

MONUMENT LAB

WEEKLY OVERVIEW

WEEK 3

Lab / Refactoring & Lab

WEEK 4

Forms & Bootstrap / Intro to Programming

WEEK 5

jQuery Part 1 / jQuery Part 2

LEARNING OBJECTIVES

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

AGENDA

Homework Review

Chrome Dev Tools

HTML Review

Monument — Part 1

CSS Review

Monument — Part 2

HOMEWORK 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 (icons in footer, form in footer, browser prefixes). If anyone in your group was able to tackle one of these, share!

USING CHROME DEV TOOLS

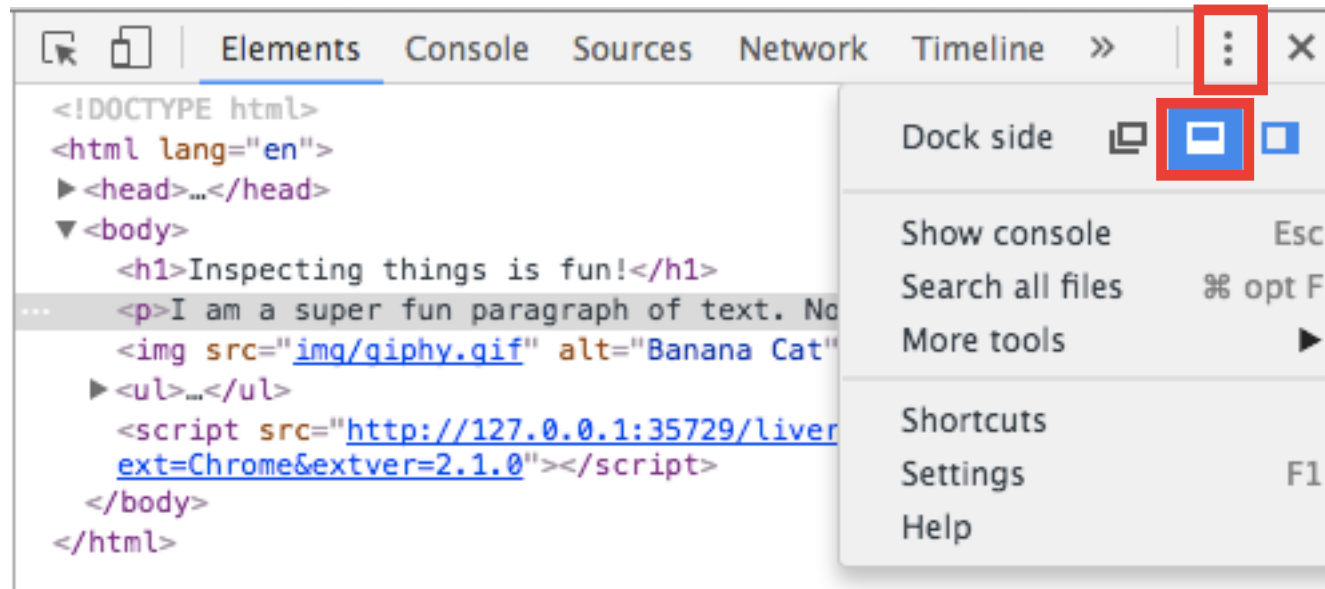
LET'S USE THE INSPECTOR!

