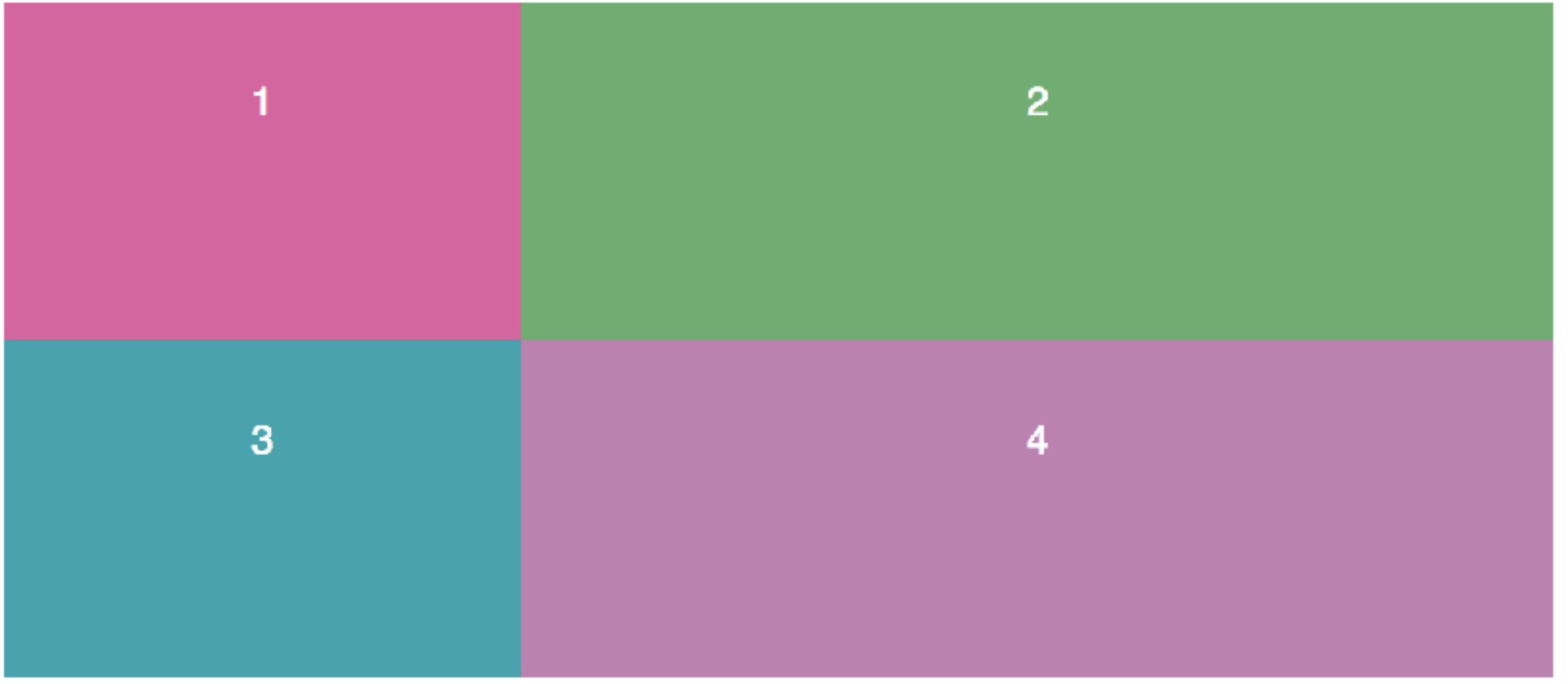
GRID-TEMPLATE-COLUMNS — NOW YOU TRY



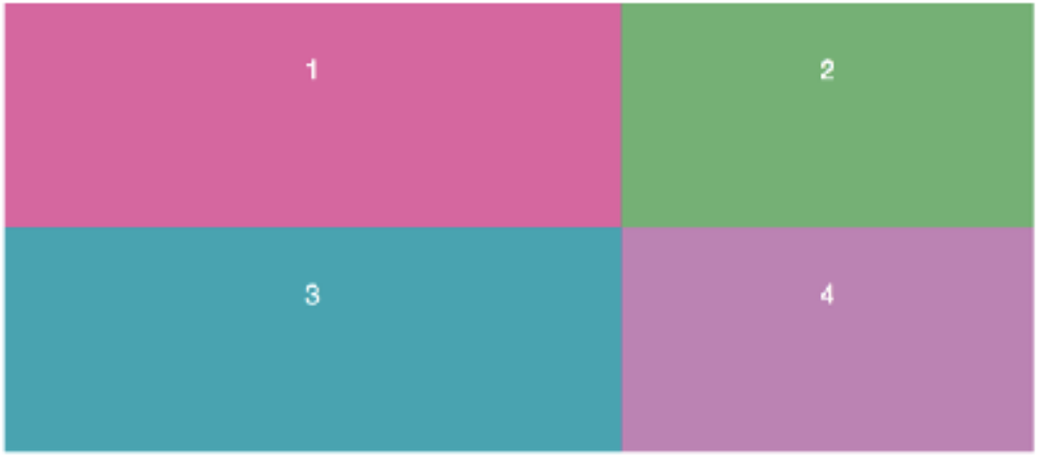
GRID-TEMPLATE-COLUMNS

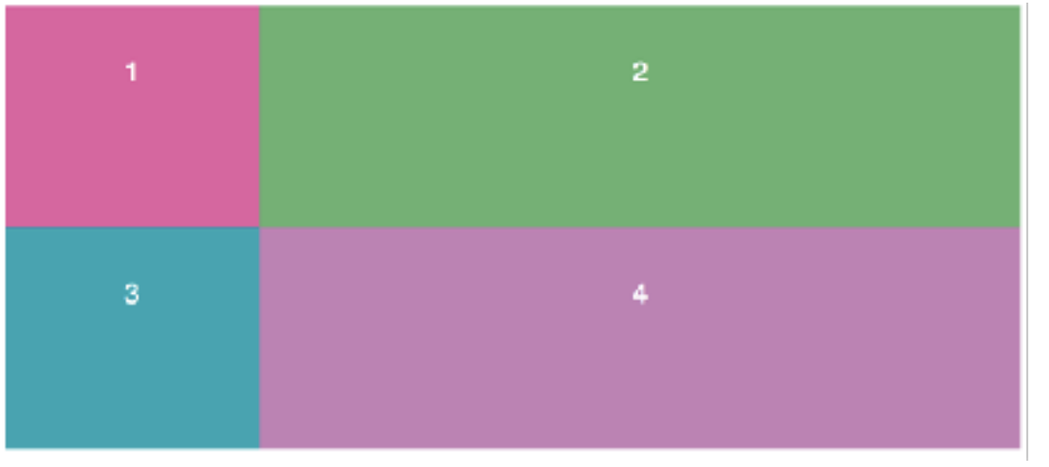
```
.container {
  display: grid;
  grid-template-columns: 1fr 2fr;
}
```

