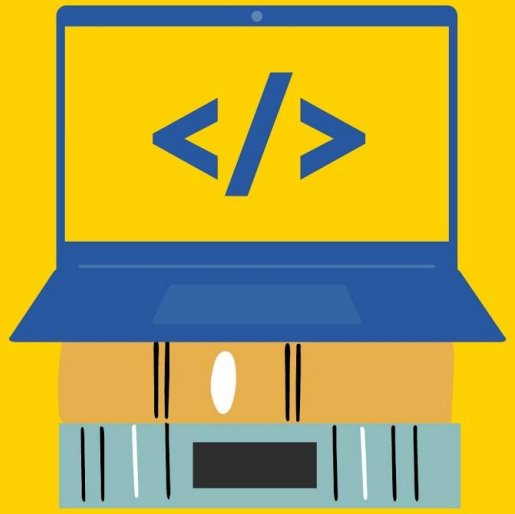


Front End Development

Session 8

Getting Loopy With ***JavaScript***


WELCOME

Learning Goals

In today's session, we will:

01 Explain how to apply a JavaScript ***for*** loop

02 Practice applying **control flow** and *conditional logic* in JavaScript

03 Define the purpose of the **DOM**

04 Demonstrate how to use JavaScript to manipulate the **DOM**

Let’s review

Q. What kind of code are we learning in this course?

Q. Can you name a role, job, or industry that uses this kind of code?

Q. What three “languages” do we use to create websites?

Q. What does "HTML" stand for?

Q. How can we examine website code on our computers?

Q. In HTML, what's the difference between the <head> and the <body>?

What is the difference between Git and Github?

What does the CLI stand for? What would we use it for?

How do you format a link in HTML?

How do we save our work with Git and Github?

What does a tag do?

What does CSS stand for? Why do we use CSS?

What is the difference between *block* and *inline*?

What does “float” do?

What are the five positioning properties?

What is the difference between *block* and *inline*?

What does “float” do?

What are the five positioning properties?

What is a framework?

What does “open-source” mean?

What is responsive design?

What is a variable? How do we declare a variable in JS?

What is a data type? What JS data types have we learned so far?

What does state refer to? Why is this useful?

What’s the difference between alerts, console.log, and document.write()?

What is a browser event?

How do we generate a random number in JavaScript?

What function do we append to round up or round down?

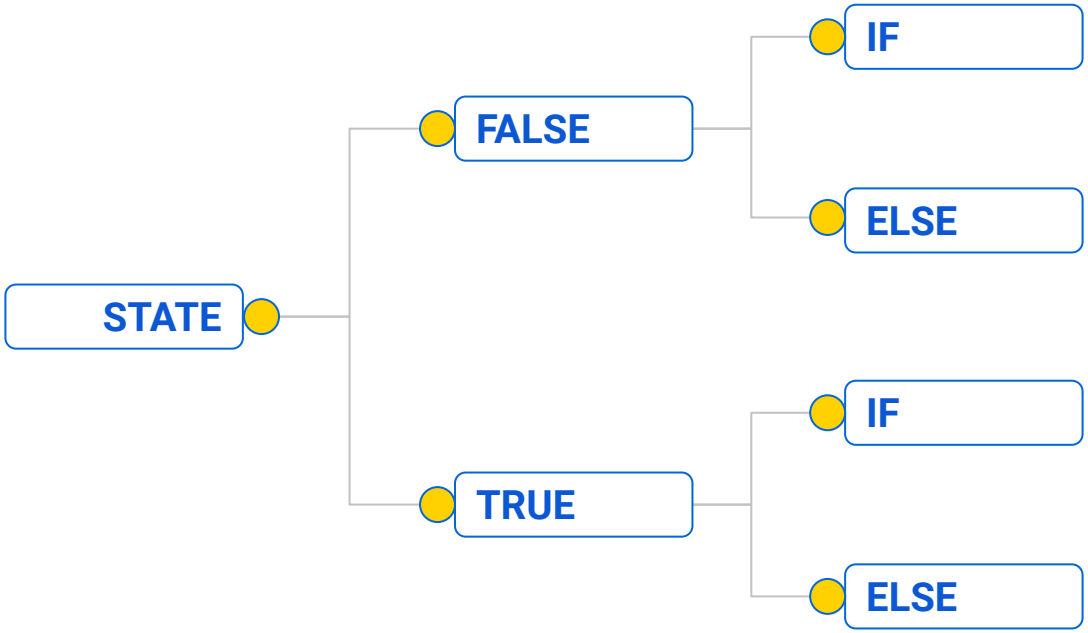
What is pseudocode and why is it useful?

