
FEWD

START SCREENCAST!!!!

LET'S GET EVERYTHING SET UP!

1. Navigate to the FEWD 42 Dashboard (saraheholden.com/fewd_dashboard/) and download the Lesson 4 starter code and slides. You'll want to keep the dashboard open for other links and resources we'll be referencing in class.
2. Move the starter code and slides from your Downloads folder to the **fewd** folder on your desktop.
3. Double-click on `starter_code_lesson_4.zip` to unzip it
4. After you've unzipped, be sure to delete the original .zip file!
5. Open the entire **fewd** folder with Sublime Text (either drag and drop the folder on the Sublime icon in the dock on Mac, or open Sublime and go to file > open... and select the **fewd** folder).
6. Log in to the FEWD 42 Slack (fewd42.slack.com) and join the class4 channel.

WEEKLY OVERVIEW

WEEK 1

HTML/CSS Basics

WEEK 2

Advanced CSS — Box Model and Layout

WEEK 3

CSS Lab Week / Advanced CSS pt. 2

WEEK 4

Intro to Programming / jQuery Basics

CLASS 3 LAB:

On the Road

The Adventure of a Lifetime

[Blog](#) [Events](#) [Snapshots](#) [Store](#) [Contact](#) [About](#)

11/12/15

Camping by Loon Lake



C ray culpa stumptown, flexitarian ex Odd Future do fugiat Wes Anderson proident 3 wolf moon officia bitters small batch. Et consequat do, nulla you probably haven't heard of them High Life scenester. Vinyl fugiat High Life, kogi do VHS in aliqua lo-fi leggings gentrify Neutra tumblr. Odio vegan PBR, Marfa forage blue bottle try-hard readymade meggings retro delectus Echo Park. Ugh consectetur farm-to-table forage, laboris blue bottle McSweeney's tattooed excepteur yr. Magna ut Schlitz flexitarian, vinyl craft beer proident yr forage 8-bit ethical sustainable placeat retro. Officia pickled beard, adipiscing gluten-free sint craft beer quis thundercats id 3 wolf moon fashion axe.

[Continues...](#)

11/11/15

A Morning Hike in Big Sur



S effies sunt Tumblr, delectus small batch DIY umami sint, Polaroid chambray selfies McSweeney's Cosby sweater, Pitchfork tattooed assumenda Wes Anderson Blue Bottle twee Carles ennui. Nisi locavore fugiat sapiente salvia aliqua. Shoreditch kogi exercitation fashion axe. Wolf semiotics Pinterest, laboris quis master cleanse tousled small batch street art bespoke fingerstache dreamcatcher ethical labore. Dreamcatcher iPhone typewriter, tote bag four loko fanny pack master cleanse organic PBR reprehenderit actually. Craft beer drinking vinegar wayfarers, pork belly nanxhal pariatur polaroid cupidatat in 8-bit distillery anim literally.

[Continues...](#)

[Home](#) | [Blog](#) | [Events](#) | [Snapshots](#) | [Store](#) | [Contact](#) | [About](#) | [Tips](#)

© 2015 Rugged Industries, Inc.

TODAY'S LAB:

On the Road The Adventure of a Lifetime

[Blog](#)[Events](#)[Snapshots](#)[Store](#)[Contact](#)[About](#)

11/12/15

Camping by Loon Lake



C ray culpa stumptown, flexitarian ex Odd Future do fugiat Wes Anderson proident 3 wolf moon officia bitters small batch. Et consequat do, nulla you probably haven't heard of them High Life sceneris. Vinyl fugiat High Life, kogi do VHS in aliqua lo-fi leggings gonrify Neutra tumblr. Odio vegan PBR, Marfa forage blue bottle try-hard readymade meggings retro-delectus Echo Park. Ugh consectetur farm-to-table forage, laboris blue bottle McSweeney's tattooed excepteur yr. Magna ut Schlitz flexitarian, vinyl craft beer proident yr forage 8-bit ethical sustainable placat retro. Officia pickled beard, adipisicing gluten-free sint craft beer quis thundercats id 3 wolf moon fashion axe.

[Continues...](#)

About Us

On the Road is u salvia, fixe mumblecore ex aesthetic qui miron/blog cliché. Retro disrupt keytar PBR, delectus consectetur flexitarian fingerstache selfies nostrud Schlitz. Tempor Wes Anderson banh mi bicycle rights. Eu occaecat Williamsburg yr letterpress, biodiesel plaid tote bag cliché messenger bag kono bespoke sapiente next level.

[More...](#)

Ad

11/11/15

A Morning Hike in Big Sur



S elfies sunt Tumblr, delectus small batch DIY umami sint, Polaroid chambray selfies McSweeney's Cosby sweater, Pitchfork tattooed assumenda Wes Anderson Blue Bottle beer Carles ennui. Nisi locavore fugiat sapiente salvia aliqua. Shoreditch kogi exorcitation fashion axe. Wolf semiotics Pinterest, laboris quis master cleanse tousled small batch street art bespoke fingerstache dreamcatcher ethical labore. Dreamcatcher iPhone typewriter, tote bag four loko fanny pack master cleanse organic PBR reprehenderit actually. Craft beer drinking vinegar wayfarers, pork belly narwhal pariatur polaroid cupidatat in 8-bit distillery anim literally.

[Continues...](#)

Popular Posts

- [Breakfast in the Bay](#)
- [Morning Crescent](#)
- [Above the Clouds](#)
- [Camper update - Kitchen Rehaul](#)
- [Journey through Muir Woods](#)
- [Playing in a Travelling Band](#)
- [Rocky Mountain High](#)
- [Winding our way through Tennessee](#)
- [A Flat Tire and a Lesson Learned](#)
- [Our Brand new Camper!](#)

GA GENERAL ASSEMBLY

LAYOUT

Sarah Holden

LEARNING OBJECTIVES

- Differentiate between block and inline elements
- Identify when HTML5 structural elements should be used
- Apply header, footer, sidebar, and multi-column layouts to build a web page.
- Experiment and predict effects of floats and clearing CSS positioning.

AGENDA



- Display
- Divs
- HTML5 Structural Elements
- Box-Sizing Part 2
- Floats
- Multi-column layouts
- Lab — Travel Blog pt. 2

FEWD

REVIEW

HTML SYNTAX — TAGS

Opening tag

Closing tag

<tag name>content</tag name>

Element

HTML SYNTAX — ATTRIBUTES

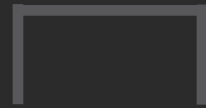
Attribute
Name

<tagName **name**=**"value"**></tagName>

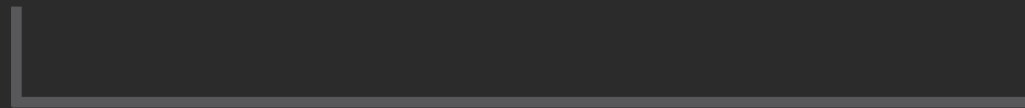
Attribute
Value

CSS SYNTAX

Selector



```
h1 { color: yellow; }
```



Declaration

CSS SYNTAX

h1, h2 {

color: yellow;

font-size: 16px;

}

Property

Value

EXIT TICKET QUESTIONS

If you have two conflicting statements in CSS, such as:

```
h1 a {  
    background-color: red;  
}  
h1 a {  
    background-color: blue;  
}
```

I know it will choose blue.

Would it be beneficial to just delete the first statement so that the browser doesn't have to do as much work. Or is it good convention to leave it in there to let developers know what you had previously done?

EXIT TICKET QUESTIONS

- ▶ How do I add padding around paragraph elements?
- ▶ Will we learn Bootstrap?
- ▶ Is there a good way to identify duplicated or otherwise unnecessary styles added to a particular element, other than delete-and-refresh? Or other ways to make the CSS as succinct as possible?
- ▶ Google font weights

SELECTORS

SELECTOR:

	MEANING:	EXAMPLE:
UNIVERSAL	Applies to all elements in the document	* {}
TYPE	Matches element names	h1, h2, h3 {}
CHILD	Matches an element that is a direct child of another element	p>a {}
DESCENDANT	Matches an element that is a descendent (not just a child) of another element	p a {}
ADJACENT SIBLING	Matches the element that is directly after another element	p+a {}
GENERAL SIBLING	Matches the element that is a sibling of another	p~a {}

CLASSES AND IDS

- Classes and ids allow us to assign 'labels' to elements so that we can target them in our stylesheets

IDS

- Ids are used to target *one specific element*
- **Important:** two elements on the same page cannot have the same id

```
<h3 id="about">Content</h3>
```

```
#about {  
  color: #ff0000;  
}
```

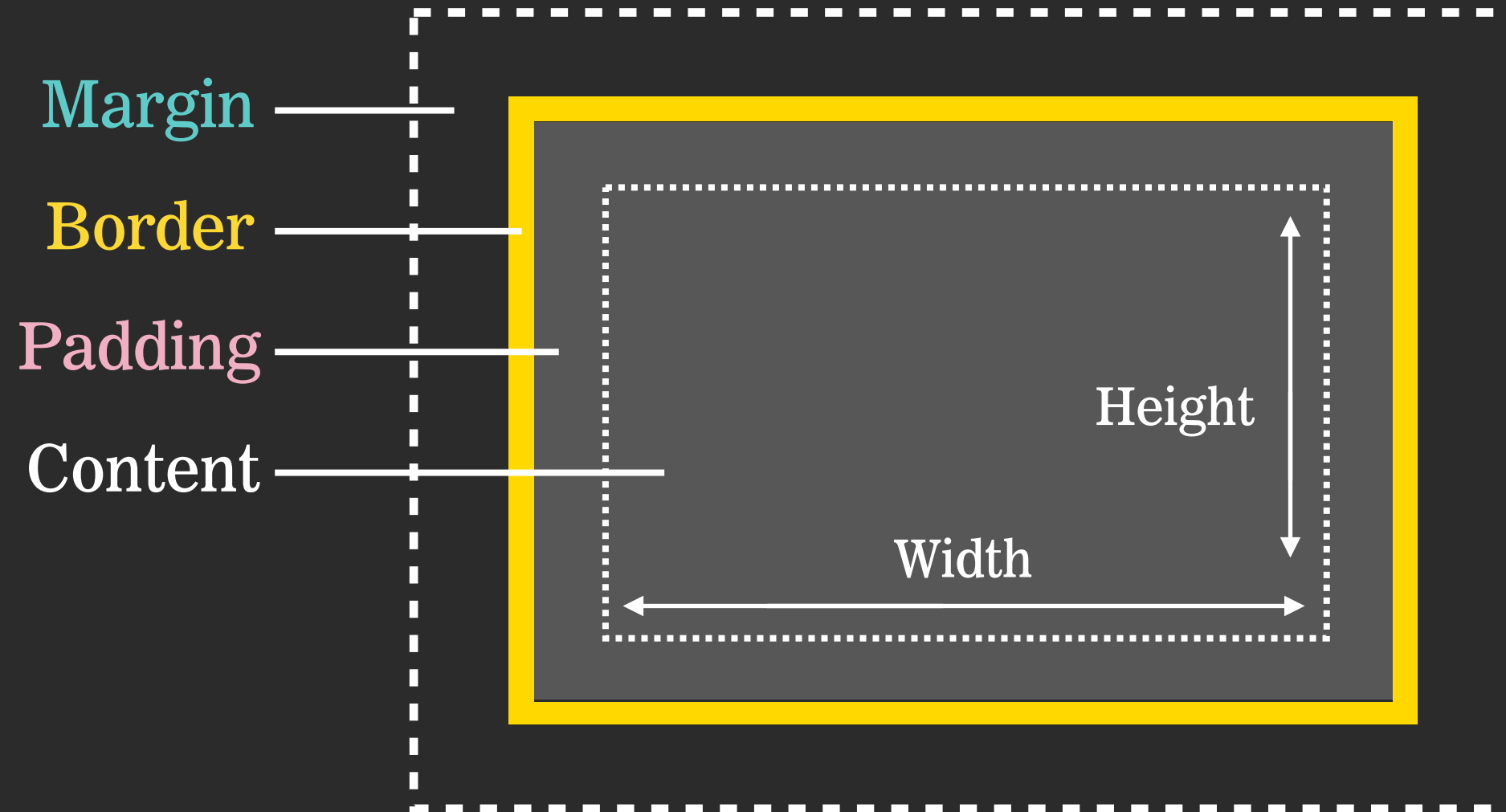
CLASSES

- Classes are used to group elements together

```
<li class="emphasis">Content</li>
```

```
.emphasis {  
  color: #ff0000;  
}
```

