LET'S GET EVERYTHING SET UP!

1. In Schoology, go to: **Courses(in the top menu) > FEWD CHI 1: Section 1**
2. Then go to the **Class Materials** folder — it's the pink one!
3. Navigate to the **Week 2 (It's the yellow folder) > Lesson 4 folder**
4. There you'll find all the materials for today's class
5. Download starter_code_lesson_4.zip
6. Move it from your Downloads folder to your Desktop
7. Double-click on starter_code_lesson_4.zip to unzip it
8. After you've unzipped, delete the original .zip to avoid confusion and make sure you don't unzip it again later!!!

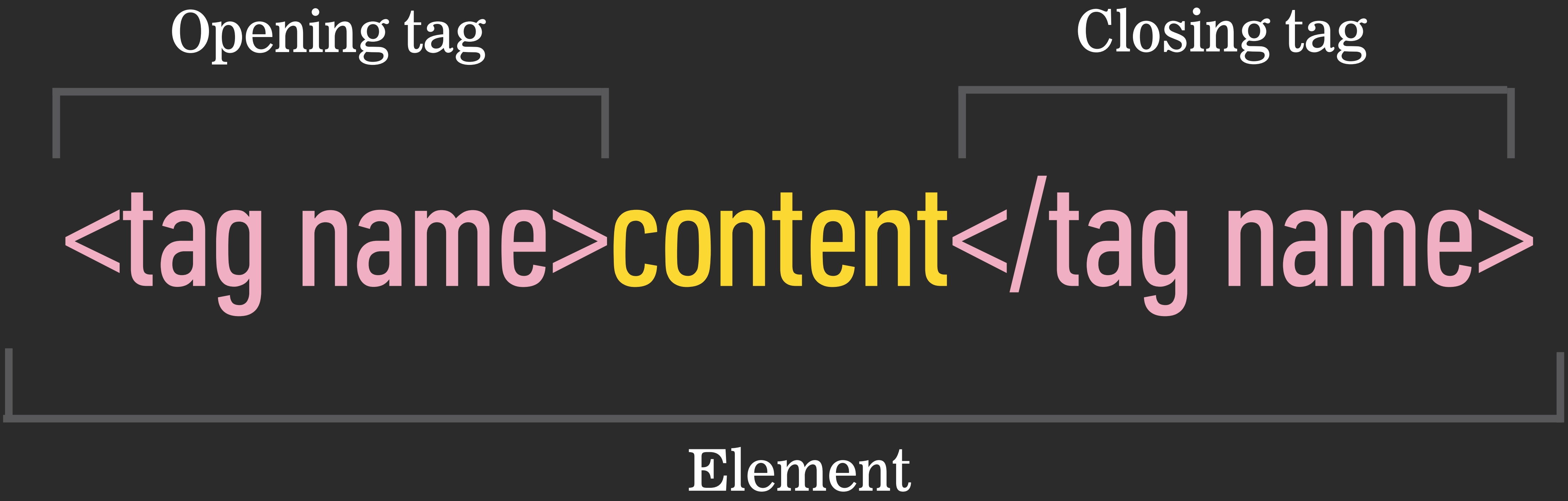
GENERAL ASSEMBLY

LAYOUT

Sarah Holden

REVIEW

HTML SYNTAX — TAGS



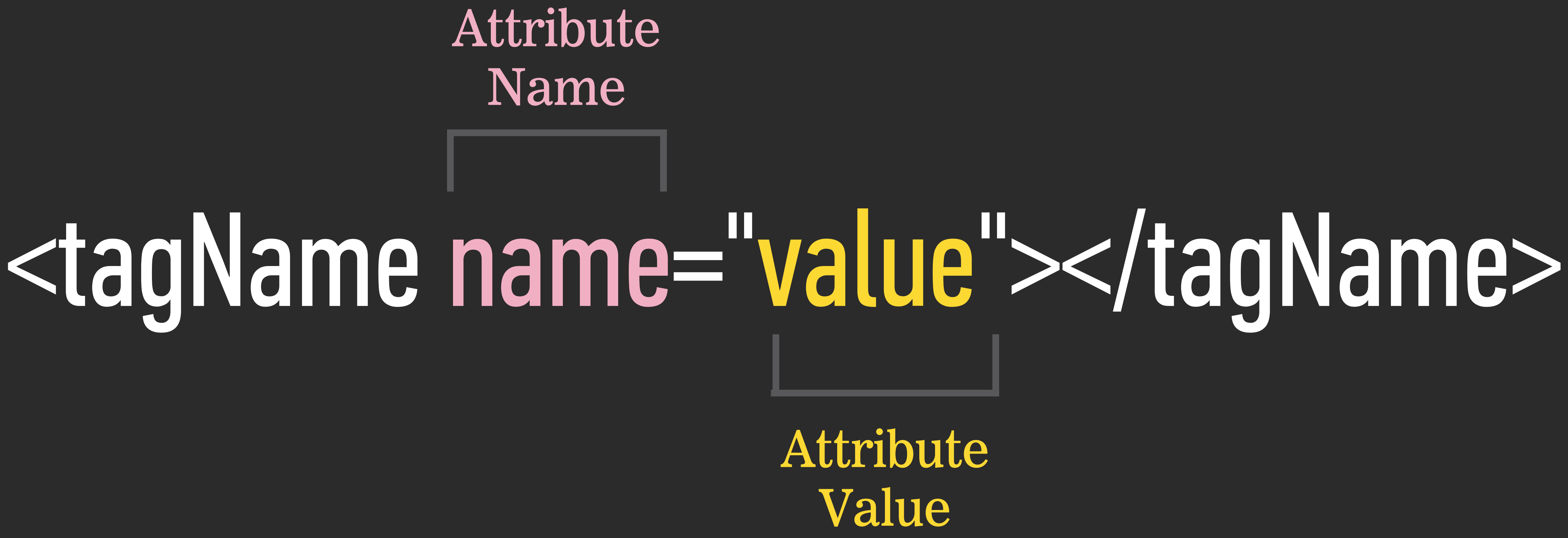
Opening tag

Closing tag

<tag name>content</tag name>

Element

HTML SYNTAX — ATTRIBUTES

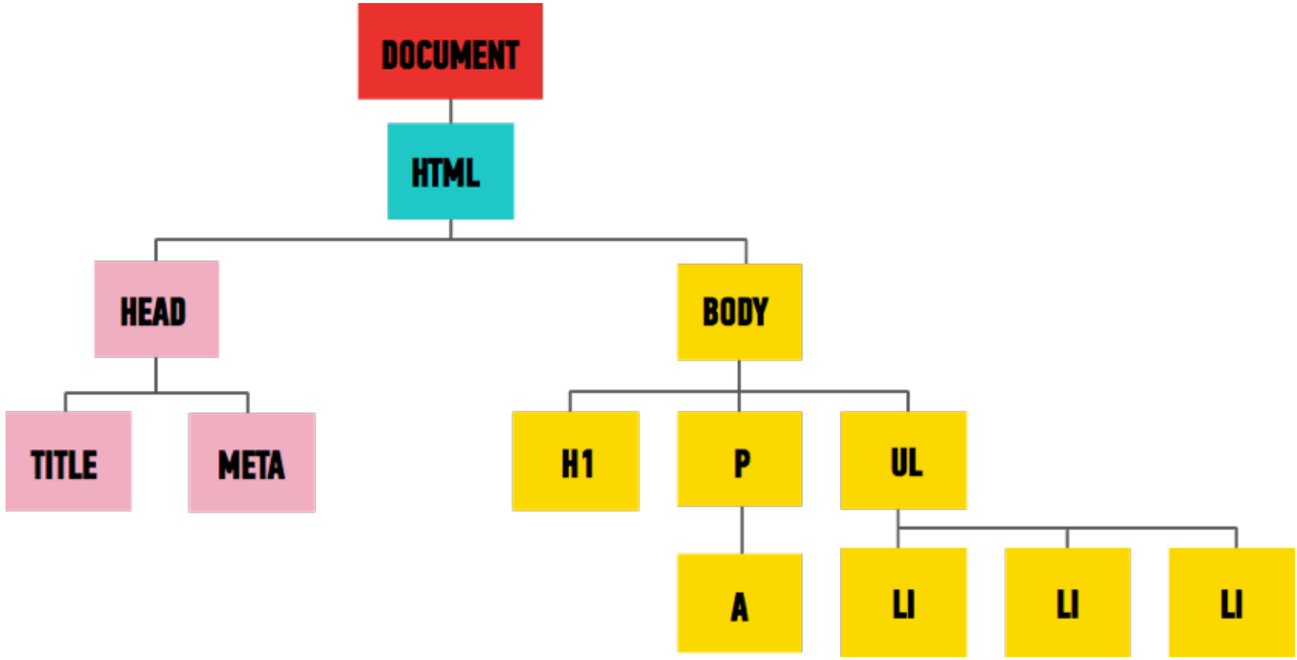


Attribute Name

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

Attribute Value

HTML STRUCTURE



DOCUMENT

HTML

HEAD

BODY

TITLE

META

H1

P

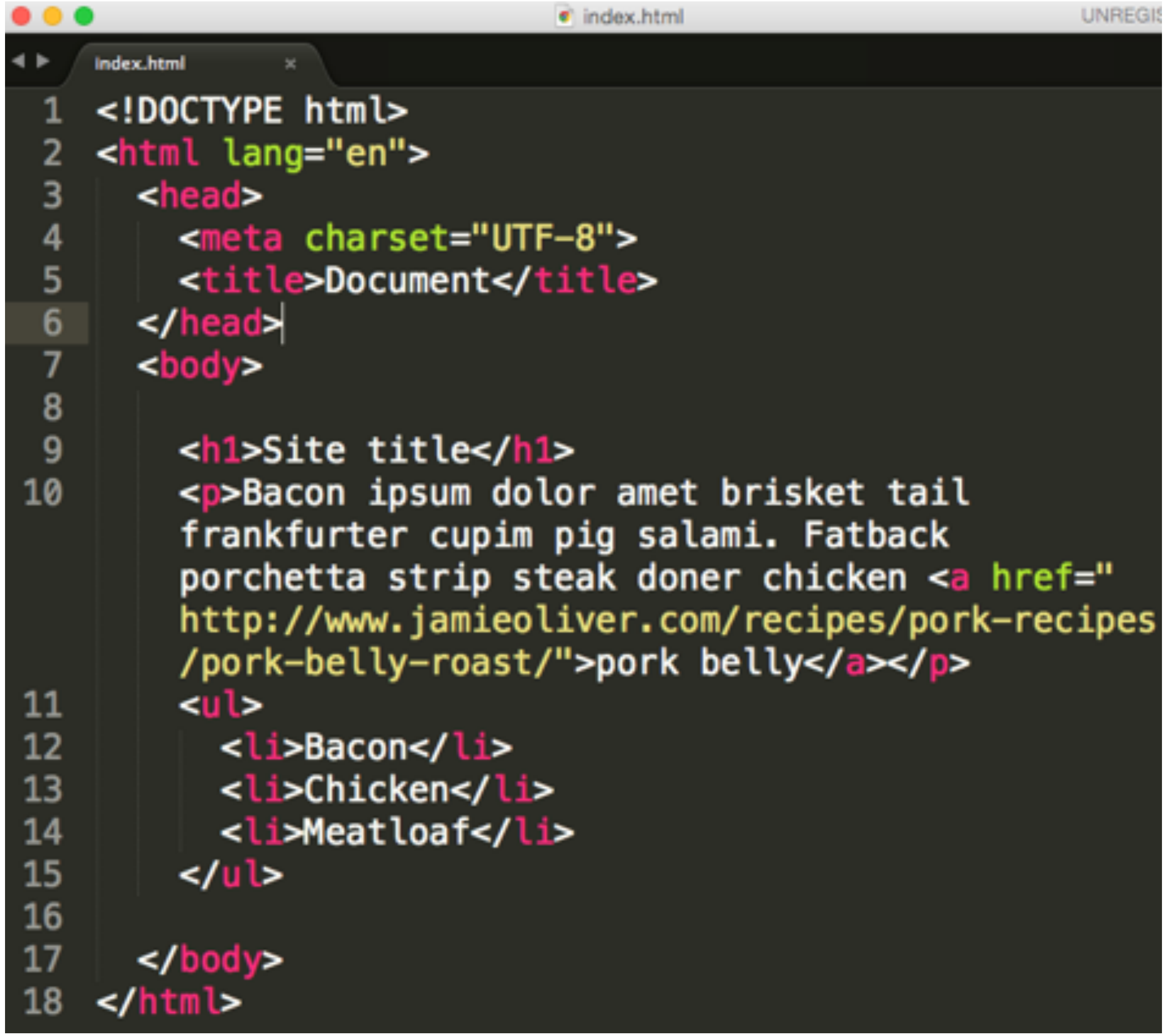
UL

A

LI

LI

LI

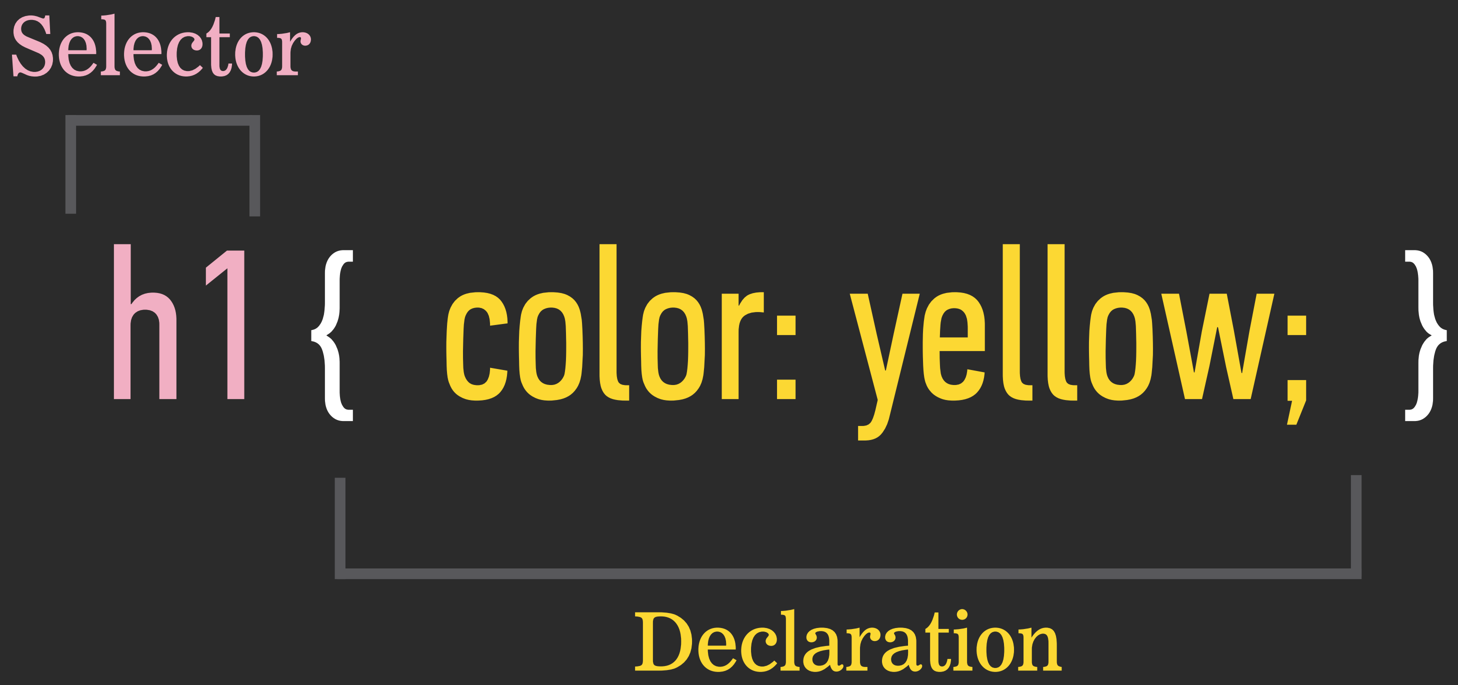


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Document</title>
  </head>
  <body>