For Loops

For loops allow a code operation to be executed repeatedly

Control Flow

Loops are a type of **control flow**, similar to conditional (IF/ELSE) statements.



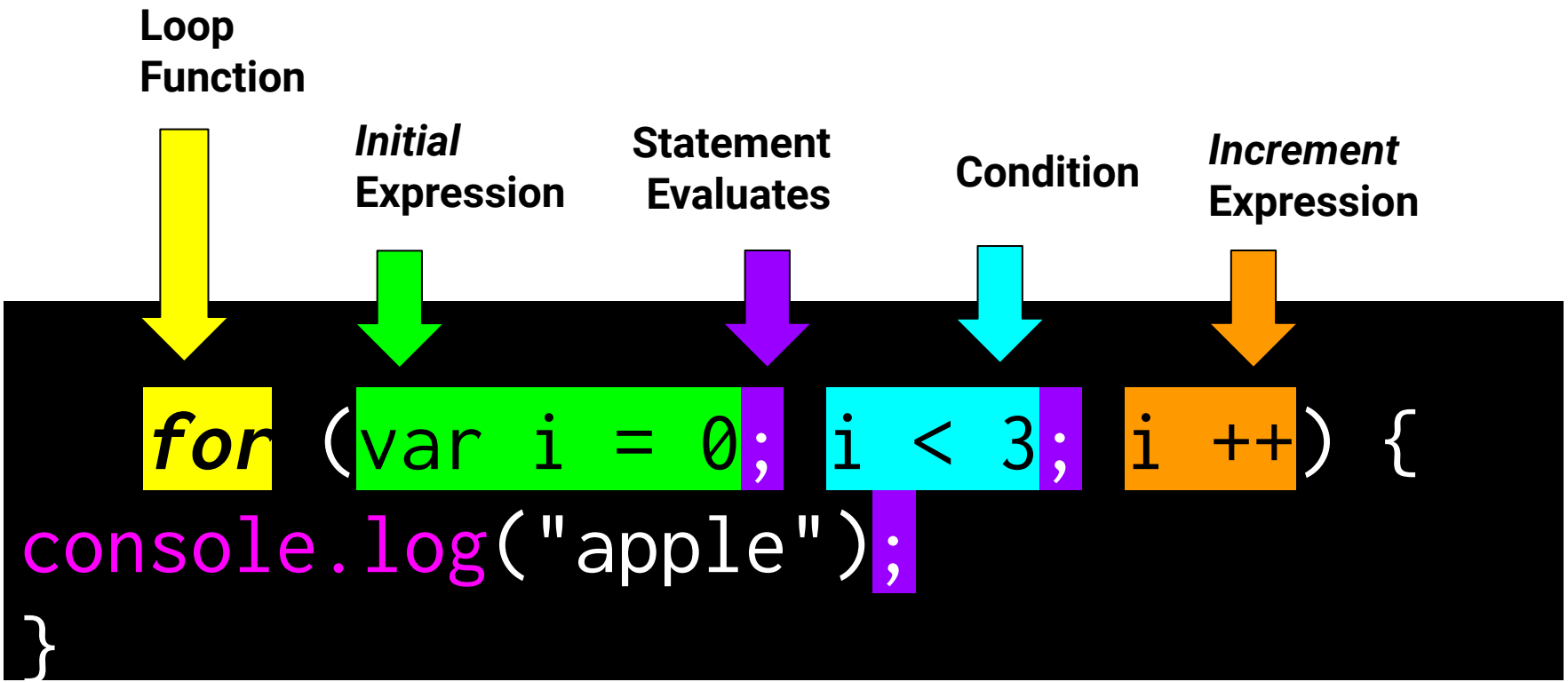
Control flow is the *order* in which a computer executes statements in a programming file.