REFRESHER — BOX MODEL



FEWD

DISPLAY

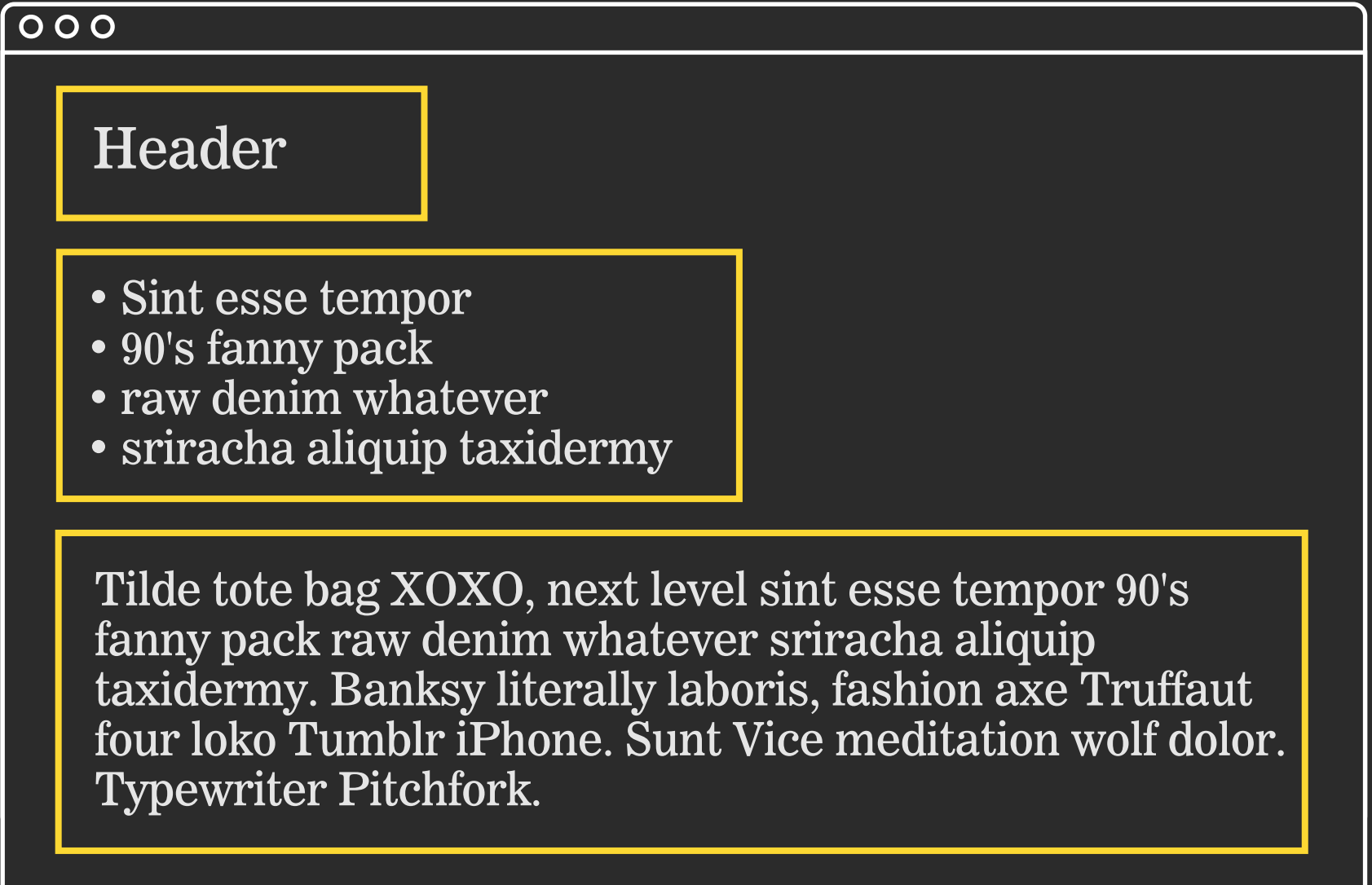
BUILDING BLOCKS

BLOCK-LEVEL ELEMENTS

- ▶ Will always start on a new line

Examples:

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



BUILDING BLOCKS

INLINE ELEMENTS

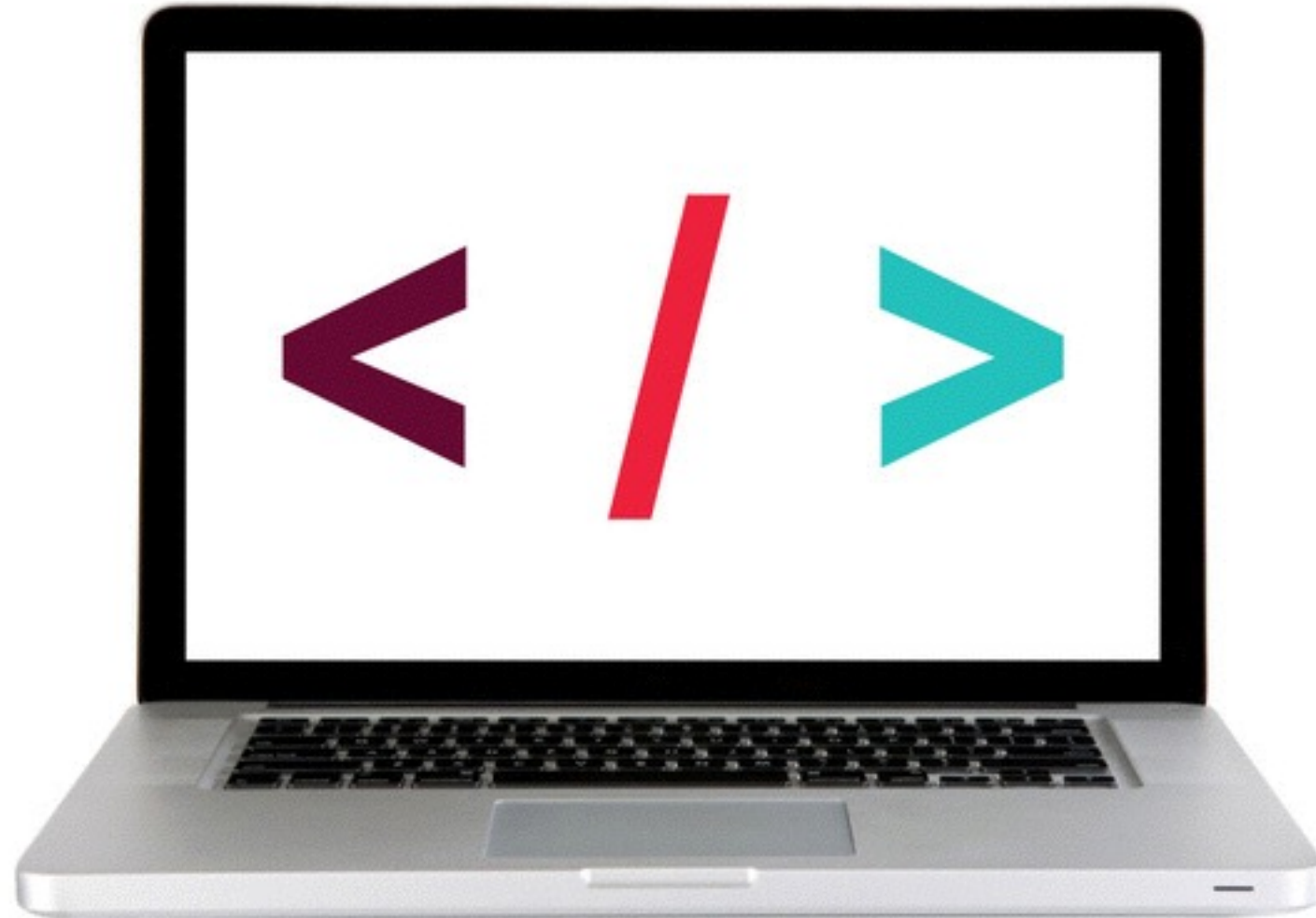
Will always appear to continue on the same line as their neighboring elements

Examples:

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



LET'S TAKE A CLOSER LOOK – PART 1



DIMENSION – A KEY DIFFERENCE BETWEEN INLINE AND BLOCK ELEMENTS

If you try to add dimension to an inline element:

- Some properties will be applied
- Some properties will be *partially* applied
- Others will *not* be applied at all

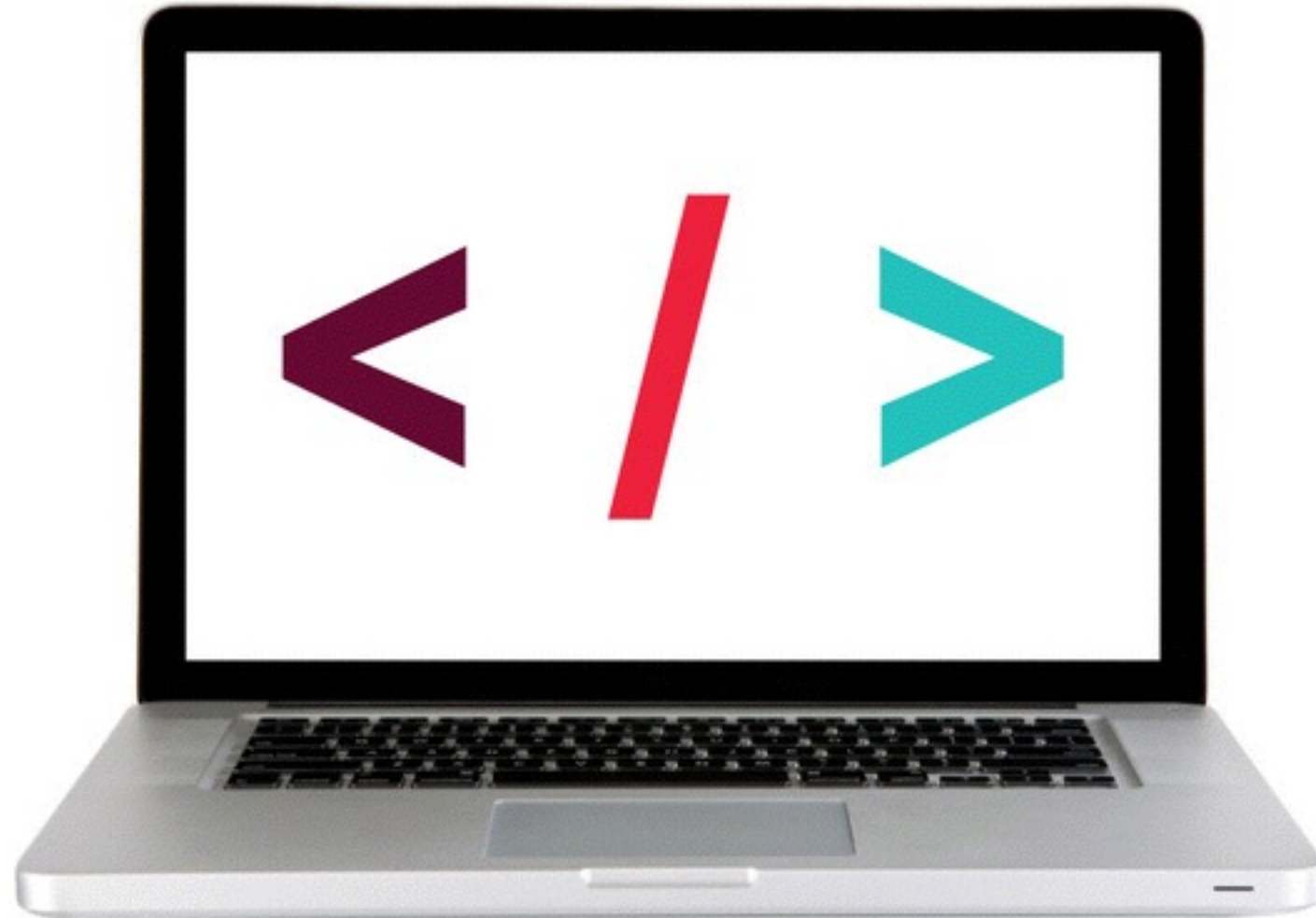
The most noticeable properties are width, height, margin and padding.

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

LET'S TAKE A CLOSER LOOK – PART 2



DISPLAY

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

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

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

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

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

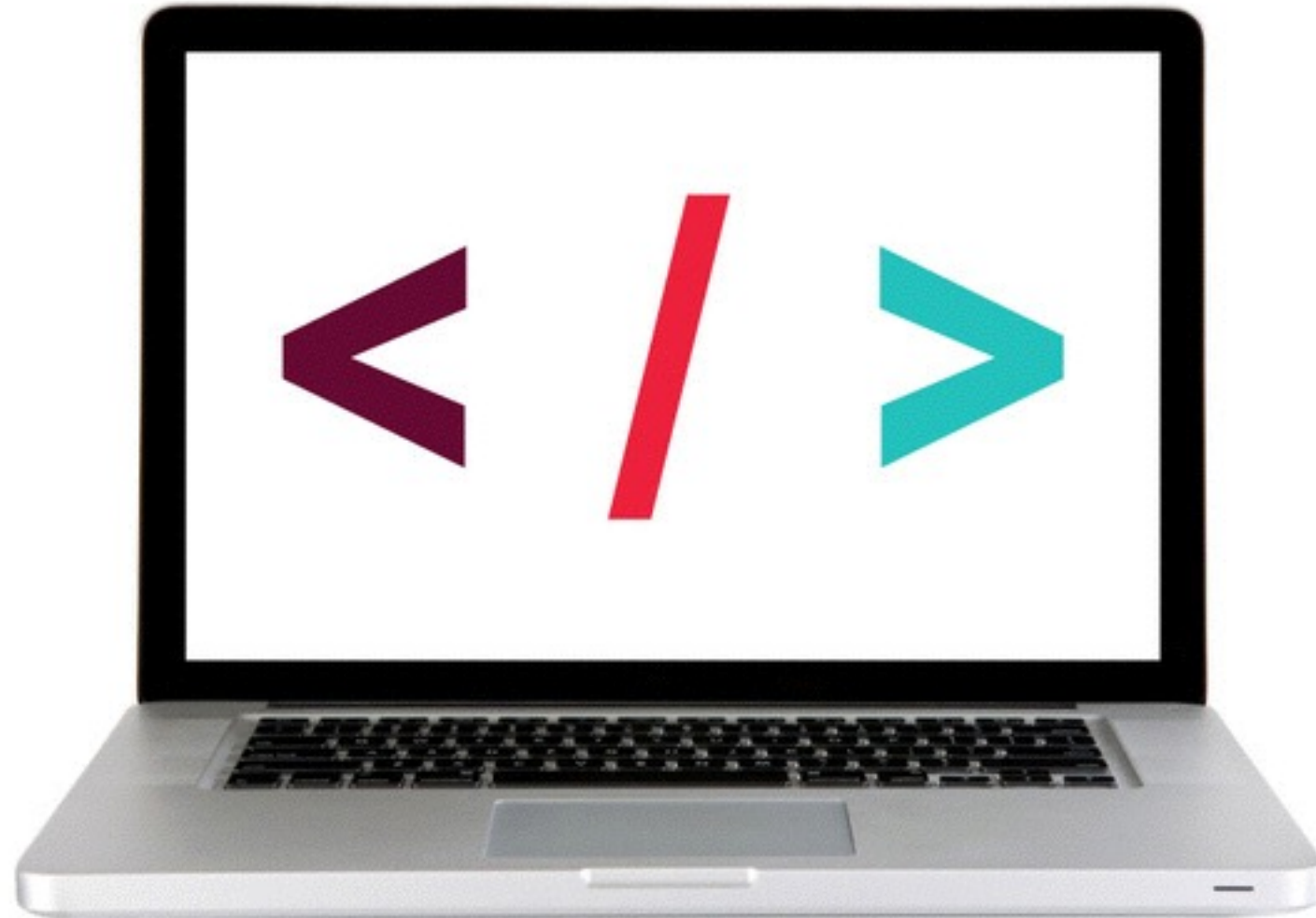
3. Make a block-level element flow like an inline element, while retaining width, height, padding, and margin:

```
h2 {  
  display: inline-block;  
}
```

4. Hide an element from a page:

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

LET'S TAKE A CLOSER LOOK



ACTIVITY — DISPLAY LAB



EXERCISE

KEY OBJECTIVE

- ▶ Get practice using the 'display' property

LOCATION OF FILES

- ▶ `starter_code_lesson_4` > **display_lab** folder

TIMING

5 min

1. Follow the instructions in steps 1-3

FEWD

DIVS

GROUPING TEXT & ELEMENTS

THE <DIV> ELEMENT

- Defines a section or division in an HTML document
- Allows us to group a set of elements together into a block-level box