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*

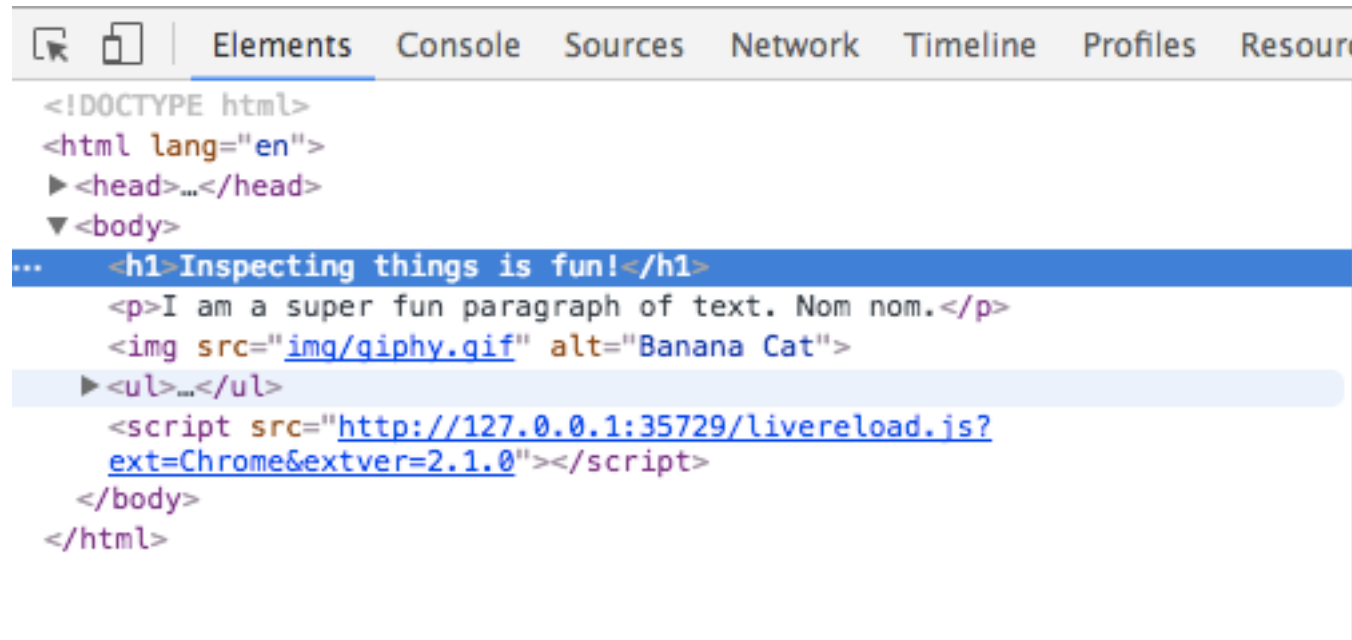
DOCK LOCATION

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



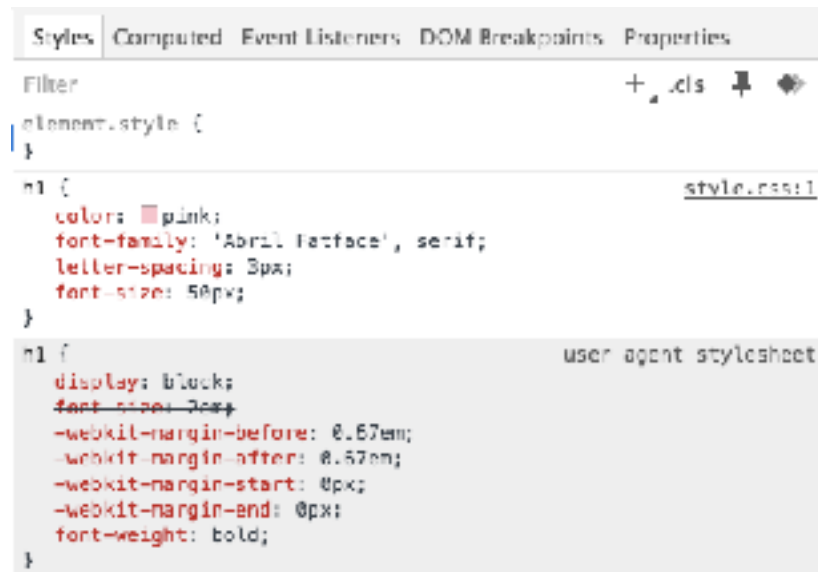
PICKING AN ELEMENT

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

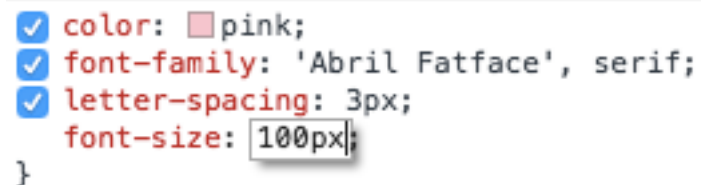


SEEING STYLES AND EXPERIMENTING

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.