For Loop Demo

Here's how we would write a **For** loop in JavaScript:

```
for (var i = 0; i < 3; i ++) {
console.log("apple");
}
```

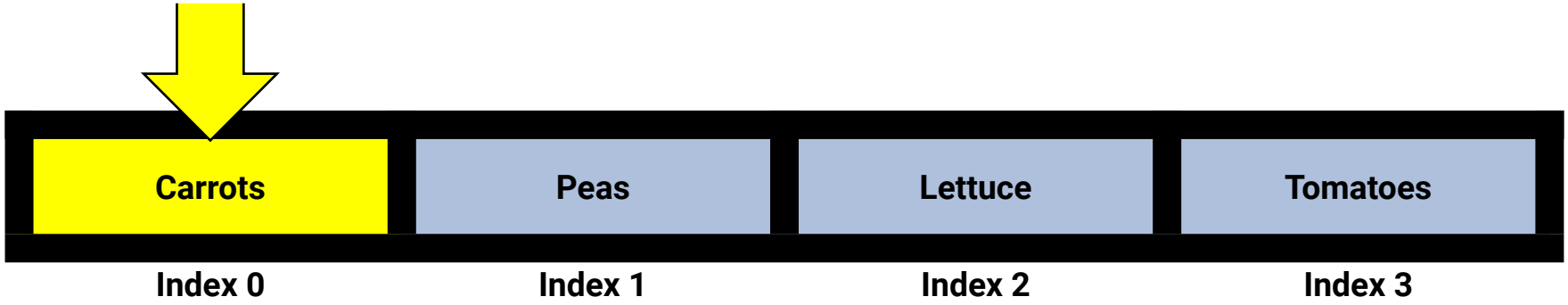
For Loop Structure



A for loop will run the code within it a given number of times until a specified condition is met.

Run That Loop!

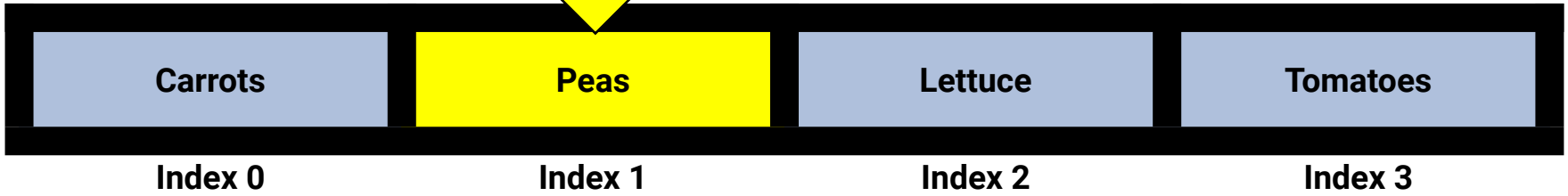
```
//Start with an Array.

var vegetables = ["Carrots", "Peas", "Lettuce", "Tomatoes];
//Loops through each index of the Array.
for (var i = 0; i < vegetables.length; i++) {
console.log("I love" + vegetables[i];
```



Run That Loop

```
//Start with an Array.

var vegetables = ["Carrots", "Peas", "Lettuce", "Tomatoes];
//Loops through each index of the Array.
for (var i = 1; i < vegetables.length; i++) {
console.log("I love" + vegetables[i];
```

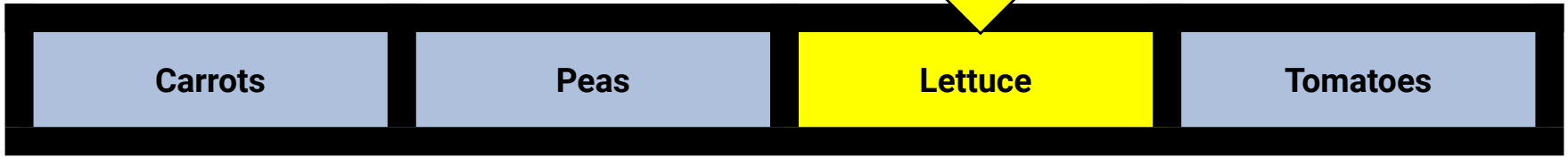
When i = 1 … console.log("I love Peas")



Run That Loop

```
//Start with an Array.

var vegetables = ["Carrots", "Peas", "Lettuce", "Tomatoes];
//Loops through each index of the Array.
for (var i = 2; i < vegetables.length; i++) {
console.log("I love" + vegetables[i];
```

When i = 2 … `console.log`(“I love Lettuce”)

