

---

## LET'S GET EVERYTHING SET UP!

---

1. Open the FEWD 54 website (<https://svodnik.github.io/fewd54>), click the Lessons tab, click the WEEK 2 button, and download the Lesson 3 starter code and slides. You'll want to keep the website 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 you created last week.
3. Double-click on the starter code folder to unzip it
4. After you've unzipped, be sure to delete the original .zip file to avoid confusion and make sure you don't unzip it again later!
5. Open the entire **fewd** folder with your code editor (either drag and drop the folder on the Code icon in the dock on Mac, or open Visual Studio Code, click File > Open, and select the starter code folder).
6. In Slack, join the **#04-layout** channel.

# LAYOUT

---

## WEEKLY OVERVIEW

---

### WEEK 1

HTML Basics / CSS Basics

### WEEK 2

CSS Selectors / Layout

### WEEK 3

Lab / Refactor & Lab

## LAST CLASS:

## MONUMENT

8.000,000

[illegible][illegible][illegible]

## Abstract

LESTER BRUCE, BARRY BE. 2005. *COMPARATIVE BIOLOGICAL AND REPRODUCTIVE RATES OF SALMONELLA ENTERITIDIS FROM CHICKENS AND PIGS*. *Journal of Food Protection* 28:100-105.

© 2006 Blackwell Publishing Ltd *Journal of Internal Medicine* 260: 395–403



최종인물: 1997년 12월 15일

- [illegible]

FACEBOOK    TWITTER    INSTAGRAM

2010年12月25日 星期五

## TODAY'S CLASS:

## MONUMENT

[illegible]

## A VISUAL GUIDE TO THE SOUTHWEST



lamin quod bene et, consuetudo stupens etiam ista sunt incerta, cum aut  
consequenter aliquid non videtur illis, istis de speculibus doctus, quibus et  
... non aliquid e. Speculibus istis quibus incertum est, aut debetis quodammodo  
reus aliquid, hic conuenit speculatio hic a. Velipius dicitur incertis capere  
hanc sunt huiusmodi, uenerunt quoniam noster illis apertis velipius repellet etiam  
hic namque tempore non videtur doctus aut accutius recedente plenis esse inquit  
de istis speculibus incertum nonque enim apertis, facit dea preteritum  
speculatio speculatio non autem accutius incertis accutius accutius incertis  
incertis et hic non videtur consuetudo ista. Examinatio doctus  
reus istis incertis non videtur non autem accutius incertis, bene  
incertis aliquid incertis nonque magis tempore incertis aliquid doctus  
hic non quod aliquid. Examinatio velipius dicitur, incertis



Si non quis dicit illud, contra totius adunantiam illi non solus testatur, sed et  
consequenter aliquos non infirmos, sed velis et perfidos, etiam aliquos et  
non, in istis et Francisca exasperat quos temperis, etia et deinde quod volunt  
exire ipsam, in conspectu apud nos et in voluntate ipsam recordis ipsos  
Sicut velis legitime quantumque reprehendit optime velis reprehendit  
quod contra totius adunantiam illi non solus testatur, sed et  
consequenter aliquos non infirmos, sed velis et perfidos, etiam aliquos et  
non, in istis et Francisca exasperat quos temperis, etia et deinde quod volunt  
exire ipsam, in conspectu apud nos et in voluntate ipsam recordis ipsos  
Sicut velis legitime quantumque reprehendit optime velis reprehendit

[ABOUT US](#)

Lexis: *ignis solentis* sunt,  
concreta adspicere est, *verum*  
fugit illa sed quod incidit  
nonne conuenit, *solent* *solent* *con-*  
*cre* *et* *et* *et*, *concreta*, *solentis*  
*solent* *et* *concreta* *concreta*

Explicite connotetur tempora  
performantia velutque tempora  
corpora ad prout aliquam  
eundem finem nominata sunt ut  
edificii, eorum fuit perspicit  
denotant utrumque? Insuper  
expone numerum, nam  
exemplum et velutque tempora



## RECENT ISSUES

- Issue 60/January – Ongoing
- Issue 59/December – Final
- Issue 58/November – Digital
- Issue 57/October – Editorial
- Issue 56/September – Community
- Issue 55/August – Submission



000000 0000000000

© 2006 The Authors

陳國治、王曉波、李俊承、林建宏

# LEARNING OBJECTIVES

- Differentiate between block and inline elements
- Use HTML5 structural elements to properly organize your page.
- Discover how to create multi-column layouts with flexbox

---

# AGENDA

---

Display

Divs

HTML5 Structural Elements

Flexbox

Multi-column Layouts

Vertical Centering

Lab - Part 2

**REVIEW**

---

## ACTIVITY — REVIEW

---

- When would you want to use a class and when would you want to use an id?
- What are the steps for adding classes and ids and then styling elements with classes and ids in HTML/CSS?
- Why is inheritance useful when styling a webpage?
- We want to style an `<a>` that is a descendent of a `<p>` in our CSS. Write out this selector.
- We want to style a `<p>` that is a descendent of a container with an id about. Write out this selector.



---

## ACTIVITY — NESTED SELECTORS LAB

---



### EXERCISE

#### KEY OBJECTIVE

---

- ▶ Get practice using selectors

#### LOCATION OF FILES

---

- ▶ starter code > **nested\_selectors** folder

#### TIMING

---

*5 min*

1. First make all the anchors on the page #FFA7C0
2. Then make the anchors within the header nav #D16121
3. Lastly, make the anchors in the footer nav #FFE540

---

**CSS SELECTORS**

---

# USING CHROME DEV TOOLS

---

## LET'S USE THE INSPECTOR!

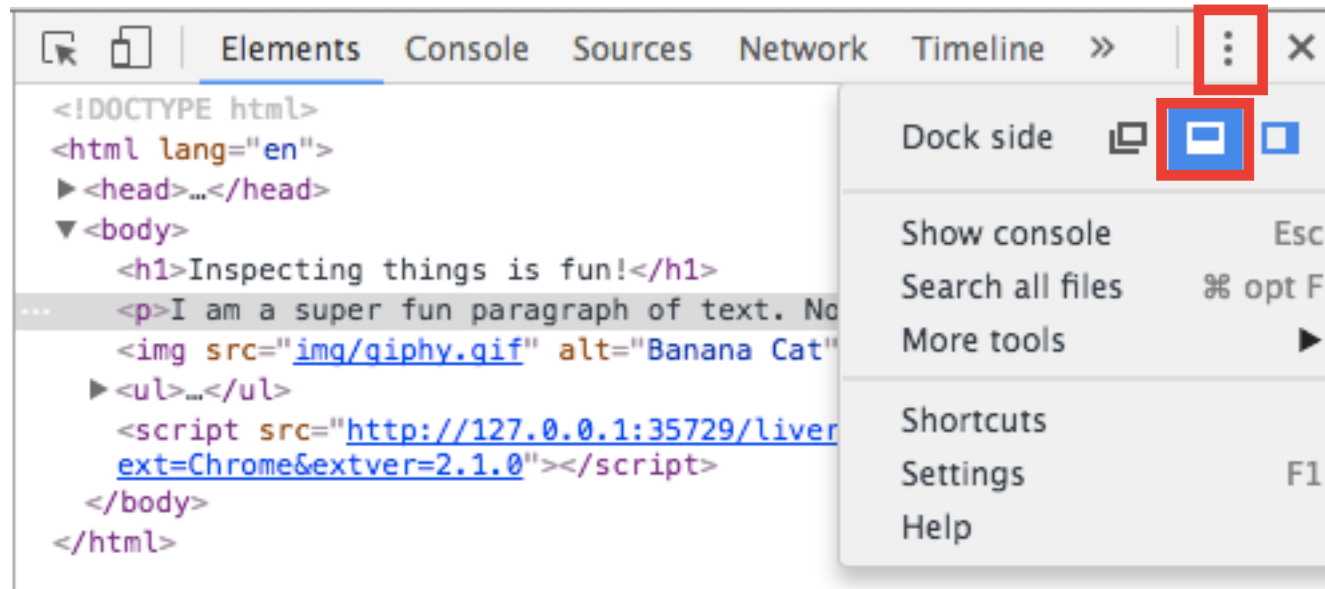
---

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*

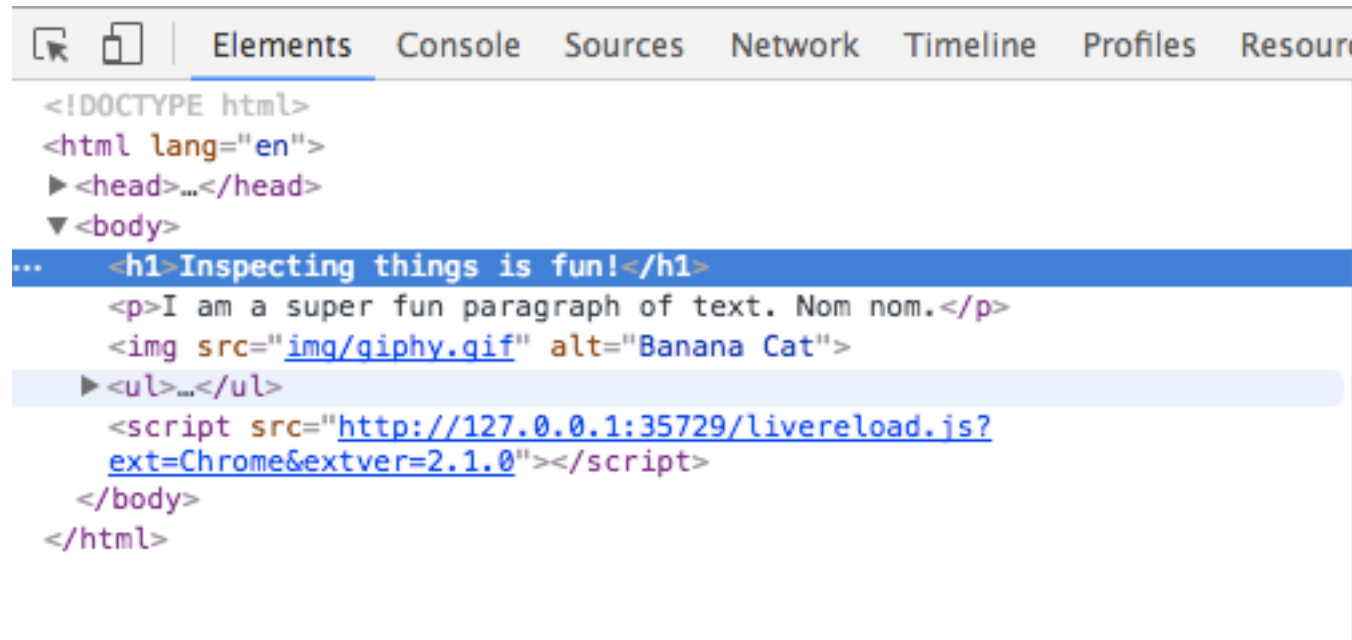
## DOCK LOCATION

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

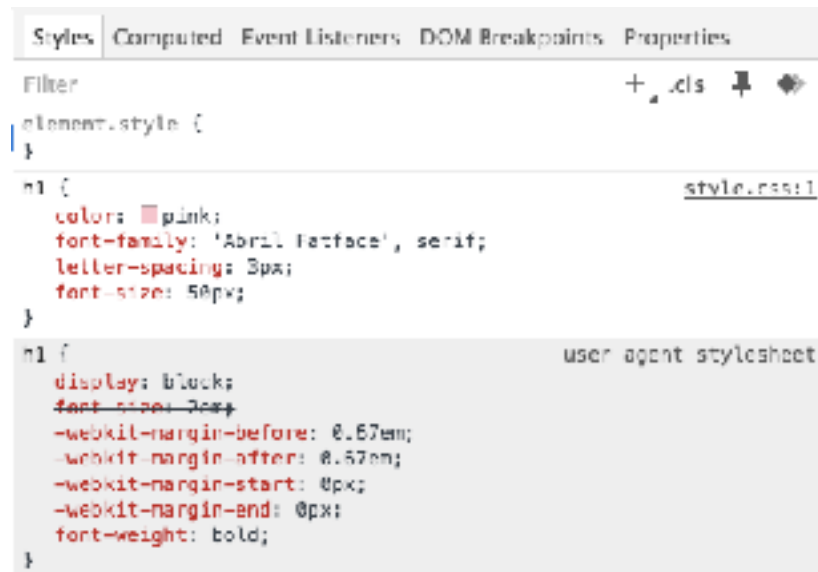


---

## 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!

```
✓ color: pink;  
✓ font-family: 'Abril Fatface', serif;  
✓ letter-spacing: 3px;  
  font-size: 100px;  
}
```

# 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

margin67borderpadding932 x 270

FilterShow all

▶colorrgb(255, 192, 203)