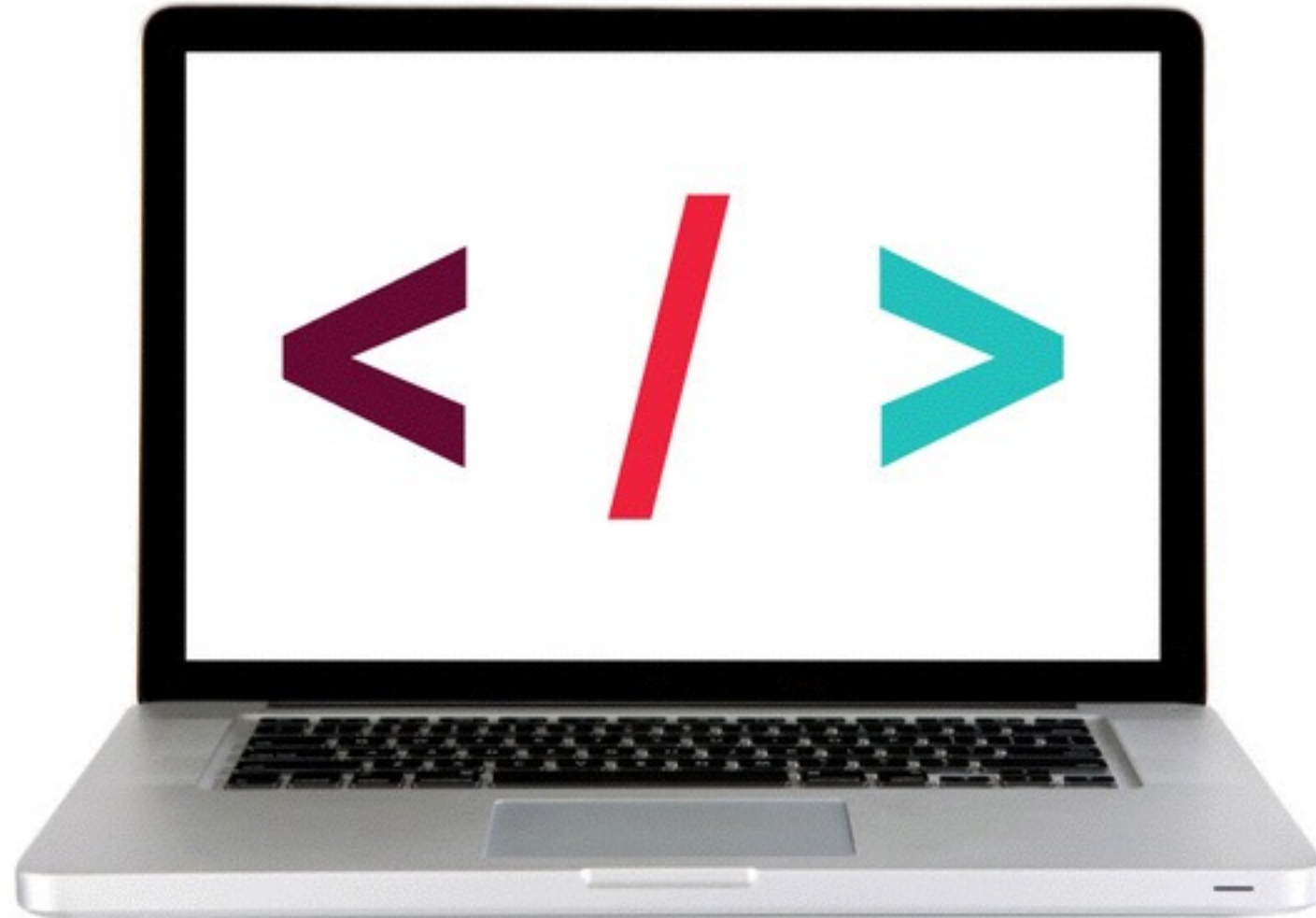
**** Divs allow developers to section off parts of a page.**

THE ELEMENT

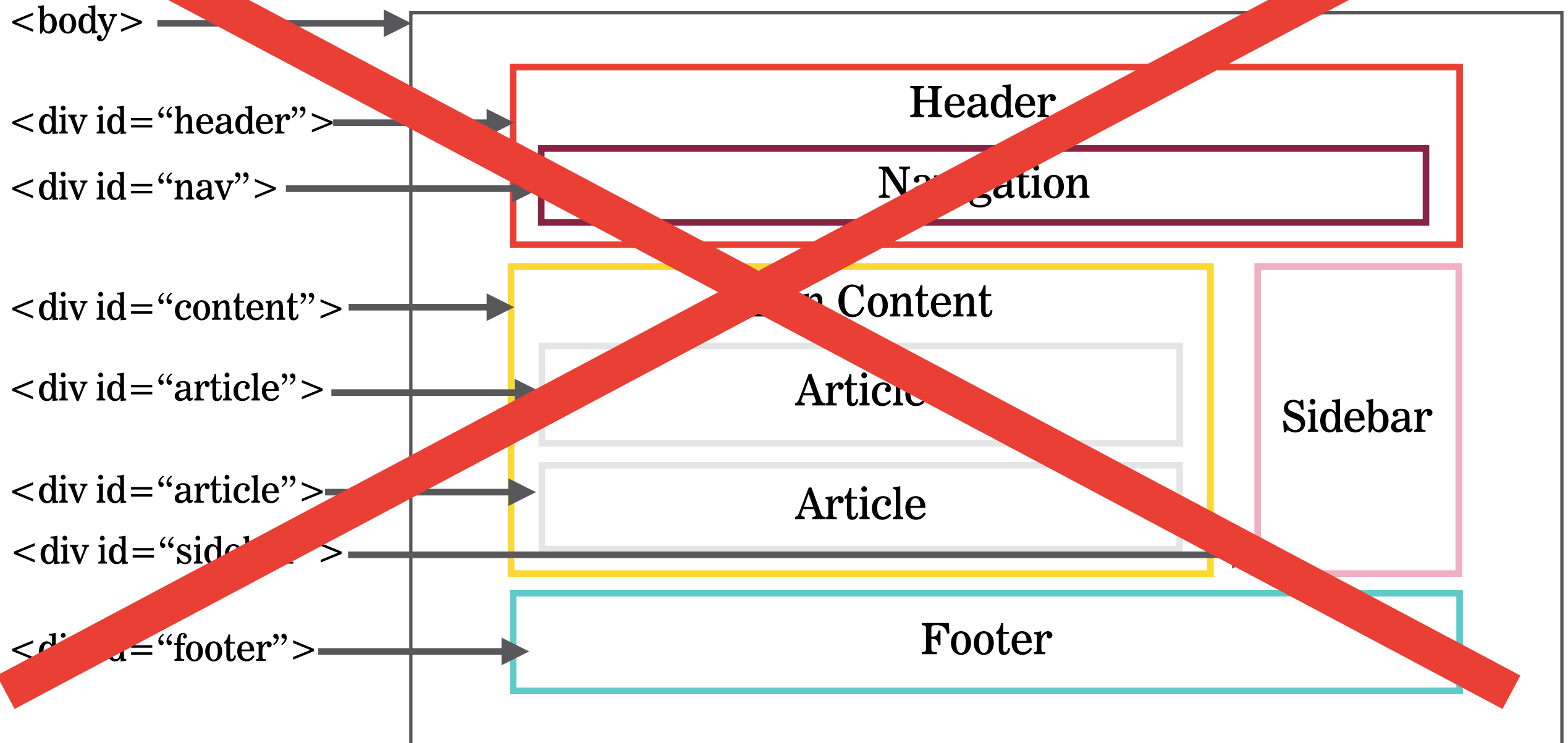
The element acts like the **inline equivalent** of the <div> element. It is used to either:

1. Style one little piece of text within a larger paragraph
2. Contain several inline elements

LET'S TAKE A LOOK



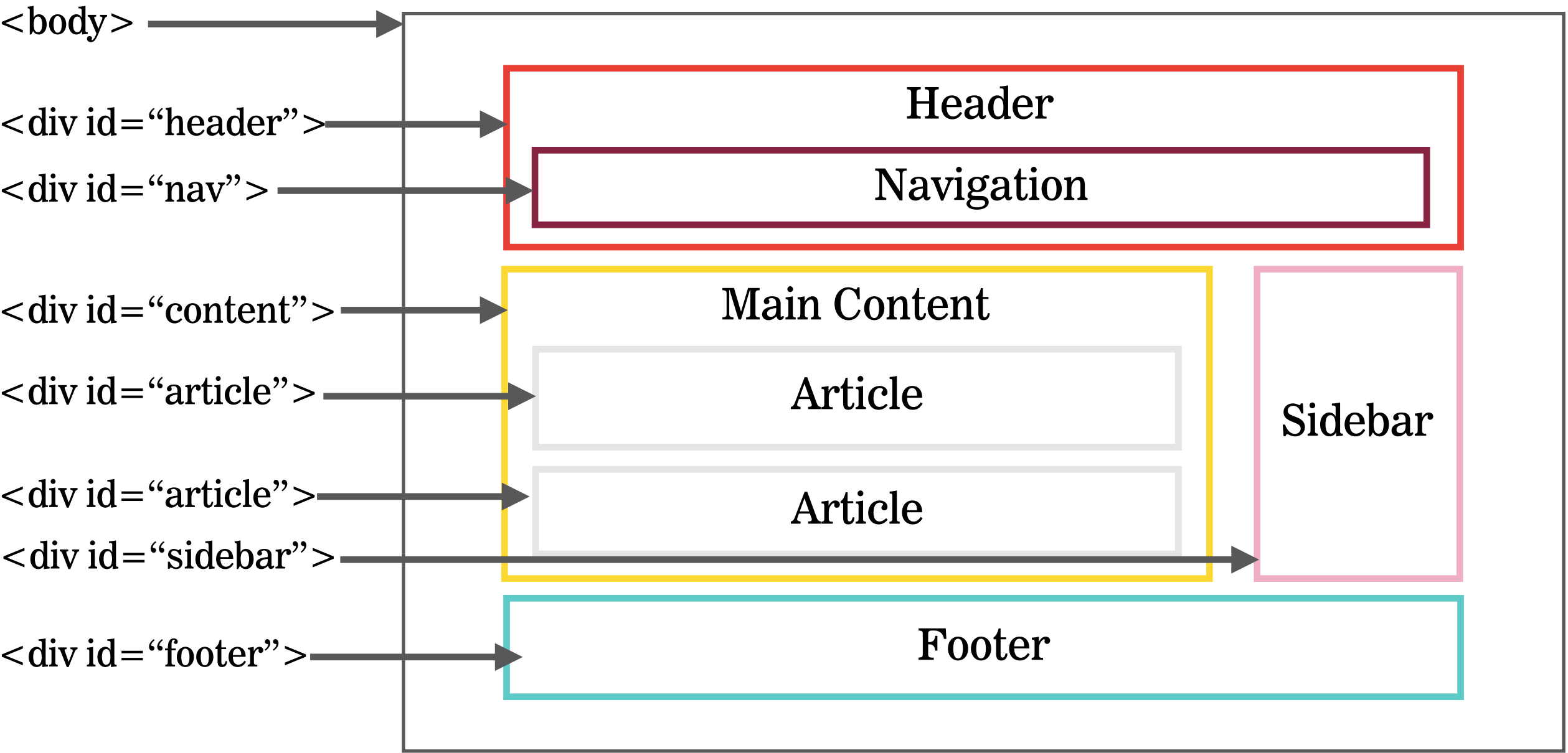
~~DIV = SECTIONING OFF PARTS OF A WEBPAGE (OLD WAY OF DOING THINGS)~~



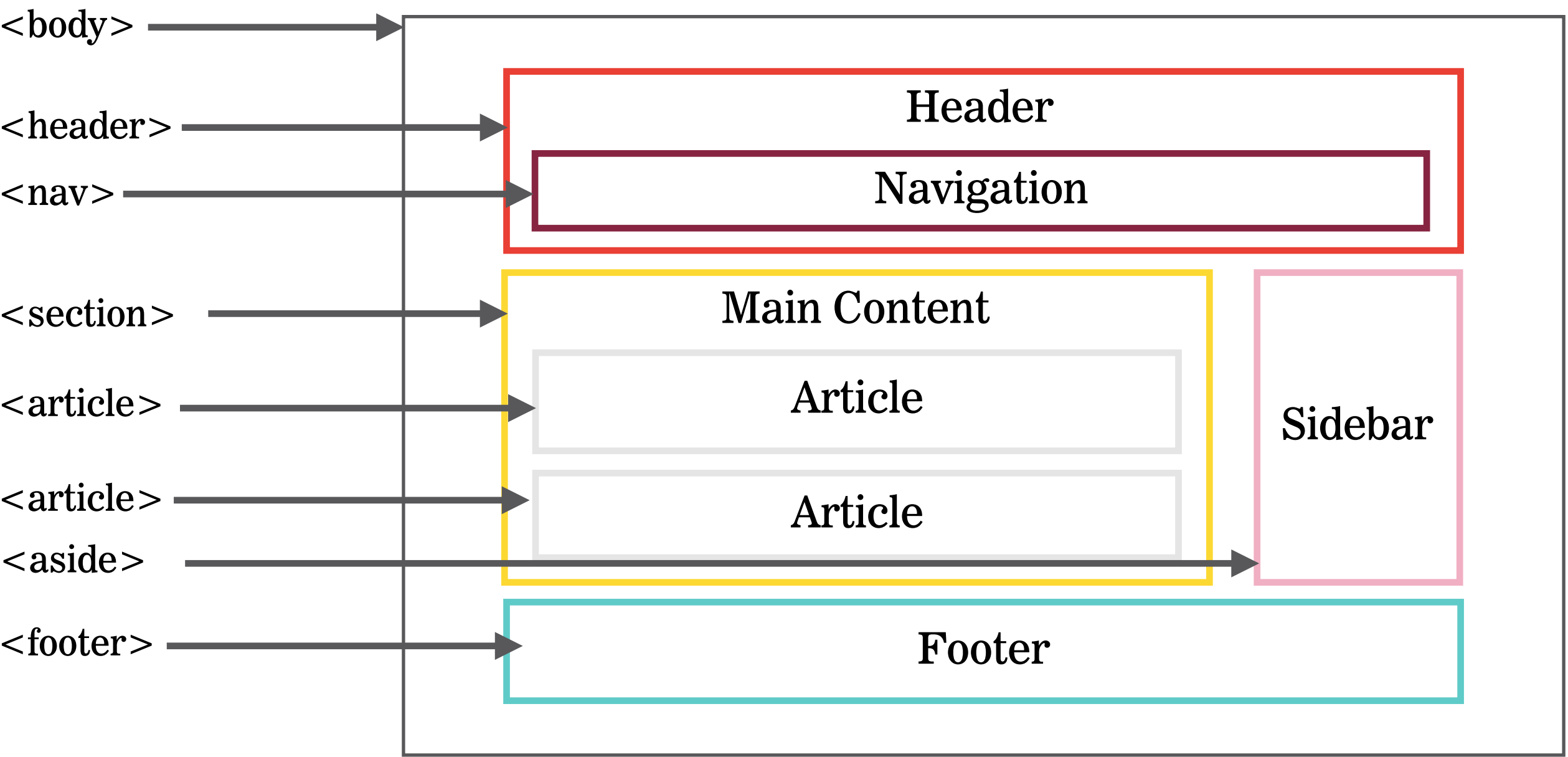
FEWD

HTML5 STRUCTURAL ELEMENTS

TRADITIONAL HTML LAYOUTS



STRUCTURAL ELEMENTS



STRUCTURAL ELEMENTS

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>HTML5 World</title>
</head>
<body>