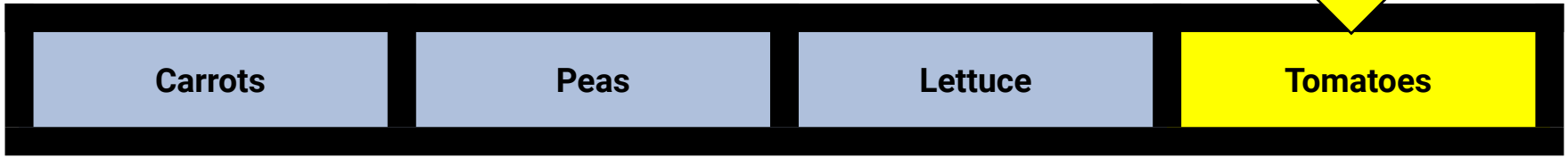


Index 0 **Index 1** **Index 2** **Index 3**

Run That Loop

```
//Start with an Array.

var vegetables = ["Carrots", "Peas", "Lettuce", "Tomatoes];
//Loops through each index of the Array.
for (var i = 3; i < vegetables.length; i++) {
console.log("I love" + vegetables[i];
```

When i = 3 … `console.log`(“I love Tomatoes”)



Index 0 **Index 1** **Index 2** **Index 3**

Instructor Demonstration:

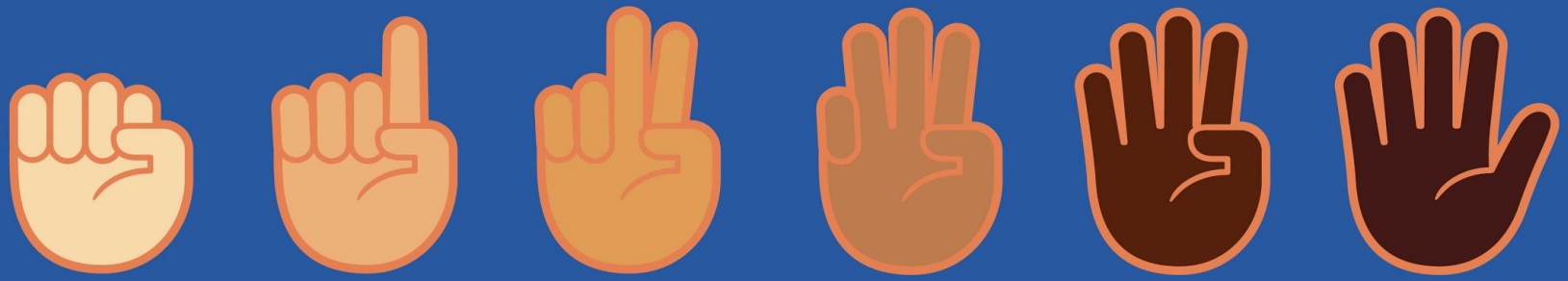
For Loops

Suggested Time:
5 minutes

Instructor Demonstration:

MyFirstLoop

Suggested Time:
5 minutes



Fist to five

Let’s

Activity: Zoo Loop

1. Open up instructions in Canvas
2. Code out your statements
3. Share with a partner

Suggested Time:
15 minutes

Let’s review

Questions?

Let’s

Activity: Conditional Looping

1. Open up instructions in Canvas
2. Pair up with a partner
3. Draft your code!

Suggested Time:
20 minutes

Let’s review

Questions?

Revisiting Arrays

Arrays let us store multiple values within a single variable (at the same time).

Single Variables vs *Arrays*

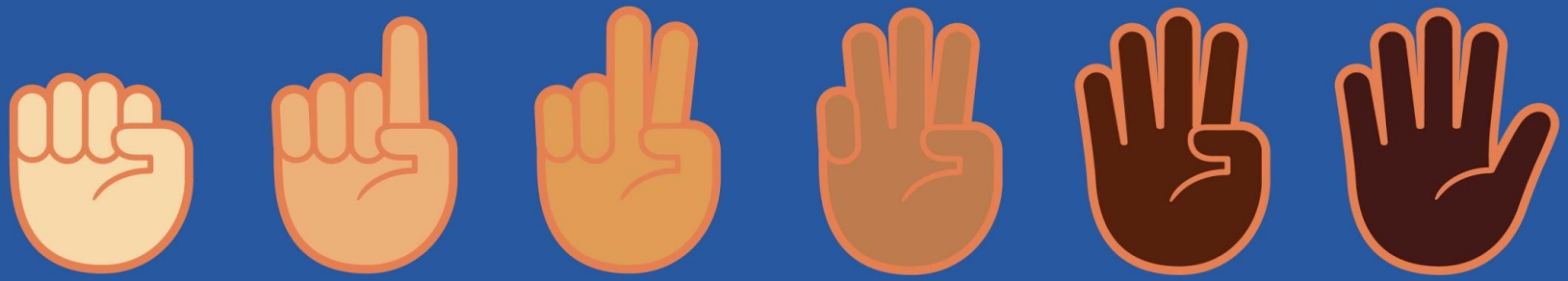
Arrays let us assign *multiple* values to the same variable!

```
var car = “Volvo”;

var cars = [“Saab”, “Volvo”, “BMW”];
// arrays let us store additional data within our variables
```