    <h1>Site title</h1>
    <p>Bacon ipsum dolor amet brisket tail frankfurter cupim pig salami. Fatback porchetta strip steak doner chicken <a href="http://www.jamieoliver.com/recipes/pork-recipes/pork-belly-roast/">pork belly</a></p>
    <ul>
      <li>Bacon</li>
      <li>Chicken</li>
      <li>Meatloaf</li>
    </ul>

  </body>
</html>
```

CSS SYNTAX



Selector

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

Declaration

CSS SYNTAX

```
h1, h2 {
        color: yellow;
        font-size: 16px;
}
```

Property Value

SELECTORS

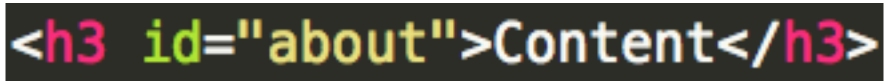
		MEANING:	EXAMPLE:
SELECTOR:	UNIVERSAL	Applies to all elements in the document	* {}
	TYPE	Matches element names	h1, h2, h3 {}
	CHILD	Matches an element that is a direct child of another element	p>a {}
	DESCENDANT	Matches an element that is a descendent (not just a child) of another element	p a {}
	ADJACENT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

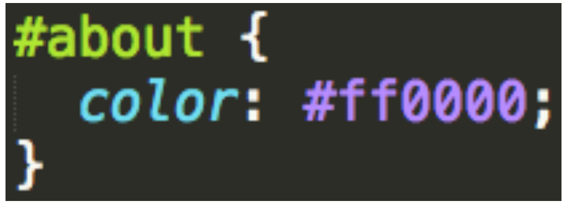
- Classes and ids allow us to assign ‘labels’ to elements so that we can target them in our stylesheets

IDS

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



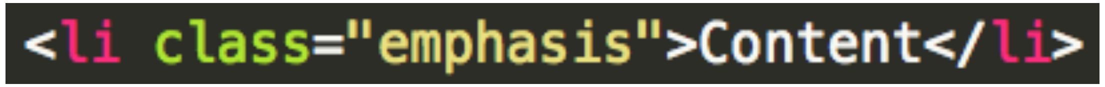
```html
<h3 id="about">Content</h3>
```



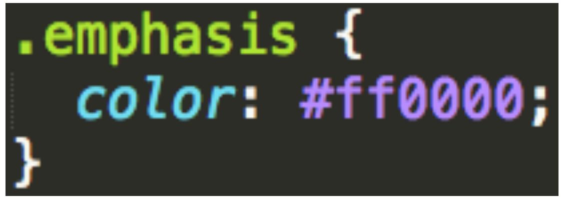
```css
#about {
  color: #ff0000;
}
```

CLASSES

- Classes are used to group elements together

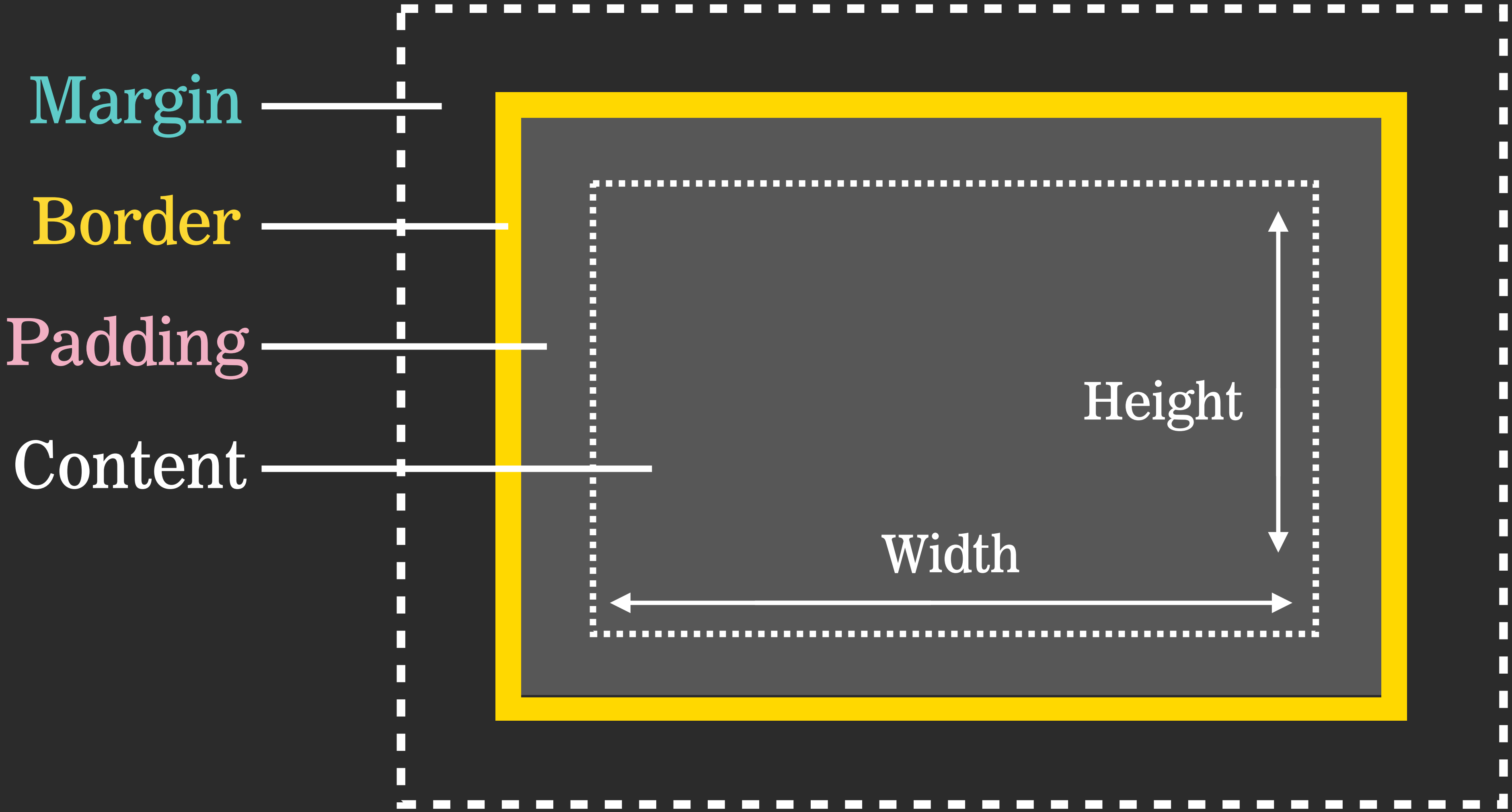