▶displayblock

▶font-family'Abril Fatface', serif

▶font-size100px

▶font-weightbold

height270px

▶letter-spacing3px

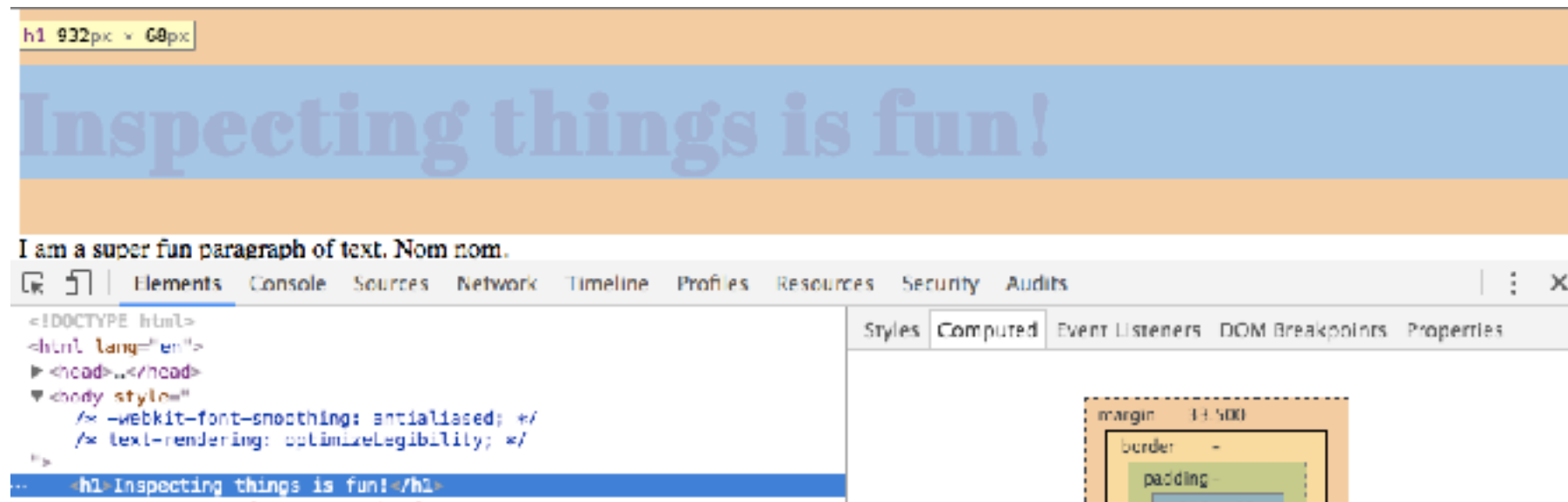
width932px

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



# LET'S USE THE INSPECTOR!

---



## EXERCISE

### LOCATION

---

► starter code > inspect\_element

### LAB SESSION

---

8 min

1. What color is the h1? Can you find the rgb value?
2. What is the width and height of the h1? How much margin does it have on each side?
3. Hover over the h1 to visualize these dimensions.
4. What is the margin-top of the p element? Margin-bottom? Are these styles added in our stylesheet or default styles given by the browser (in the user agent stylesheet)
5. What is the background-color for each li? How much padding and margin does each li have?
6. Hover over an li to visualize these dimensions.
7. What is the width/height of the img?

# INSPECT ELEMENT



To access the elements panel (in Chrome): Right click > Inspect Element

---

**FEWD**

---

**DISPLAY**

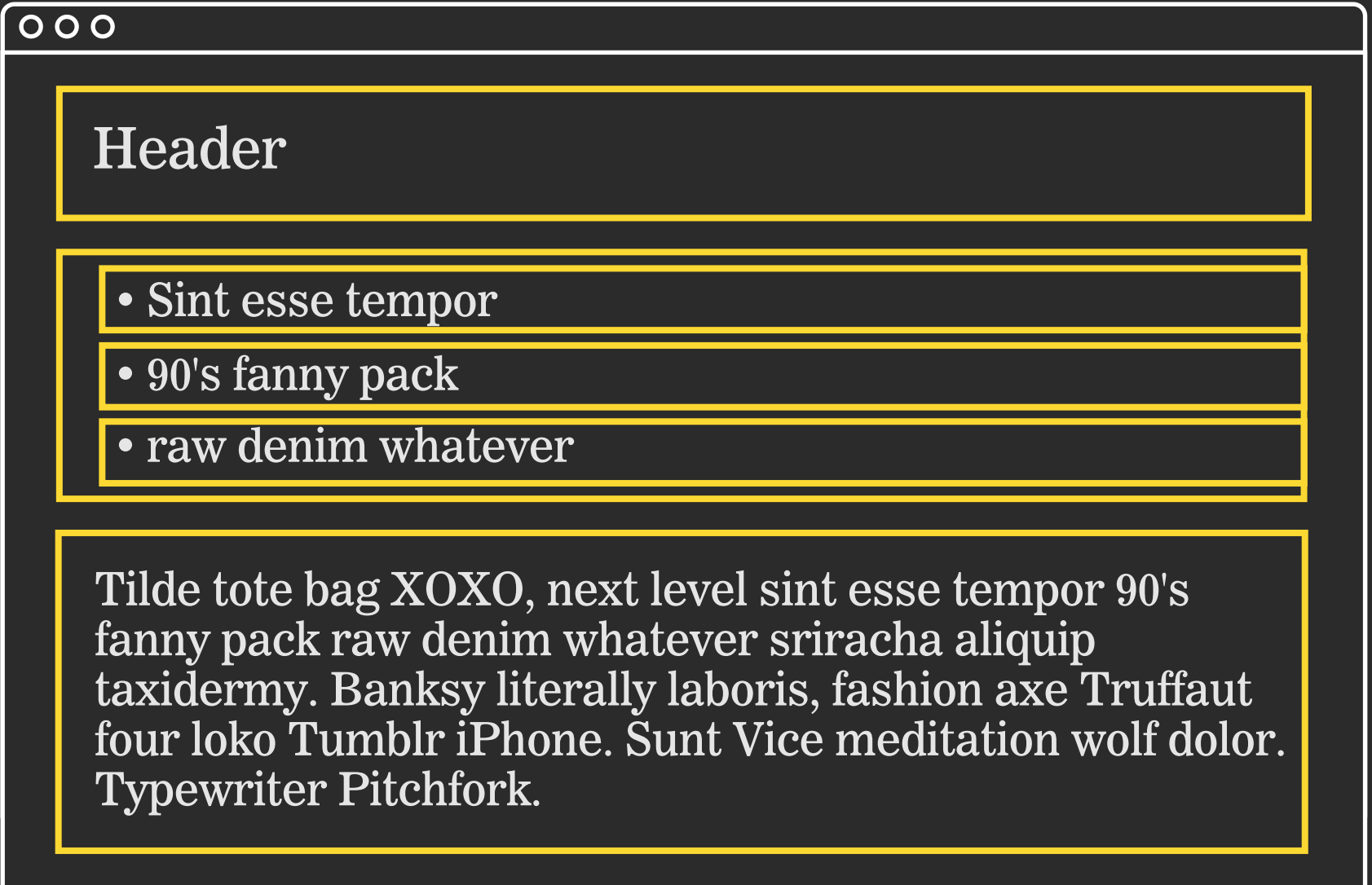
# BUILDING BLOCKS

## BLOCK-LEVEL ELEMENTS

- ▶ Will always start on a new line

Examples:

- ▶ `<h1>-<h6>`
- ▶ `<ul>`
- ▶ `<li>`
- ▶ `<p>`
- ▶ `<ol>`
- ▶ `<div>`



## BUILDING BLOCKS

### INLINE ELEMENTS

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

Examples:

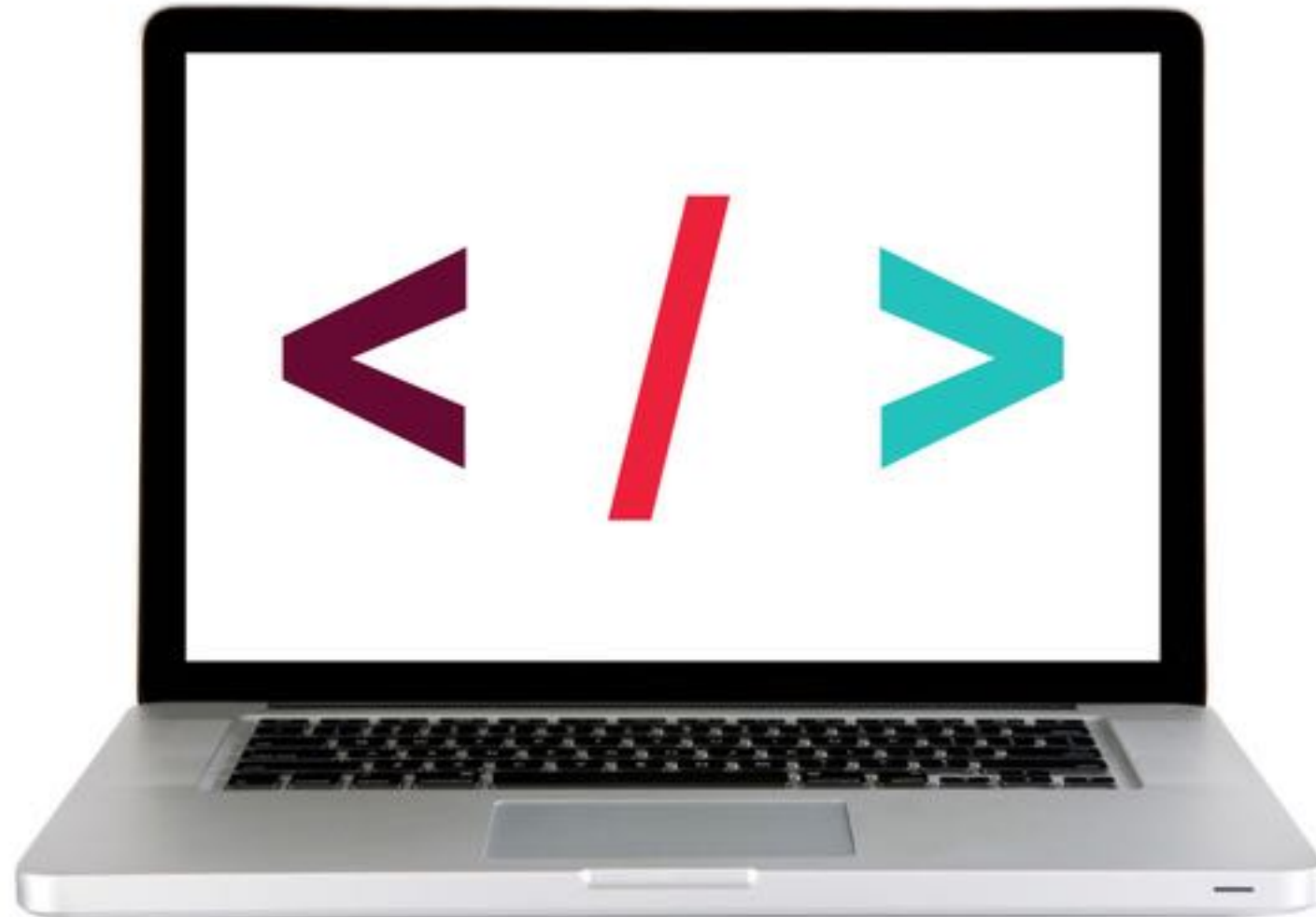
- ▶ `<a>`
- ▶ `<img>`
- ▶ `<em>`
- ▶ `<strong>`
- ▶ `<span>`



