



# Jumping for JavaScript

Web Development Boot Camp  
Lesson 3.2



# Today's Class

# Announcements

---

- Algorithms
- Pre-class activities
- Holiday Schedule



# Objectives

---

In today's class, we'll cover:



Array assignments



The concept of `for` loops



The art of pseudocoding



Building Rock-Paper-Scissors

# Basics Recap

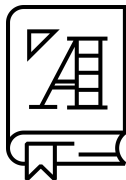


# **What is JavaScript?**

And what is it used for?

# JavaScript Definitions

---



JavaScript is the third of the three fundamental programming languages of the modern web (along with HTML and CSS).



JavaScript allows developers to create dynamic web applications capable of taking in user inputs, changing what's displayed to users, animating elements, and much more.



**What is a variable?**  
And how do we declare one?



# Variable Basics

---



**Variables** are the “nouns” of programming.



They are “things” (numbers, strings, Booleans, etc.).



A variable is composed of a variable name and a value.

```
var name = "Snow White";  
var dwarfCount = 7;  
var isSleeping = true;
```



**What is meant by `console.log`?**

And how does it differ from an alert, prompt, or confirm?

# Console.log vs. JavaScript popup boxes



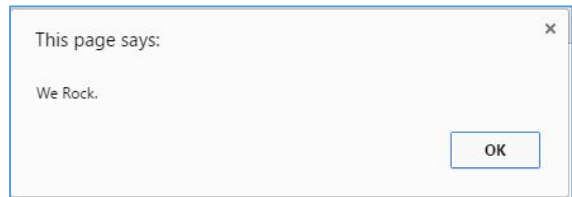
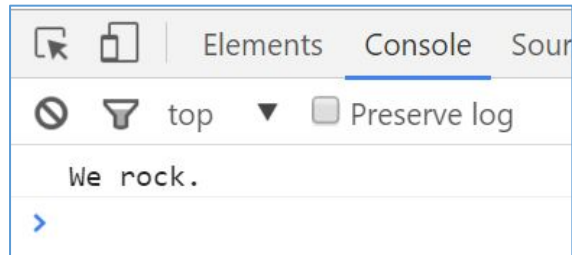
`console.log` displays discreetly to the debugger.



`alert` displays a pop-up message to the user.

```
console.log("We rock.");
```

```
alert("We Rock.");
```



# Console.log vs. JavaScript popup boxes



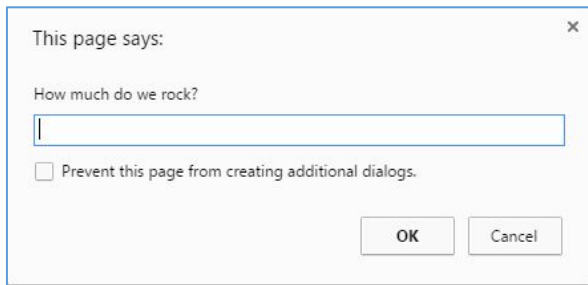
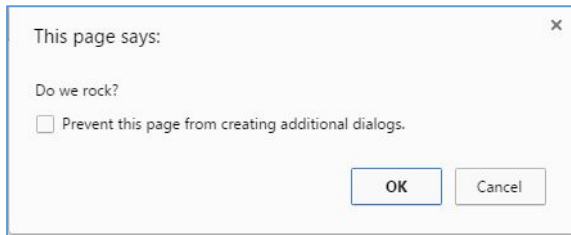
`confirm` displays a true/false popup.



`prompt` displays a popup with a text-box input.

```
confirm("Do we rock?");
```

```
prompt("How much do we rock?");
```





How do we **check conditions**?

# If/Else Statements Are Critical

---

Each statement is composed of an if, else-if, or else (keyword), a condition, and the resulting code in { } curly brackets.

```
// If the user likes sushi (confirmSushi === true), we run the following block of code.  
if (confirmSushi) {  
    alert("You like " + sushiType + "!");  
}  
// If the user likes ginger tea (confirmGingerTea === true), we run the following block of code.  
else if (confirmGingerTea) {  
    alert("You like ginger tea!!");  
}  
// If neither of the previous condition were true, we run the following block of code.  
else {  
    alert("You don't like sushi or ginger tea.");  
}
```



What is an **array**?