```html
<li class="emphasis">Content</li>
```



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

REFRESHER — BOX MODEL



LEARNING OBJECTIVES

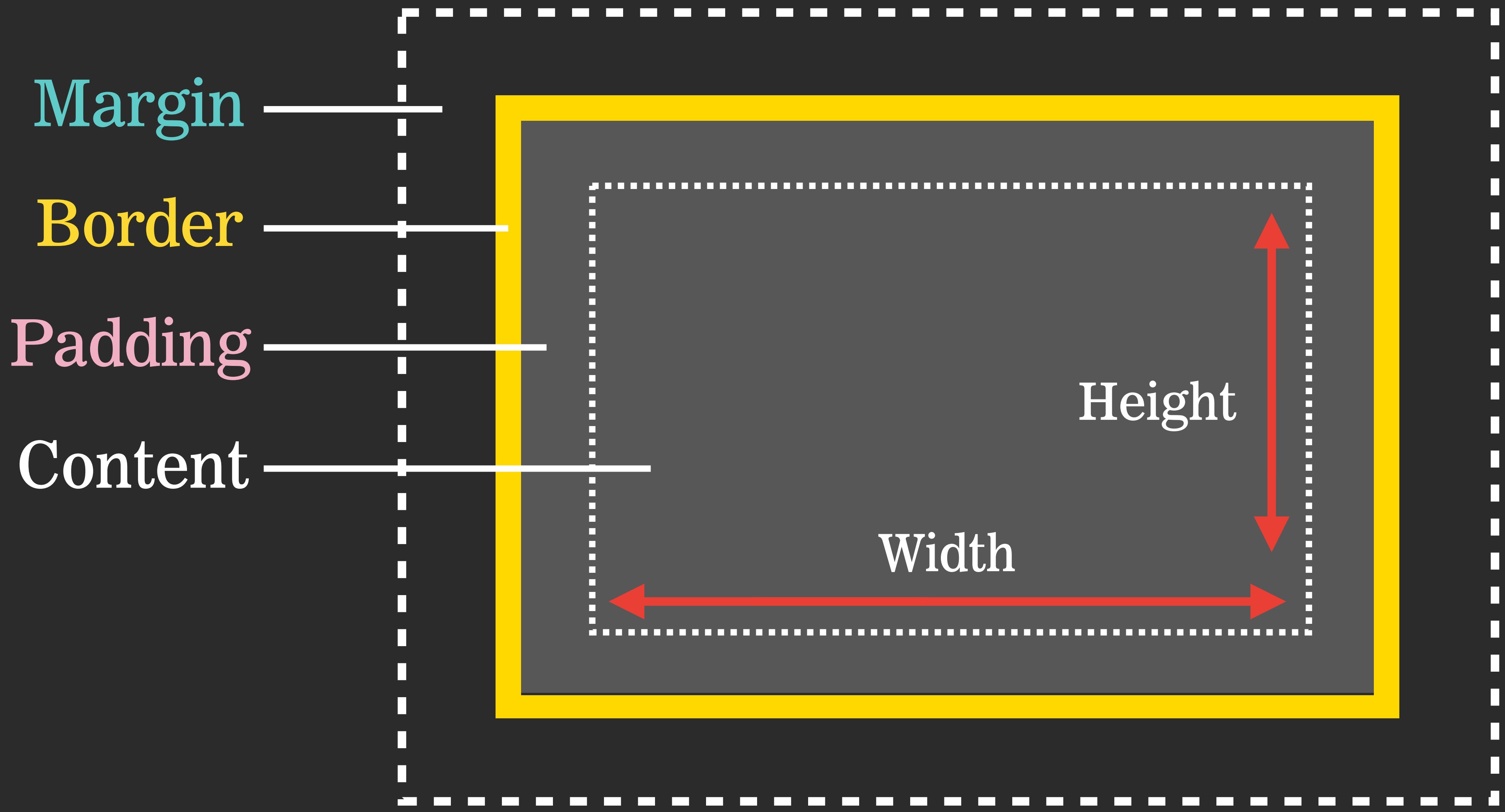
- Identify when HTML5 structural elements should be used
- Apply header, footer, sidebar, and multi-column layouts to develop a web page.
- Experiment and predict effects of floats and clearing CSS positioning.
- Describe the use of Normalize.css

AGENDA

- Divs
- HTML5 Structural Elements
- Floats
- Lab — Travel Blog pt. 2

BOX-SIZING FTW!

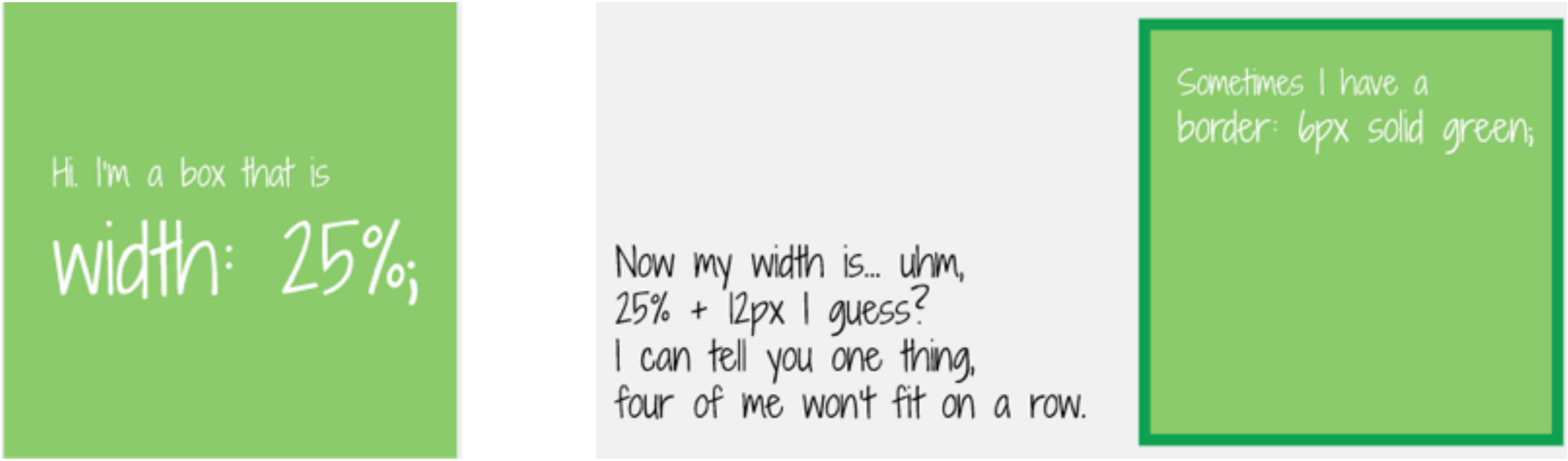
REFRESHER — BOX MODEL



THE DEFAULT WAY — ANNOYING!!

- With the 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



Hi. I'm a box that is

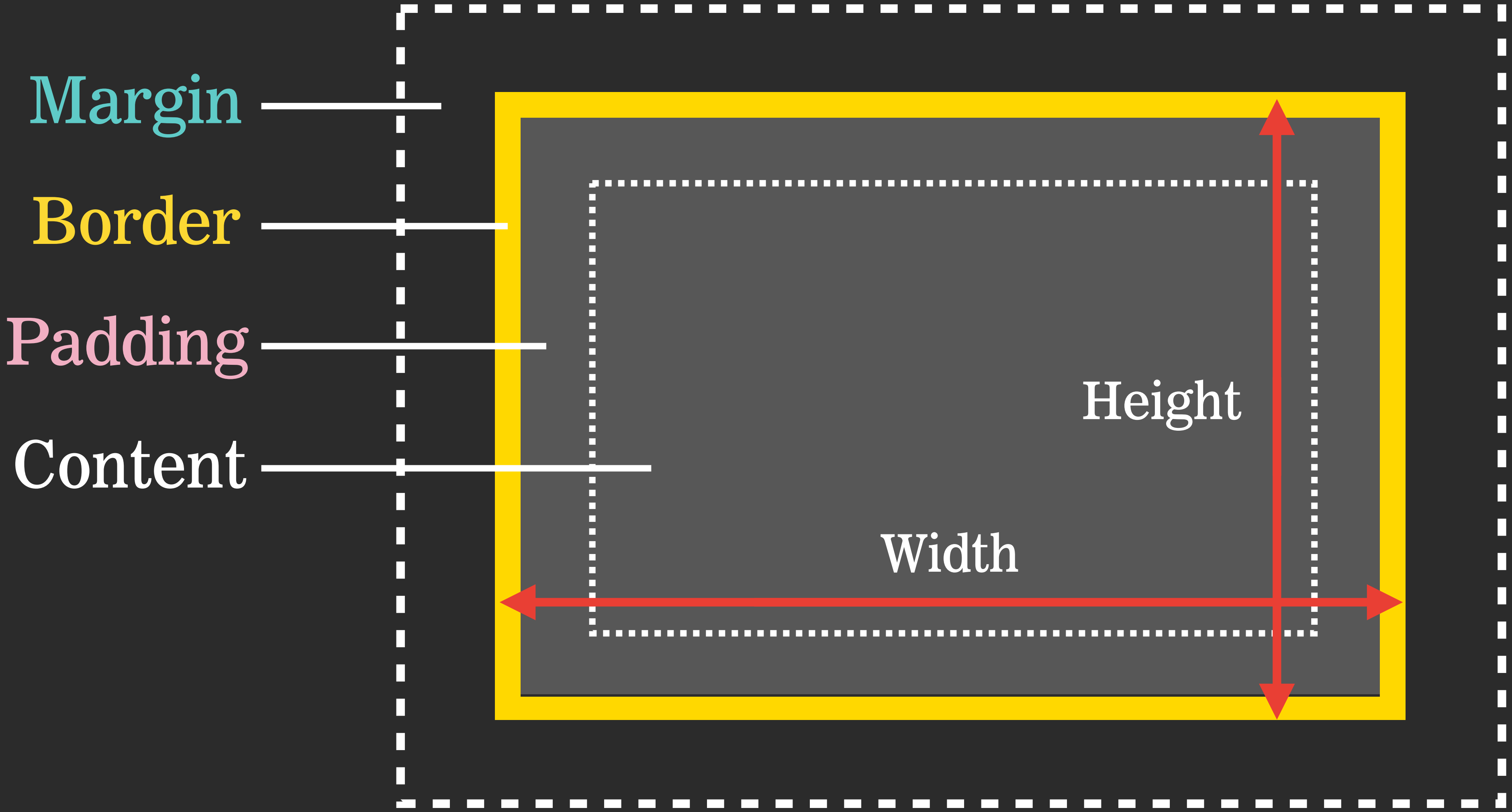
width: 25%;

Now my width is... uhm,
25% + 12px I guess?
I can tell you one thing,
four of me won't fit on a row.

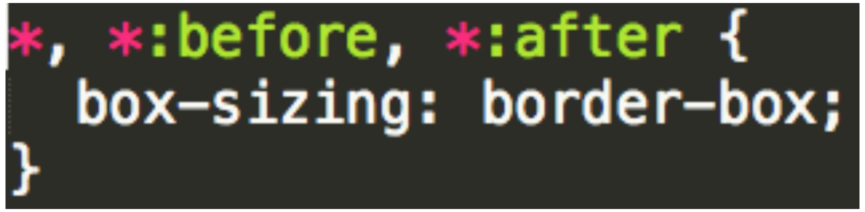
Sometimes I have a
border: 6px solid green;

Image credit: Chris Coyier's [International Box Sizing Awareness Day]

BOX-SIZING: BORDER-BOX



HERE'S THE SYNTAX

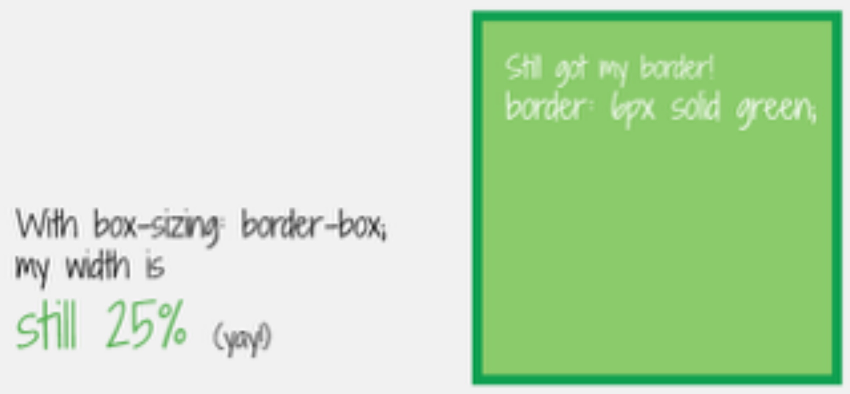