GRID-TEMPLATE-COLUMNS — CODE ALONG RESULT



GRID-TEMPLATE-COLUMNS — NOW YOU TRY



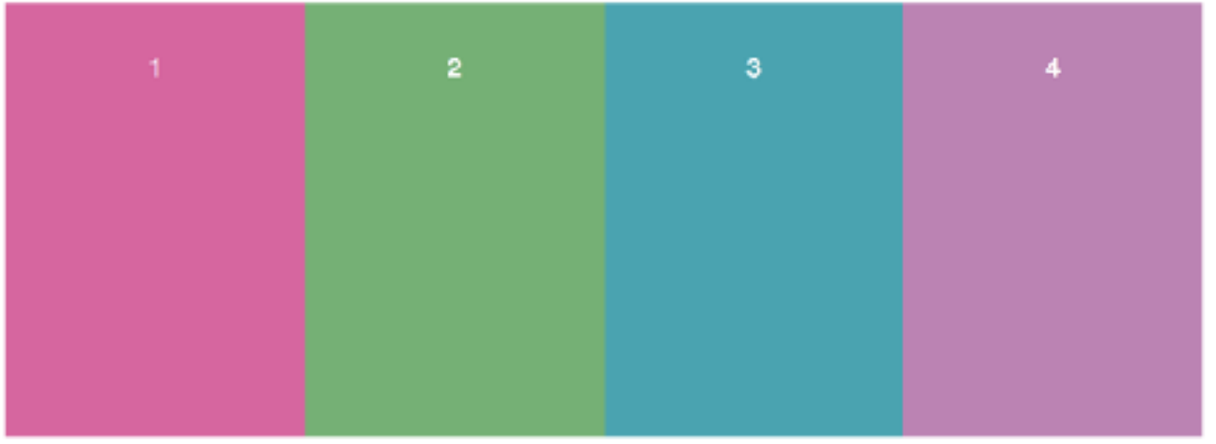


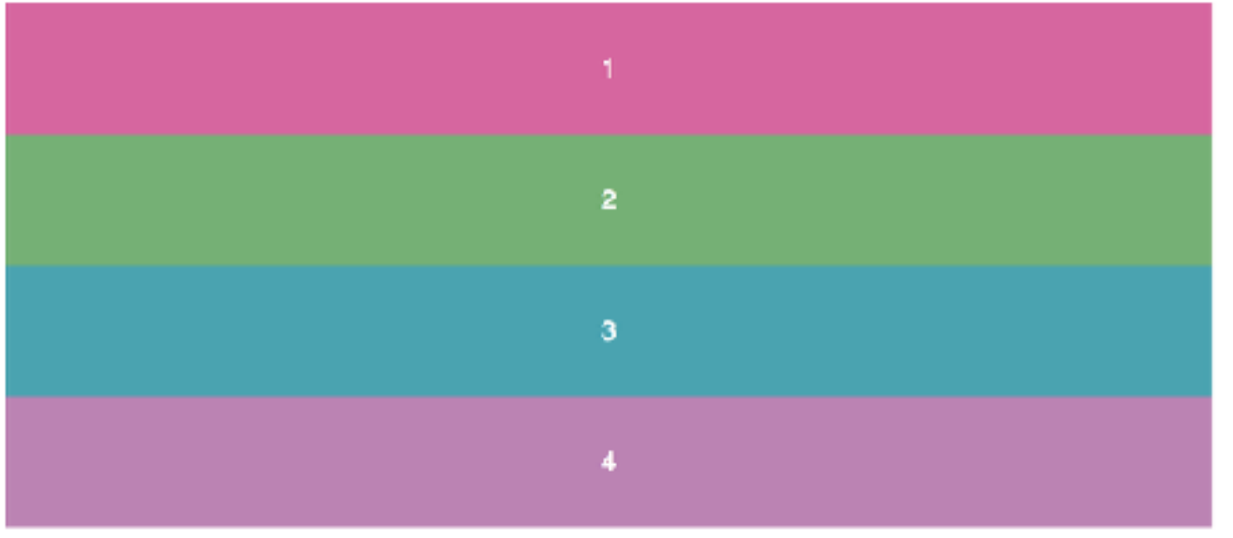
GRID-TEMPLATE-COLUMNS

```
.container {
  display: grid;
  grid-template-columns: repeat(2, 50%);
}
```