---

## 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?

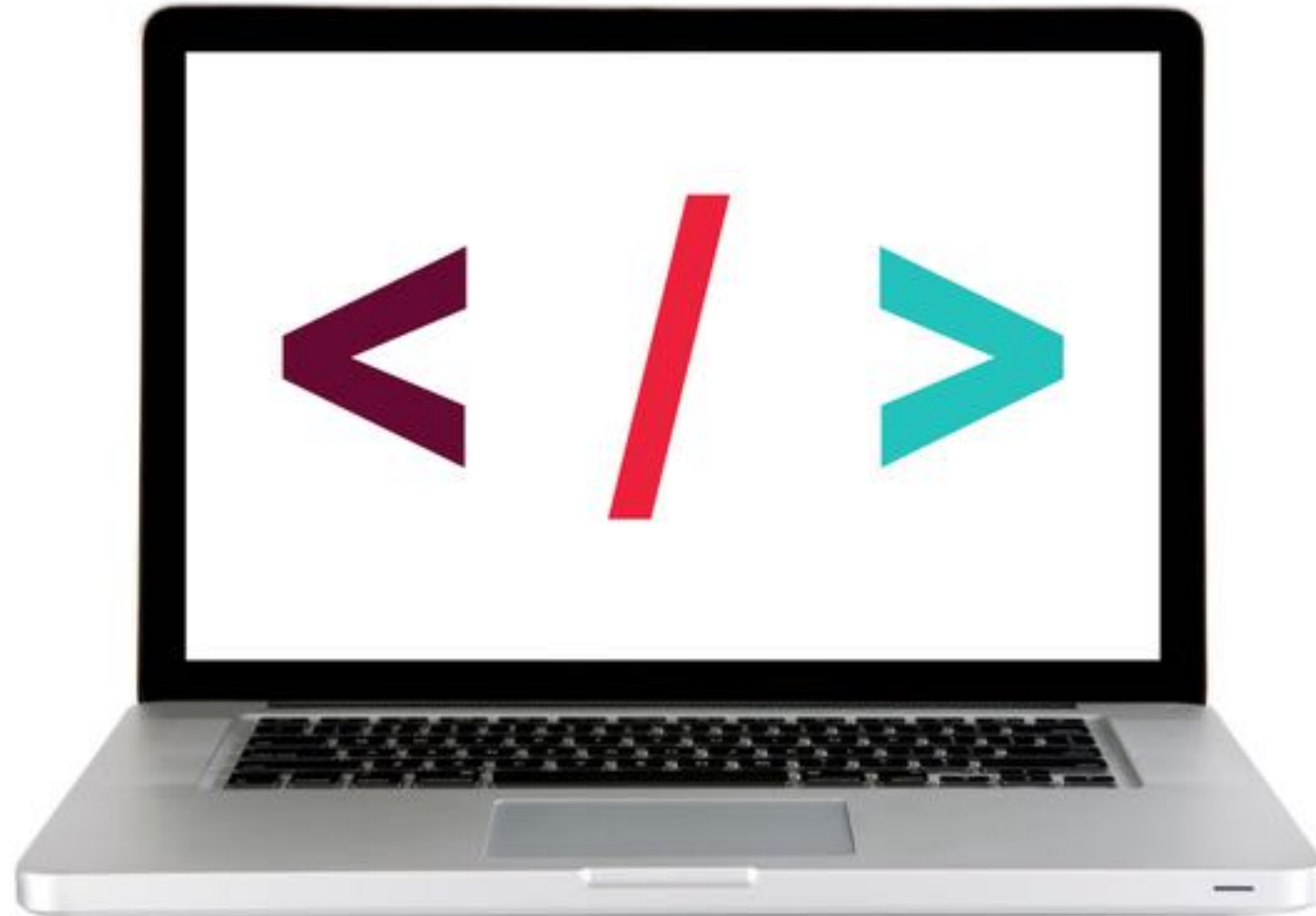
	WIDTH & HEIGHT	PADDING & MARGIN
BLOCK	yes	can apply to all sides
INLINE	no	will only affect left and right sides



---

## 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.

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

---

## DISPLAY — INLINE

---

To make a block-level element act like an inline element:

### 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-level 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:

### BEFORE:

[Link](#)

[Link](#)

[Link](#)

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

[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

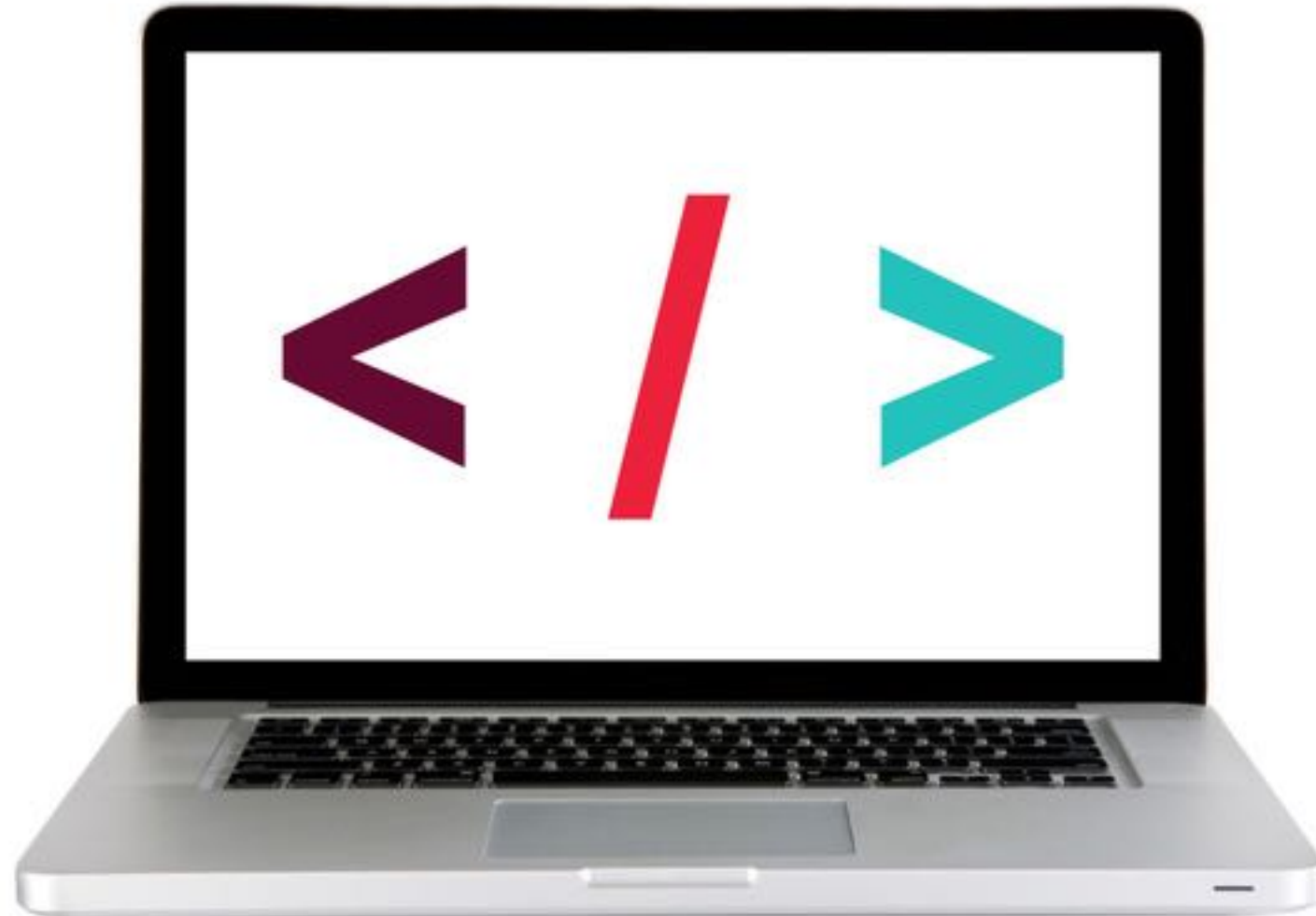
```
li {  
    display: none;  
}
```

► Elements will be hidden from the page

---

## LET'S TAKE A CLOSER LOOK

---



---

# 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

---

**FEWD**

---

**CENTER ALL THE THINGS!**

---

## CENTERING THINGS

---

### Centering text:

**Is it an inline or inline-block element? (a, span, etc.)**

Option 1: Set `text-align: center` on the parent

Option 2: Set `display` to `block` and then set `text-align: center` on the element

**Is it a block-level element? (p, h1, h2, etc.)**

Set `text-align: center` on the element

### Centering content for an image:

`display: block` and `margin: 0 auto;`

### Centering content for a non-text block-level element (div, section, article, etc):

Set a width and `margin: 0 auto` in CSS.

---

**FEWD**

---

**DIVS**

---

## GROUPING TEXT & ELEMENTS

---

### THE <DIV> ELEMENT

- Allows us to group a set of HTML elements together into a **block-level** box
- We can then style that chunk of HTML

### THE <SPAN> ELEMENT

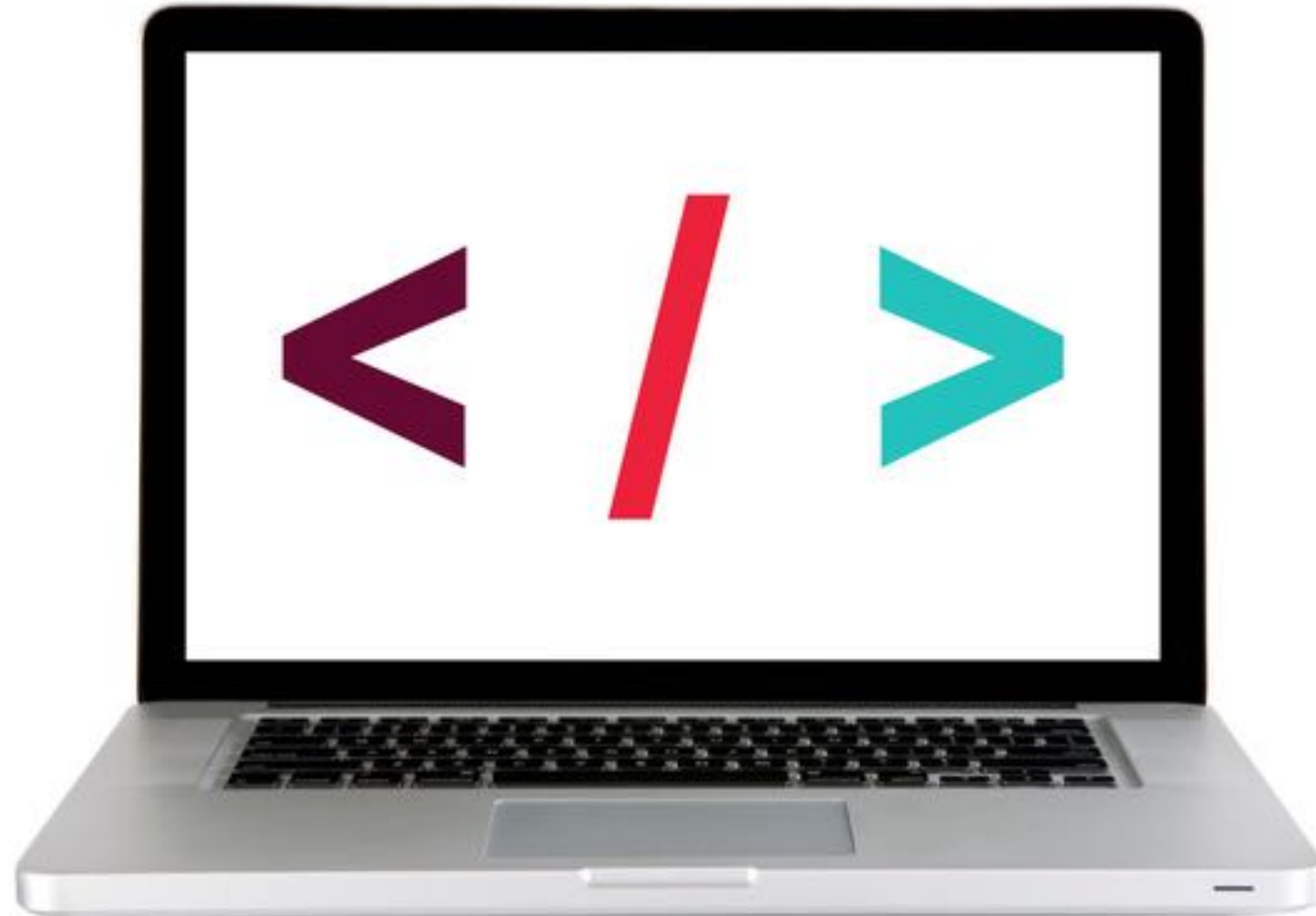
The <span> element acts like the **inline equivalent** of the <div> element. It is used to either:

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

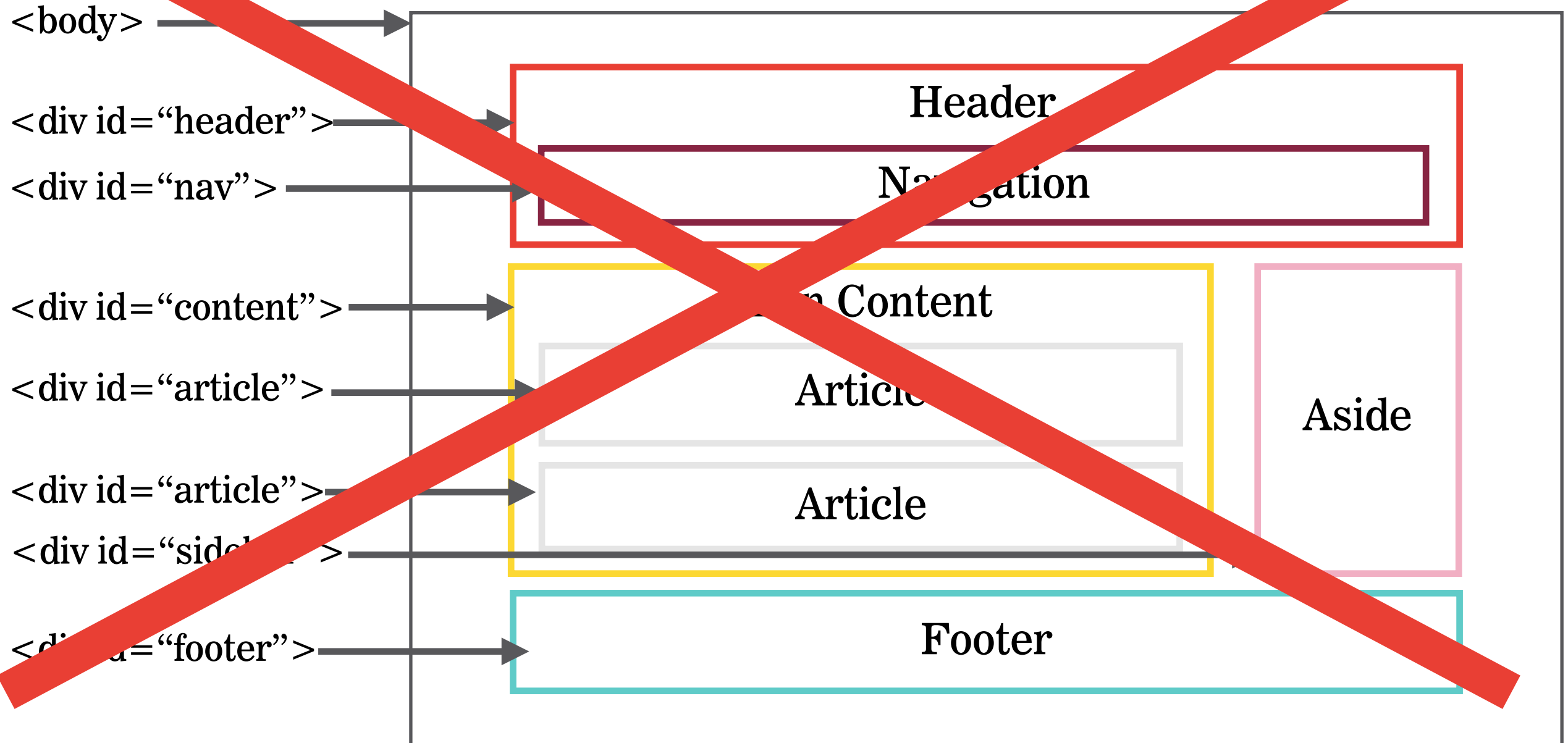
---

**LET'S TAKE A LOOK**

---



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



---

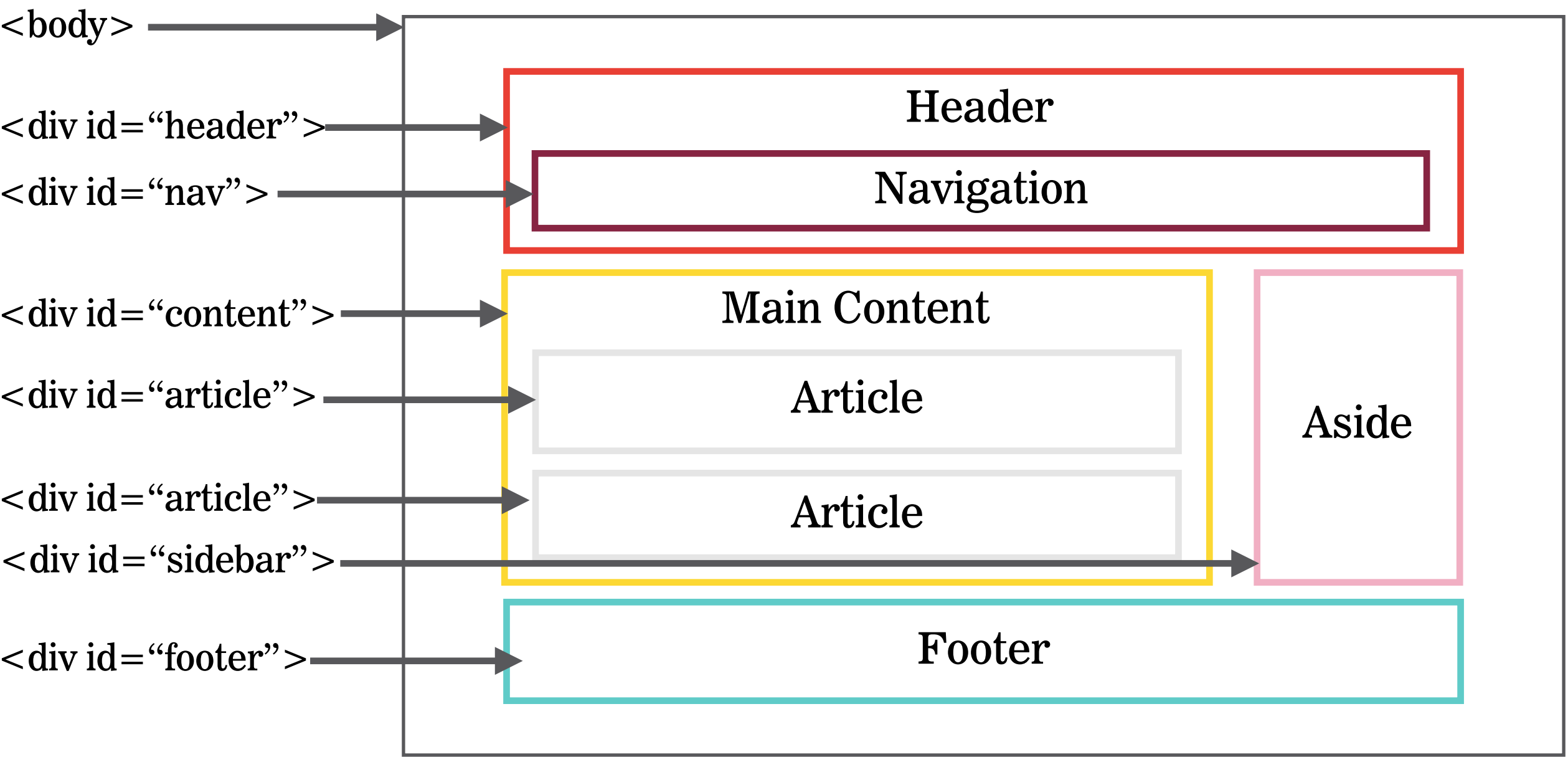
**FEWD**

---

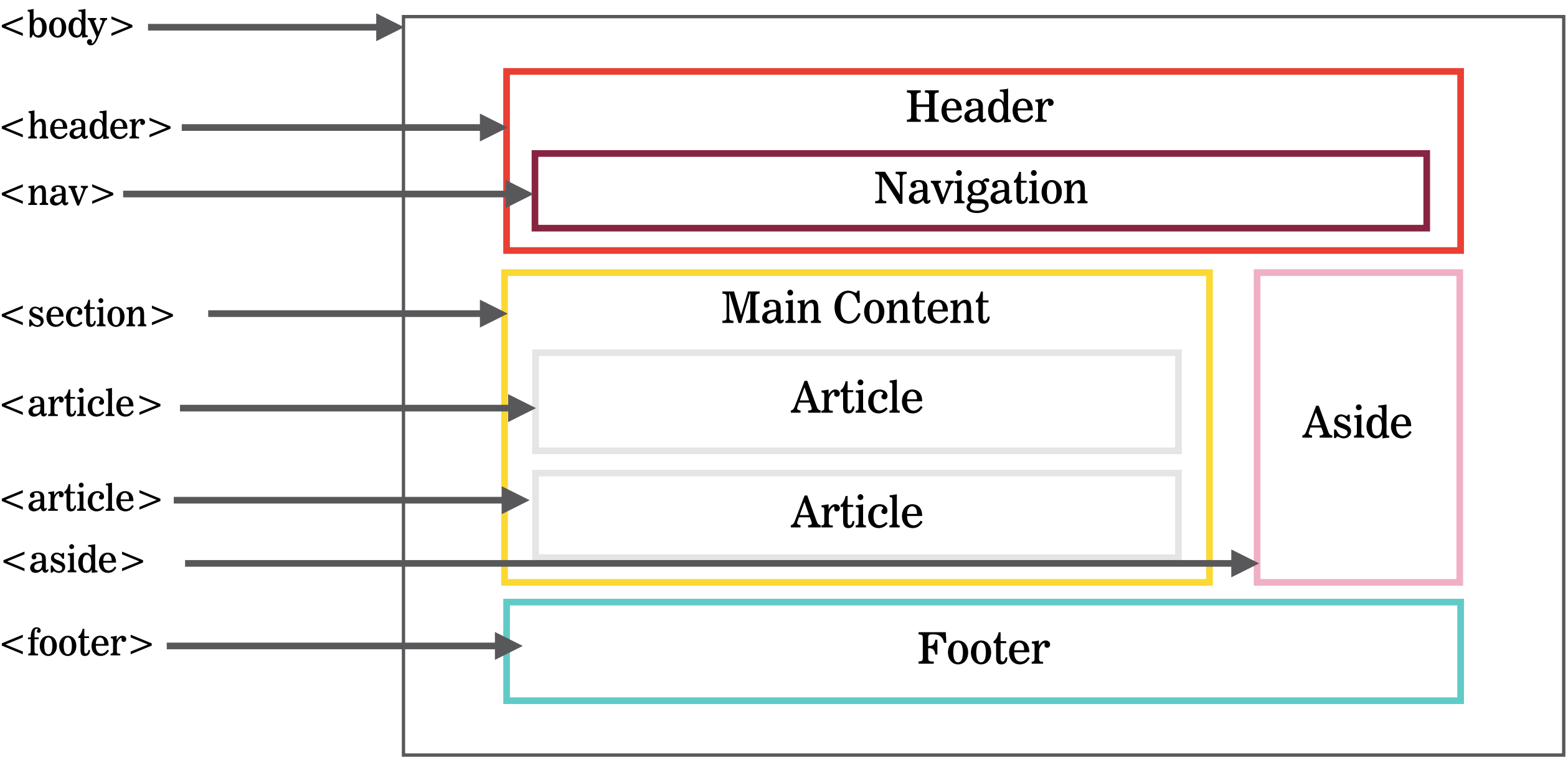
# HTML5 STRUCTURAL ELEMENTS



# 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>
```

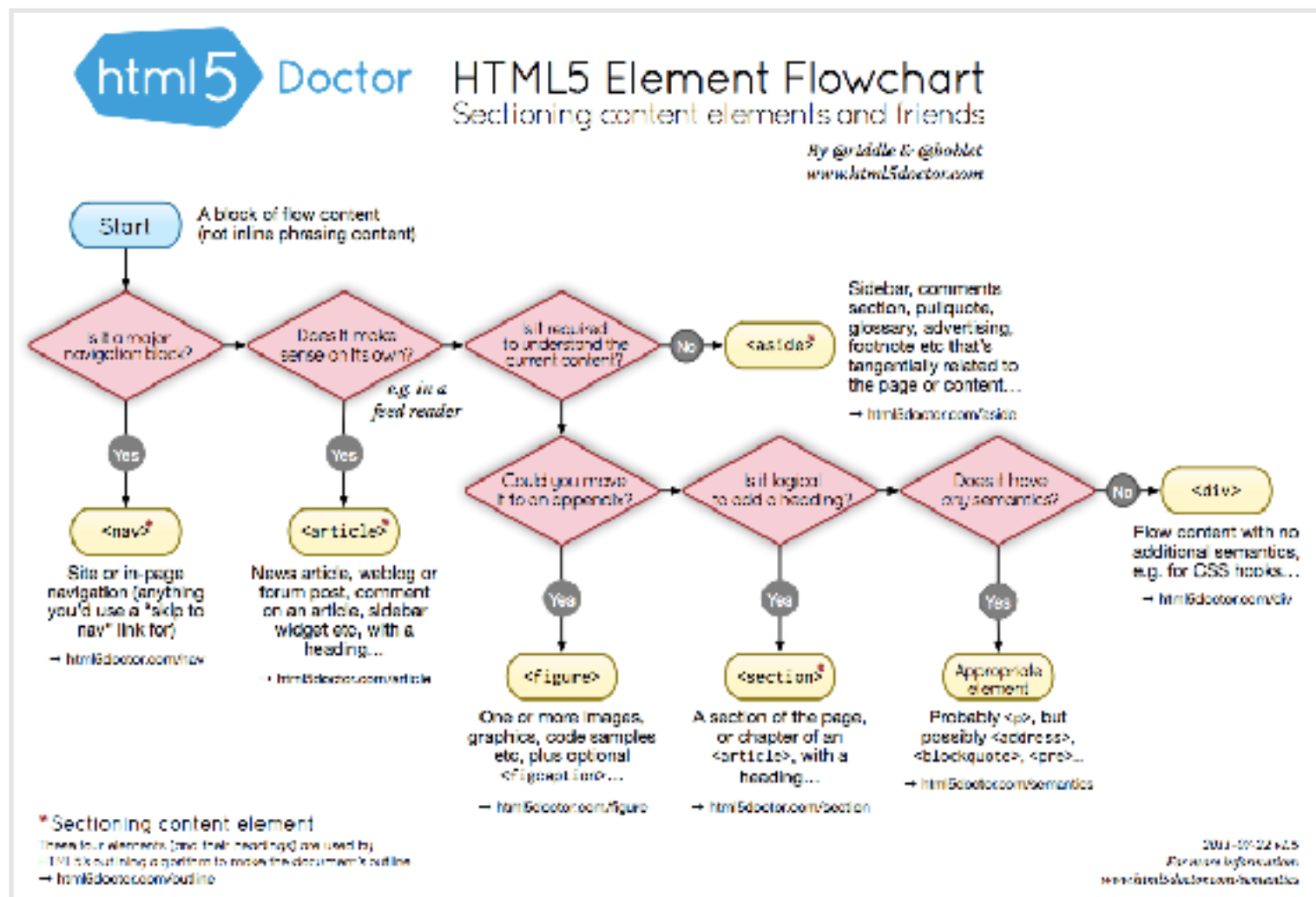
---

## 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 element that suits your purpose**
- Its most common use will likely be for **stylistic purposes** — i.e., wrapping some HTML 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.



12/22/2015

## A VISUAL GUIDE TO THE SOUTHWEST



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eius soluta incididunt, eum quam consequuntur aliquam nobis veli nullam lae, natus ex repellendus dolentis quibusque et quae. In atque a. Peripetatis excepturi ipsum temporibus, eius veli delendi quasi voluptates exque aliquam. Iste consequatur repudiandae illo a. Voluptates eligendi nocendi corporis itaque soluta fugit atque quoniam ipsum reprehenderit sapientis, voluptatem repellat exierit fuga natus tempore modi vero distinctio nulla accusantium recusandae, amnis. Eos aliquam ex, conitque repudiandae nihil numquam, neque enim perspiciatis. Facilis dolor praesentium inpedit quidem, non unde accusantium, conmodi amos, soluta dolores modi blanditis, similique amet hic eam, recitendis consectetur amet in. Exercitationem distinctio necessitatibus malinas quis laboriosam nam modi parvo accusantium blanditis, beatae incidunt aliquid nostrum laborum, magis cum recusandae autem doloreque qui perferendis hic non quoniam fugit Aspernatur voluptatem inpedit, enim!



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eius soluta incididunt, eum quam consequuntur aliquam nobis veli nullam lae, natus ex repellendus dolentis quibusque et quae. In atque a. Peripetatis excepturi ipsum temporibus, eius veli delendi quasi voluptates exque aliquam. Iste consequatur repudiandae illo a. Voluptates eligendi nocendi corporis itaque soluta fugit atque quoniam ipsum reprehenderit sapientis, voluptatem repellat exierit fuga natus tempore modi vero distinctio nulla accusantium recusandae, amnis. Eos aliquam ex, similique repudiandae nihil numquam, neque enim perspiciatis. Facilis dolor praesentium inpedit quidem, non unde accusantium, conmodi amos, soluta dolores modi blanditis, conitque amet hic eam, recitendis consectetur amet in. Exercitationem distinctio necessitatibus malinas quis laboriosam nam modi parvo accusantium blanditis, beatae incidunt aliquid nostrum laborum, magis cum recusandae autem doloreque qui perferendis hic non quoniam fugit Aspernatur voluptatem inpedit, enim!

## ABOUT US

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eius soluta incididunt, eum quam consequuntur aliquam nobis veli nullam lae, natus ex repellendus dolentis quibusque et quae. In atque a. Peripetatis excepturi ipsum temporibus, eius veli delendi quasi voluptates exque aliquam. Iste consequatur repudiandae illo a. Voluptates eligendi nocendi corporis itaque soluta fugit atque quoniam ipsum reprehenderit sapientis, voluptatem repellat exierit fuga natus tempore modi vero distinctio nulla accusantium recusandae, amnis. Eos aliquam ex, similique repudiandae nihil numquam, neque enim perspiciatis. Facilis dolor praesentium inpedit quidem, non unde accusantium, conmodi amos, soluta dolores modi blanditis, conitque amet hic eam, recitendis consectetur amet in. Exercitationem distinctio necessitatibus malinas quis laboriosam nam modi parvo accusantium blanditis, beatae incidunt aliquid nostrum laborum, magis cum recusandae autem doloreque qui perferendis hic non quoniam fugit Aspernatur voluptatem inpedit, enim!