Accessing Data From An Array

Array data can be accessed with an **index**, but here's a quirky thing: the count starts at *zero*!

```
var cars = ["Saab", "Volvo", "BMW"];
console.log(cars[2])

// This will return our third entry
```



Fist to five

Let’s

Activity: TV Array Building

1. Open instructions in Canvas
2. Run the code for this activity
3. Pair with a partner and fill in the missing code!

Suggested Time:
10 minutes

Let’s review

Questions?

Practice
Makes
Perfect!

Let’s

Activity: Movie Ratings

1. Open instructions in Canvas
2. Pair up with a partner
3. Follow instructions and work on solving each prompt
4. Try to complete as many as possible!

Suggested Time:
30 minutes

Let’s review

Instructor Demonstration:

Movie Ratings Review

Suggested Time:
5 minutes

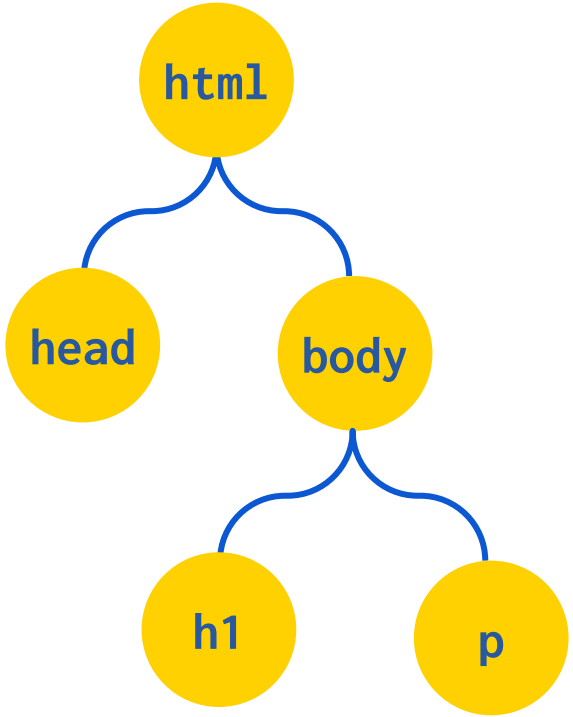
Questions?