# The Zoo Pen

---

**Array Name:** zooAnimals

**Zebra**

**Index 0**

**Rhino**

**Index 1**

**Giraffe**

**Index 2**

**Owl**

**Index 3**



# The Zoo Pen: Coded

---

**Array Name:** zooAnimals

Zebra

Index 0

Rhino

Index 1

Giraffe

Index 2

Owl

Index 3

Coded in JavaScript using an array:

```
// Our array of zoo animals.  
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];
```

# Arrays

---



Arrays are a type of variable that are *collections*.



These collections can be made up of strings, numbers, Booleans, other arrays, objects ... anything.



Each element of the array is marked by an index. Indexes always start with 0.

```
var nickCharacters = ["Tommy", "Doug", "Oblina"];
```

```
var diceNumbers = [1, 2, 3, 4, 5, 6];
```

```
var mixedArray = ["Zoo", 12, "Carrot", 3];
```

# Arrays: Indices

---



To recover the value at any specific index, include the name of the array with a square bracket `[]` and inside the bracket is the element's index.



You can easily grab the number of elements in the array using the method `array.length`.

```
// Our array of zoo animals.  
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];  
  
// Prints 4 to the console because there are 4 items in our zooAnimals array.  
console.log(zooAnimals.length);  
  
// Prints Rhino to the console. Remember, the first item in an array has an index position of 0!  
console.log(zooAnimals[1]);  
  
// Prints undefined...because the last index ("Owl") is 3.  
console.log(zooAnimals[4]);
```



# Instructor Demonstration

## Arrays



## **Activity:** Array Logging

**Suggested Time:**  
5 minutes



# Activity: Array Logging

---



Follow the instructions provided in the file to `console.log` each of the names in the `coolPeople` variable.



**Hint:** You should be repeating the same line six times.



Be prepared to share when time is up.

**Suggested Time:** 5 minutes





## **Activity:** Array Setting

**Suggested Time:**  
5 minutes



# Activity: Array Setting



Follow the instructions in the file provided to convert each item in the array to lowercase.



Make sure to only add in lines of code where instructed.



**Hint:** You will need to use the method `.toLowerCase()`. Research if you don't remember how to use it.



Be prepared to share when time is up.

Suggested Time: 5 minutes





# Back to The Zoo Pen

---

Array name: zooAnimals

Zebra

Index 0

Rhino

Index 1

Giraffe

Index 2

Owl

Index 3

Coded in JavaScript using an array

```
// Our array of zoo animals.  
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];
```

# Back to The Zoo Pen

Array name: zooAnimals

Zebra

Rhino

Giraffe

Owl

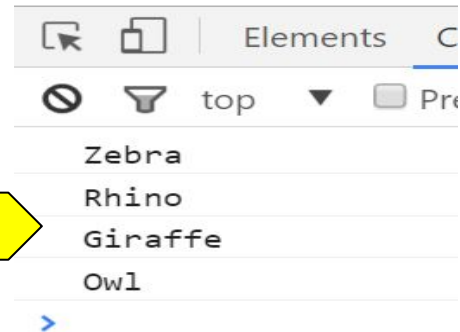
Index 0

Index 1

Index 2

Index 3

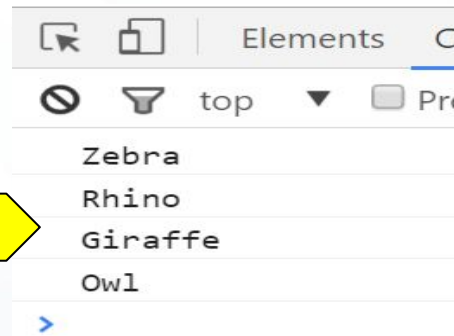
```
// Array of zoo animals.  
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];  
  
console.log(zooAnimals[0]);  
console.log(zooAnimals[1]);  
console.log(zooAnimals[2]);  
console.log(zooAnimals[3]);
```