StylesComputedEvent ListenersDOM BreakpointsProperties

margin67

border-

padding-

932 x 270

-

-

67

Filter☐ Show all

▶colorrgb(255, 192, 203)

▶displayblock

▶font-family'Abril Fatface', serif

▶font-size100px

▶font-weightbold

height270px

▶letter-spacing3px

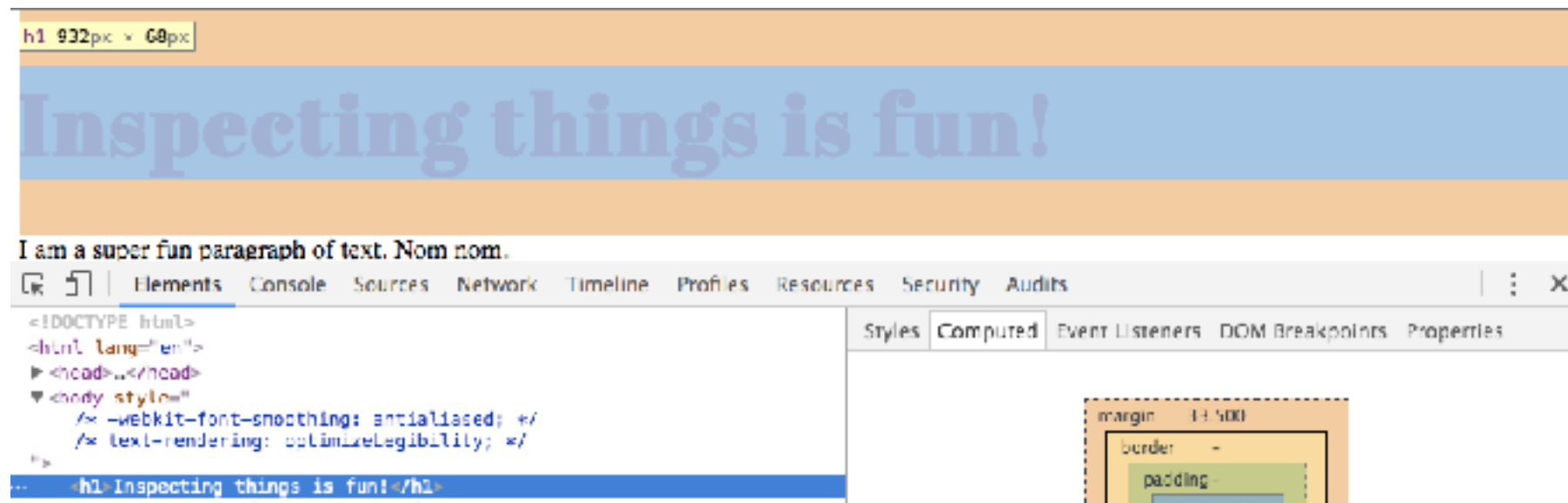
width932px

Rendered Fonts

Abril Fatface — 24 glyphs

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

TWO COLUMN LAYOUT (FOR LAB)

Why is Flexbox not working?!

This should be the left column

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fugiat, beatae. Hic quo sequi, fugiat nesciunt fuga illo, dolores optio itaque ipsum nam impedit excepturi, facilis consequuntur dicta omnis, deserunt quaerat?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fugiat, beatae. Hic quo sequi, fugiat nesciunt fuga illo, dolores optio itaque ipsum nam impedit excepturi, facilis consequuntur dicta omnis, deserunt quaerat?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fugiat, beatae. Hic quo sequi, fugiat nesciunt fuga illo, dolores optio itaque ipsum nam impedit excepturi, facilis consequuntur dicta omnis, deserunt quaerat?