```
*, *:before, *:after {
  box-sizing: border-box;
}
```

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.
- The result is a box the exact width you set it to be in your css.

Actual width = width YAY!



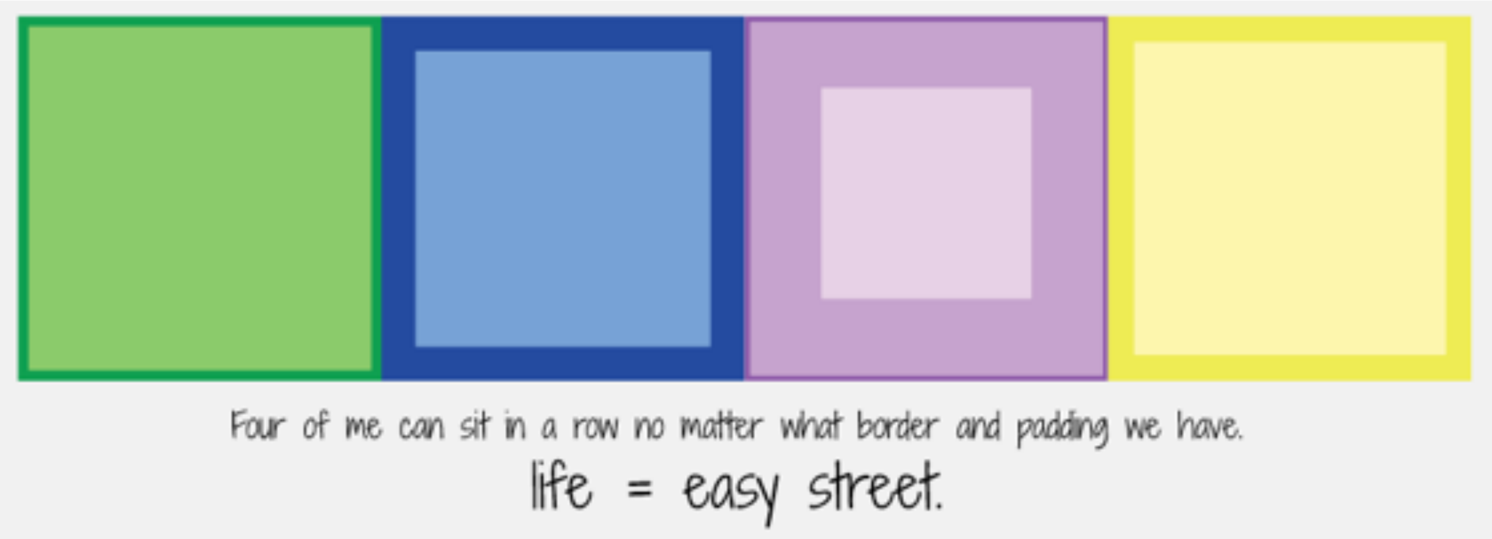
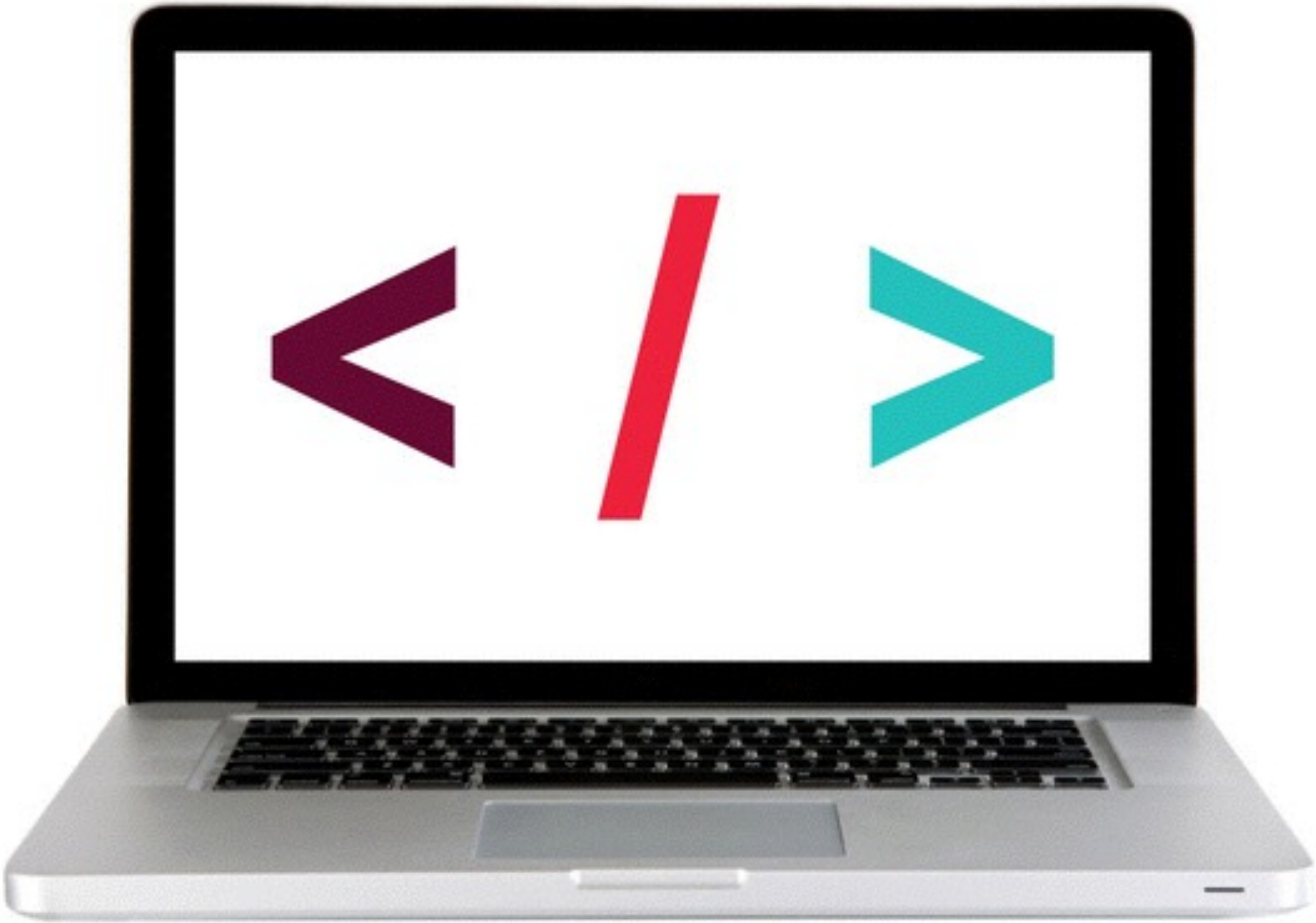

Four of me can sit in a row no matter what border and padding we have.
life = easy street.

Image credit: Chris Coyier's International Box Sizing Awareness Day

LET'S TAKE A CLOSER LOOK



DISPLAY

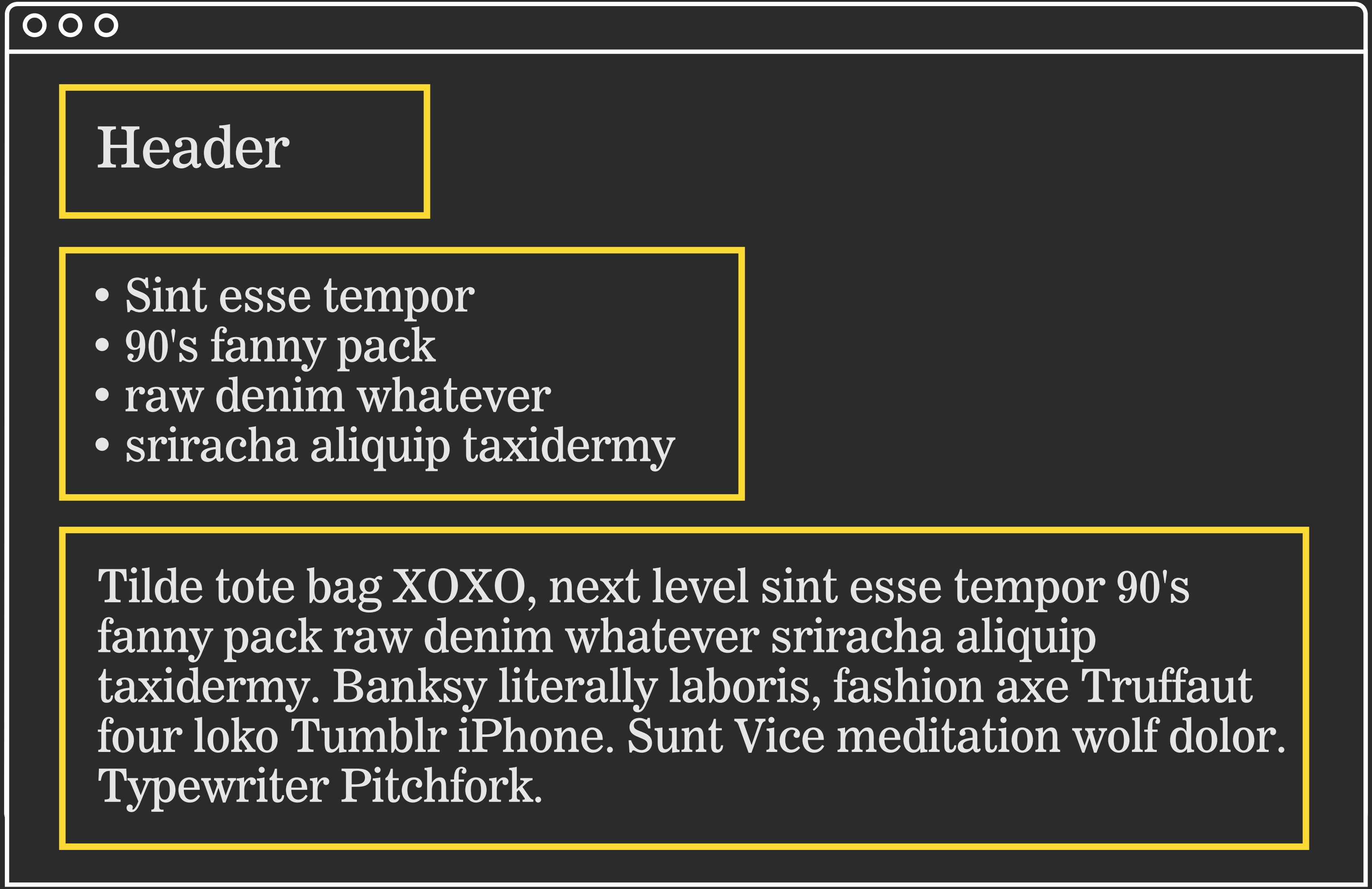
BUILDING BLOCKS

BLOCK-LEVEL ELEMENTS

- Will always start on a new line

Examples:

- <h1>-<h6>
- <ul>
- <li>
- <p>
- <ol>
- <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.

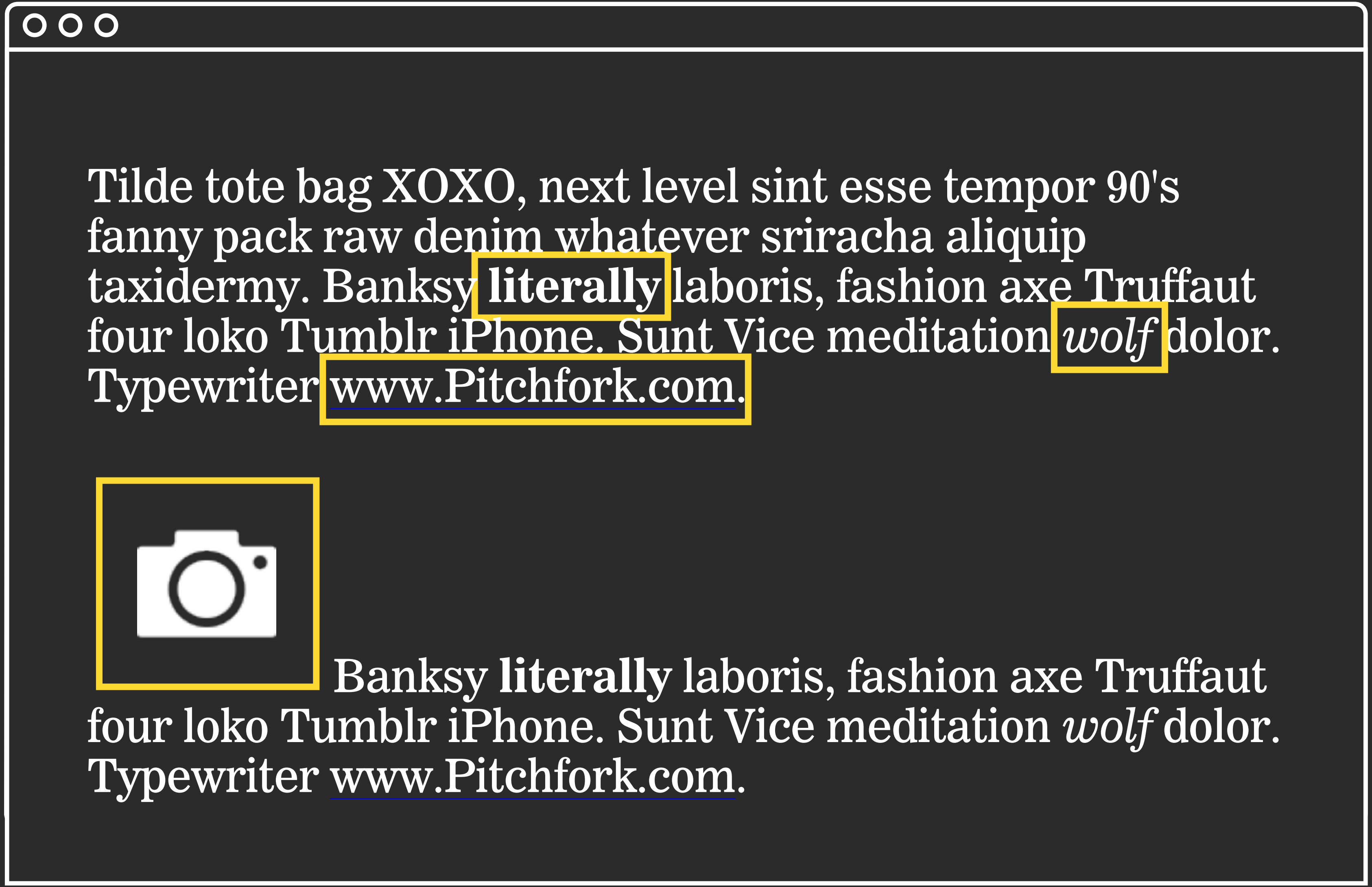
BUILDING BLOCKS

INLINE ELEMENTS

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

Examples:

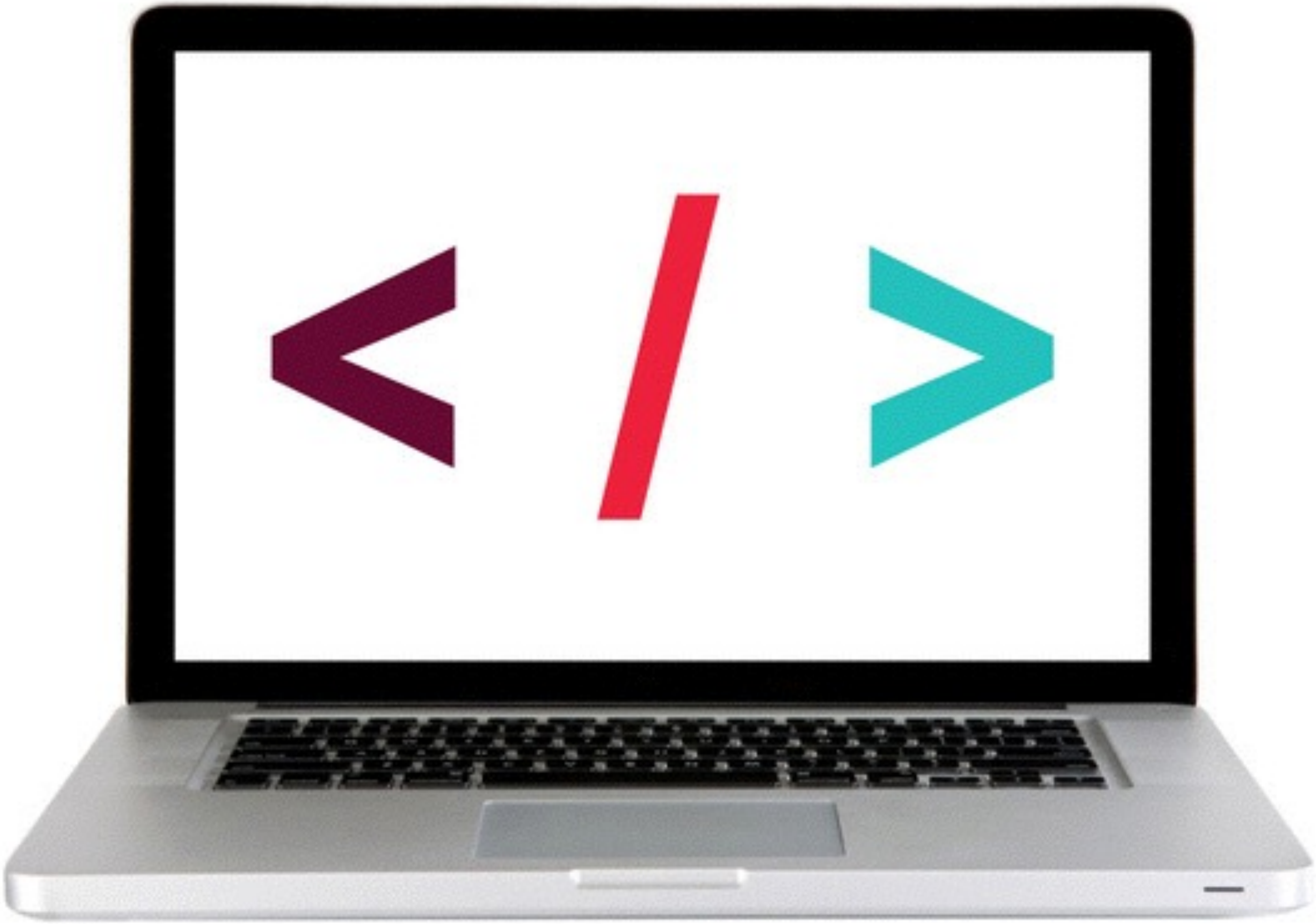
- <a>
- <img>
- <em>
- <strong>
- <q>
- <span>



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.

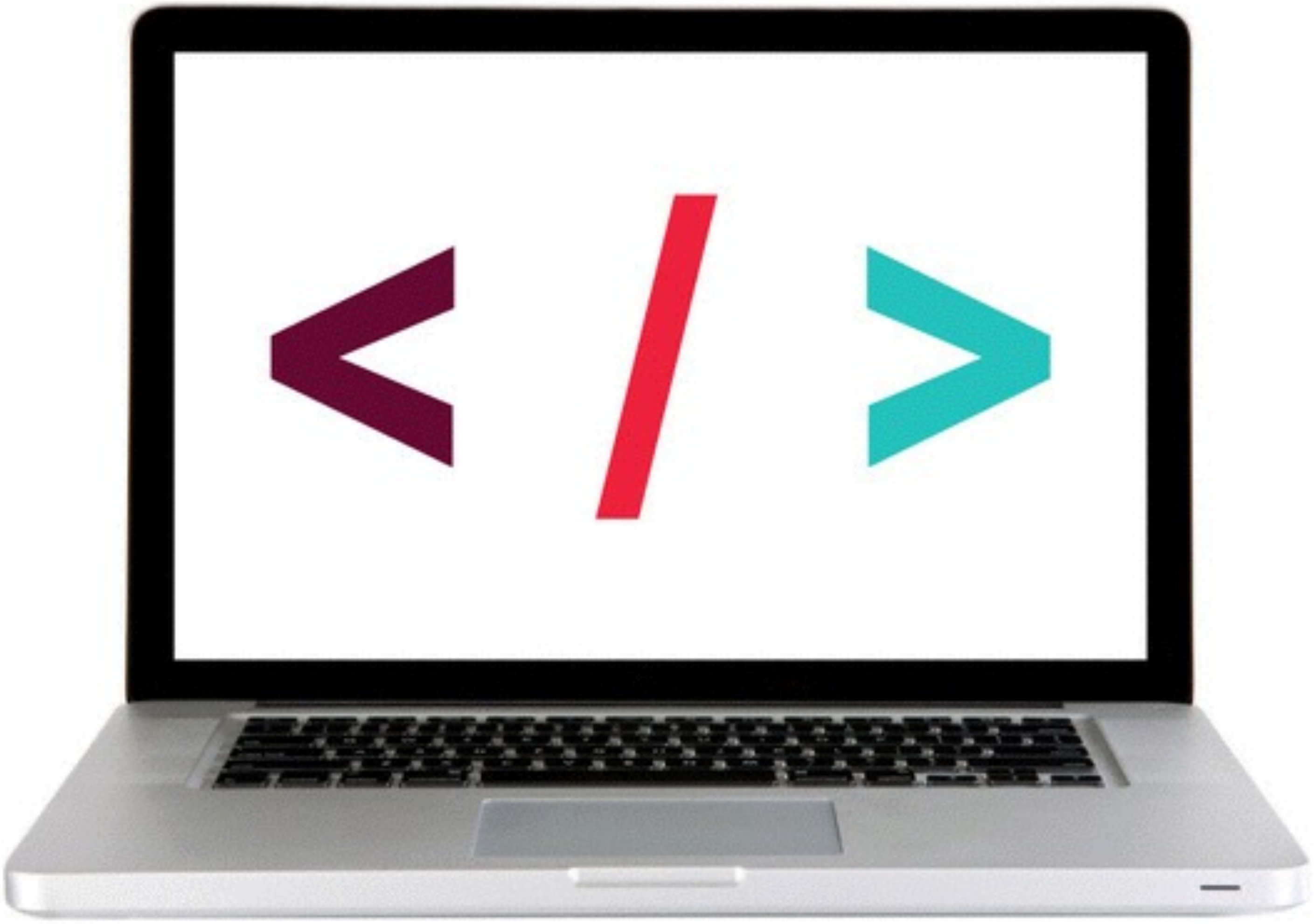
LET'S TAKE A CLOSER LOOK - PART 1



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 and others will not be applied at all. The most noticeable properties are width, height, margin and padding.

LET'S TAKE A CLOSER LOOK - PART 2



DIMENSION – A KEY DIFFERENCE BETWEEN INLINE AND BLOCK ELEMENTS

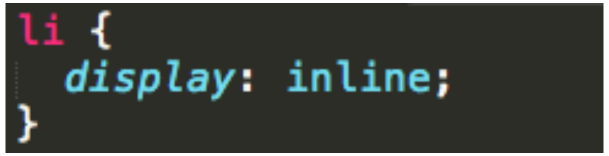
SUMMARY — WHICH DIMENSIONS CAN BE CHANGED?

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

DISPLAY

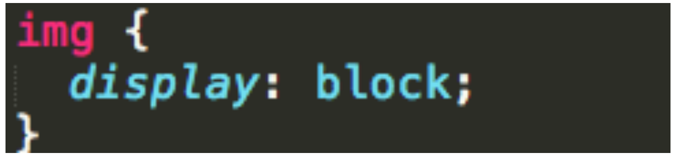
You can affect whether elements are displayed as inline or block elements by using the **display** property.

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



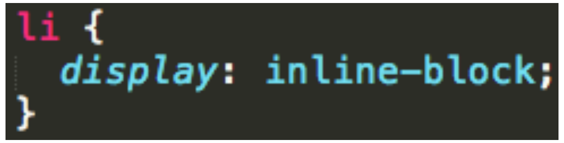
```
li {
  display: inline;
}
```

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



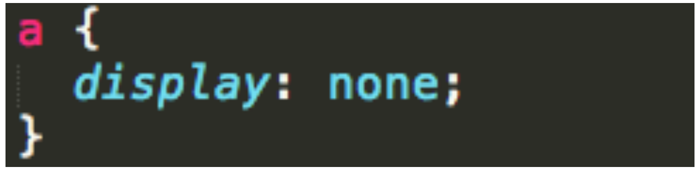
```
img {
  display: block;
}
```

3. Make a block-level element flow like an inline element, while retaining other features of a block-level element (width, height, padding, margin):



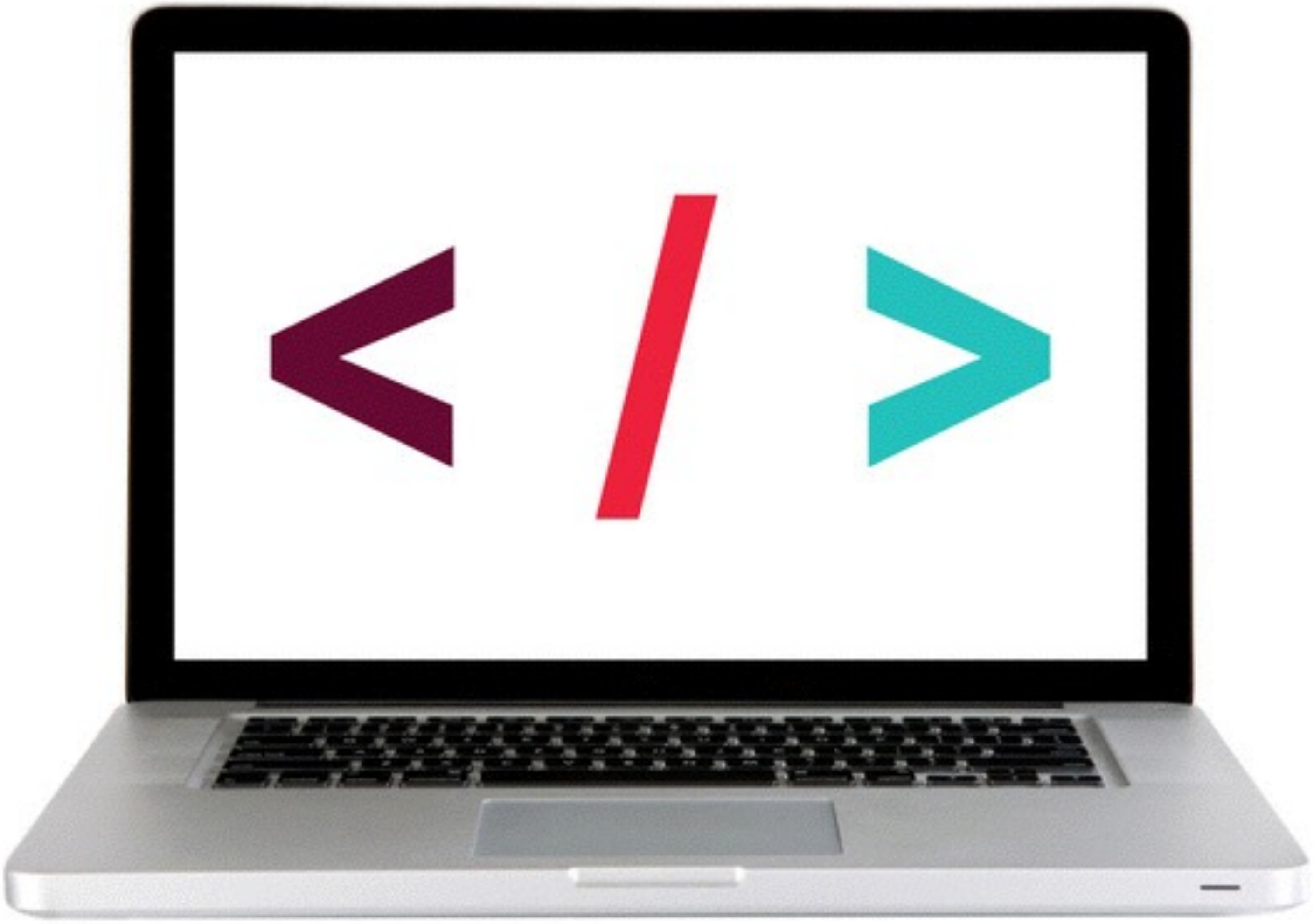
```
li {
  display: inline-block;
}
```

4. Hide an element from a page:



```
a {
  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

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

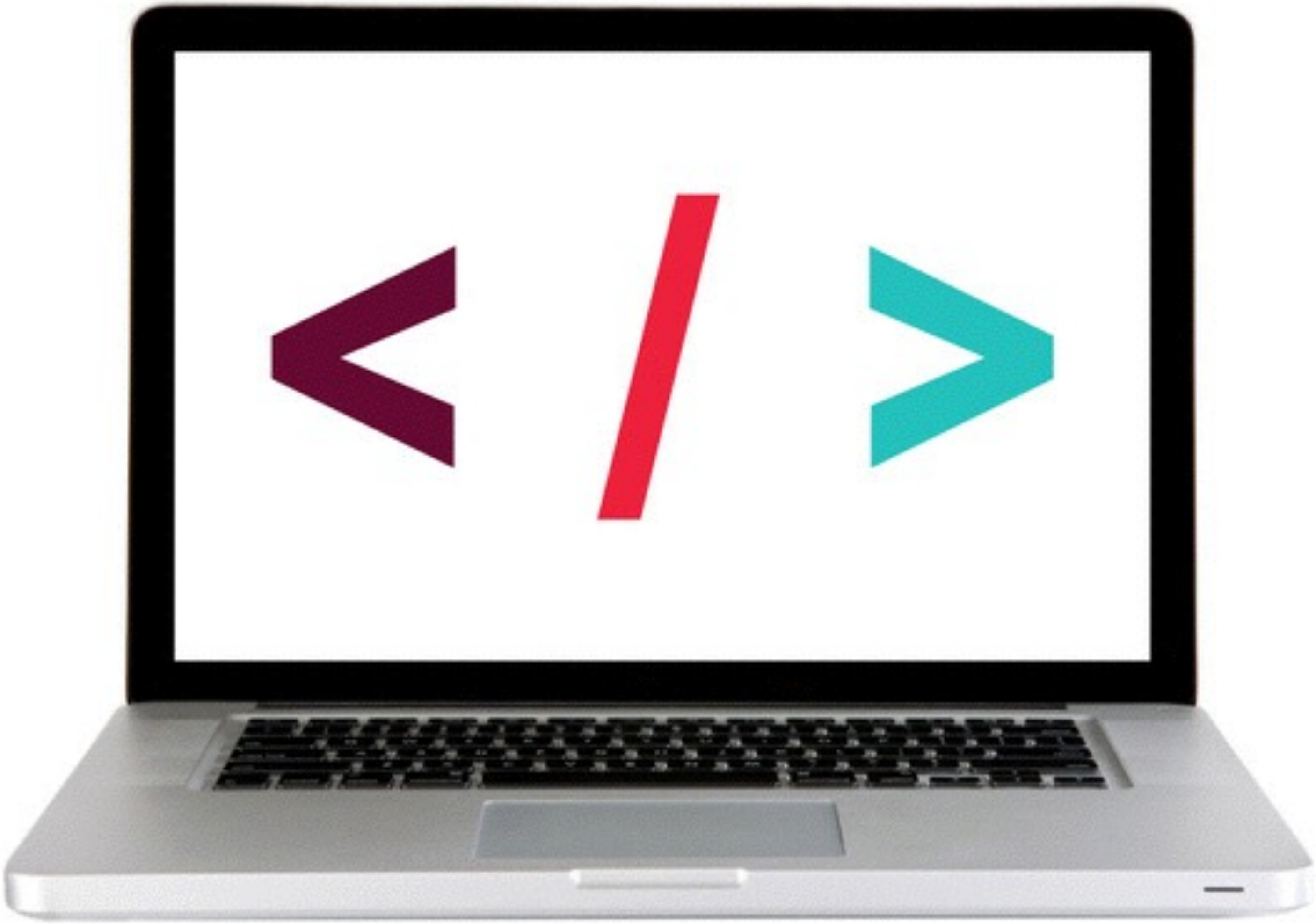
** Divs allow developers to section off parts of a page.

THE <SPAN> ELEMENT

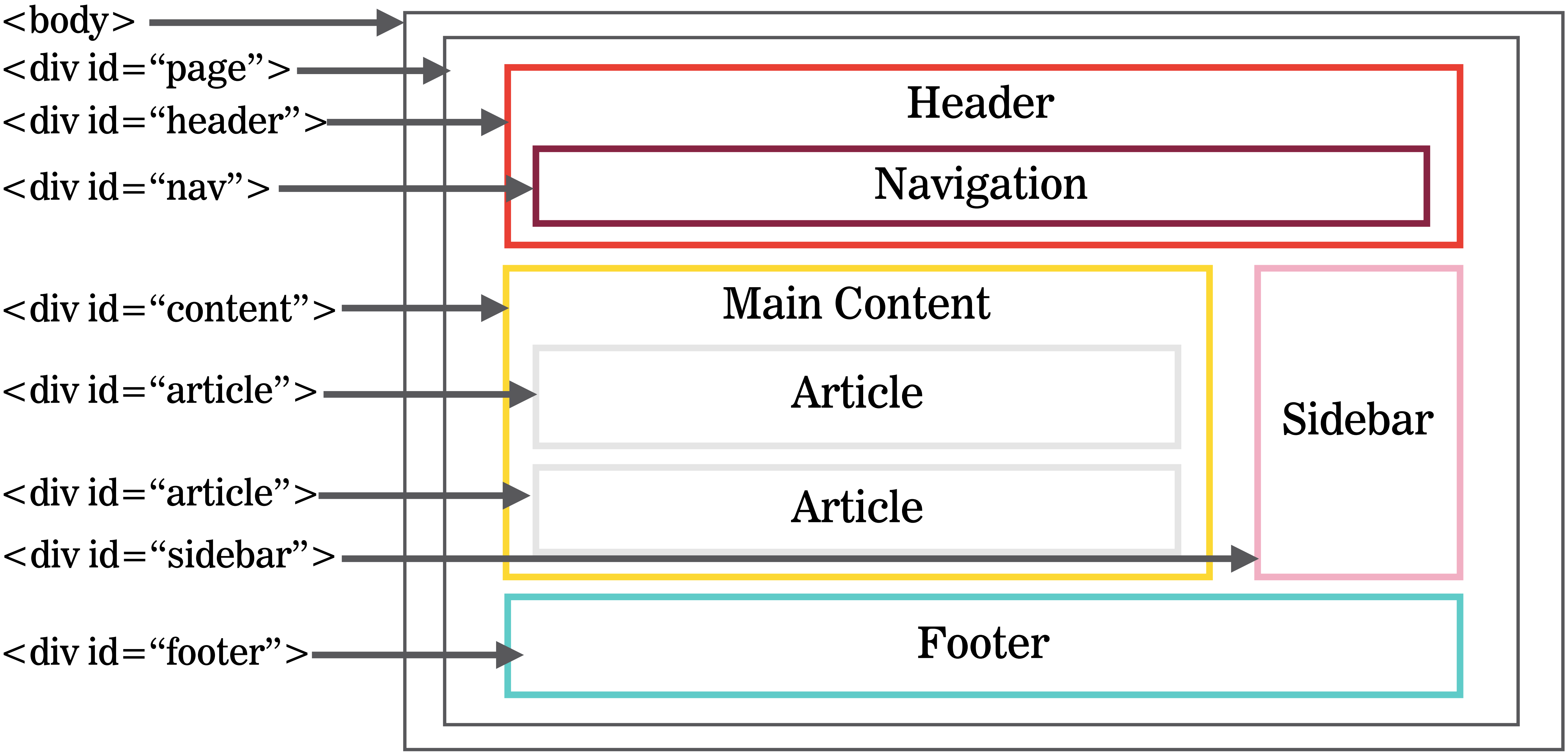
The <span> element acts like the inline equivalent of the <div> element. It is used to either:

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

LET'S TAKE A LOOK



DIVS — SECTIONING OFF PARTS OF A WEBPAGE



<body>

<div id="page">

<div id="header">

Header

<div id="nav">

Navigation

<div id="content">

Main Content

<div id="article">

Article

<div id="article">

Article

<div id="sidebar">

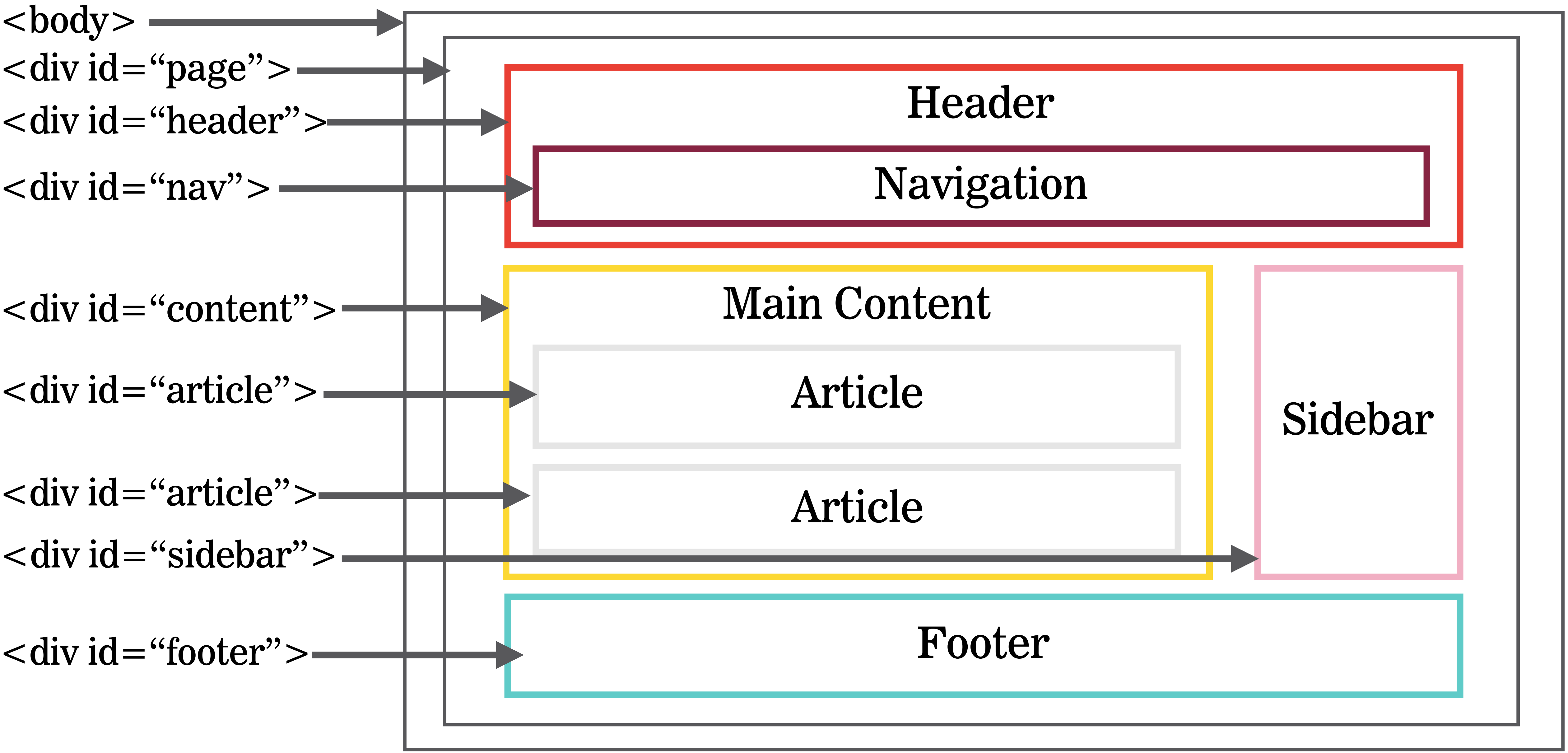
Sidebar

<div id="footer">

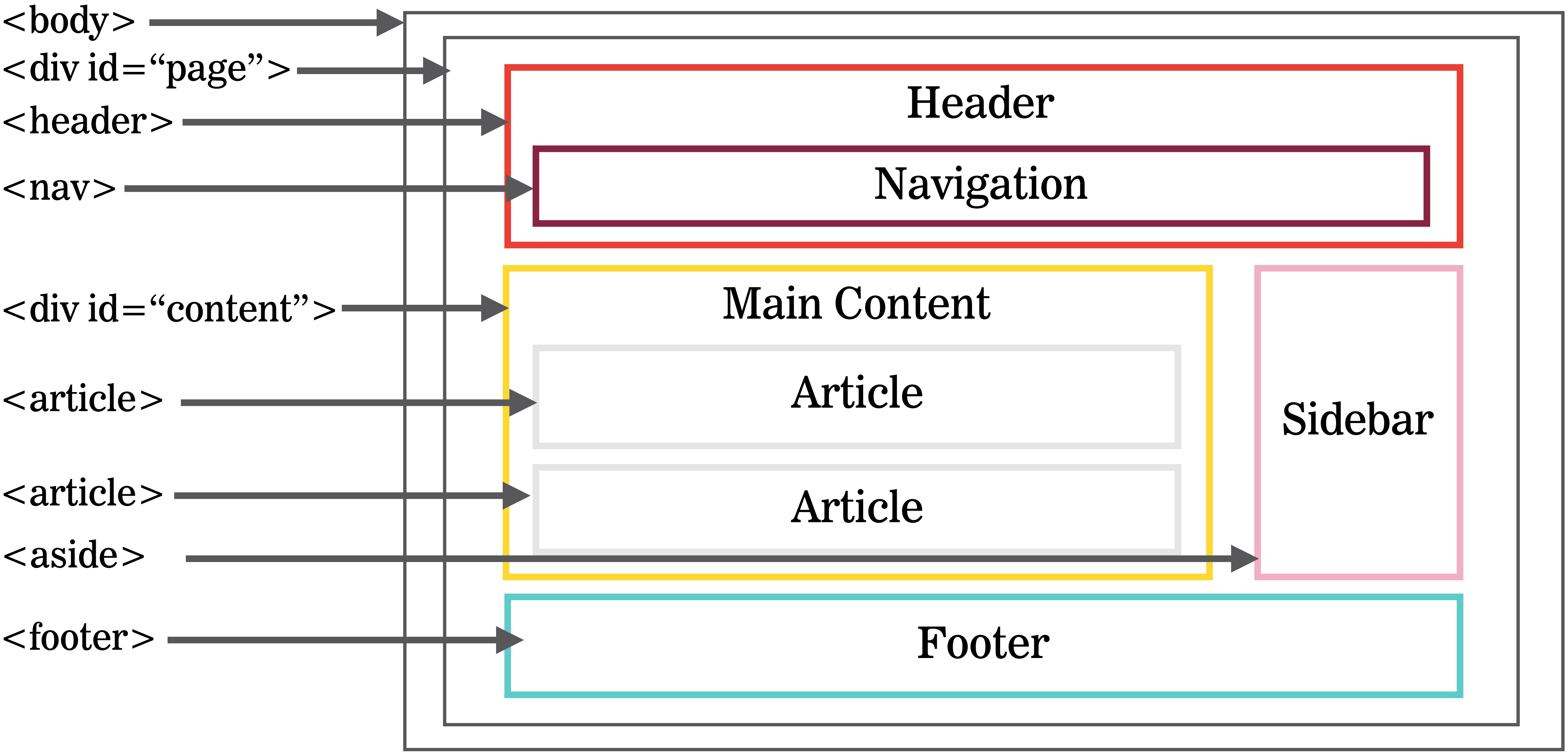
Footer

HTML5 STRUCTURAL ELEMENTS

TRADITIONAL HTML LAYOUTS



STRUCTURAL ELEMENTS



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.

ACTIVITY — 'DIV' UP THE CONTENT

EXERCISE

KEY OBJECTIVE

- Identify content sections

TYPE OF EXERCISE

- Partner

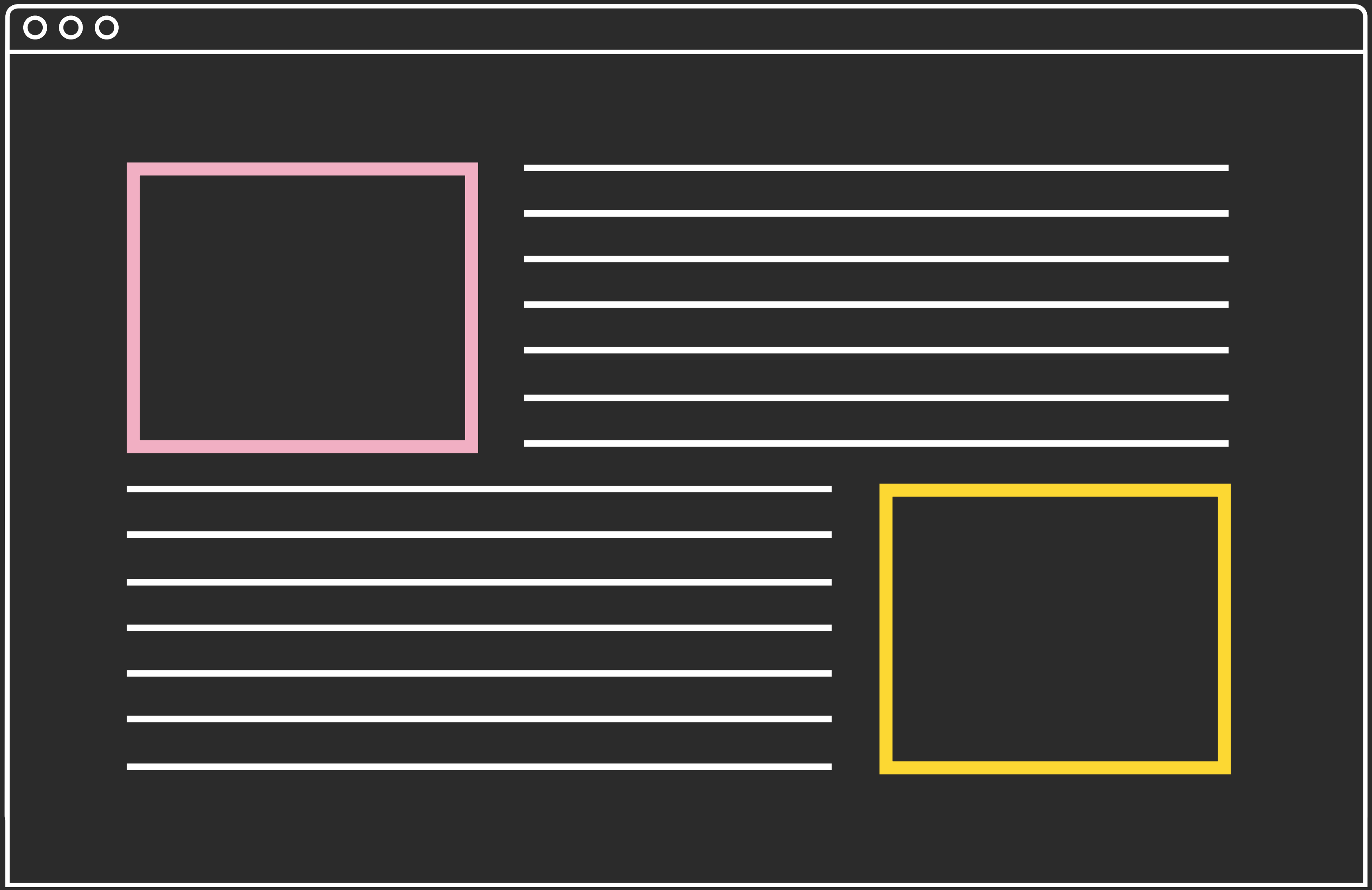
TIMING

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.

FLOATS

CSS — FLOATS

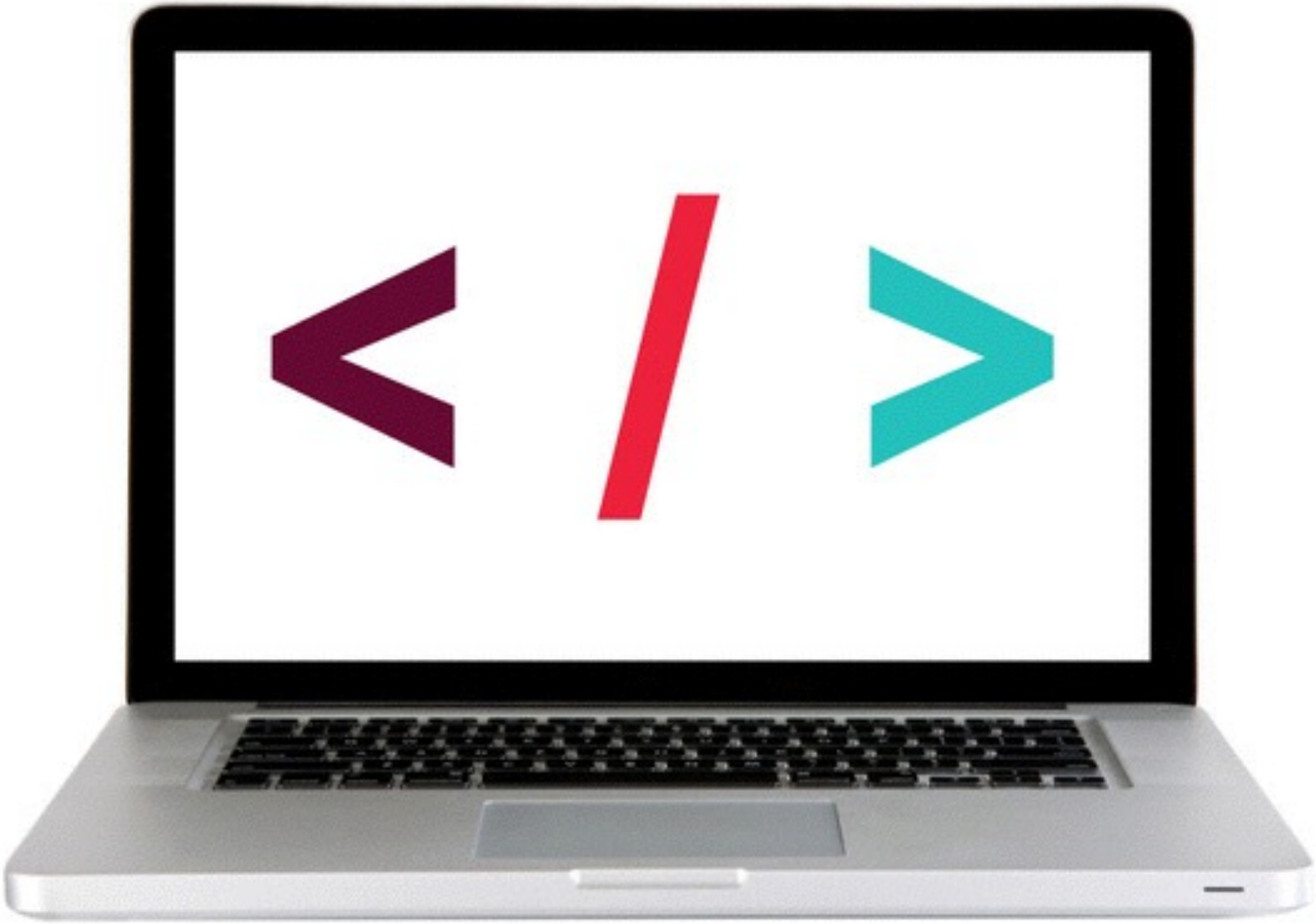


FLOATS

There are four valid values for the float property:

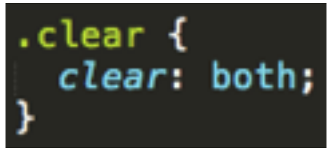
- **Left** and **Right** float elements those directions respectively
- **None** (the default) ensures the element will not float
- **Inherit** will assume the float value from that elements parent element

LET'S TAKE A LOOK



CLEARING FLOATS

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



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

LEFT

- No floating elements allowed on the left side

RIGHT

- No floating elements allowed on the right side

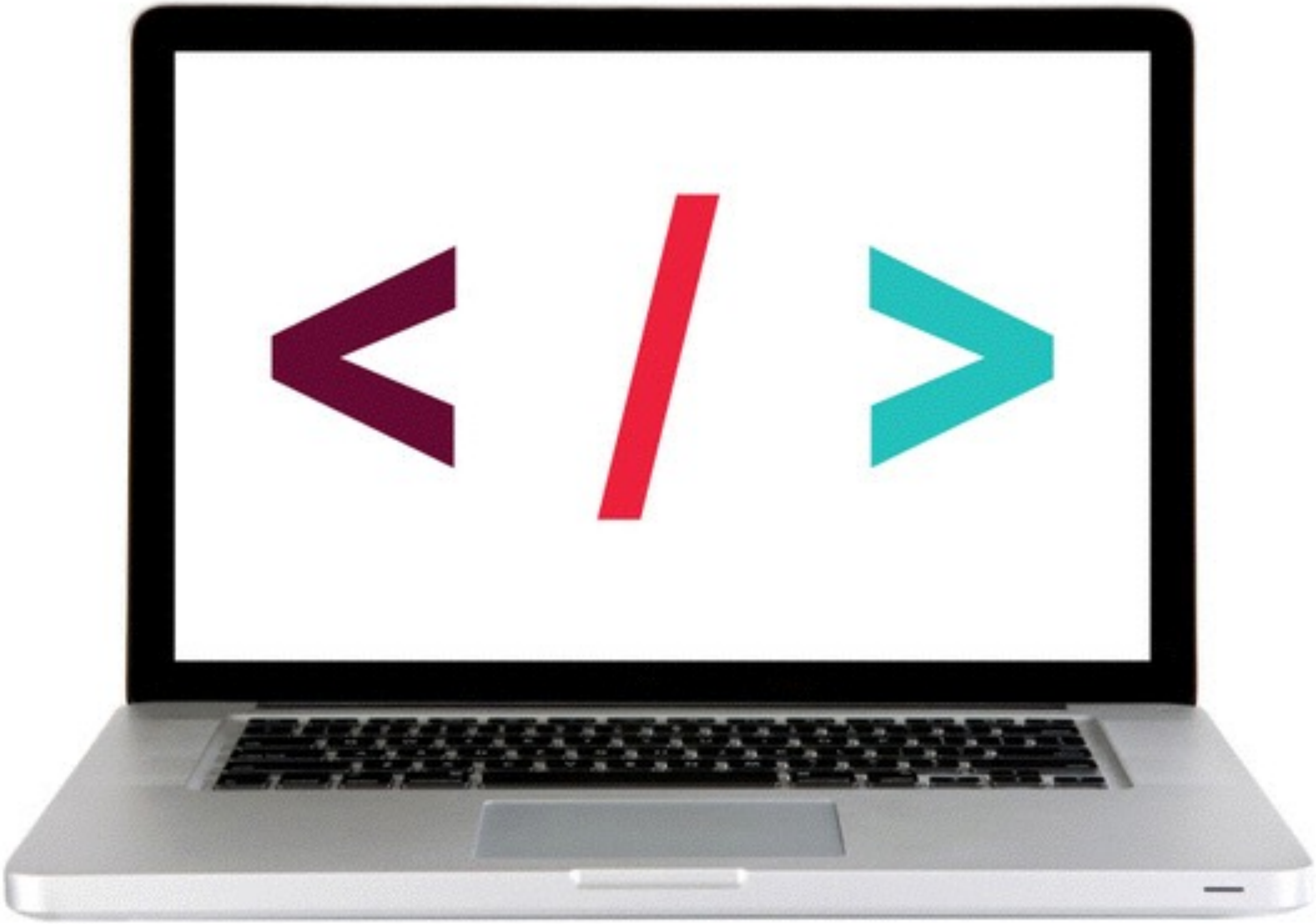
BOTH

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

NONE

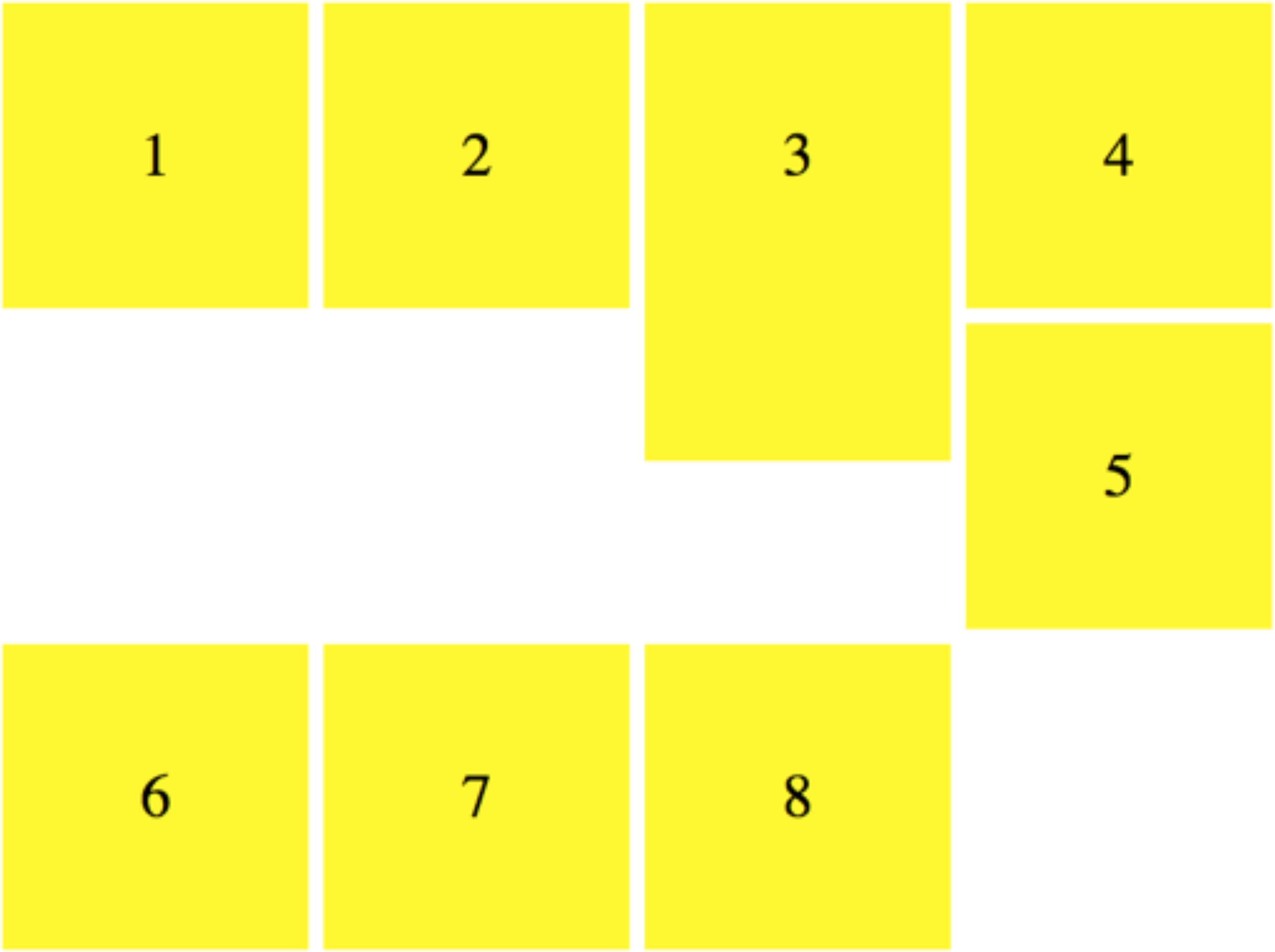
- Allows floating elements on both sides

LET'S TAKE A LOOK



LET'S TAKE A LOOK

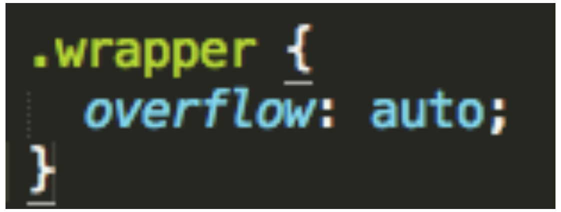
- I've added the example to Codepen so you can refer to it later if needed



PARENTS OF FLOATED ELEMENTS

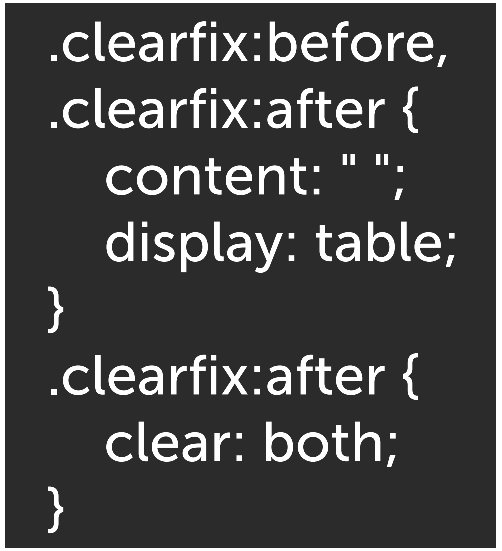
- If a containing element only contains floated elements, some browsers will treat it as if it is zero pixels tall.
- There are two common ways to solve this problem:

1. Method One - set overflow property on parent element to 'auto'



```
.wrapper {
  overflow: auto;
}
```

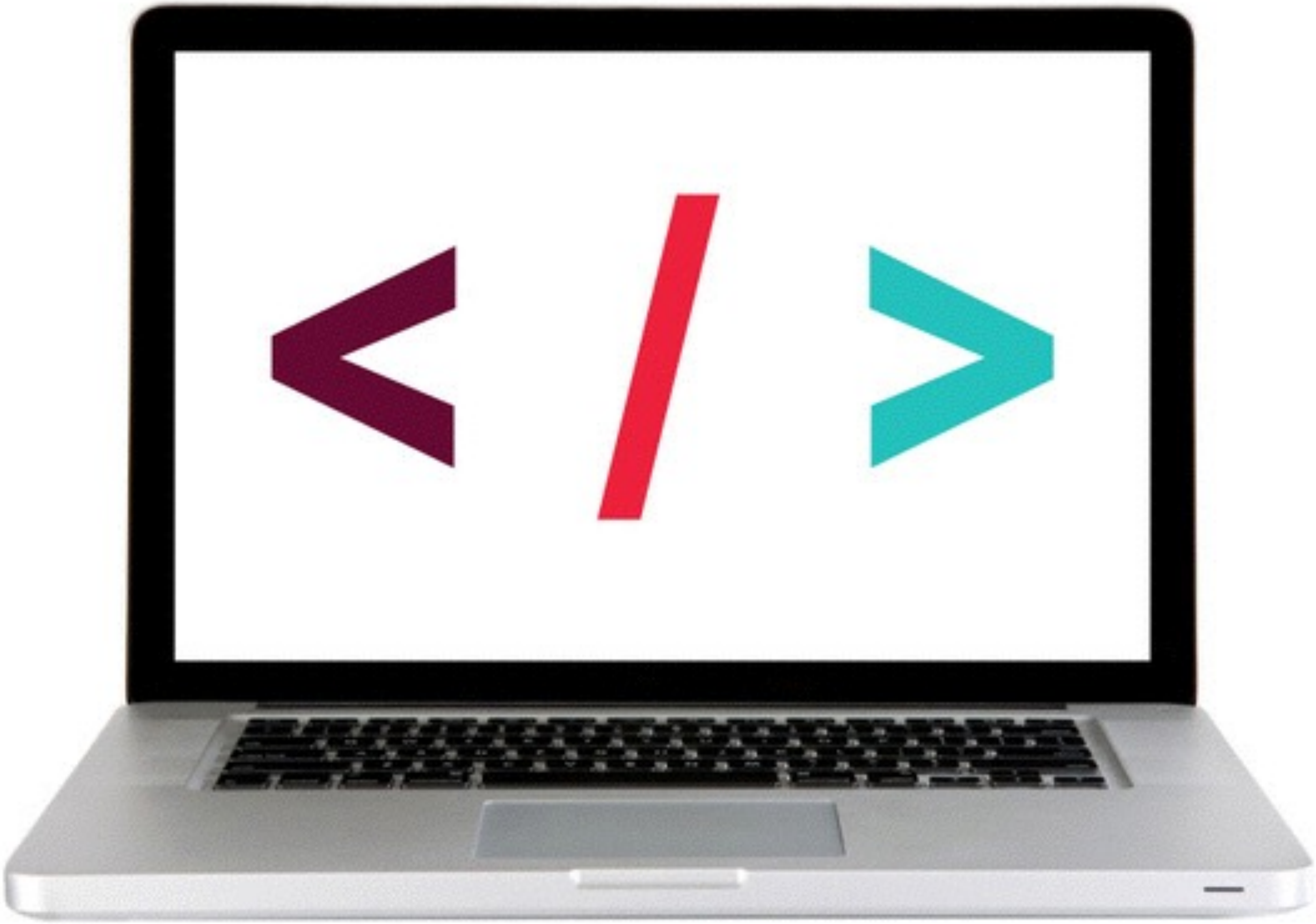
2. Method two (preferred!) - [The Micro Clearfix Hack]