GRID-TEMPLATE-COLUMNS — CODE ALONG RESULT

GRID-TEMPLATE-COLUMNS

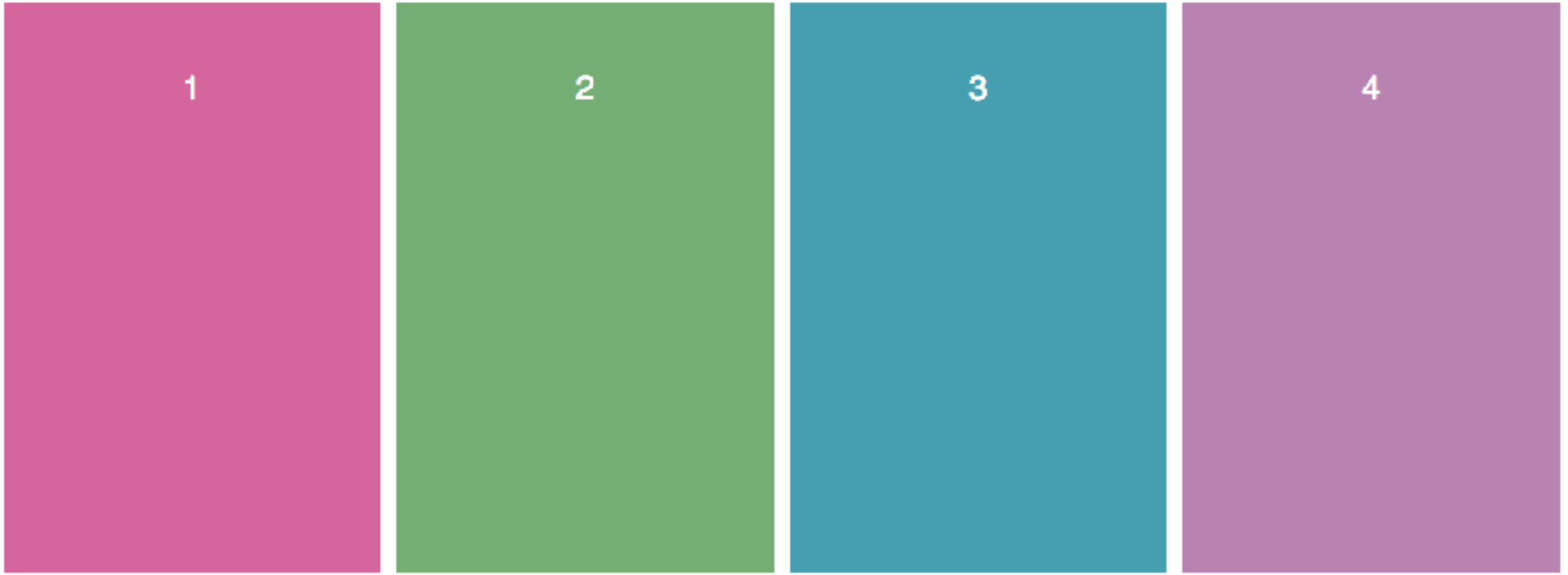




GRID-TEMPLATE-COLUMNS

```
.container {
  display: grid;
  grid-column-gap: 20px;
  grid-row-gap: 20px;
}
```

COLUMN AND ROW GAPS



COLUMN AND ROW GAPS

ACTIVITY

EXERCISE

KEY OBJECTIVE

Utilize CSS grid to create a responsive, grid-based layout.