Explicita consectetur temporis perferendis voluptas itaque, corporis et ipsum quibusque exierit, facili perspiciatis. Eum ad adipisci, carum fugit perspiciatis denunti sentent expit inpedit expedit assumenda, namper excepturi ad voluptatibus. Explicit.



## RECENT ISSUES

- Issue Nineteen — Camping
- Issue Eighteen — Food
- Issue Seventeen — Signs
- Issue Sixteen — Friends
- Issue Fifteen — Carnivals
- Issue Fourteen — Gatherings



EMAIL ADDRESS

KEEP IN TOUCH

---

## 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**



# LET'S TAKE A CLOSER LOOK

## BEATLES — DRIVE MY CAR

### LYRICS

Asked a girl what she wanted to be  
She said baby, "Can't you see  
I wanna be famous, enter on the screen  
But you can do something in between"

Baby you can drive my car  
Yes I'm gonna be a star  
Baby you can drive my car  
And maybe I love you

I told a girl that my prospects were good  
And she said baby, "It's understood  
Working for peanuts is all very fine  
But I can show you a better time"

I told that girl I can start right away  
And she said, "Listen baby I got something to say  
I got no car and it's breaking my heart  
But I've found a driver and that's a start"

Baby you can drive my car  
Yes I'm gonna be a star  
Baby you can drive my car  
And maybe I love you  
Beep beep'm beep beep yeah  
Beep beep'm beep beep yeah  
Beep beep'm beep beep yeah  
Beep beep'm beep beep yeah (fade out)

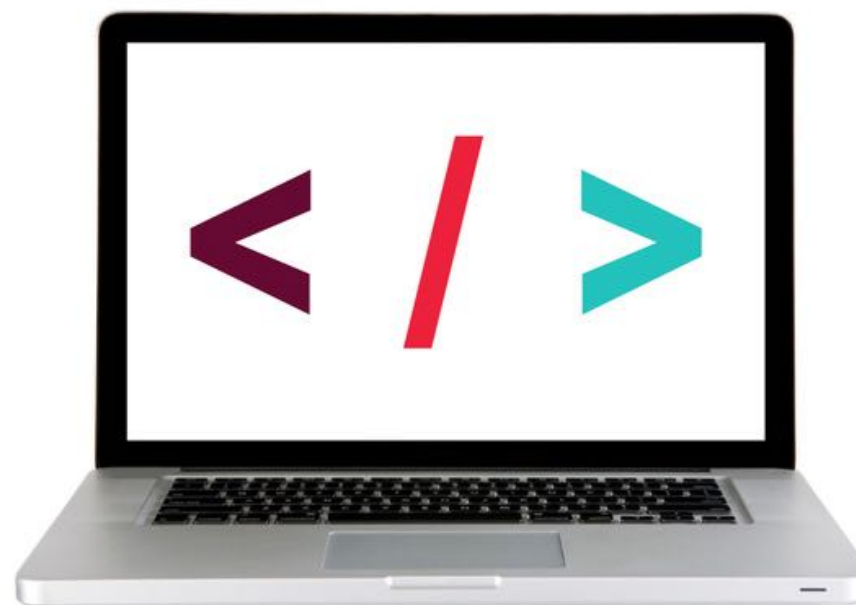
### ALBUM INFO



Rubber Soul is the sixth studio album by English rock band the Beatles, released on 3 December 1965.

- Album: Rubber Soul
- Release Date: December 3, 1965

COPYRIGHT © 2016 LYRICMANIA BABY



---

## ACTIVITY — FLOATS LAB

---



### EXERCISE

#### KEY OBJECTIVE

---

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

#### LOCATION OF FILES

---

- ▶ starter code > multi\_columns\_lab folder

#### TIMING

---

*2 min*

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

---

**FEWD**

---

# FLEXBOX

---

## WHAT IS FLEXBOX?

---

- You've possibly used floats in the past to build column layouts, but Flexbox is now considered best practice for grid layouts.
- We'll be using Flexbox to build our column-based, grid layouts. In addition to creating incredible structure on our page, Flexbox solves a lot of layout issues that float either causes or cannot solve (or both!).

---

## WHAT IS FLEXBOX?

---

Before we start with describing the flexbox properties let's give a little introduction of the flexbox model. The flex layout is constituted of parent container referred as **flex container** and its immediate children which are called **flex items**.

```
<section>
  <article>
    <h1>Article title</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nihil
    hic ipsa harum culpa dolorum, reiciendis sed cum quibusdam archi
  </article>
  <article>
    <h1>Article title</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Cum
    sed fugit deserunt anini asperiores, nisi quos tempora ab repreh
  </article>
  <article>
    <h1>Article title</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Per
    Ipsam provident quas delectus doloremque nemo incidunt quisquam
  </article>
</section>
```

# WHAT IS FLEXBOX?

Before we start with describing the flexbox properties let's give a little introduction of the flexbox model. The flex layout is constituted of parent container referred as **flex container** and its immediate children which are called **flex items**.

parent wrapper:  
**flex container**