Aside for my page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fugiat, beatae. Hic quo sequi, fugiat nesciunt fuga illo, dolores optio itaque ipsum nam impedit excepturi, facilis consequuntur dicta omnis, deserunt quaerat?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fugiat, beatae. Hic quo sequi, fugiat nesciunt fuga illo, dolores optio itaque ipsum nam impedit excepturi, facilis consequuntur dicta omnis, deserunt quaerat?

LET'S USE THE INSPECTOR!



EXERCISE

LOCATION

‣ Starter Code > inspect_element

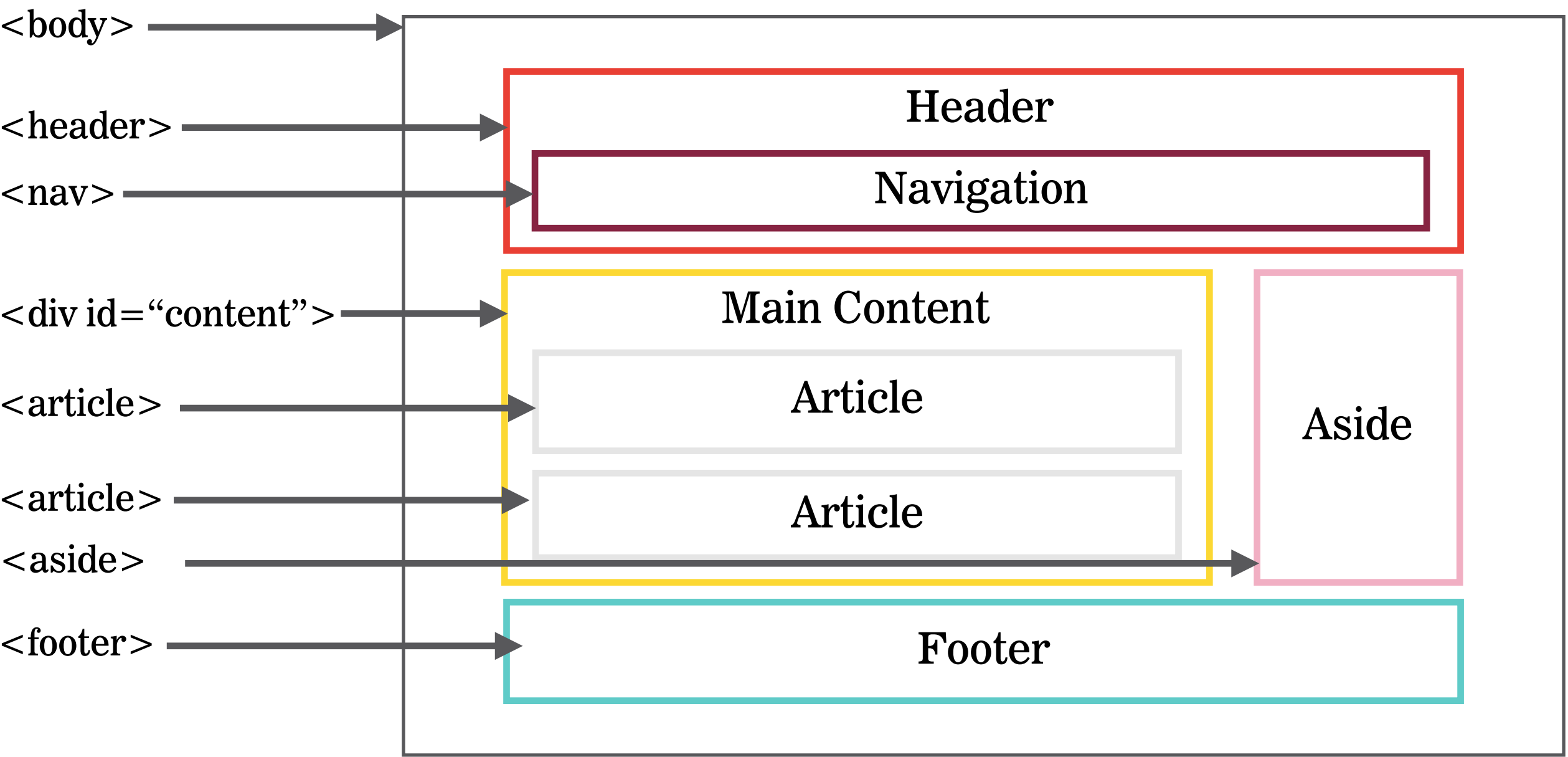
LAB SESSION

10 min

1. What color is the h1? Can you find the rgb value?
2. What is the margin-top of the h1 element? Margin-bottom? Are these styles added in our stylesheet or default styles given by the browser (in the user agent stylesheet).
3. Hover over the h1 to visualize these dimensions.
4. If we didn't want margin on top, what could we do? Try to update the h1 so there is no margin on top — in your inspector.
5. Our two-column layout isn't working! Using your inspector can you figure out why? Can you fix this — in your inspector.
6. What is the background-color for each li? How much padding and margin does each li have? What is the width and height?
7. Hover over an li to visualize these dimensions.
8. Bonus - can you figure out why the image isn't loading? Hint: look in the "Console" tab and compare with what you see in the folder in Finder. What should the image path be?

HTML REVIEW

STRUCTURAL ELEMENTS



STRUCTURAL ELEMENTS — FLOWCHART

	DESCRIPTION
NAV	Is it a major navigational block?
ARTICLE	Does it make sense on its own?
ASIDE	Is it not required to understand the rest of the content?
SECTION	Is it logical to add a heading?
FOOTER	Does it contain footer content?
DIV	Do you need a wrapper for styling and do none of the above apply?