TIMING

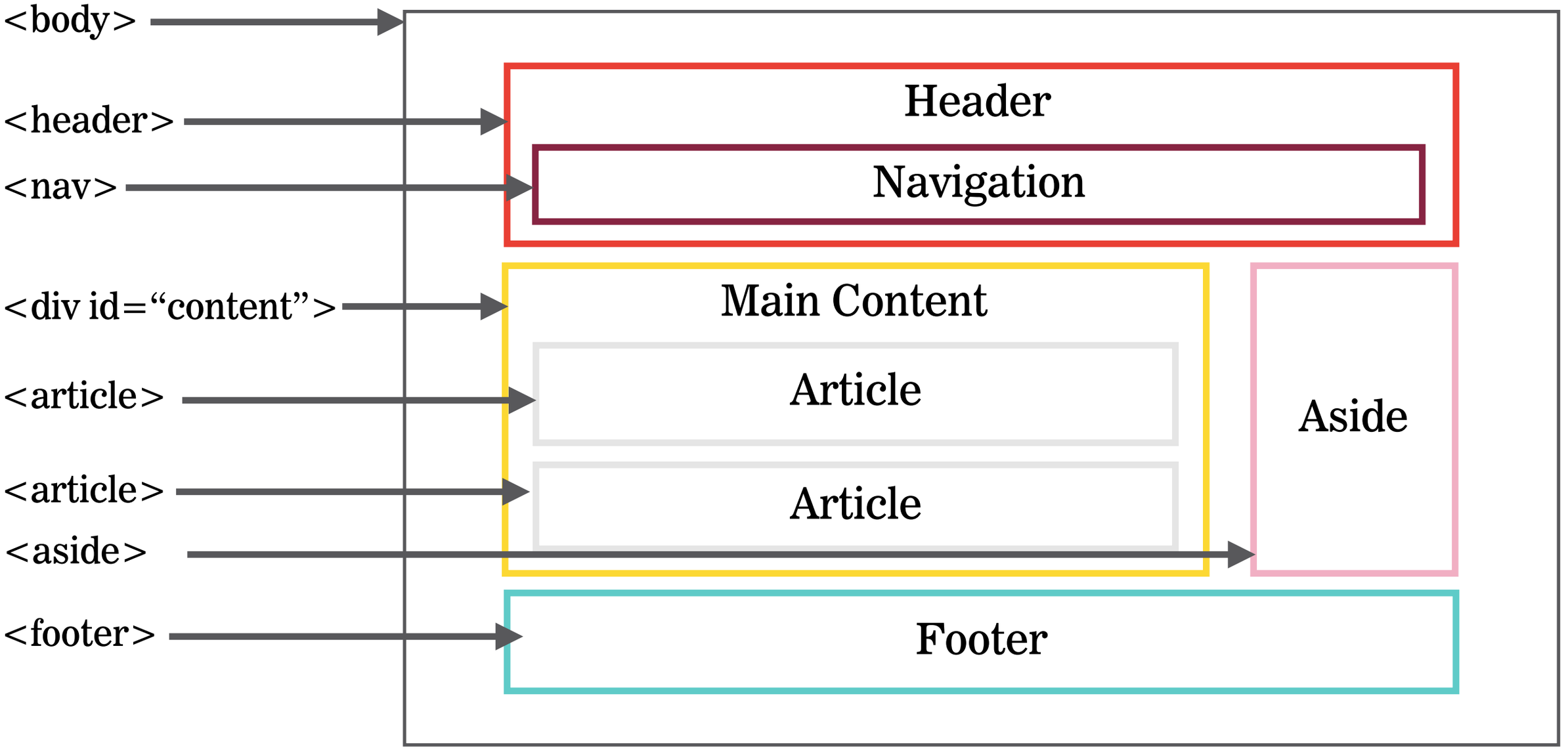
10 min

1. Recreate the image in the reading list folder
2. Bonus: Complete beatles_columns

```css
/*Whatever the selector is for your grid container*/
.grid-container {
  /*To set up CSS grid*/
  display: grid;

  /*To set up columns*/
  /*This is a space-separated list*/
  /*If there are 2 items in the list, it will be a 2 column layout*/
  grid-template-columns: 1fr 1fr;

  /*Space between columns*/
  grid-column-gap: 20px;

  /*Space between rows*/
  grid-row-gap: 20px;

  /*Shorthand for the above two properties if you have even space between*/
  grid-gap: 20px;
}
```

MONUMENT LAB

STRUCTURAL ELEMENTS



	DESCRIPTION
HEADER	Does it contain introductory content, such as the site title and navigation?
NAV	Does it contain a major navigational block?
FOOTER	Does it contain footer content?
MAIN	Does it contain content that is unique to that particular page?
ARTICLE	Is it self-contained, standalone content? Would it make sense on its own?
ASIDE	Is the content **not** required to understand the rest of the page?
SECTION	Is it a general section of the page that could have a heading?
DIV	Do you need a wrapper for styling and none of the above apply?

PLANNING - MONUMENT

EXERCISE

KEY OBJECTIVE

- Demonstrate the ability to plan and build a website

PLANNING

8 min

1. Plan out divs and HTML5 Sections
2. Where will you need to use flexbox for columns?
3. What are the base text styles for the page? (font-family, color, background-color, text-transform, text-align, etc.)
4. What are the base styles for headings? (Font, color, text-transform, text-align, etc.)
5. Are there base styles for anchors on the page? (font, color, text-decoration, text-transform, text-align)

LAB - MONUMENT

KEY OBJECTIVE

- Demonstrate the ability to plan and build a website

LAB SESSION

30 min

1. Write HTML for provided text in the starter code for the lab.

LAB - MONUMENT

EXERCISE

KEY OBJECTIVE

- Demonstrate the ability to plan and build a website

LAB SESSION