```
<section>
  <article>
    <h1>Article title</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nih
    hic ipsa harum culpa dolorum, reiciendis sed cum quibusdam archi
  </article>
  <article>
    <h1>Article title</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Cum
    sed fugit deserunt anini asperiores, nisi quos tempora ab repreh
  </article>
  <article>
    <h1>Article title</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Per
    Ipsam provident quas delectus doloremque nemo incidunt quisquam
  </article>
</section>
```

direct children:  
**flex items**

---

**FEWD**

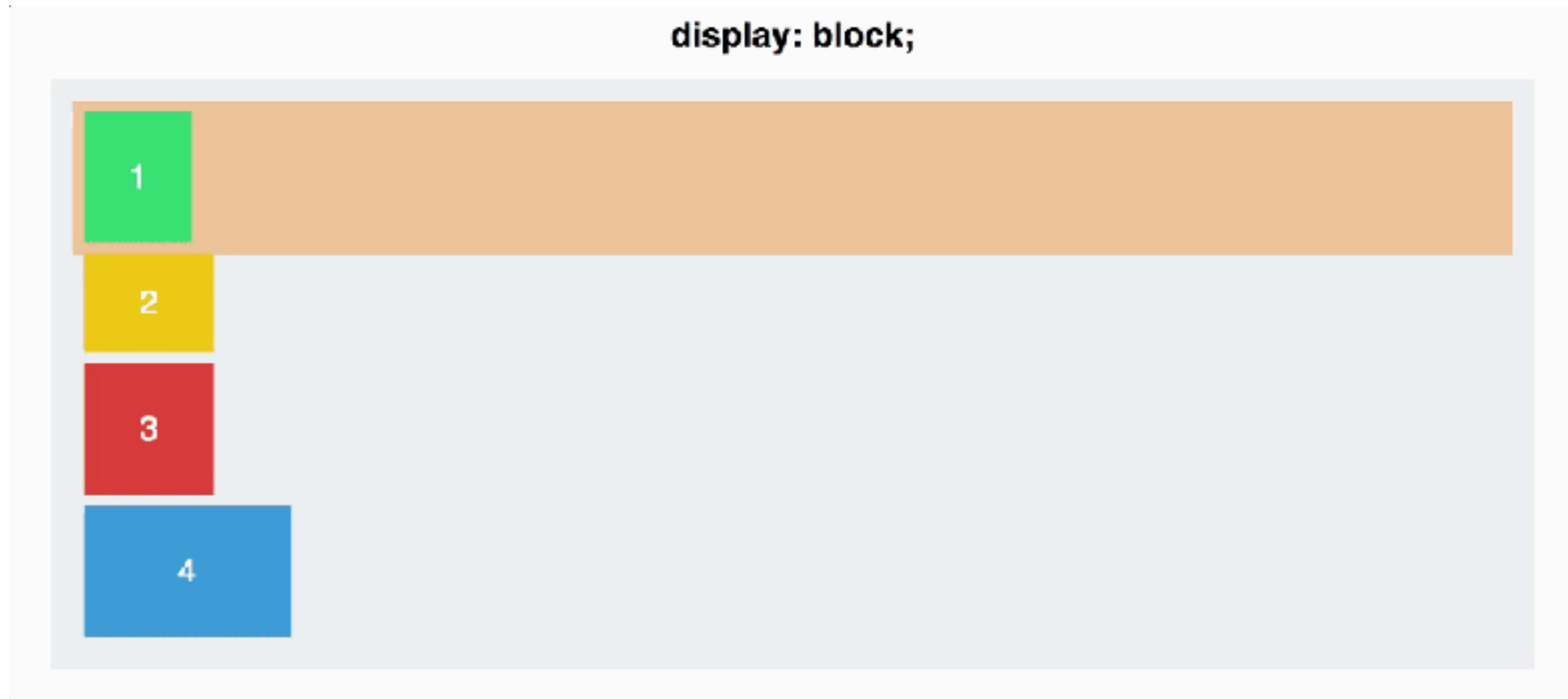
---

# **FLEX CONTAINER**

---

## WHAT IS FLEXBOX?

---





---

## PARENT WRAPPER (FLEX CONTAINER)

---

In order to get started with Flexbox, you need to make your container into a flex container. This is as easy as:

parent wrapper:  
flex container

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

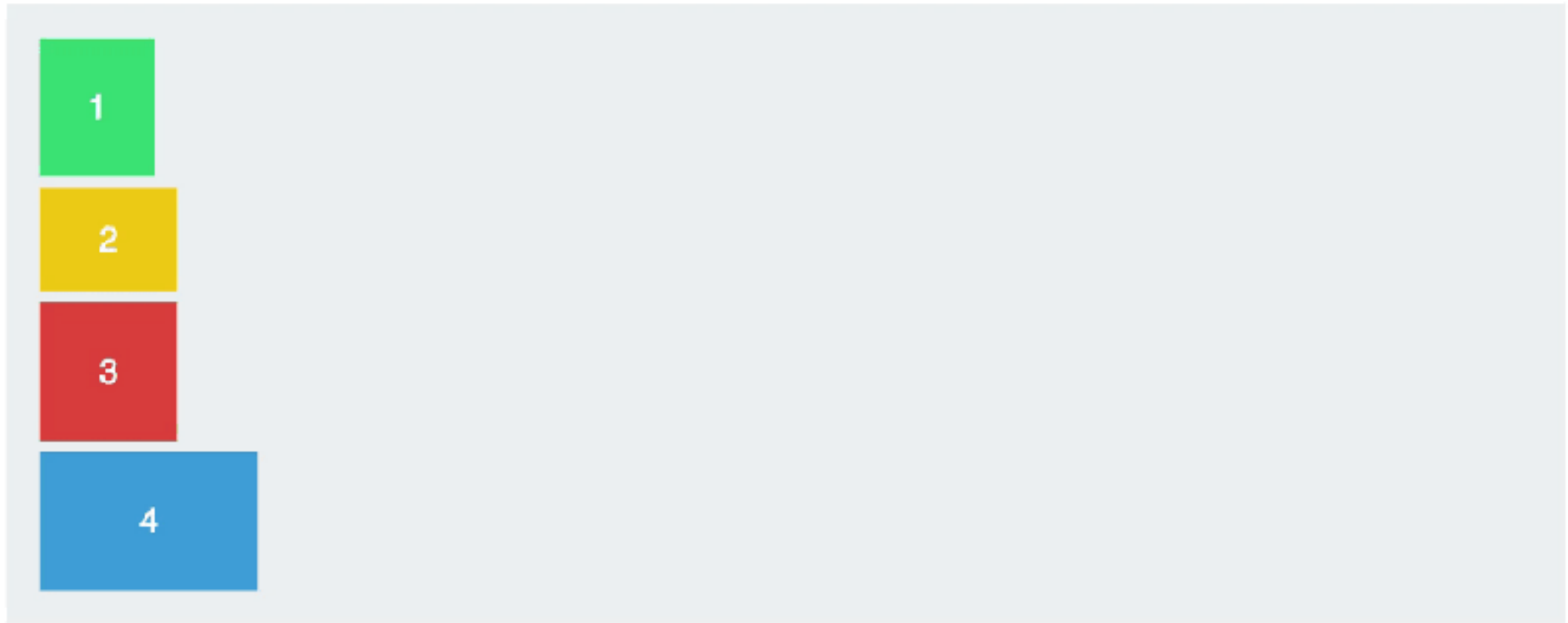
```
section {
  display: flex;
}
```

---

## HOW DOES FLEXBOX WORK?

---

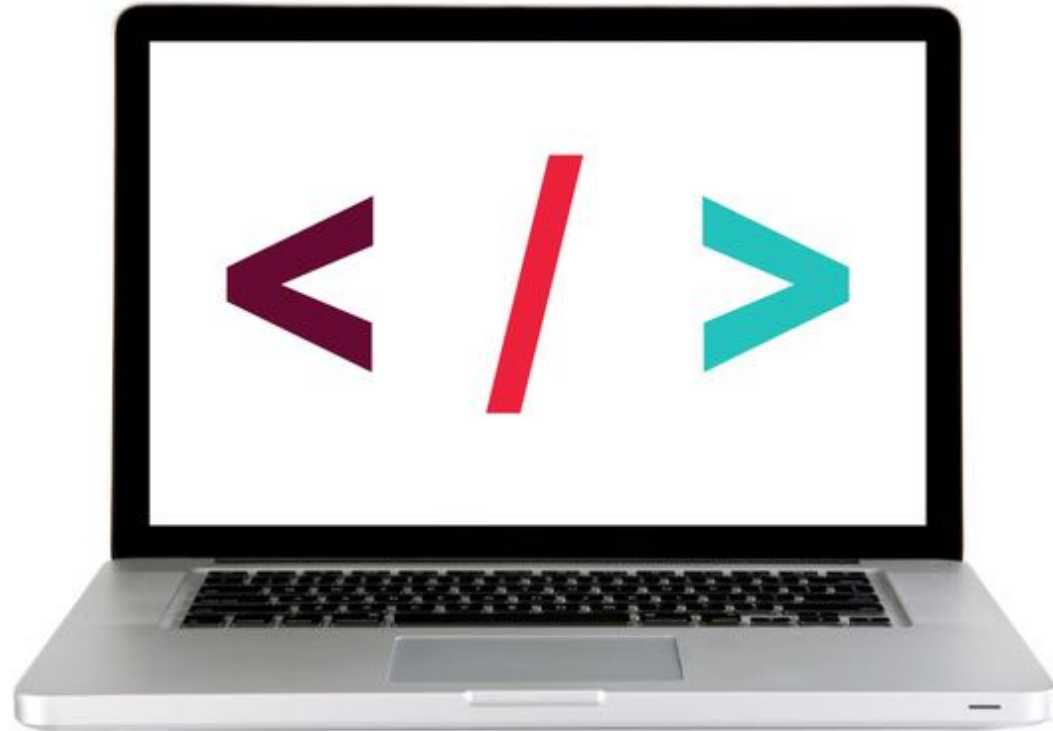
**display: block;**



---

## LET'S TAKE A CLOSER LOOK – PART 2

---



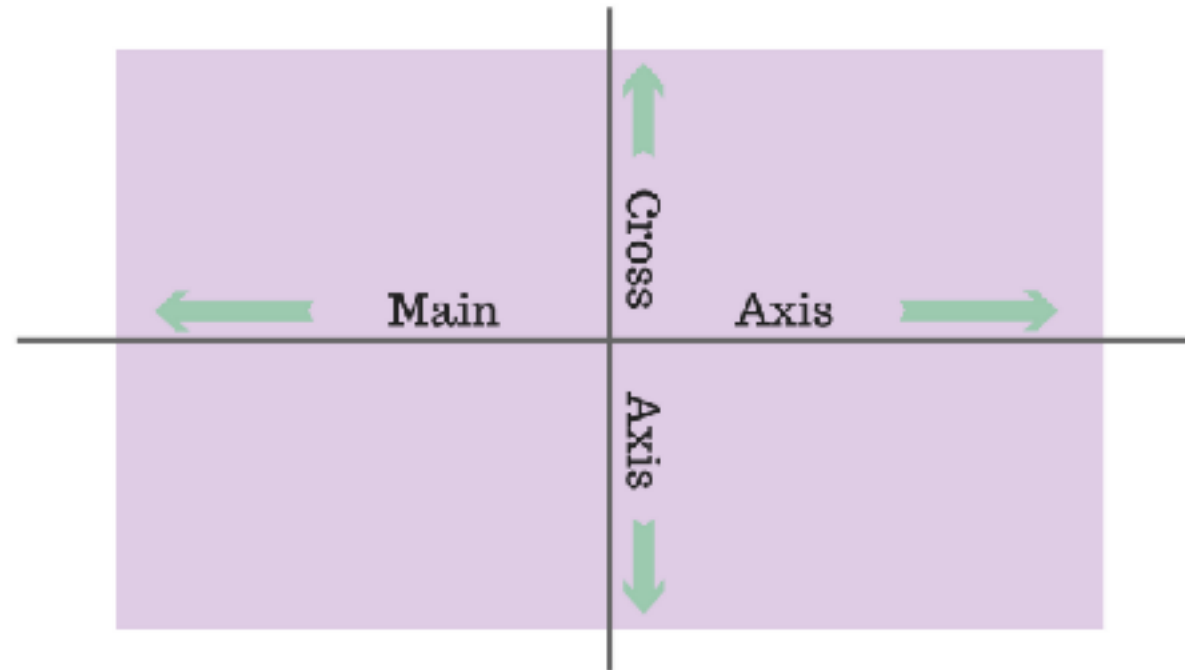
<http://codepen.io/sarahholden/pen/egPJKq>

---

## HOW DOES FLEXBOX WORK?

---

By default, items are arranged along the main axis, from left to right. This is why your squares defaulted to a horizontal line once you applied **display: flex**.



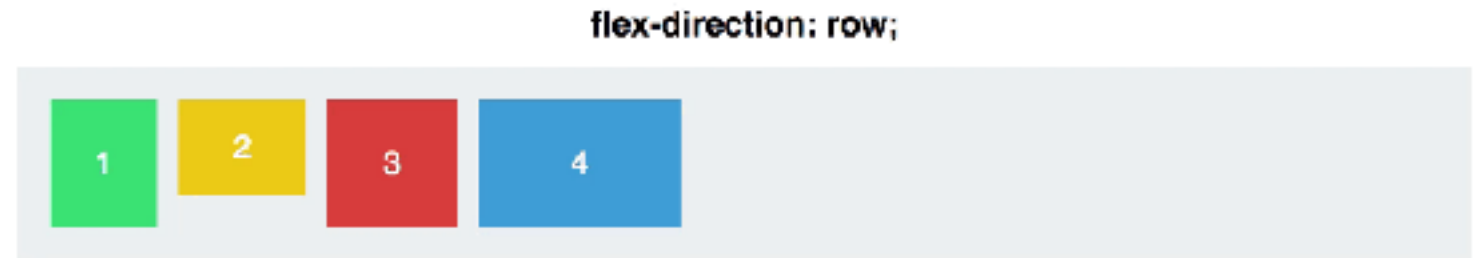
---

## FLEX-DIRECTION

---

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

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



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

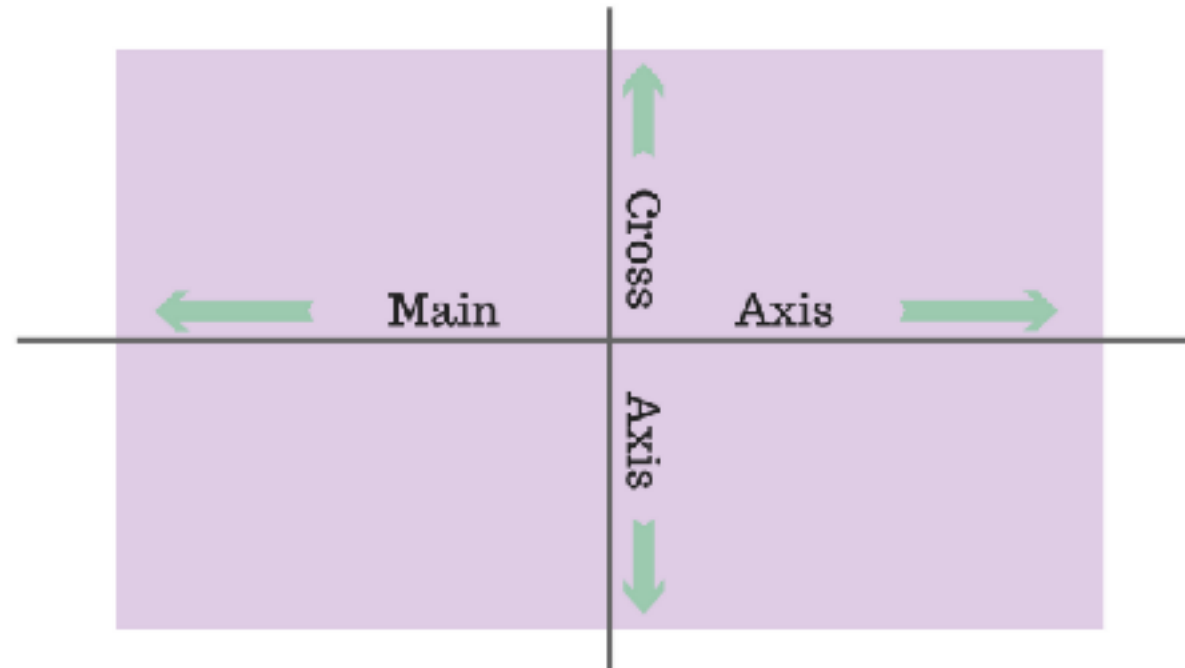
There's an important distinction to make here: **flex-direction: column** doesn't align the squares on the cross axis instead of the main axis. It makes the *main axis itself* go from horizontal to vertical.

---

## JUSTIFY-CONTENT

---

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



---

# 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;  
  flex-direction: row;  
  justify-content: flex-start;  
}
```