REFACTOR — FILE HANDLING

- Files and folder names should not contain odd characters like spaces " " — instead use dashes (-) or underscores (_)
- Files and folder names should be written in all lower-case (about-me.html not About-Me.html)

REFACTOR — HTML

- Always indent nested elements with 2-space tabs
- Close tags immediately after you open them so you don't forget later
- Nothing should be naked — all text should be wrapped in an HTML tag
- Keep project folders organized — the index.html should be in the root directory, images etc should be in sub-directories named as such
- Always opt for the more semantic HTML5 tags instead of plain `<div>`s when appropriate
- Before you finish a project, run your code through a [validator](#) to catch any errors.
- Don't use `
` tags to add space, only when you need a line break. Instead, use CSS padding and margin.
- Headings should be hierarchical (h1 for most important heading, h2 - h6 for subheadings).

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



EXERCISE

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.

CSS REVIEW

BUILDING BLOCKS

BLOCK-LEVEL ELEMENTS

- ▶ Will always start on a new line

Examples:

- ▶ `<h1>-<h6>`
- ▶ ``
- ▶ ``
- ▶ `<p>`
- ▶ ``
- ▶ `<div>`

○ ○ ○

Header

- Sint esse tempor
- 90's fanny pack
- raw denim whatever
- sriracha aliquip taxidermy

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 Pitchfork.

BUILDING BLOCKS

INLINE ELEMENTS

Will always appear to continue on the same line as their neighboring elements

Examples:

- ▶ `<a>`
- ▶ ``
- ▶ ``
- ▶ ``
- ▶ `<q>`
- ▶ ``



DIMENSION – A KEY DIFFERENCE BETWEEN INLINE AND BLOCK ELEMENTS

SUMMARY — WHICH DIMENSIONS CAN BE CHANGED?

	WIDTH & HEIGHT	PADDING & MARGIN
BLOCK	yes	can apply to all sides
INLINE	no	will only affect left and right sides