8 min

1. Hook up Google Fonts (Lato 400 and Raleway 400 and 600).
2. Add the FontAwesome Icons

GOOGLE FONTS

Web font services — like Google Fonts — allow us to ensure that all users can see the same font.

1. Go to the Google Fonts website and click the + button next to the fonts you want to use
2. Click the tab that pops up at the bottom:

3. Click the "customize" tab and choose the font weights you'll need
4. Click the @import tab
5. Copy the code inside of the <style> tags and place it at the TOP of the CSS panel in CodePen.
6. To use these fonts, use the `font-family` property and the name of the font within quotes:

```
body {
    font-family: 'Raleway';
}
```

```
p {
    font-family: 'Lato';
}
```

LAB - MONUMENT

EXERCISE

KEY OBJECTIVE

- Demonstrate the ability to plan and build a website

LAB SESSION

1. Get everything where it needs to be! Add styles for page structure (flexbox-containers, wrappers for columns). Use "*Steps to achieve a multi-column layout*" in review section or your handout for reference
2. Add base Styles (base fonts, colors, etc.)
3. Add base styles for headers, anchors, text
4. Style everything else!

LAB - MONUMENT

EXERCISE

KEY OBJECTIVE

- Demonstrate the ability to plan and build a website

BONUS TASKS (GIFS IN STARTER CODE)

1. Add a hover effect to anchors, buttons, social media icons. Extra bonus: Look up "transition effect hover CSS) to add a smoother transition to hover effects.
2. Create a fixed background effect on scroll for header image
3. Turn the nav into a "sticky nav" so that it stays fixed while the user scrolls.
4. Add an effect to the images in the gallery on hover.
5. Google and implement "Jump Links" so that clicking on a nav item jumps you to that section of the page

LEARNING OBJECTIVES

- Practice "debugging" using Chrome Dev Tools
- Practice web development by transforming a design comp into a webpage.

WEEKLY OVERVIEW

WEEK 3	Layout Lab / Layout Lab & Wireframing
WEEK 4	Responsive Design / CSS Positioning & Animation

EXIT TICKETS