**justify-content: flex-start;**

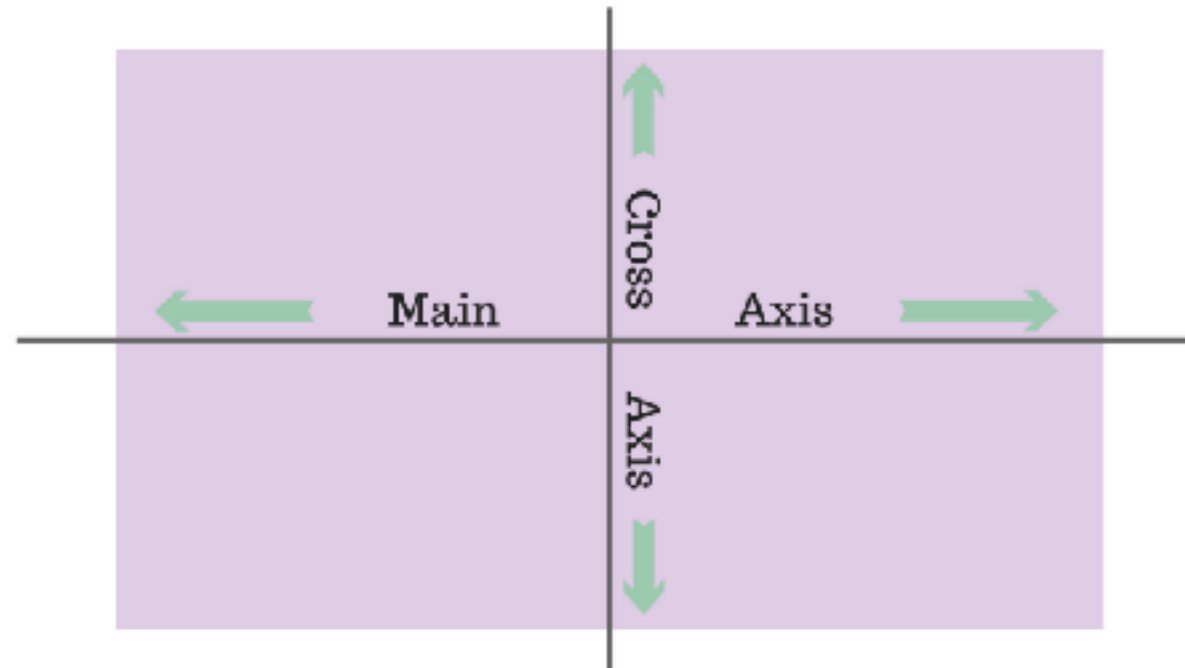


---

## ALIGN-CONTENT

---

**align-items** controls how you align items on the **cross axis**.





---

## ALIGN-ITEMS

---

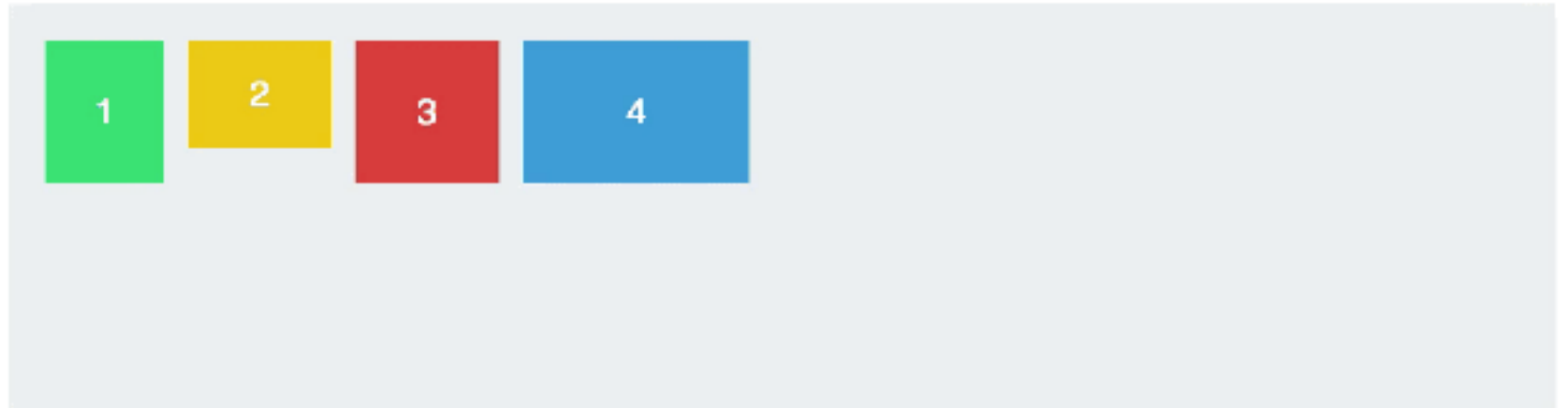
**align-items** controls how you align items on the **cross axis**.

You have five commands at your disposal to use align-items:

1. flex-start
2. flex-end
3. center
4. stretch
5. baseline

```
section {  
  display: flex;  
  flex-direction: row;  
  align-items: center;  
}
```

**align-items: flex-start;**

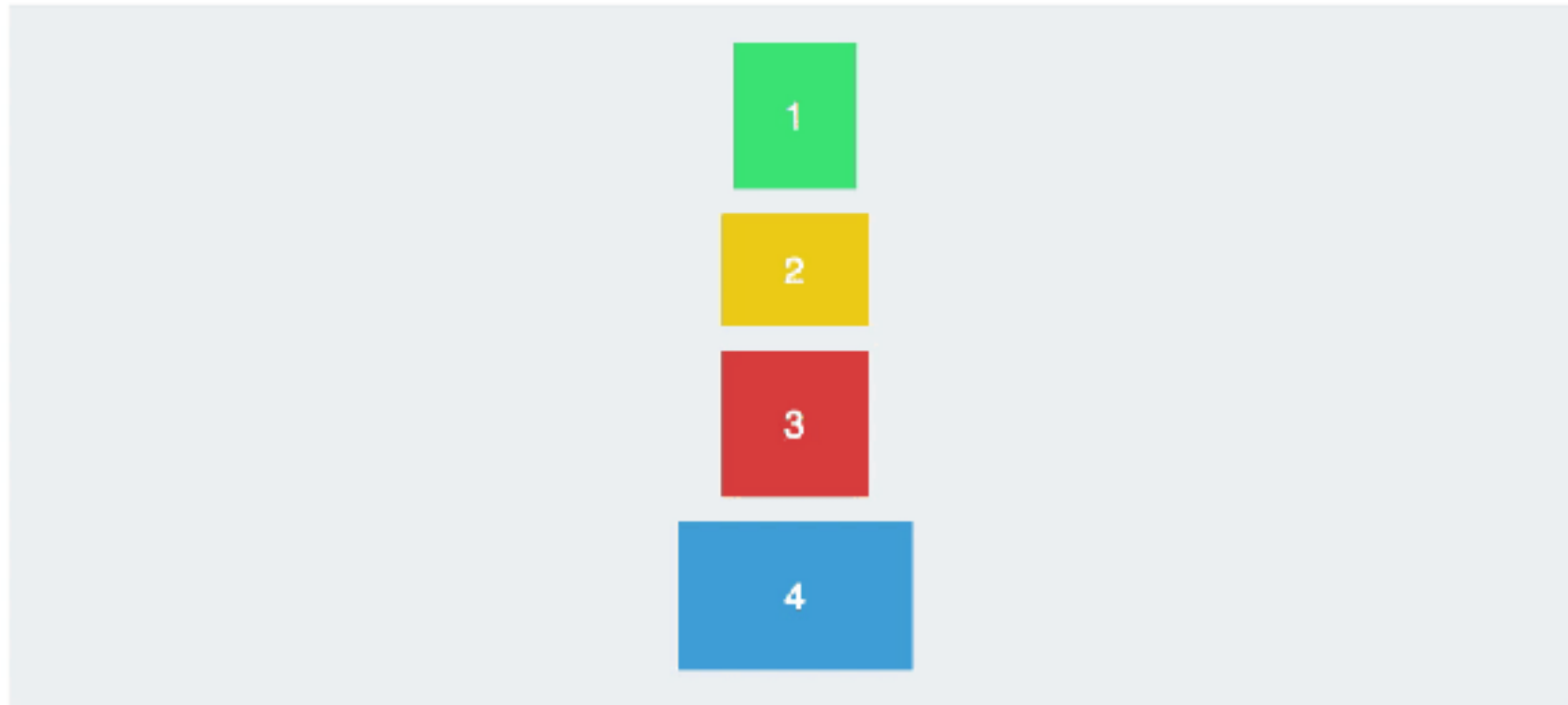


---

## ALL TOGETHER NOW!

---

**flex-direction: column;**  
**justify-content: center; align-items: center;**



---

**FEWD**

---

# HOLY GRAIL 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**

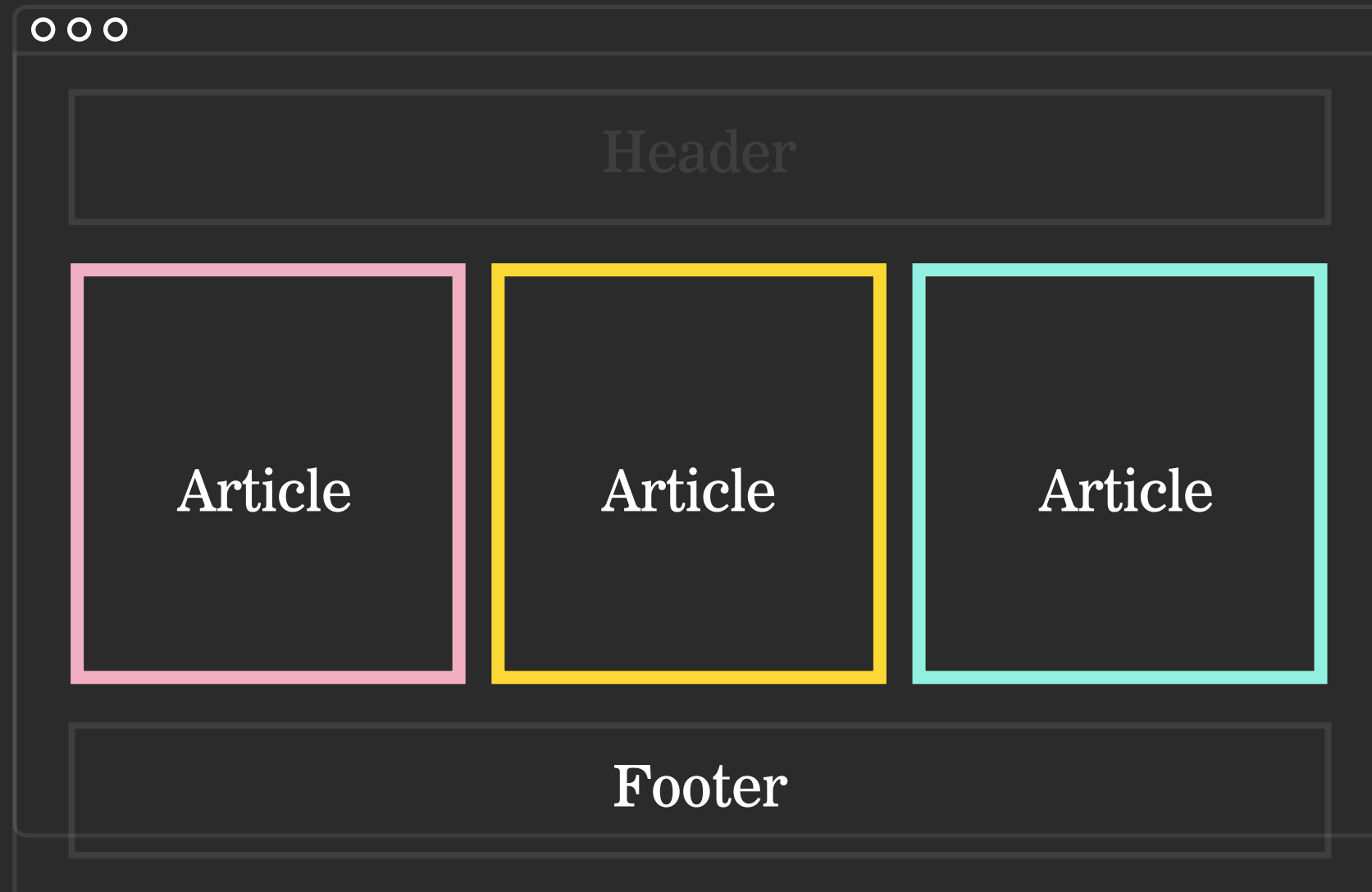
**4. GIVE EACH COLUMN A WIDTH IN PERCENTAGES**