TEXT-ALIGN

	TEXT-ALIGN
BLOCK	yes
INLINE / INLINE-BLOCK	no

STEPS TO ACHIEVE A MULTI-COLUMN LAYOUT

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

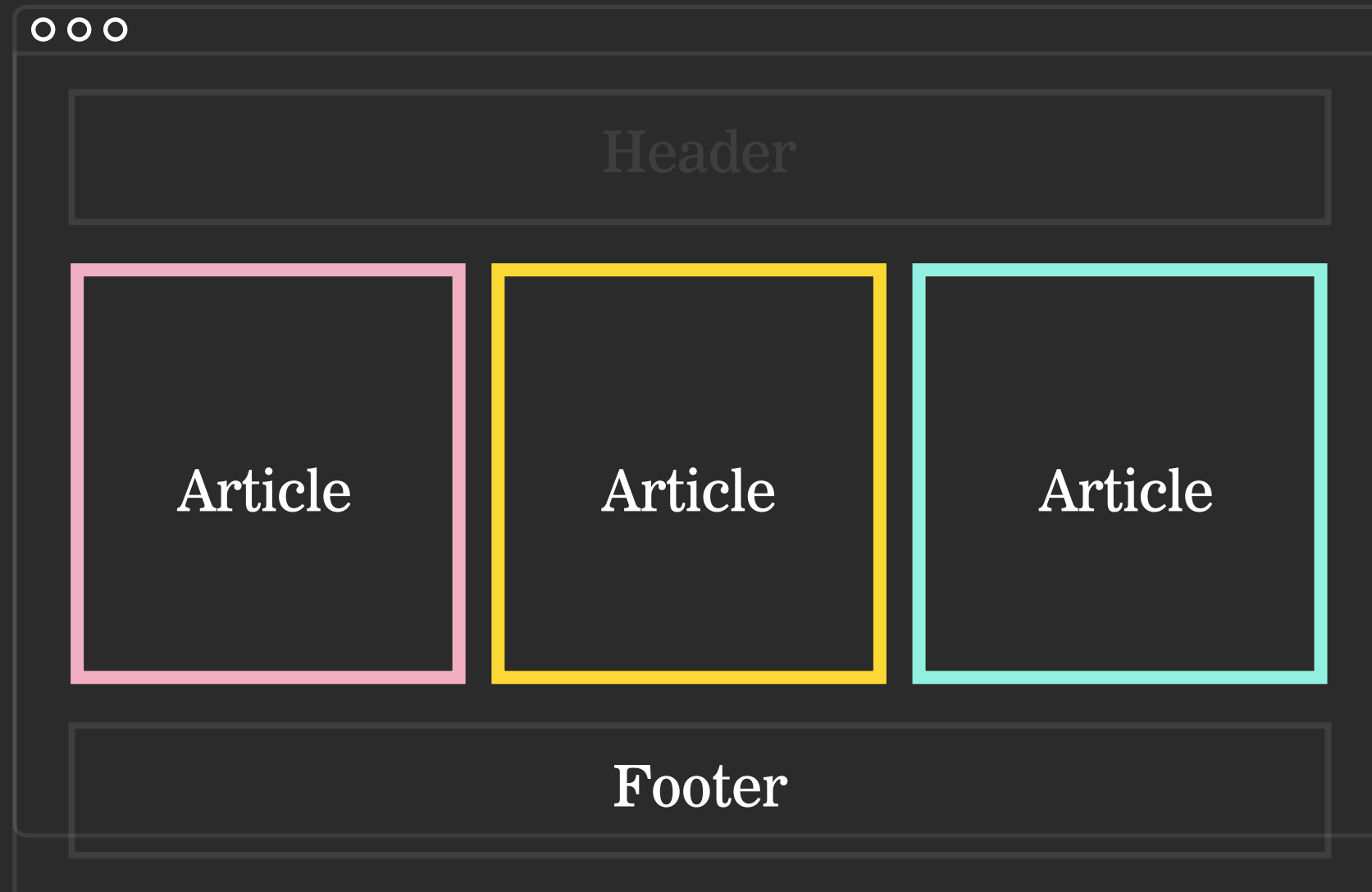
2. WRAP THE COLUMNS WITH A PARENT CONTAINER, THE "FLEX CONTAINER"