Introducing the DOM

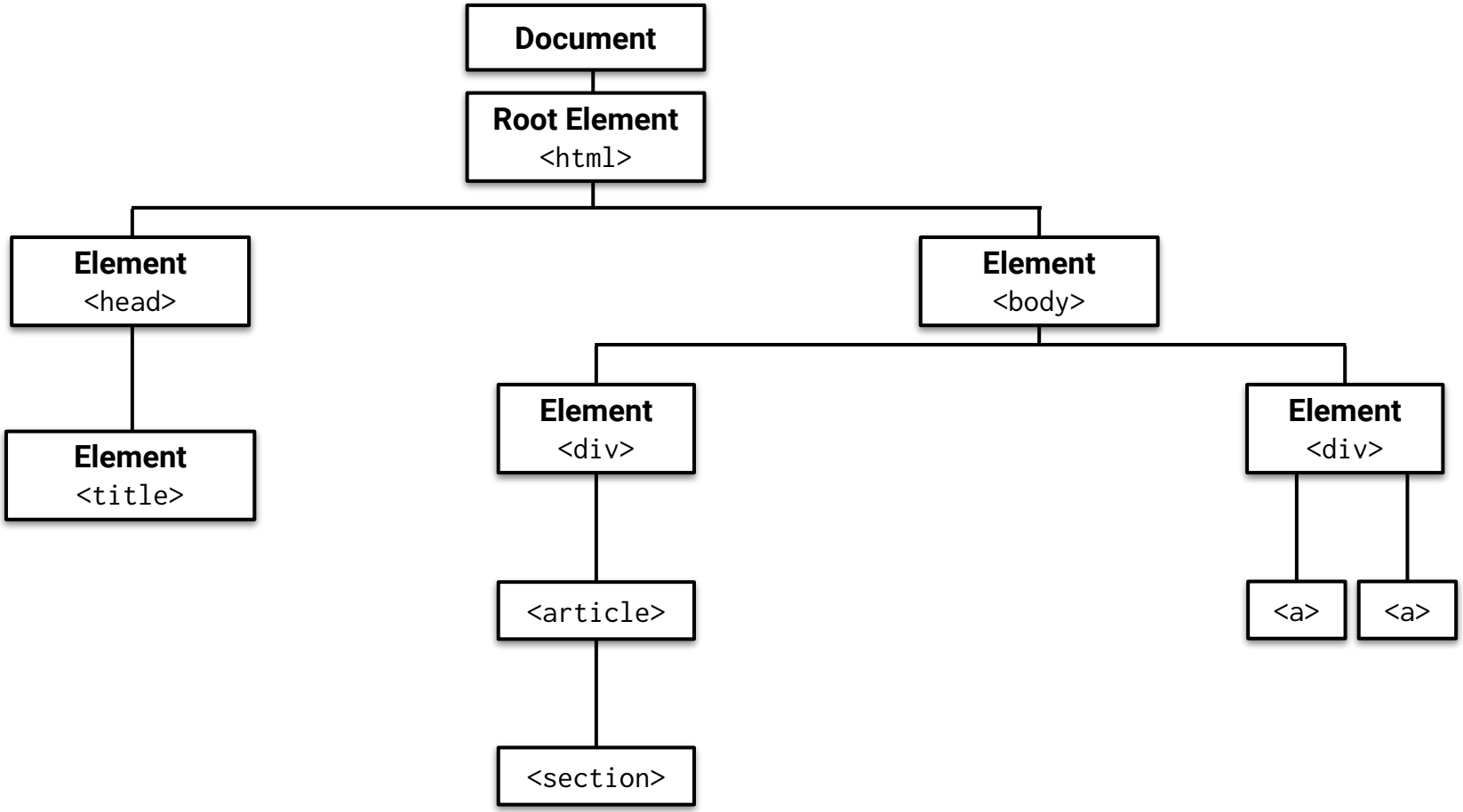
The **DOM** is an *object-oriented* representation of HTML

Remember HTML?