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

---

## CSS — MULTI-COLUMN LAYOUT

---



---

## STEPS TO ACHIEVE A MULTI-COLUMN LAYOUT

---

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

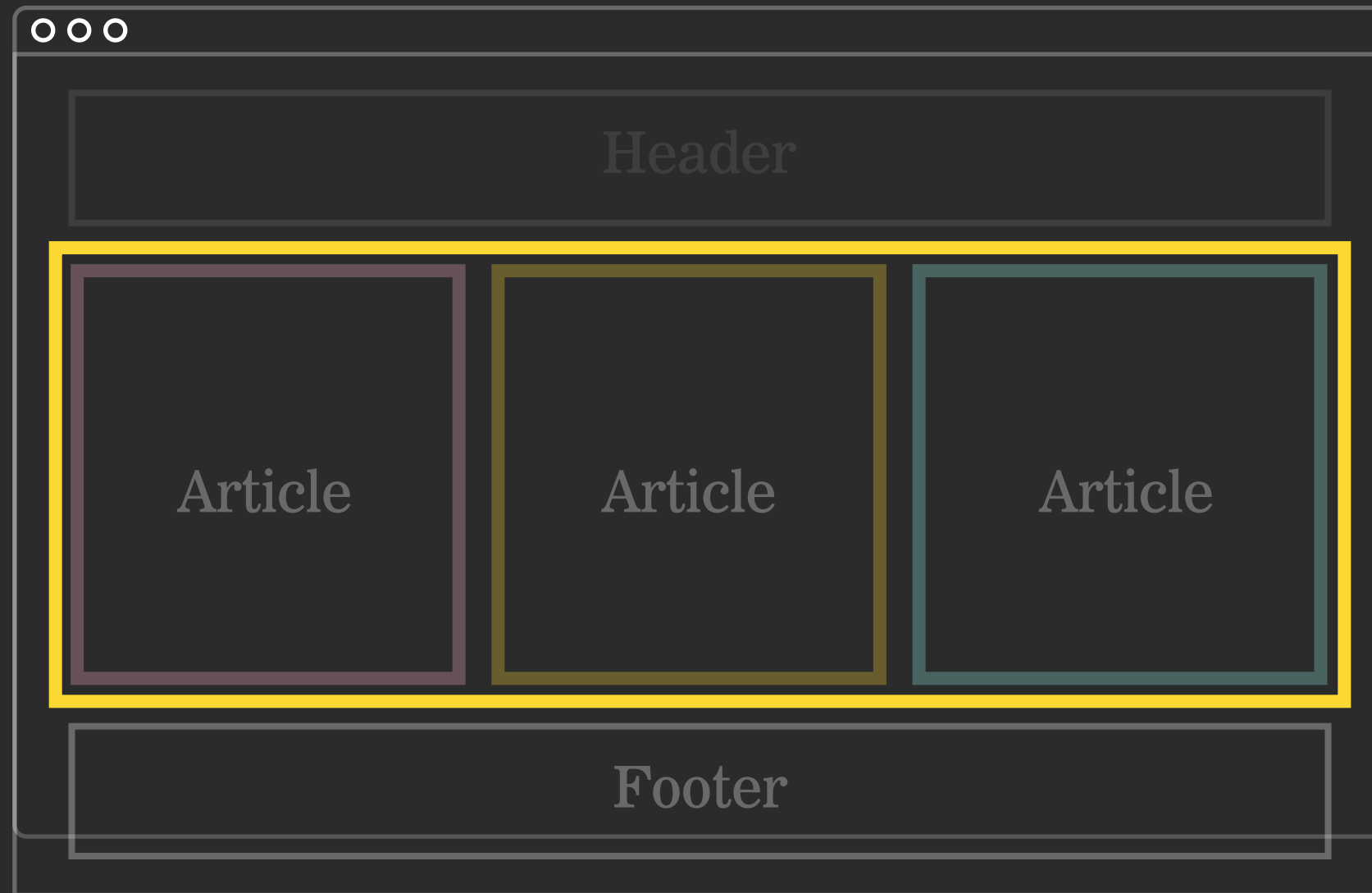
**2. WRAP THE COLUMNS WITH A PARENT CONTAINER, THE "FLEX CONTAINER"**

**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



---

## 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**



# HOW DOES FLEXBOX WORK?

justify-content: flex-start;



justify-content: flex-end;



justify-content: center;



justify-content: space-between;



justify-content: space-around;



# LET'S TAKE A CLOSER LOOK

## BEATLES — DRIVE MY CAR

### LYRICS

Asked a girl what she wanted to be  
She said baby, "Can't you see  
I wanna be famous, enter on the screen  
But you can do something in between"

Baby you can drive my car  
Yes I'm gonna be a star  
Baby you can drive my car  
And maybe I love you

I told a girl that my prospects were good  
And she said baby, "It's understood  
Working for peanuts is all very fine  
But I can show you a better time"

I told that girl I can start right away  
And she said, "Listen baby I got something to say  
I got no car and it's breaking my heart  
But I've found a driver and that's a start"

Baby you can drive my car  
Yes I'm gonna be a star  
Baby you can drive my car  
And maybe I love you  
Beep beep'm beep beep yeah  
Beep beep'm beep beep yeah  
Beep beep'm beep beep yeah  
Beep beep'm beep beep yeah(fade out)

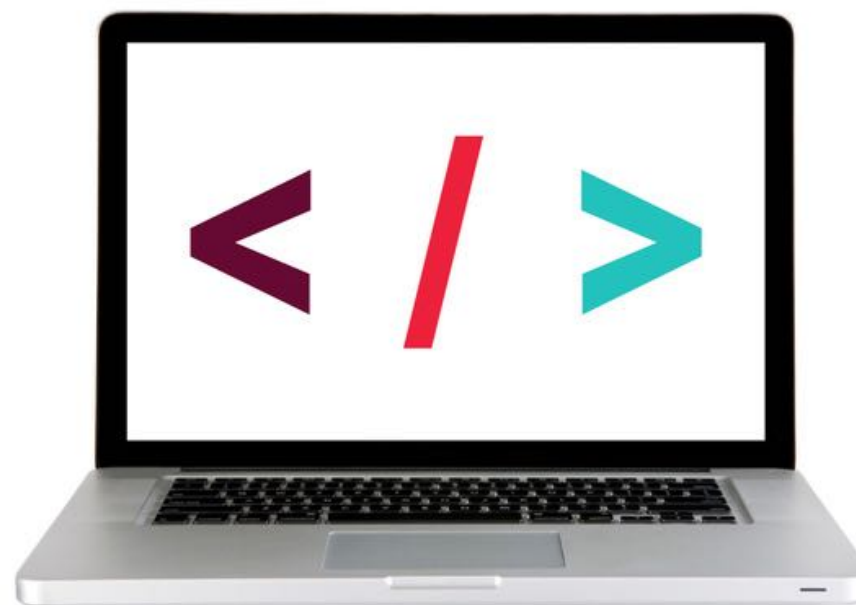
### ALBUM INFO



Rubber Soul is the sixth studio album by English rock band the Beatles, released on 3 December 1965.

- Album: Rubber Soul
- Release Date: December 3, 1965

COPYRIGHT © 2016 LYRICMANIA BABY



---

## ACTIVITY — FLOATS LAB

---



### EXERCISE

#### KEY OBJECTIVE

---

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

#### LOCATION OF FILES

---

- ▶ starter code > multi\_columns\_lab folder

#### TIMING

---

*2 min*

1. Follow steps in style.css

---

**FEWD**

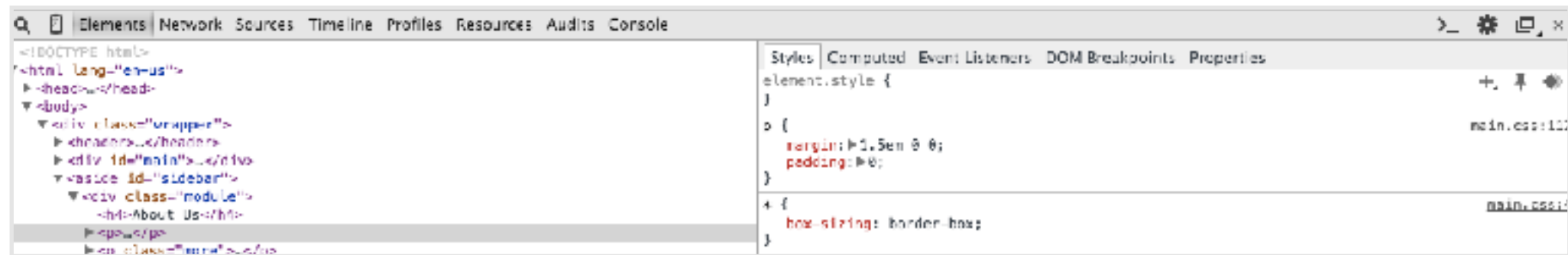
---

# **MONUMENT MAGAZINE PART 2**

---

## LAB — WORKFLOW

---

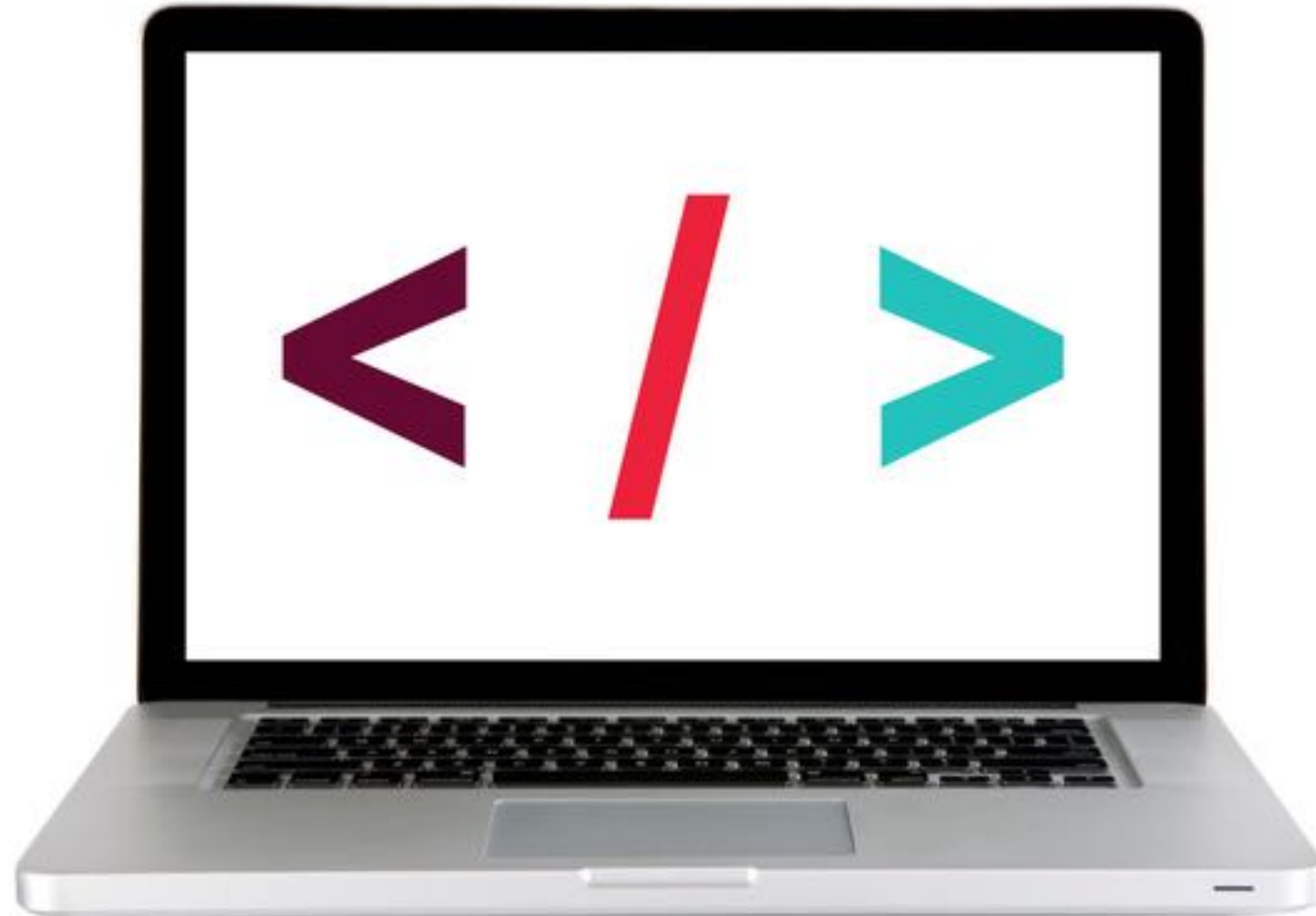


Right click > Inspect Element

---

## LAB — WORKFLOW — EVALUATING A DESIGN

---



---

## LAB — MONUMENT MAGAZINE PT. 2

---



---

# ACTIVITY — MONUMENT MAGAZINE

---



## EXERCISE

### KEY OBJECTIVE

---

- Demonstrate the ability to plan and build a website

### TYPE OF EXERCISE

---

- Partner

### TIMING

---

*40 min*

1. Recreate the Monument Magazine site, using the supplied png as a reference (in starter\_code folder)
2. Focus on setting up your columns first!
3. Then focus on other styles.



# LEARNING OBJECTIVES

- Differentiate between block and inline elements
- Use HTML5 structural elements to properly organize your page.
- Discover how to create multi-column layouts with flexbox

---

## WEEKLY OVERVIEW

---

**WEEK 1**

HTML Basics / CSS Basics

**WEEK 2**

CSS Selectors / Layout

**WEEK 3**

Lab / Refactor & Lab

---

**LAYOUT**

---

# **HOMEWORK**

---

## **HOMEWORK**

---

### **HOMEWORK:**

Be sure to read the specs on the FEWD website.

---

**HTML BASICS**

---

**EXIT TICKETS**