  <div id="header">
    <h1>Page title</h1>
    <div id="nav">
      <a href="">Link to page</a>
      <a href="">Link to page</a>
    </div>
  </div>

  <div id="content">
    <div id="article">
      <p>Hi! I'm an article</p>
    </div>
    <div id="article">
      <p>Hi! I'm an article</p>
    </div>
  </div>

  <div id="sidebar">
    <p>Hi, I'm the sidebar</p>
  </div>

  <div id="footer">
    <p>Copyright info</p>
  </div>

</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>HTML5 World</title>
</head>
<body>

  <header>
    <h1>Page title</h1>
    <nav>
      <a href="">Link to page</a>
      <a href="">Link to page</a>
    </nav>
  </header>

  <section>
    <article>
      <p>Hi! I'm an article</p>
    </div>
    <article>
      <p>Hi! I'm an article</p>
    </div>
  </section>

  <aside>
    <p>Hi, I'm the sidebar</p>
  </aside>

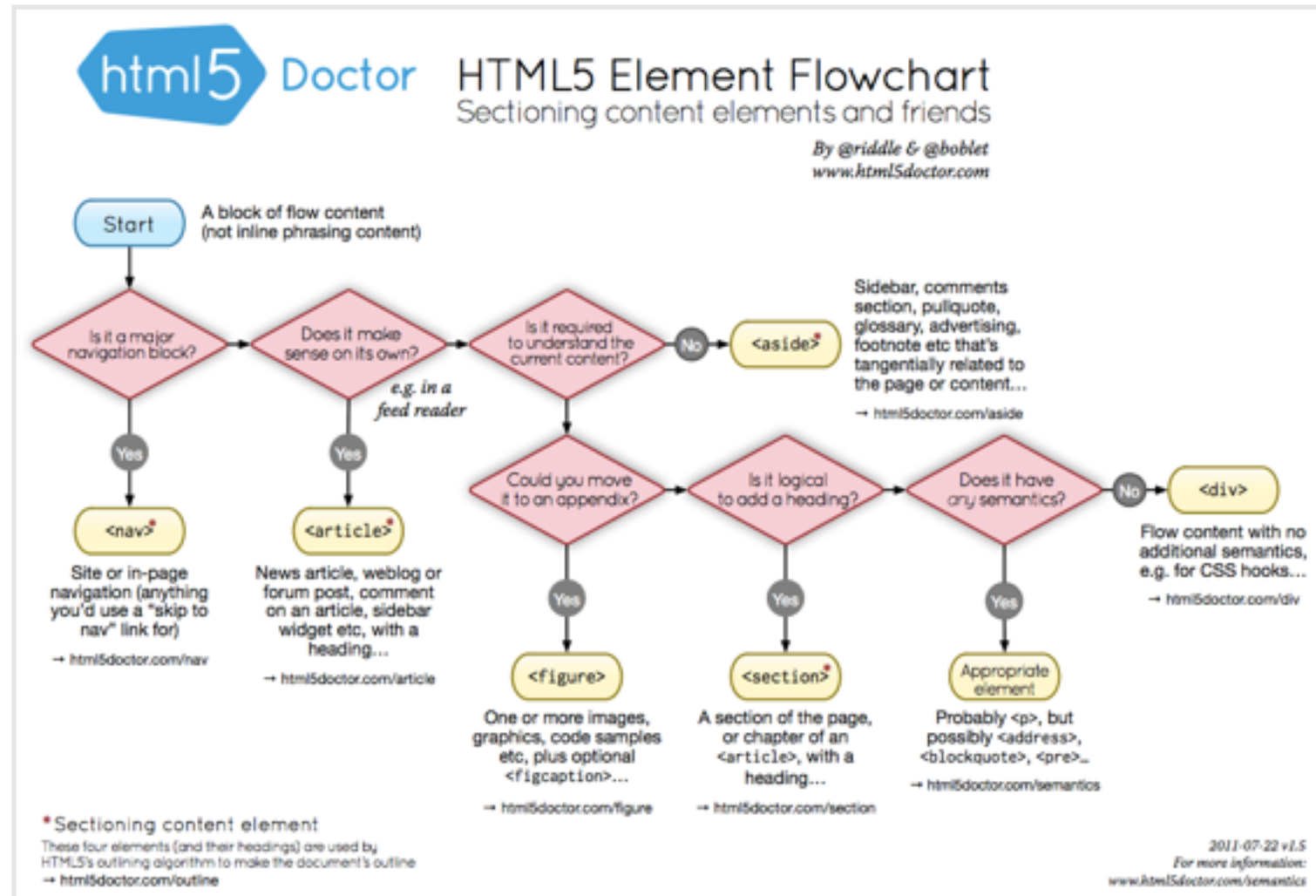
  <footer>
    <p>Copyright info</p>
  </footer>

</body>
</html>
```

SO...IS THERE STILL A PLACE FOR DIVS IN AN HTML5 WORLD?

- Yes! The `<div>` still has a place in the HTML5 world
- You should use `<div>` when there is **no other more semantically appropriate element that suits your purpose**
- Its most common use will likely be for stylistic purposes — i.e., wrapping some semantically marked-up content in a CSS-styled container.

HTML5 ELEMENT FLOWCHART



ACTIVITY — 'DIV' UP THE CONTENT



EXERCISE

KEY OBJECTIVE

- Identify content sections

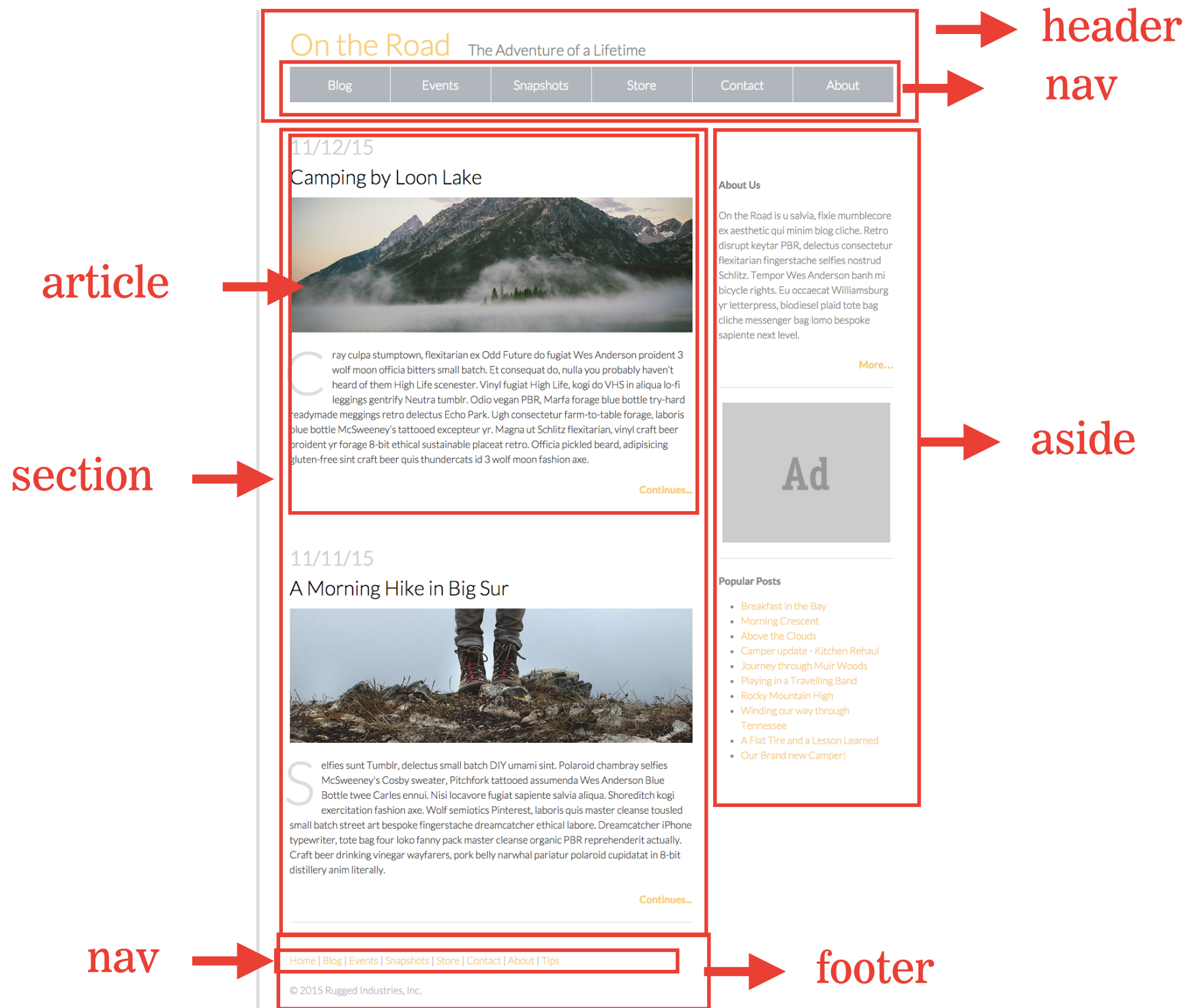
TYPE OF EXERCISE

- Partner

TIMING

8 min

1. First draw boxes around the content you think should live inside a sectioning element — a div, header, footer, etc.
2. Then determine which boxes/divs should have a class or id. Look for similarities to determine what should be a class.



STEPS TO ACHIEVE A MULTI-COLUMN LAYOUT

1. MAKE SURE EACH COLUMN HAS A WRAPPER AROUND IT IN YOUR HTML

2. ADD BOX-SIZING: BORDER-BOX; TO EVERYTHING

3. GIVE A WIDTH TO EACH COLUMN

4. FLOAT EACH COLUMN TO LEFT

5. USE PADDING TO ADD SPACE BETWEEN COLUMNS

**6. CLEAR ANYTHING UNDERNEATH YOUR COLUMNS I.E. A FOOTER
USING THE CSS CLEAR PROPERTY (CLEAR: BOTH;)**

LET'S TAKE A CLOSER LOOK

Header

Main Content

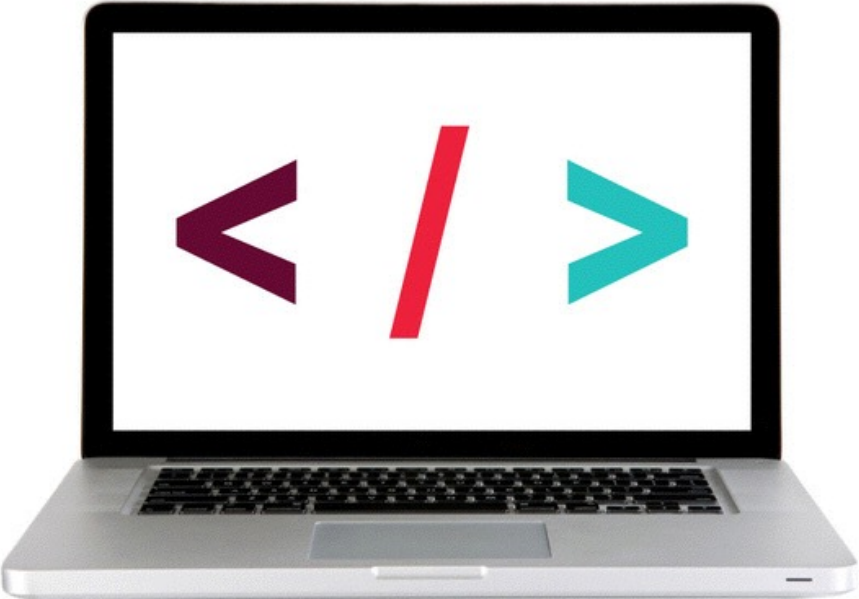
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nam dicta explicabo illum sint, incidunt aut suscipit recusandae, ut voluptatibus, sunt ex placeat? Dolores adipisci id animi soluta eligendi, voluptatibus, et.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Debitis optio harum consequatur dolorem rem beatae ducimus possimus tempore, sequi provident eaque sed explicabo veritatis libero maiores vel praesentium. Sit, nulla.

Sidebar Content



Footer



ACTIVITY — DISPLAY LAB



EXERCISE

KEY OBJECTIVE

- ▶ Work through steps to create a multi-column layout

LOCATION OF FILES

- ▶ `starter_code_lesson_4` > **floating_sections_lab** folder

TIMING

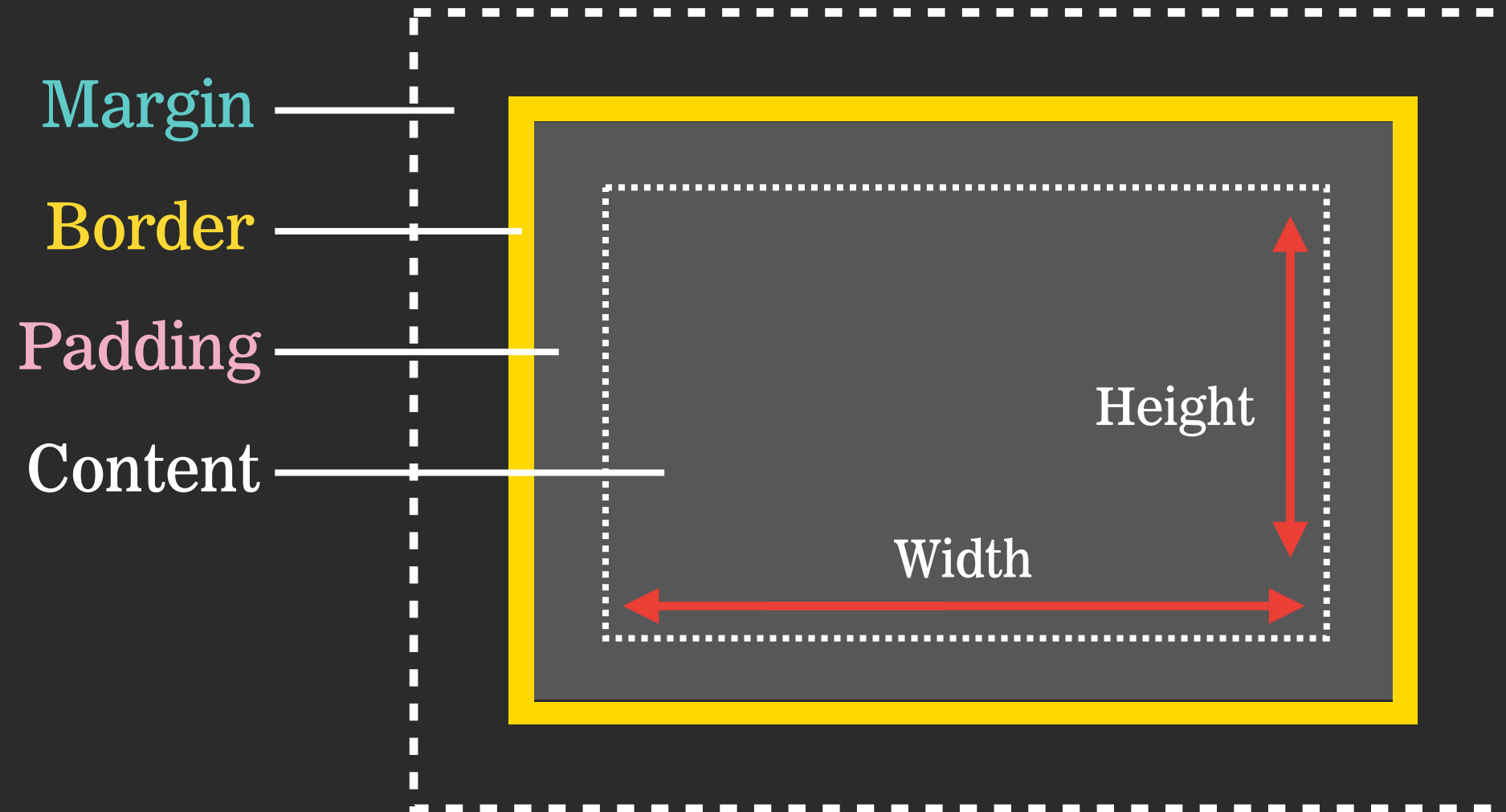
2 min

1. Follow steps #1 and #2 in `index.html`

FEWD

BOX-SIZING FTW!

REFRESHER — BOX MODEL



THE DEFAULT WAY — BOX-SIZING: CONTENT-BOX;

- ▶ **Default box-sizing** (box-sizing: content-box): As soon as an element has either padding or border applied, the actual rendered width is wider than the width you set in your CSS.

Actual width = *width* + *border-left* + *border-right* + *padding-left* + *padding-right*



WIDTH: 25%;

TOTAL WIDTH: 25%

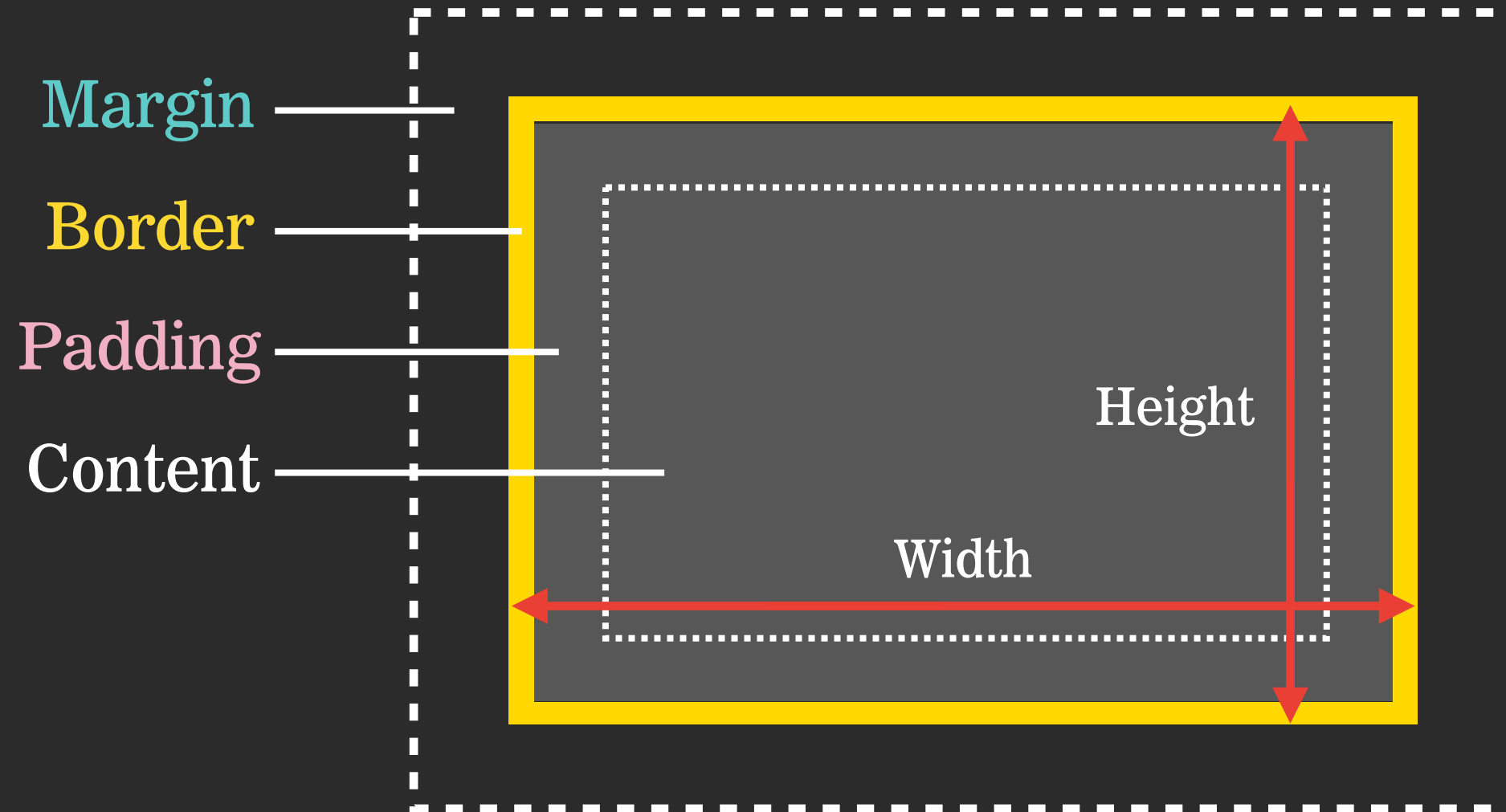


**WIDTH: 25%;
BORDER: 6PX SOLID PINK;**

TOTAL WIDTH: 25% + 6px + 6px

BOO!

BOX-SIZING: BORDER-BOX



HERE'S THE SYNTAX