3. ADD `DISPLAY: FLEX;` TO FLEX-CONTAINER

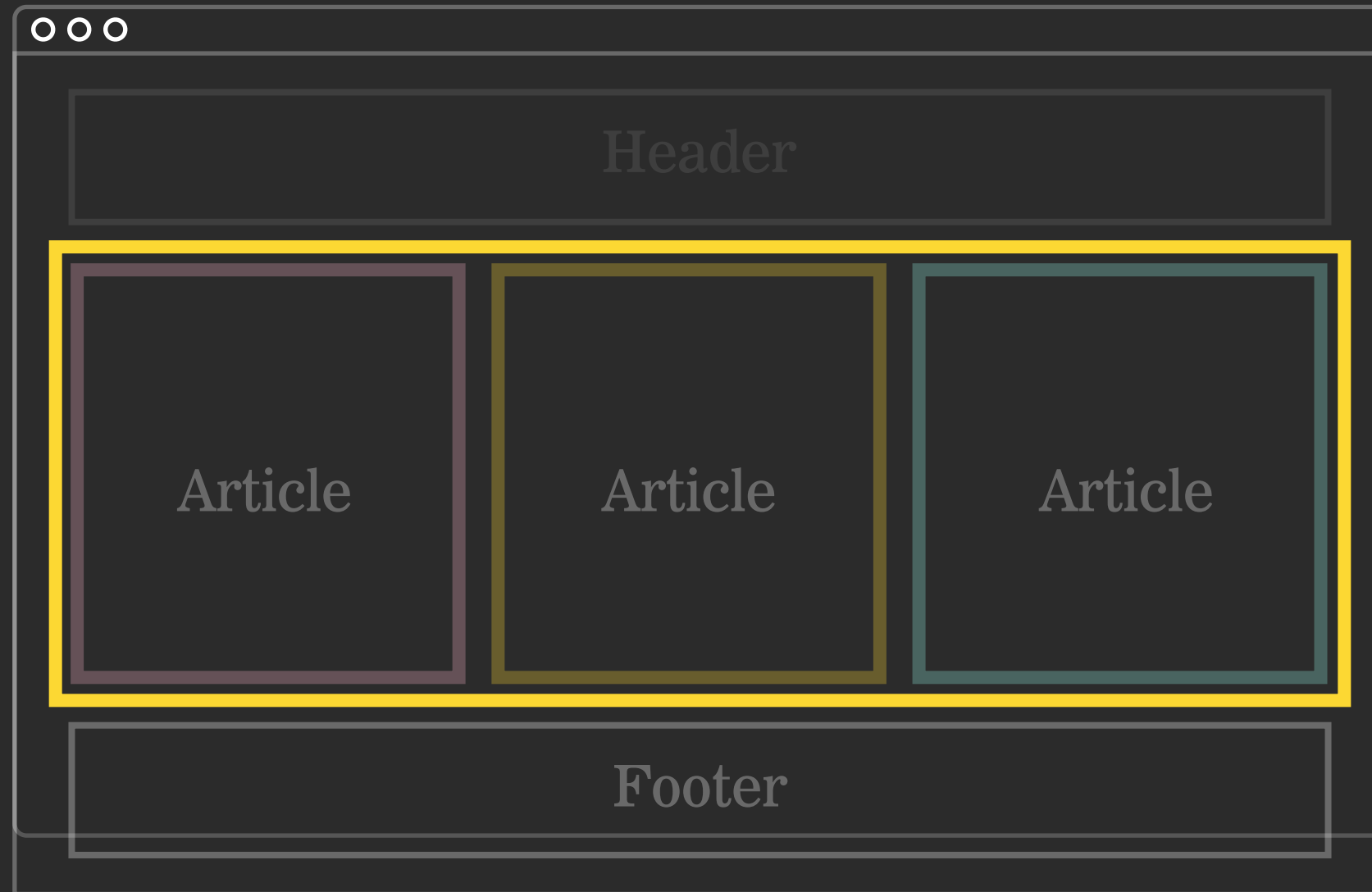
4. GIVE EACH COLUMN A WIDTH IN PERCENTAGES

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

CSS — MULTI-COLUMN LAYOUT



CSS — MULTI-COLUMN LAYOUT



HOW DOES FLEXBOX WORK?