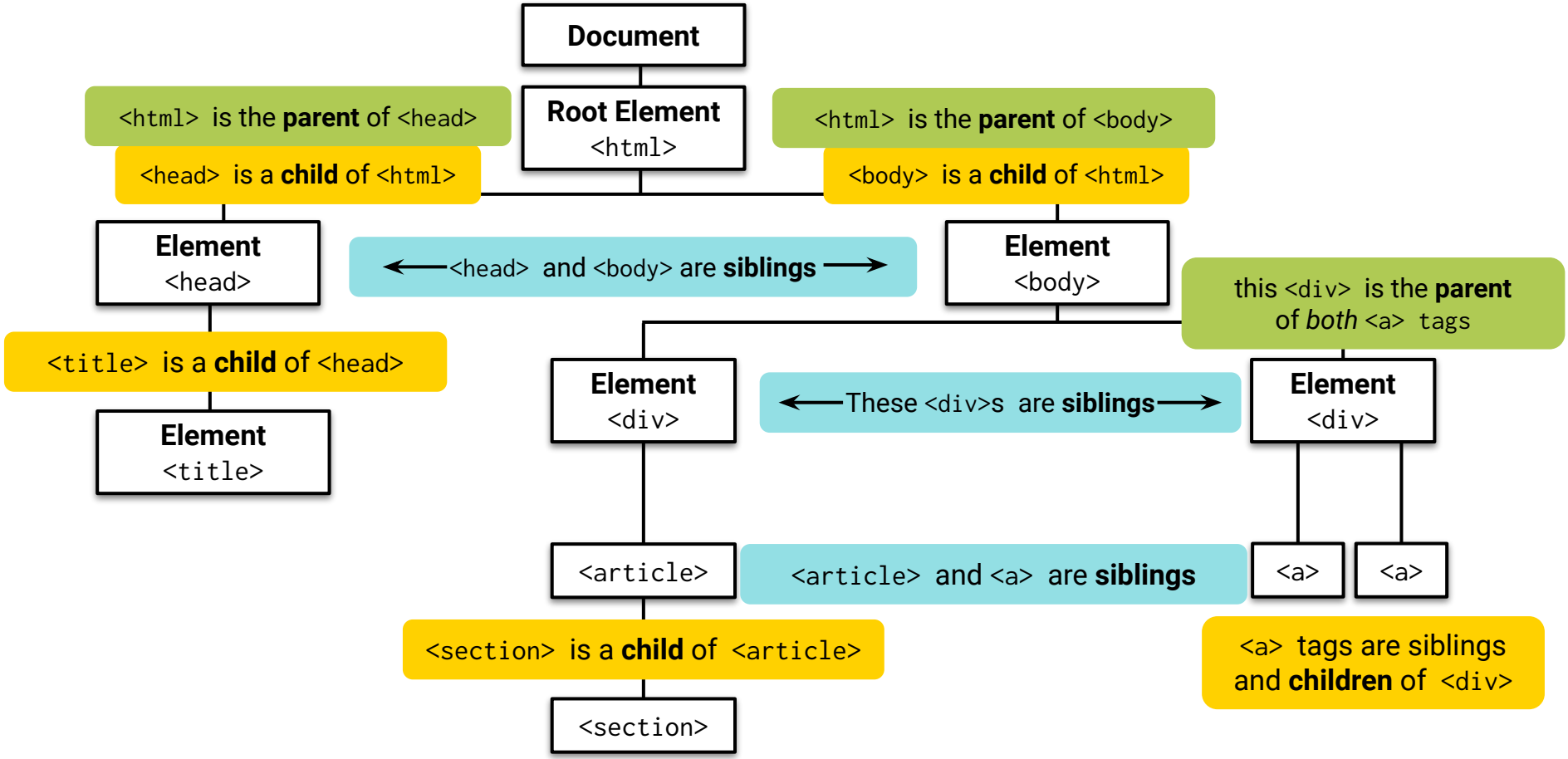
HTML elements have **parent-child** relationships