```
* {  
  box-sizing: border-box;  
}
```

* You'll want to add this at the very top of your CSS file.

WHY IS THIS SO AWESOME?

- ▶ With **box-sizing: border-box** — the padding and border press their way inside the box instead of expanding the box.

Actual width = *Width set in CSS*

YAY!

```
* {  
  box-sizing: border-box;  
}
```

**25% WIDTH AND
BORDER: 6PX SOLID PINK**

TOTAL WIDTH: 25%

YAY!

WHY IS THIS SO AWESOME?

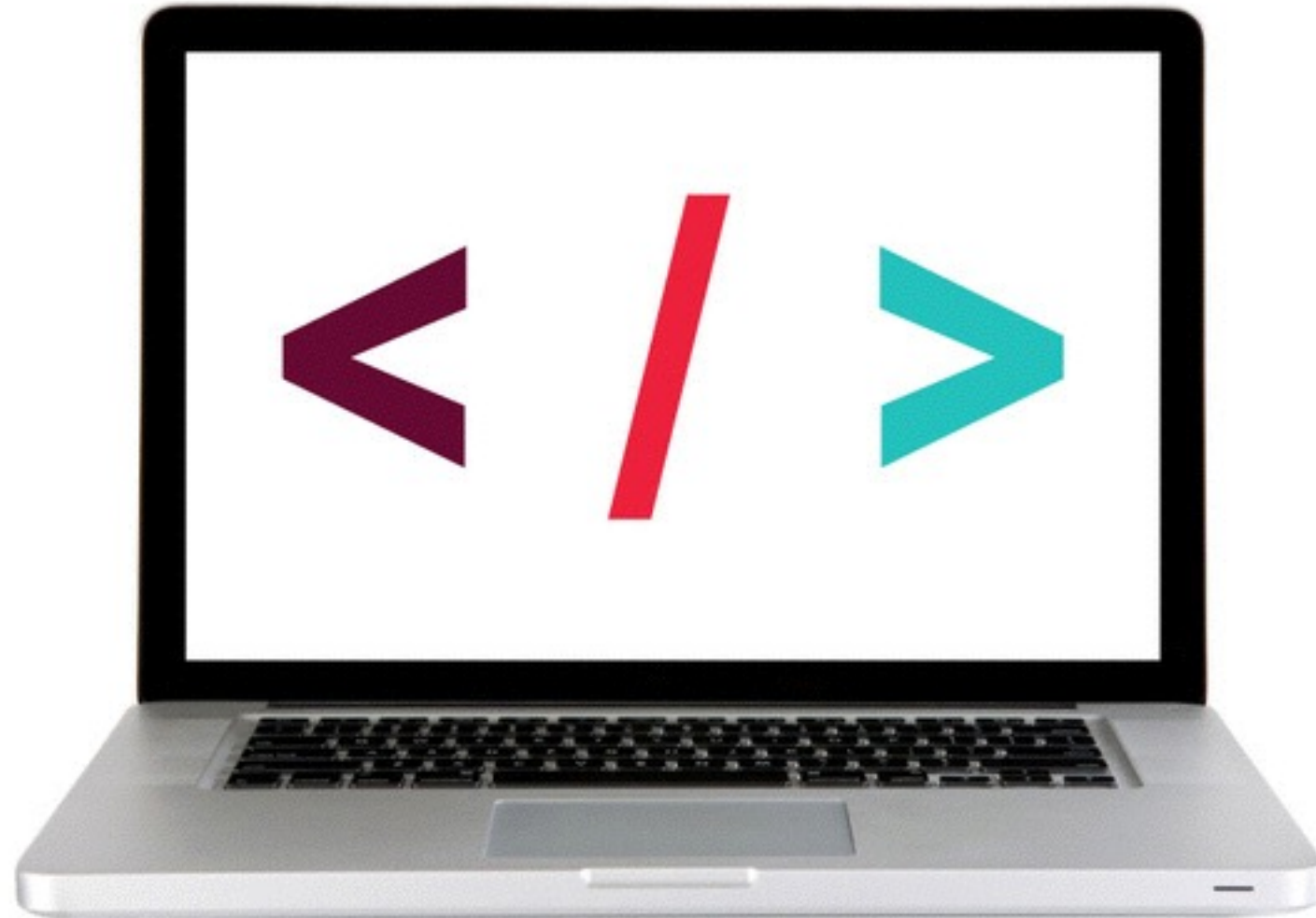
```
* {  
  box-sizing: border-box;  
}
```



**NOW FOUR OF US CAN SIT IN A ROW SIDE BY SIDE NO MATTER
HOW MUCH PADDING AND BORDER WE HAVE.**

YAY!

LET'S TAKE A CLOSER LOOK



STEPS TO ACHIEVE A MULTI-COLUMN LAYOUT

1. MAKE SURE EACH COLUMN HAS A WRAPPER AROUND IT IN YOUR HTML

2. ADD BOX-SIZING: BORDER-BOX; TO EVERYTHING (USE THE * CSS SELECTOR)

3. GIVE A WIDTH TO EACH COLUMN (PREFERABLY IN %)

4. FLOAT EACH COLUMN TO LEFT

5. USE PADDING TO ADD SPACE BETWEEN COLUMNS

**6. CLEAR ANYTHING UNDERNEATH YOUR COLUMNS I.E. A FOOTER
USING THE CSS CLEAR PROPERTY (CLEAR: BOTH;)**

LET'S TAKE A CLOSER LOOK

Header

Main Content

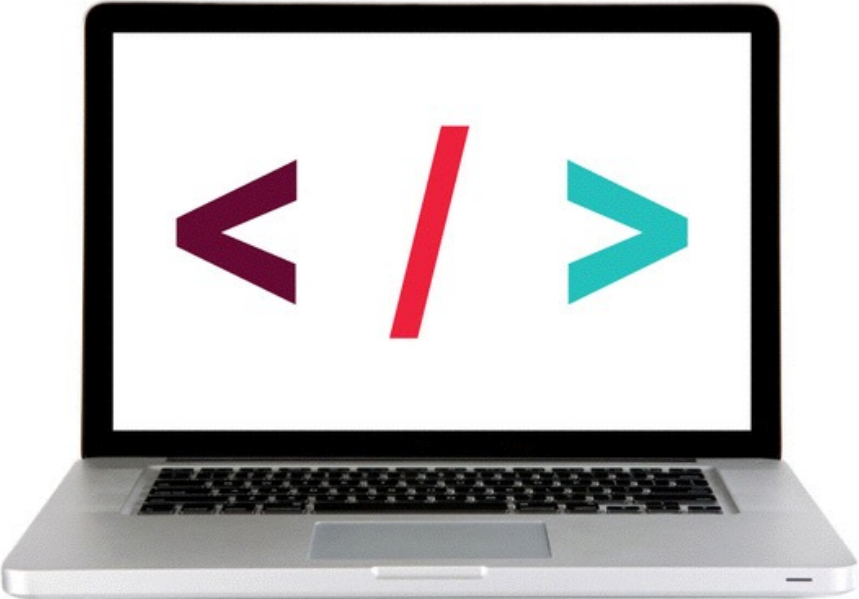
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nam dicta explicabo illum sint, incidunt aut suscipit recusandae, ut voluptatibus, sunt ex placeat? Dolores adipisci id animi soluta eligendi, voluptatibus, et.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Debitis optio harum consequatur dolorem rem beatae ducimus possimus tempore, sequi provident eaque sed explicabo veritatis libero maiores vel praesentium. Sit, nulla.

Sidebar Content



Footer



ACTIVITY — DISPLAY LAB



EXERCISE

KEY OBJECTIVE

- ▶ Work through steps to create a multi-column layout

LOCATION OF FILES

- ▶ `starter_code_lesson_4` > **floating_sections_lab** folder

TIMING

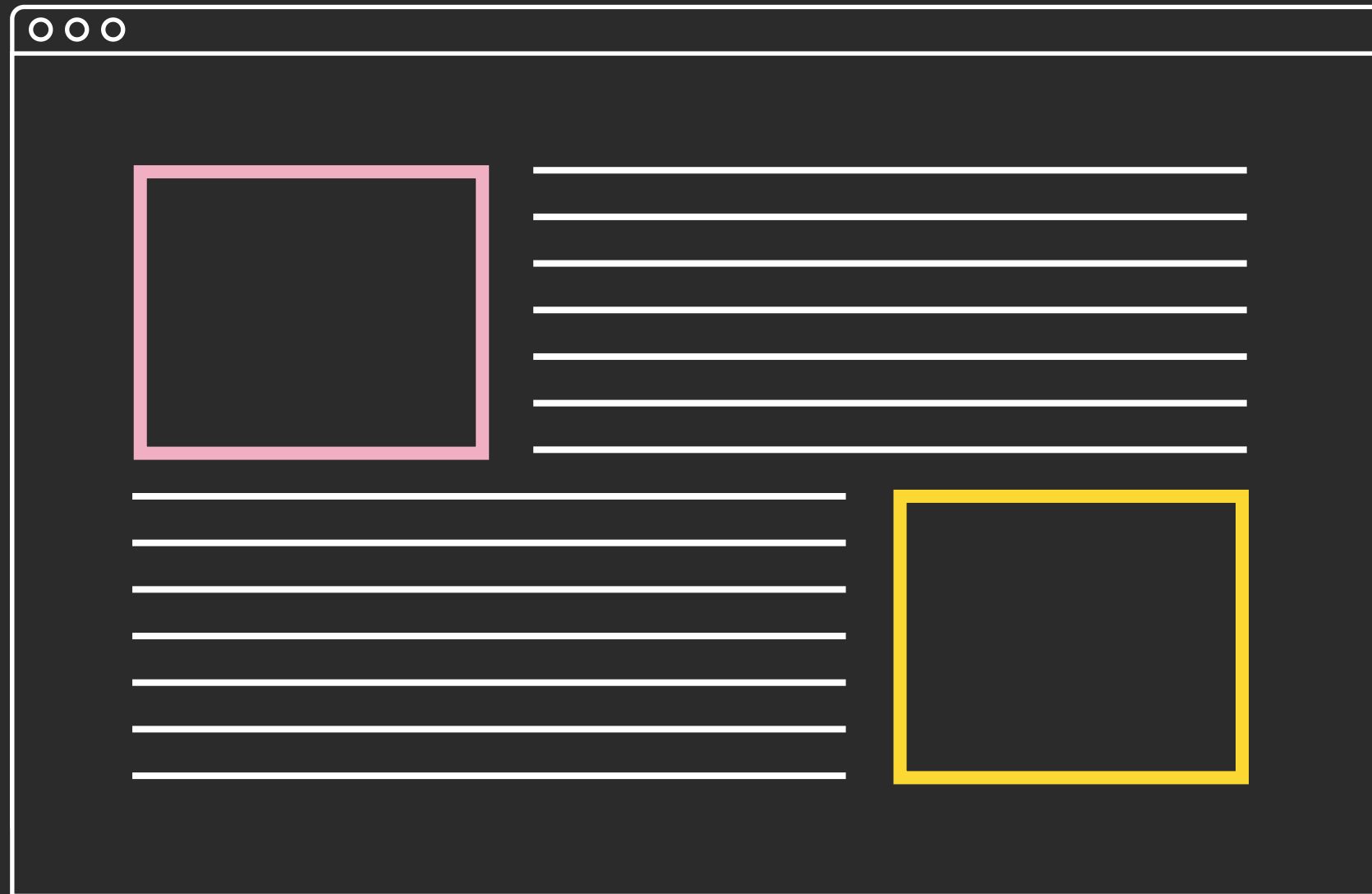
2 min

1. Follow steps #3, #4, and #5 in `style.css`. Steps are a little out of order! You can use command + f to open a search window and search for a step.

FEWD

FLOATS

CSS — FLOATS



FLOATS

Values for the float property:

```
p {  
  float: left;  
}
```

▸ **float: left;** Floats an element to the left side

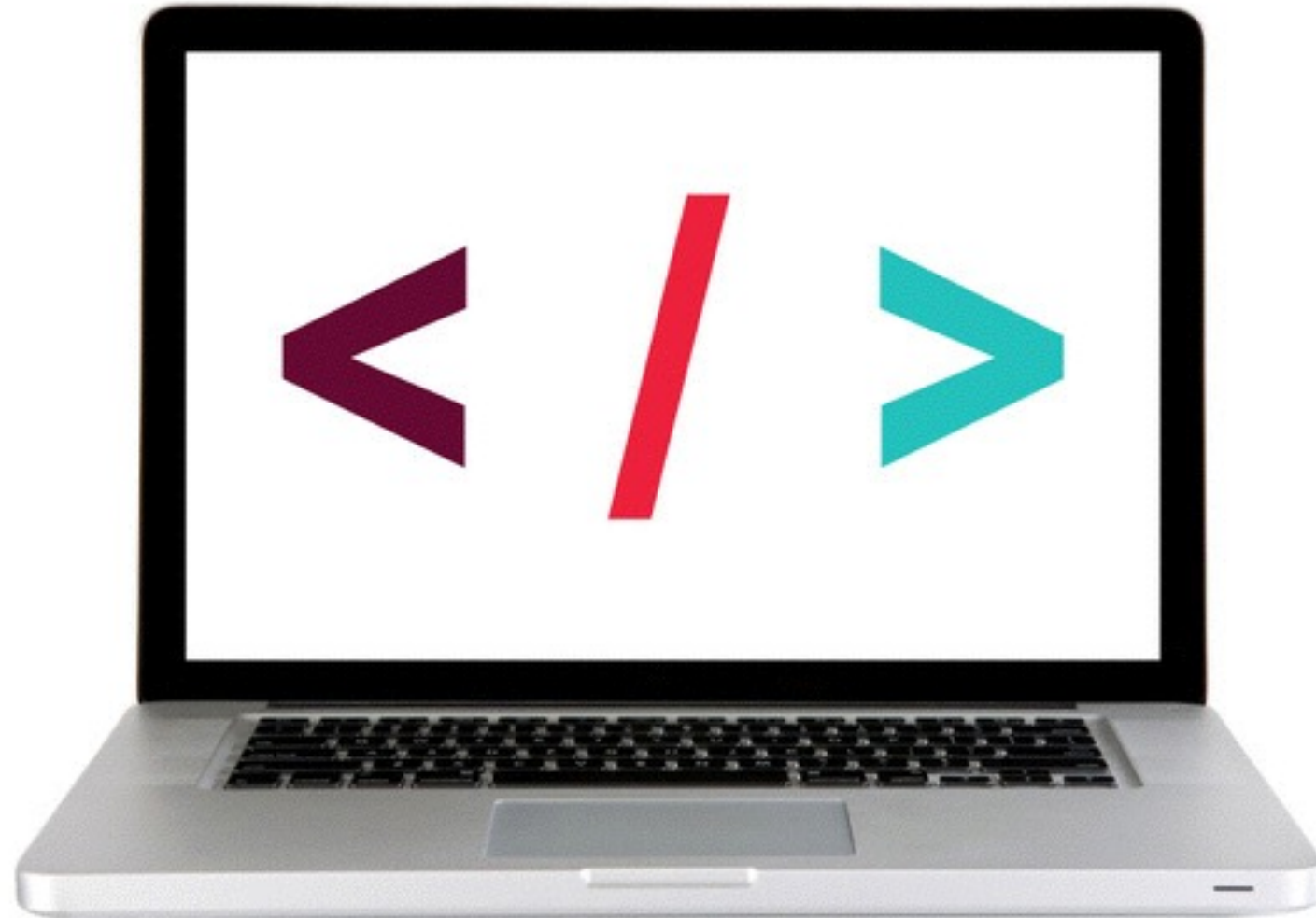
```
h1 {  
  float: right;  
}
```

▸ **float: right;** Floats an element to the right side

```
div {  
  float: none;  
}
```

▸ **float: none;** (the default) ensures the element will not float

LET'S TAKE A LOOK



CLEARING FLOATS

- The **clear** property specifies which side(s) of an element other floating elements are not allowed

LEFT

- No floating elements allowed on the left side

RIGHT

- No floating elements allowed on the right side

```
.clear {  
  clear: both;  
}
```

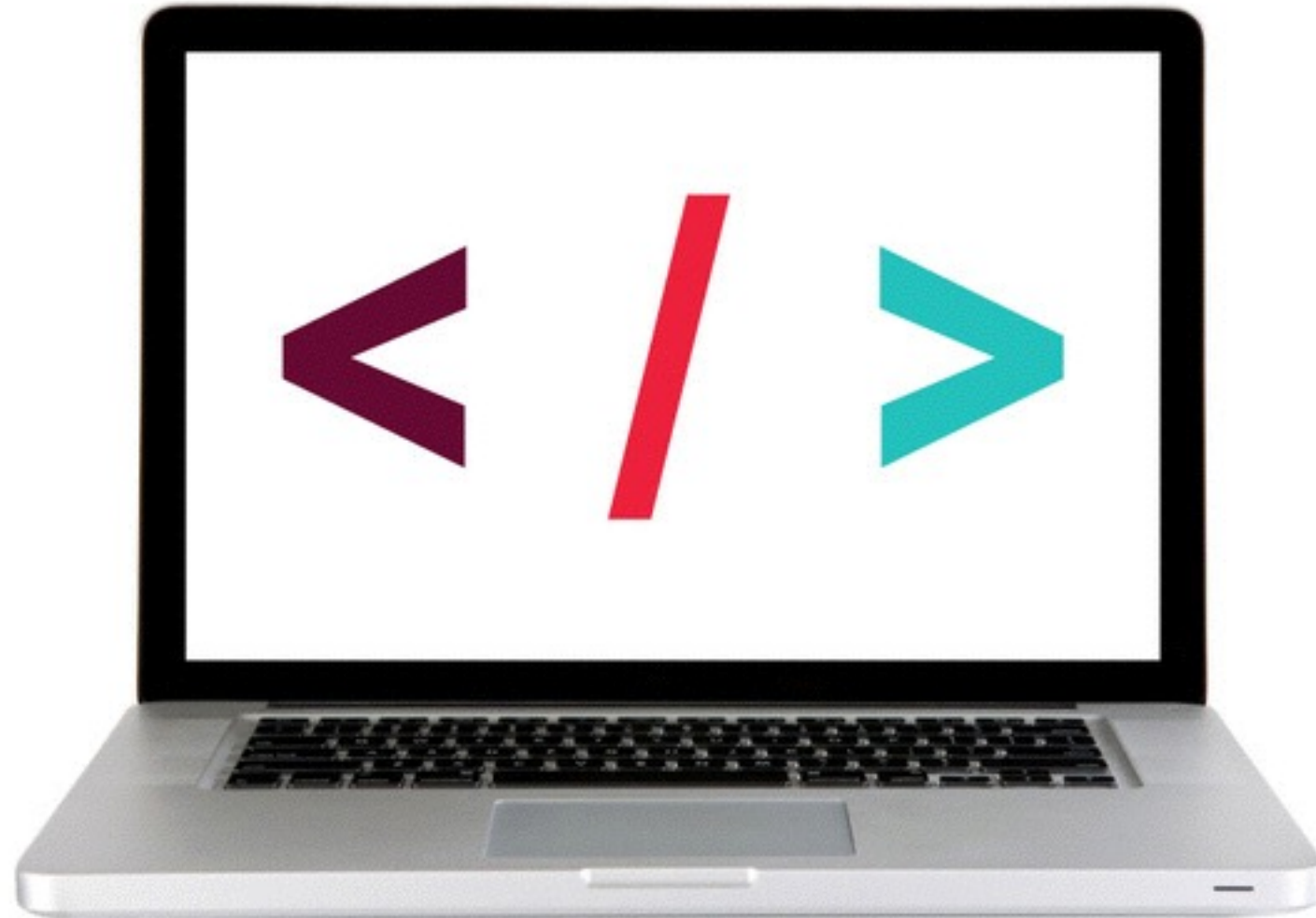
BOTH

- No floating elements allowed on either the left or right side

NONE

- Allows floating elements on both sides

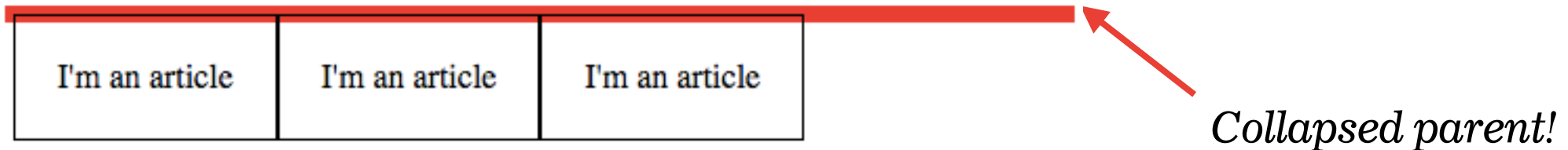
LET'S TAKE A LOOK



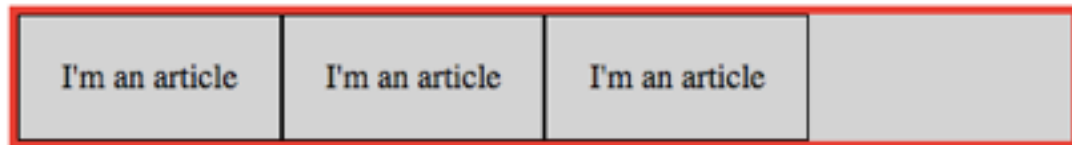
PARENTS OF FLOATED ELEMENTS

- ▶ If a containing element **only contains floated elements**, some browsers will treat it as if it is zero pixels tall.

PROBLEM:



SOLUTION:



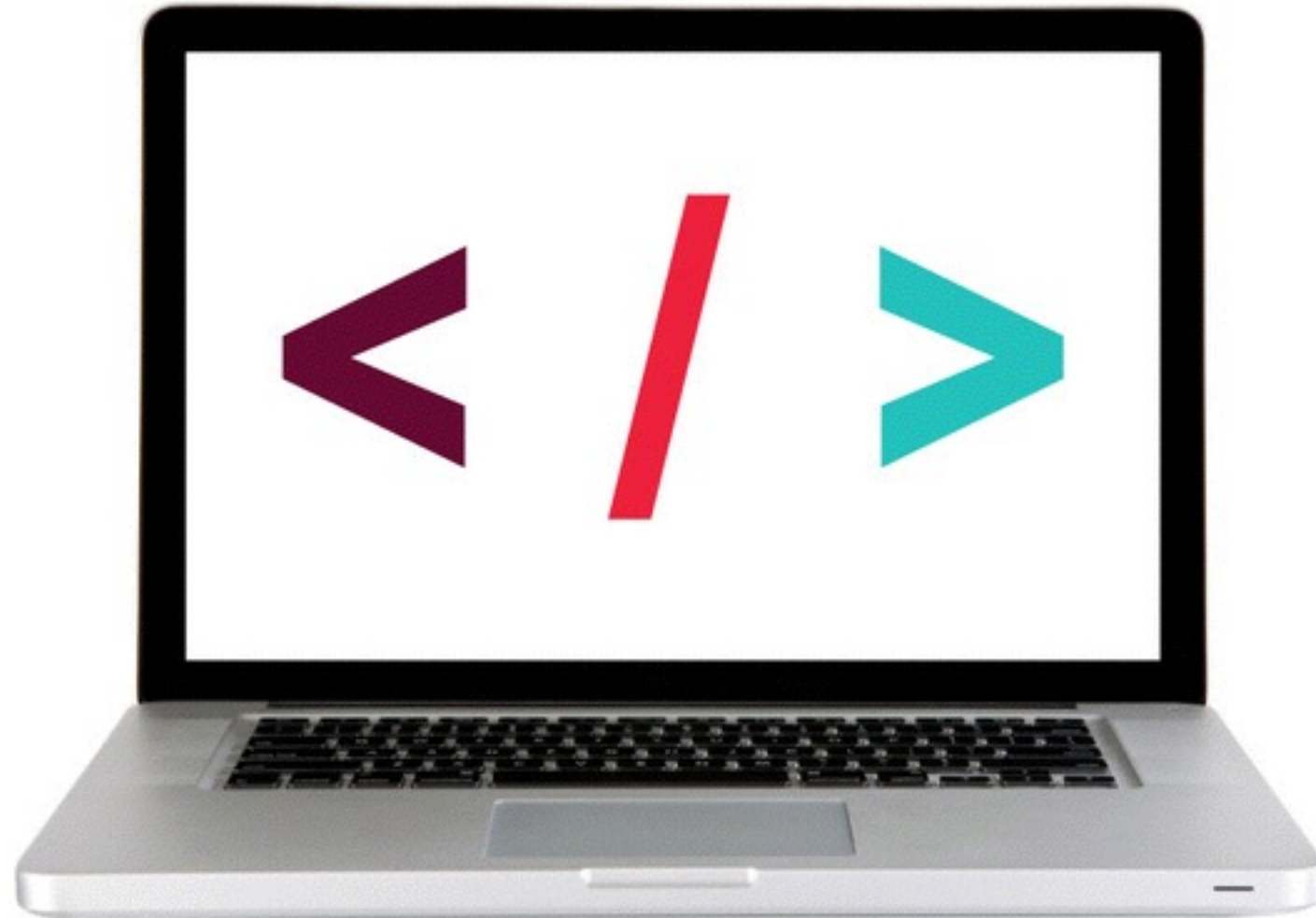
PT. 1 — ADD CSS CLASS:

```
.clearfix:after {  
  content: "";  
  display: table;  
  clear: both;  
}
```

PT. 2 — ADD CLASS TO HTML:

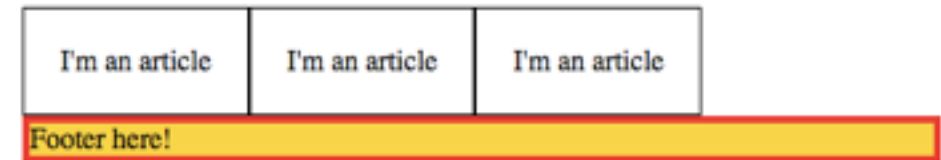
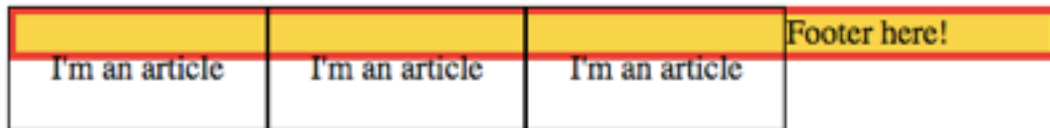
```
<div class="clearfix">  
  <p>1</p> <!-- float: left -->  
  <p>2</p> <!-- float: left -->  
  <p>3</p> <!-- float: left -->  
</div>
```

LET'S TAKE A LOOK



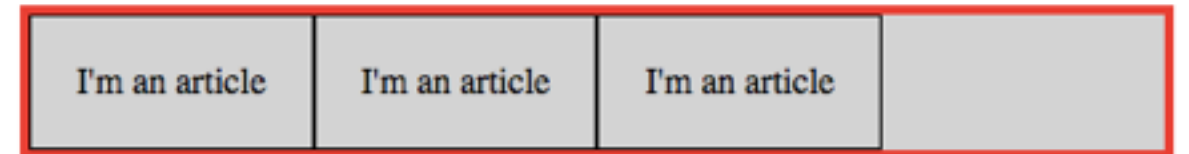
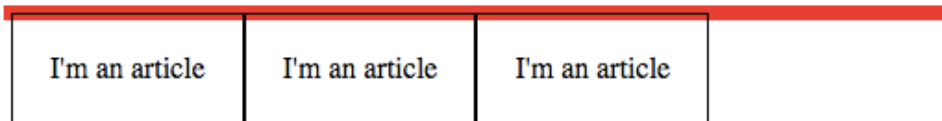
CONFUSING NAMES — KEEPING THINGS STRAIGHT

CLEAR: BOTH; *Make sure an element starts on a new line*



```
footer {  
  clear: both;  
}
```

CLEARFIX: *Fixes collapsed parent*



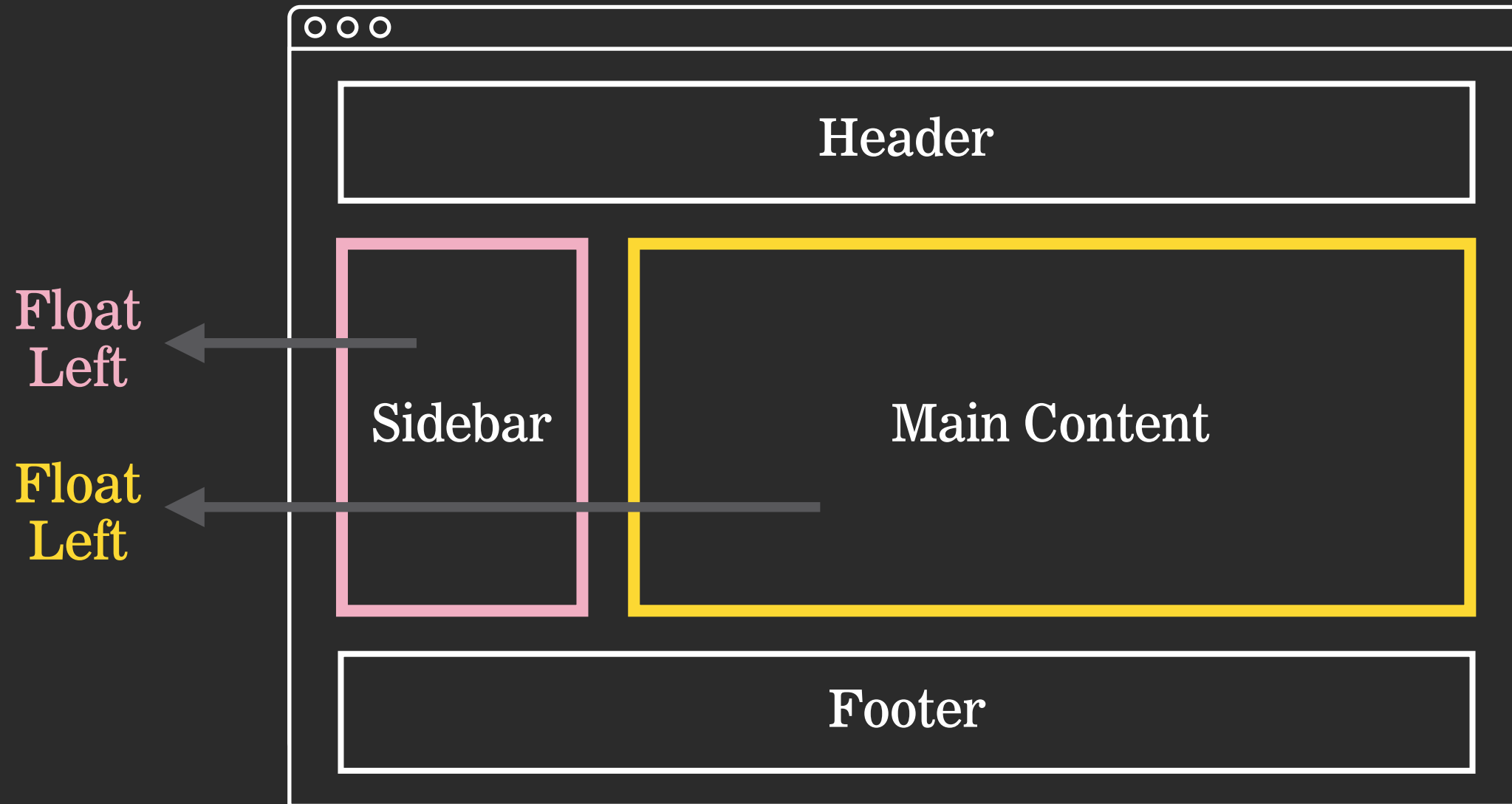
```
.clearfix:after {  
  content: "";  
  display: table;  
  clear: both;  
}
```