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

**Don't feel the need to memorize! You can look this up.

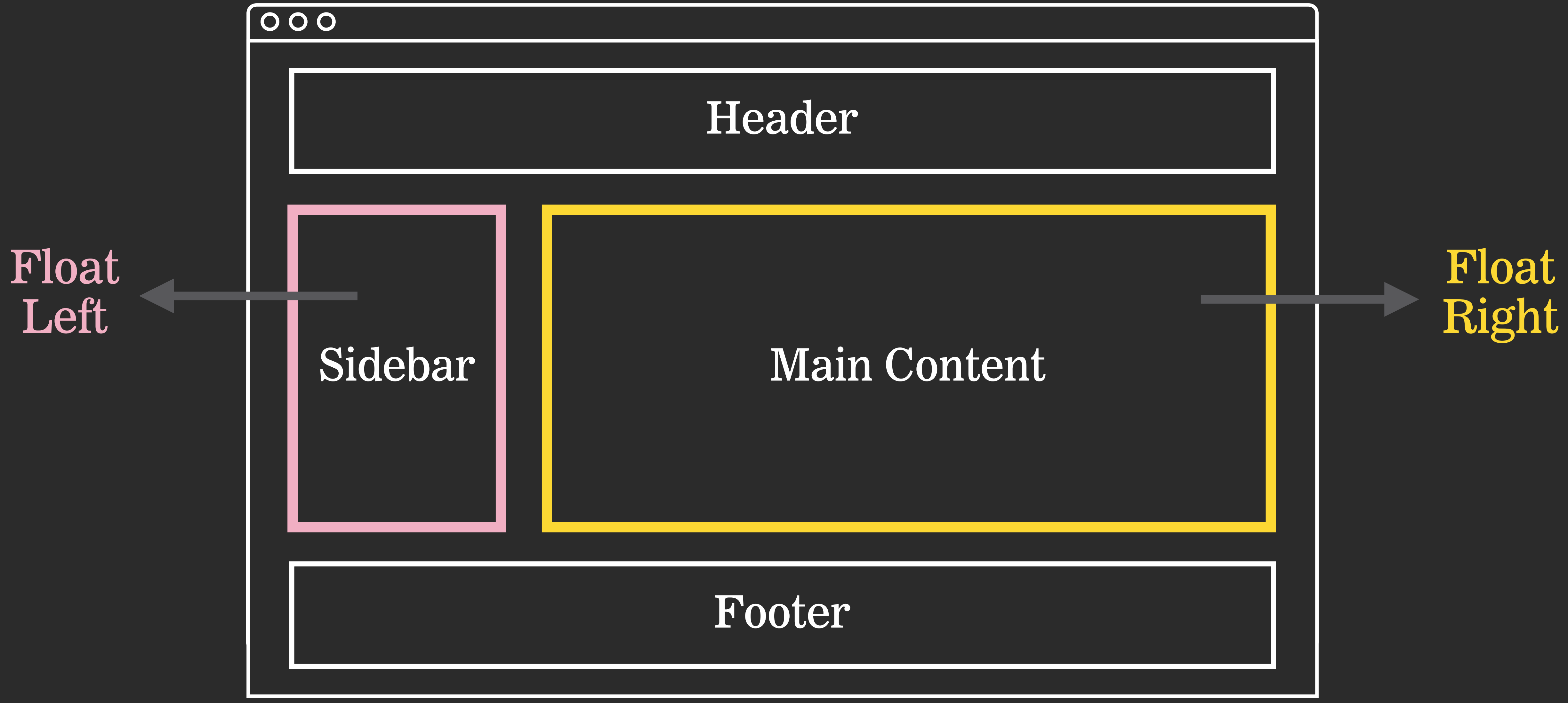
LET'S TAKE A LOOK



FEWD

MULTI-COLUMN LAYOUT

CSS — MULTI-COLUMN LAYOUT

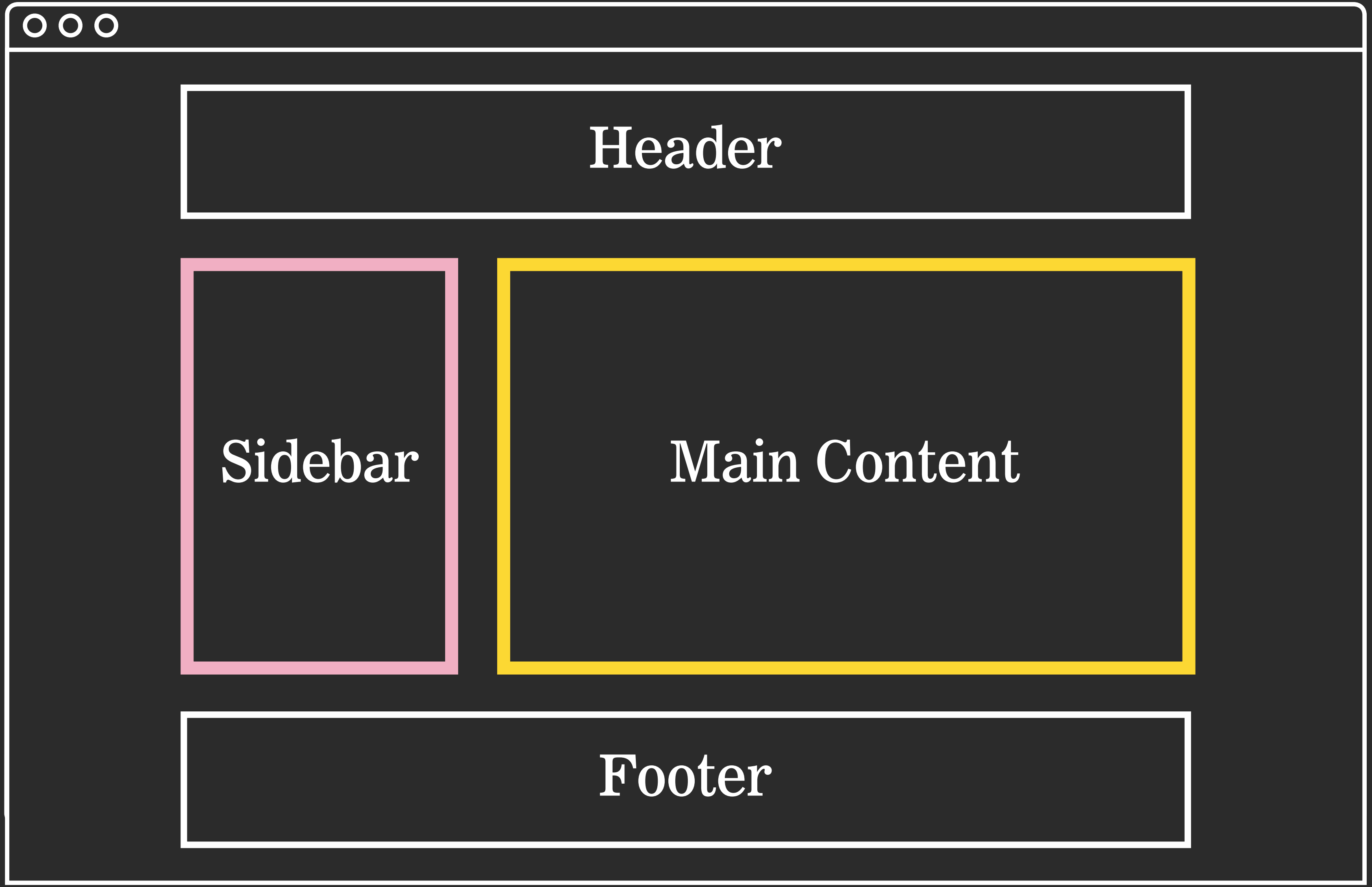


FIXED WIDTH LAYOUT

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

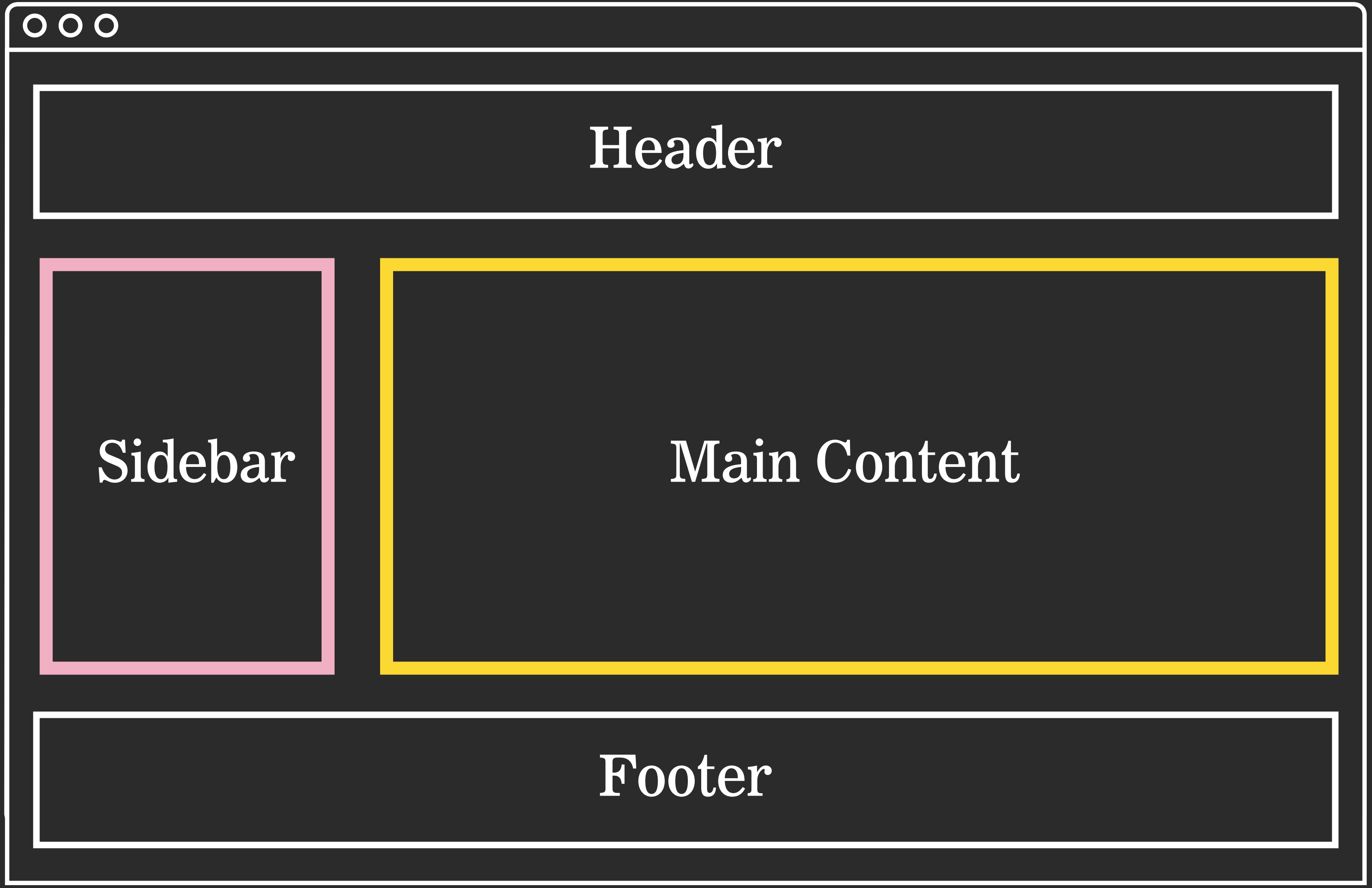


LIQUID LAYOUT

Liquid 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



LET'S TAKE A LOOK — FLOATING SECTIONS CODE ALONG

STARTER TEMPLATE

NORMALIZE.CSS

You can use a **reset** or **normalize** file to give you a blank slate as far as default css styles.

1. Normalize.css makes browsers render all elements more consistently and in line with modern standards. It precisely targets only the styles that need normalizing
2. The common reset from MeyerWeb will take away all the styles foe every HTML tag.

To add to your project:

1. Include the stylesheet (either normalize.css or reset.css) in your css directory
2. Add a <link> to the stylesheet in the head of your HTML. You'll want to include it **above** any other stylesheets so that your styles will be able to override the defaults.

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>
3. You'll also want to use classes and IDs as discussed in class

LEARNING OBJECTIVES

- Identify when HTML5 structural elements should be used.
- Apply header, footer, sidebar, and multi-column layouts to develop a web page.
- Experiment and predict effects of floats and clearing CSS positioning.
- Describe the use of Normalize.css

HOMEWORK

HOMEWORK

BEFORE WEDNESDAY:

- Finish Travel Blog Part 2

REQUIRED READING (BEFORE NEXT MONDAY):

If you purchased the textbook - HTML & CSS by Jon Duckett

- Chapter 13: Boxes and Chapter 15: Layout

Otherwise:

- Read Chris Coyier's All About Floats

BONUS READING:

- From the textbook — Chapter 17: HTML5 Layout

FINAL PROJECTS

WHERE CAN I FIND MORE INFO AND WHEN EACH MILESTONE IS DUE?

- Final Project folder on Schoology

WHERE CAN I GET SOME INSPIRATION FROM WHAT PAST STUDENTS HAVE DONE?

- Visit the General Assembly Gallery

EXIT TICKETS