Parent, Child, and Sibling Relationships



Parent, Child, and Sibling Relationships

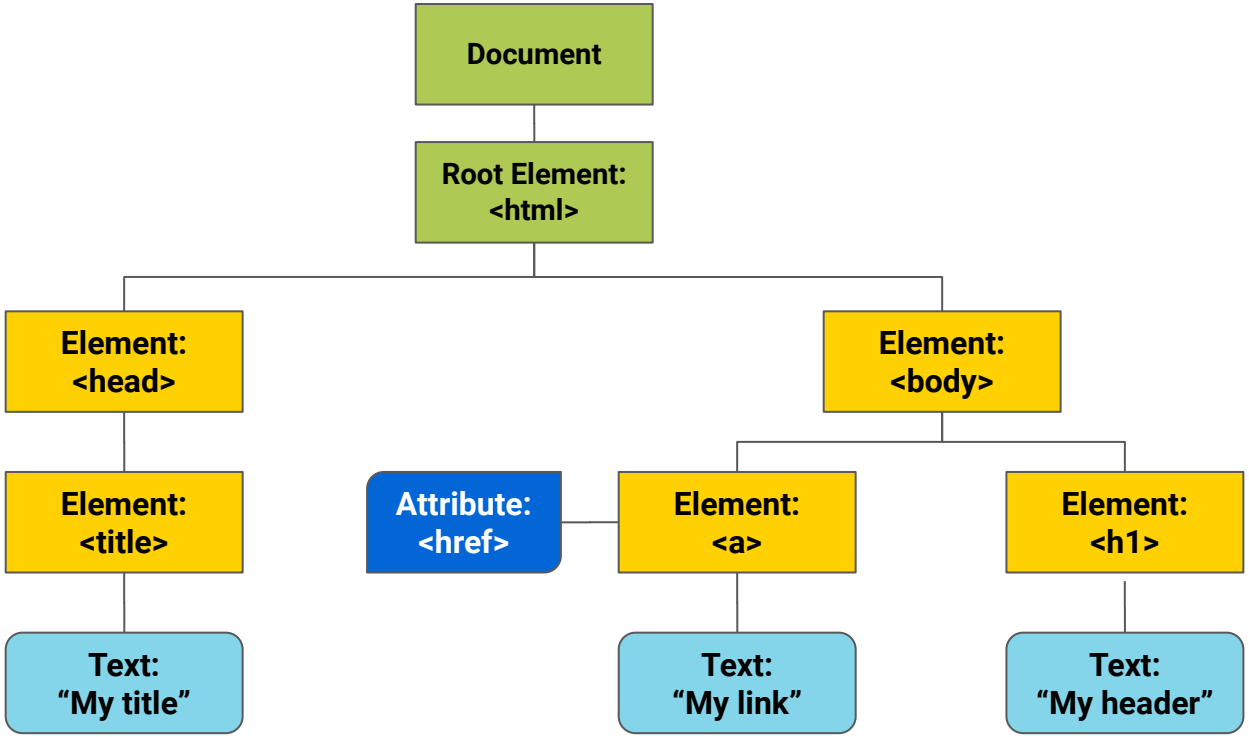


The Document Object Model

The **DOM** is a representation of HTML modeled as *JavaScript objects*:

Each element is a ***node***.

Nodes are organized in a ***tree***.



Example HTML DOM

```
<!DOCTYPE html>
<html lang="en">
   <head>
      <title>Document</title>
   </head>
   <body>
      <div>Main div
        <article>
           <section>

           </section>
        </article>
      </div>
      <div>
        <a href="myImg"></a>
        <a href="secondImg"></a>
      </div>
   </body>
</html>
```

Instructor Demonstration:

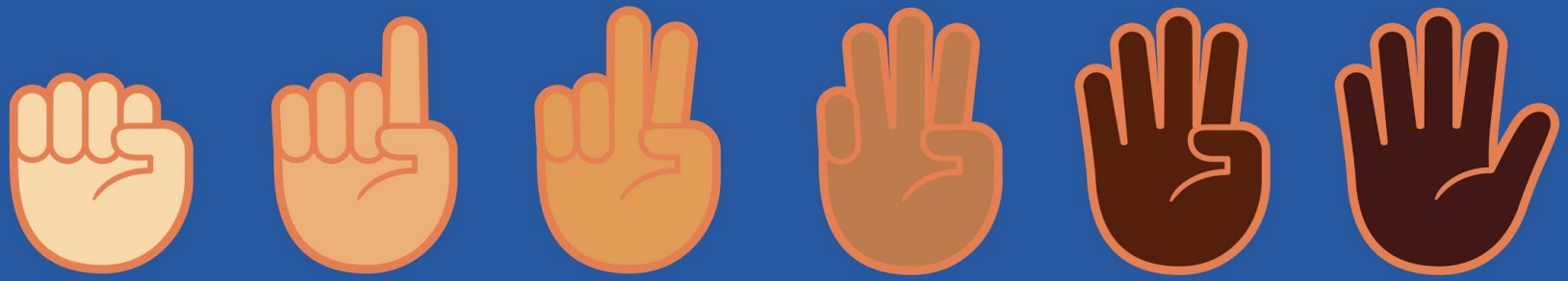
To Do List

Suggested Time:
5 minutes

Instructor Demonstration:

DOM Manipulation

Suggested Time:
10 minutes



Fist to five

Let’s

Activity: Generating HTML with Plain JS

1. Pair up and open `drinklist-unsolved.html`
2. Take turns adding in the missing code, so that your JavaScript generates HTML content that displays all of the drink options for the coffee shop!
3. Make sure you can both explain how the code works.

Suggested Time:
25 minutes

Let’s review

Time to Recap

Learning Goals

Our objectives for today's session:

01 Explain how to apply a JavaScript ***for*** loop

02 Practice applying **control flow** and *conditional logic* in JavaScript

03 Define the purpose of the **DOM**

04 Demonstrate how to use JavaScript to manipulate the **DOM**

Reflection

What was your favorite part of today's session?

What was the most interesting thing we covered today?

What do you still have questions about?

Sneak Preview

Tomorrow we'll work in groups to code another fully functioning game in JavaScript.

Afterward, you'll have a second sample application to add to your portfolios!

Questions?

The End