```
<div class="clearfix">  
  <p>1</p> <!-- float: left -->  
  <p>2</p> <!-- float: left -->  
  <p>3</p> <!-- float: left -->  
</div>
```

FEWD

MULTI-COLUMN LAYOUT

CSS — MULTI-COLUMN LAYOUT



STEPS TO ACHIEVE A MULTI-COLUMN LAYOUT

1. MAKE SURE EACH COLUMN HAS A WRAPPER AROUND IT IN YOUR HTML

2. ADD BOX-SIZING: BORDER-BOX; TO EVERYTHING (USE THE * CSS SELECTOR)

3. GIVE A WIDTH TO EACH COLUMN (PREFERABLY IN %)

4. FLOAT EACH COLUMN TO LEFT

5. USE PADDING TO ADD SPACE BETWEEN COLUMNS

**6. CLEAR ANYTHING UNDERNEATH YOUR COLUMNS I.E. A FOOTER
USING THE CSS CLEAR PROPERTY (CLEAR: BOTH;)**

LET'S TAKE A CLOSER LOOK

Header

Main Content

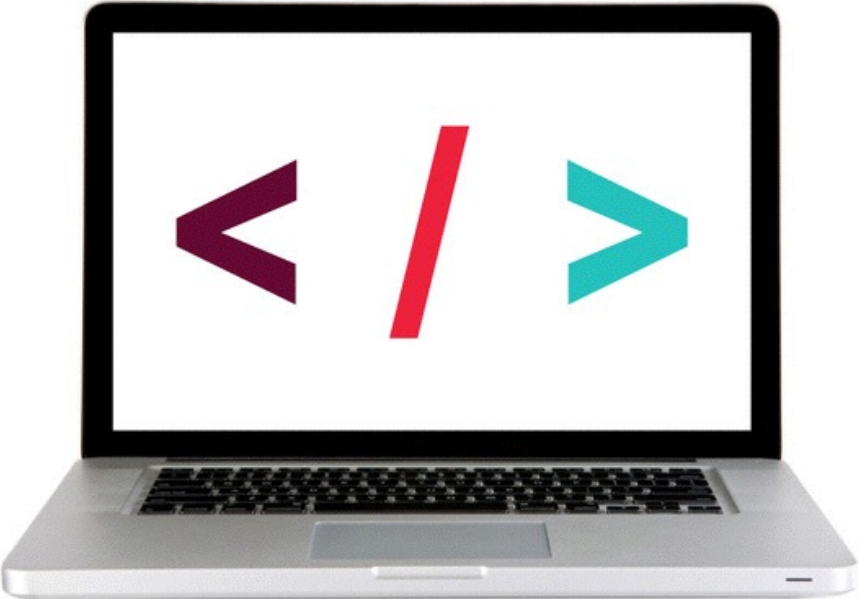
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nam dicta explicabo illum sint, incidunt aut suscipit recusandae, ut voluptatibus, sunt ex placeat? Dolores adipisci id animi soluta eligendi, voluptatibus, et.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Debitis optio harum consequatur dolorem rem beatae ducimus possimus tempore, sequi provident eaque sed explicabo veritatis libero maiores vel praesentium. Sit, nulla.

Sidebar Content



Footer



ACTIVITY — DISPLAY LAB



EXERCISE

KEY OBJECTIVE

- ▶ Work through steps to create a multi-column layout

LOCATION OF FILES

- ▶ `starter_code_lesson_4` > **floating_sections_lab** folder

TIMING

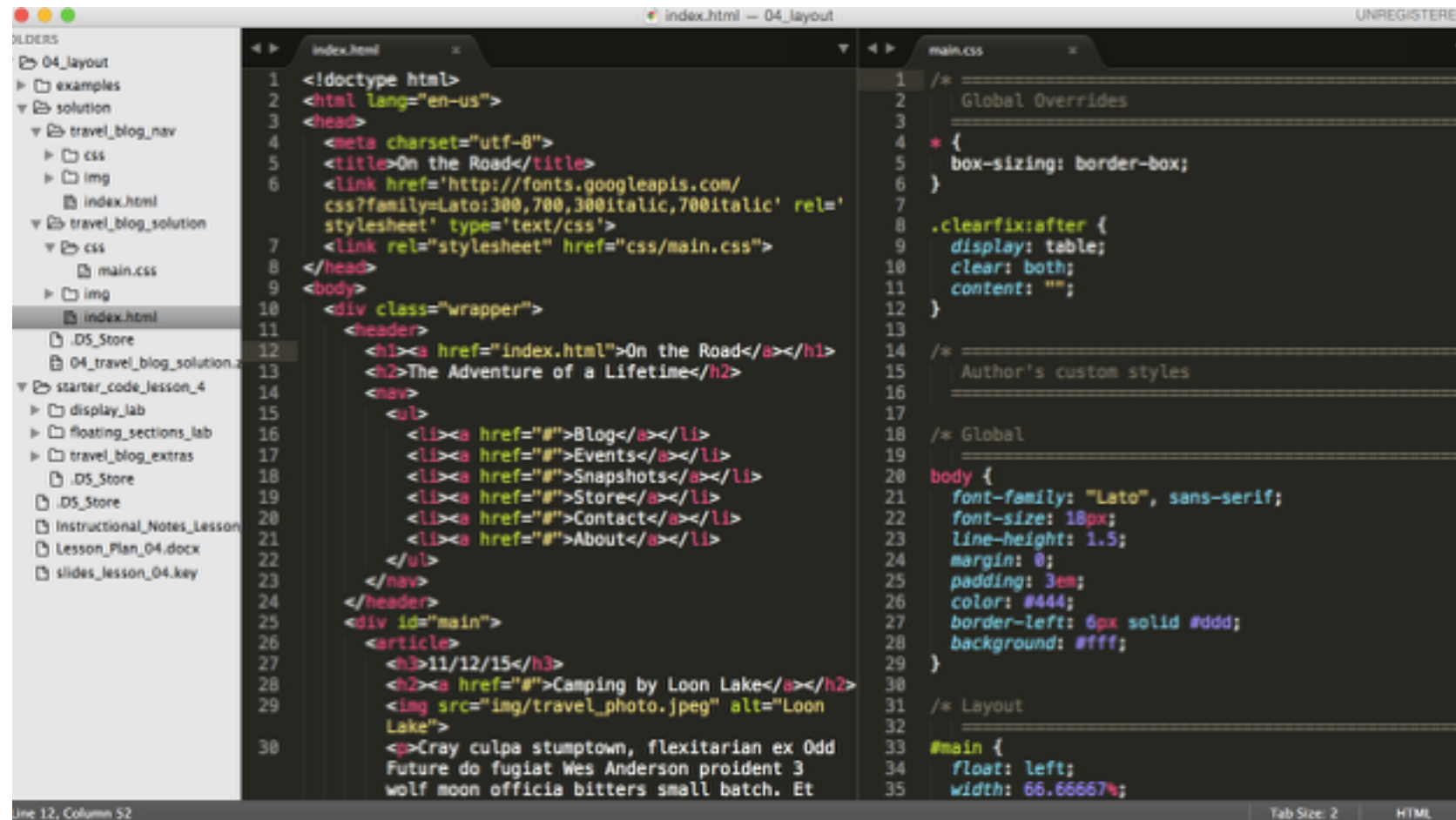
2 min

1. Follow steps 6-9 in `style.css`.

FEWD

TRAVEL BLOG PART 2

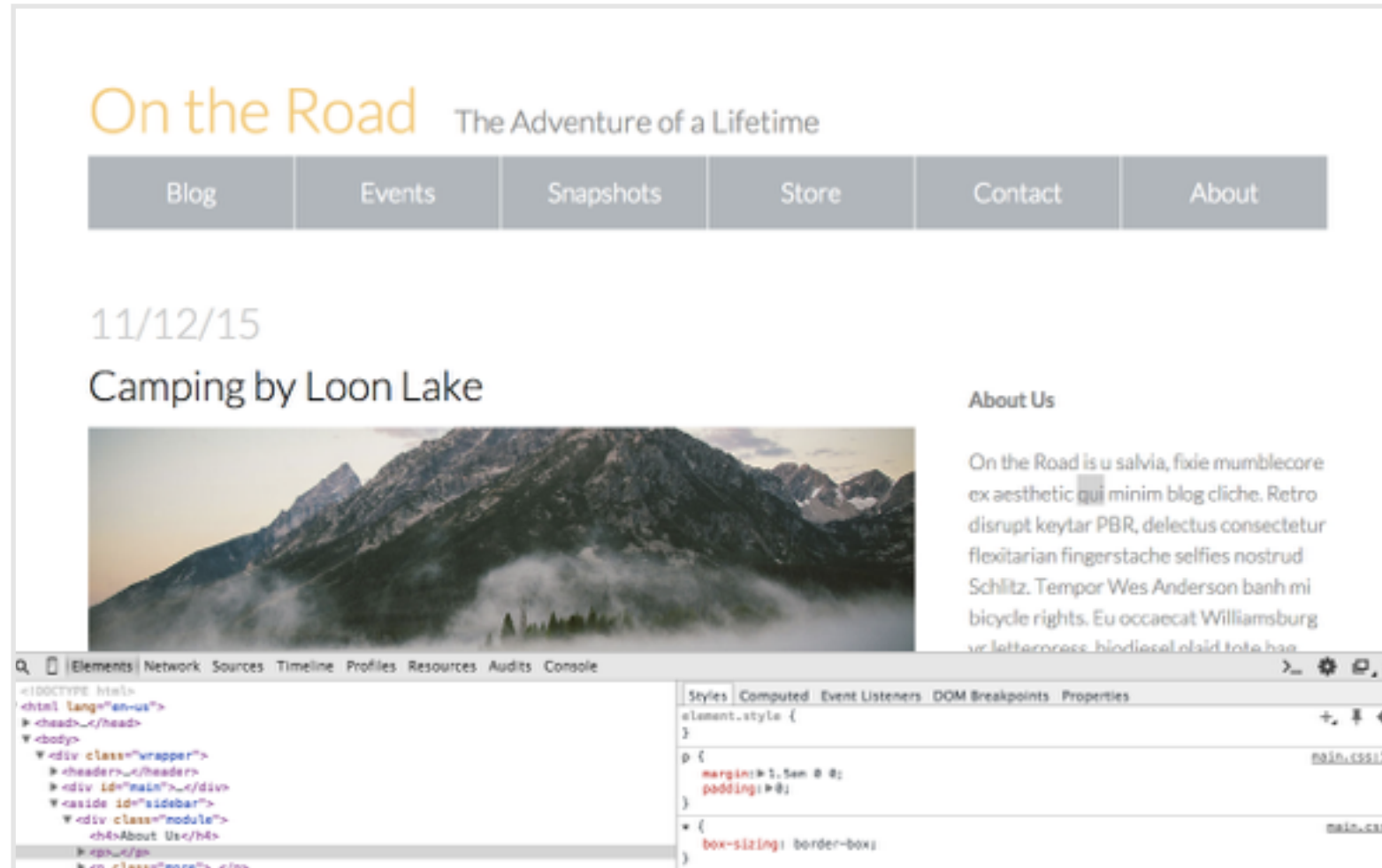
LAB — WORKFLOW



Mac: command + option + 2

Windows: Alt + ↑ + 2

LAB — WORKFLOW



Right click > Inspect Element

LAB — WORKFLOW



FEWD LESSON 4

LAYOUT CSS:

<code>clear</code>	left, right, both, none
<code>float</code>	left, right, none
<code>display</code>	inline, inline-block, block, none

BOX MODEL:



GROUPING TEXT & ELEMENTS:

<code><div></code>	Defines a section in a document (block-level)
<code></code>	Defines a section in a document (block-level)

HTML5 SECTIONING ELEMENTS

<code><header></code>	Defines a header for a section or page
<code><nav></code>	Defines navigation links
<code><article></code>	Defines an article
<code><aside></code>	Defines content away from the page content
<code><footer></code>	Defines a footer for a section or page

CSS:

<code>height</code>	auto, length, %
<code>width</code>	auto, length, %
<code>margin</code>	margin top margin right
<code>margin-top</code>	auto, length, %
<code>margin-right</code>	
<code>margin-bottom</code>	
<code>margin-left</code>	
<code>padding</code>	padding top padding right padding bottom padding left
<code>padding-top</code>	auto, length, %
<code>padding-right</code>	
<code>padding-bottom</code>	
<code>padding-left</code>	

<code>border</code>	border-width border-style border-color
<code>border-top</code>	border-width border-style border-color
<code>border-right</code>	
<code>border-bottom</code>	
<code>border-left</code>	
<code>border-width</code>	width (example: 2px)
<code>border-style</code>	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset
<code>border-color</code>	color

STEPS TO ACHIEVE A MULTI-COLUMN LAYOUT:

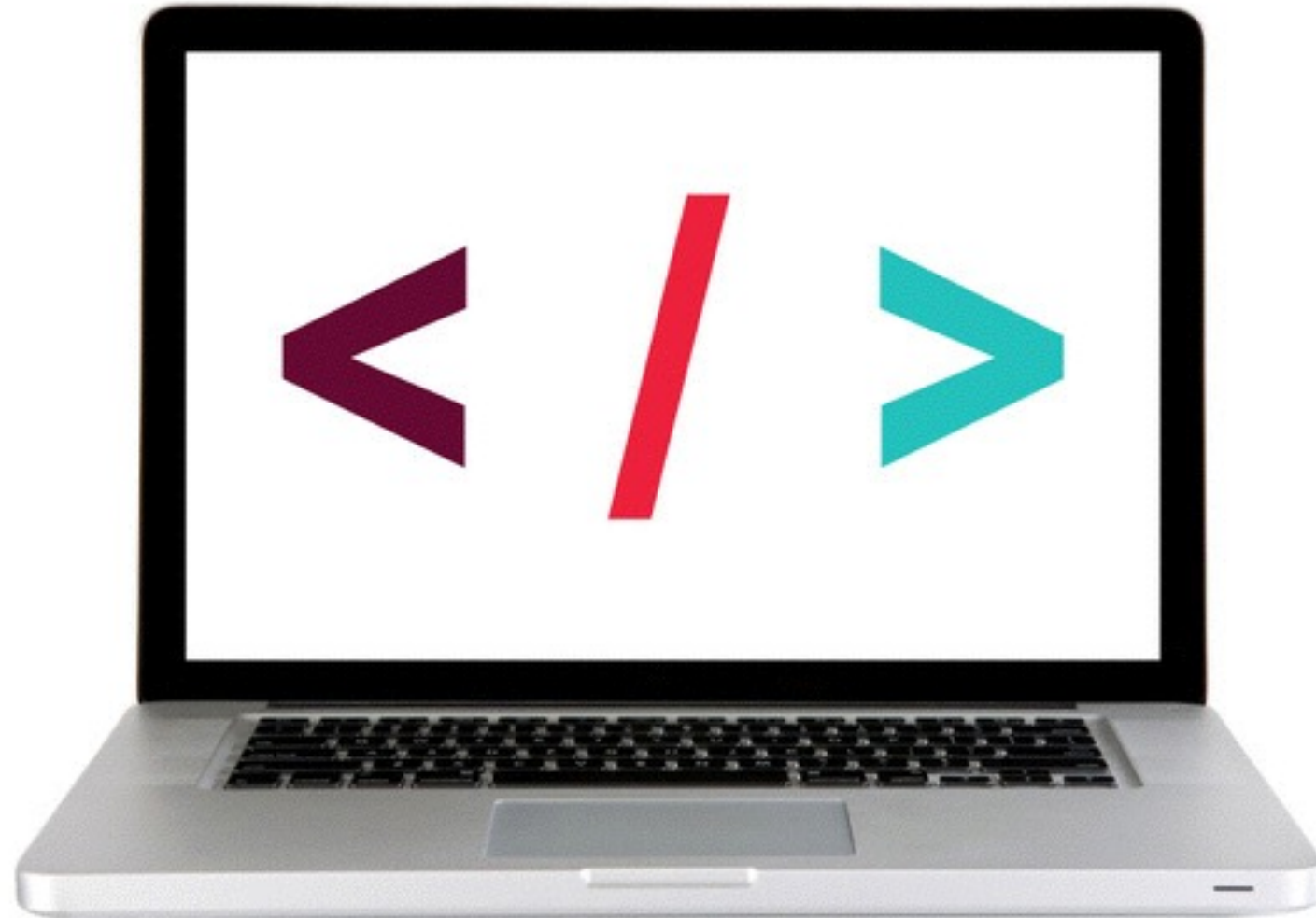
1. Make sure each column has a wrapper around it in your HTML.
2. Give a width to each column (preferably in %).
3. Float each column to left.
4. Use padding to add space between columns.
5. Add box-sizing: border-box; to everything (use the * CSS selector).
6. Clear anything underneath your columns i.e. a footer using the CSS clear property (clear: both;).

LAB — WORKFLOW

Temporarily add a border to everything on the page using the `*` selector so that you can easily see how all the elements on the page line up.

```
* {  
  box-sizing: border-box;  
  border: 1px solid black;  
}
```

LAB — WORKFLOW — EVALUATING A DESIGN



LAB — TRAVEL BLOG PT. 2



ACTIVITY — TRAVEL BLOG



EXERCISE

KEY OBJECTIVE

- Demonstrate the ability to plan and build a website

TYPE OF EXERCISE

- Partner

TIMING

40 min

1. Recreate the Travel Blog site, using Travel_Blog.png as a reference (in starter_code folder)
2. Use HTML structural tags such as <header>, <aside>, <article> and <footer>

LEARNING OBJECTIVES

- Differentiate between block and inline elements
- Identify when HTML5 structural elements should be used
- Apply header, footer, sidebar, and multi-column layouts to build a web page.
- Experiment and predict effects of floats and clearing CSS positioning.

LAYOUT

HOMEWORK

HOMEWORK

HOMEWORK:

Be sure to read the specs on the [FEWD 42 Dashboard](#).

FINAL PROJECTS

- Visit the General Assembly [Gallery](#) and start to get ideas
- Brainstorm and jot down a few ideas

LAYOUT

EXIT TICKETS

<http://goo.gl/forms/vPhCOlfESf>