First, you use `flex-direction` to indicate whether you want the items in the container -- the flex items — to "read" left-to-right (row), right-to-left (row-reverse), top-to-bottom (column), **or** bottom-to-top (column-reverse).

parent wrapper:
flex container

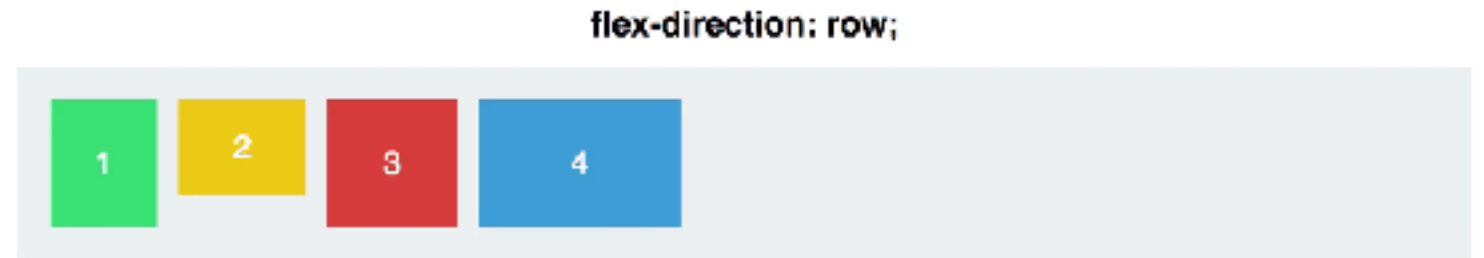
```
<section>
  <article>
    ...
  </article>
  <article>
    ...
  </article>
  <article>
    ...
  </article>
</section>
```

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

FLEX-DIRECTION

flex-direction, however, lets you rotate the **main axis**.

```
section {  
  display: flex;  
  flex-direction: row;  
}
```



```
section {  
  display: flex;  
  flex-direction: column;  
}
```

There's an important distinction to make here: **flex-direction: column** doesn't align the squares on the cross axis instead of the main axis. It makes the *main axis itself* go from horizontal to vertical.

JUSTIFY-CONTENT

`justify-content` controls how you align items on the **main axis**.

You have five commands at your disposal to use `justify-content`:

1. `flex-start`
2. `flex-end`
3. `center`
4. `space-between`
5. `space-around`

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

`justify-content: flex-start;`



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

REFACTORING CSS

- Organize CSS
- Format CSS declarations with each property on a new line, indented with 2-space tabs (ex: <http://d.pr/i/17ISs>)
- Avoid being super specific with your CSS declarations - keep to two to three levels of indentation when nesting
- Try and write your CSS declarations from the most generic declarations (body, class-less elements), to the most specific (classes, ids, specific sections/pages) from top-to-bottom
- Be wary of creating classes or ID names similar to names used by CSS and HTML e.g. a class named 'body' or 'header'
- When wrapping up a project, run CSS through [validator](#) to check for errors.
- When wrapping up a project, run CSS through Autoprefixer to add browser prefixes.

TOOLS OF THE TRADE!

- [Sip](#) — Color Picker for Mac
- [Tape](#) — Chrome browser extension (Be sure to go to extensions and make sure it has access to your file URLs in order for it to work!)
- **Inspect Element!!!**

LAB – MONUMENT



EXERCISE

KEY OBJECTIVE

- Demonstrate the ability to plan and build a website

LAB SESSION

10 min

1. Hook up Google Fonts (Lato 400 and Raleway 400 and 600).

30 min

2. 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

Until 9:15

3. Add base Styles (base fonts, colors, etc.)
4. Add base styles for headers, anchors, text
5. Style everything else!

LEARNING OBJECTIVES

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

WEEKLY OVERVIEW

WEEK 3

Lab / Refactor & Lab

WEEK 4

Forms & Advanced Positioning / Intro to Programming

WEEK 5

jQuery Part 1 / jQuery Part 2

EXIT TICKETS!