## What's wrong here?

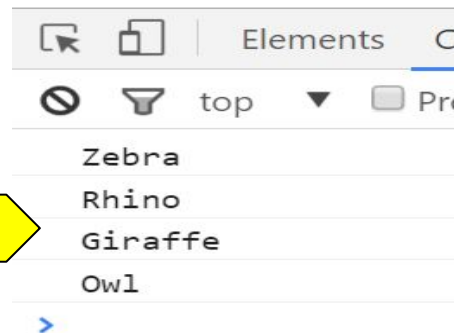
```
// Array of zoo animals.  
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];  
  
console.log(zooAnimals[0];  
console.log(zooAnimals[1];  
console.log(zooAnimals[2];  
console.log(zooAnimals[3];
```



# Don't Repeat Yourself (DRY)

**Repeated code!** Let's be more efficient.

```
// Array of zoo animals.  
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];  
  
console.log(zooAnimals[0];  
console.log(zooAnimals[1];  
console.log(zooAnimals[2];  
console.log(zooAnimals[3];
```





## **Activity:** Dissecting for Loops

**Suggested Time:**  
5 minutes



# Activity: Dissecting for Loops

---



With a partner, spend a few moments trying to dissect the code sent to you.



Try to explain to one another what is happening in each line of code.



Feel free to do research if you are stumped. As a **hint**, look into the phrase “for loop”.



Be prepared to share when time is up.

**Suggested Time:** 5 minutes



# for Loops

for loops are **critical** in programming. We use them to run **repeated blocks of code** over a set period.

Each for loop is composed of a:



Variable declaration or counter (iterator)



Loop condition



Iteration (addition)

```
// 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]);  
}
```

# for Loops

```
// 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]);  
}  
  
// Logs:  
// I love Carrots  
// I love Peas  
// I love Lettuce  
// I love Tomatoes
```



Iterator

Condition

Increment



# for Loops

Code between the `{ }` gets repeated each time the iterator is smaller than the condition (in this case, as long as `i < 4`).

```
// 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]);  
}  
  
// Logs:  
// I love Carrots  
// I love Peas  
// I love Lettuce  
// I love Tomatoes
```

# for Loops

---

Running the code “loops” through and prints each element in the array.

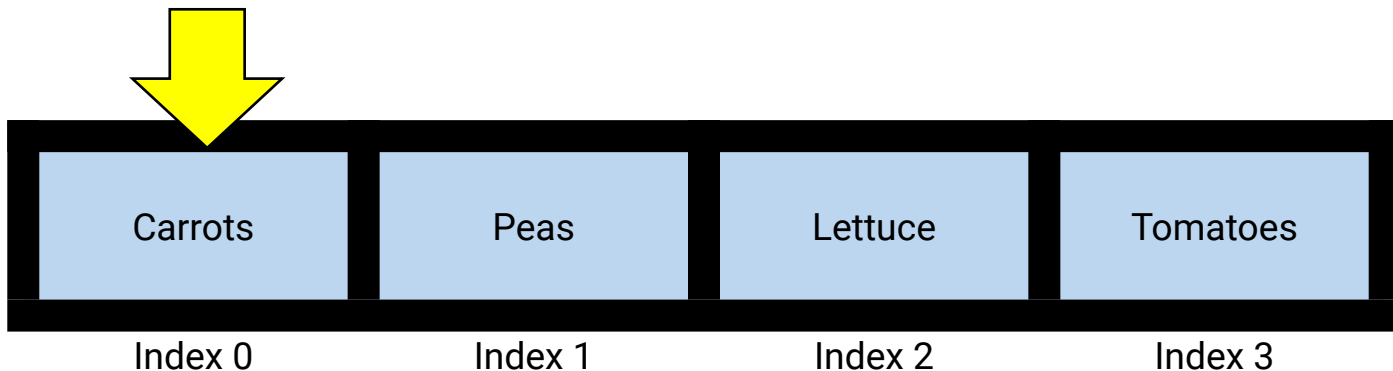
```
// Start with an Array.  
var vegetables = ["Carrots", "Peas", "Lettuce", "Tomatoes"];  
  
// Loops through each index of the Array.  
for (var i = 0; i < vegetables.length; i++) {  
    console.log("I love " + vegetables[i]);  
}
```

```
// Logs:  
// I love Carrots  
// I love Peas  
// I love Lettuce  
// I love Tomatoes
```

# 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]);  
}
```

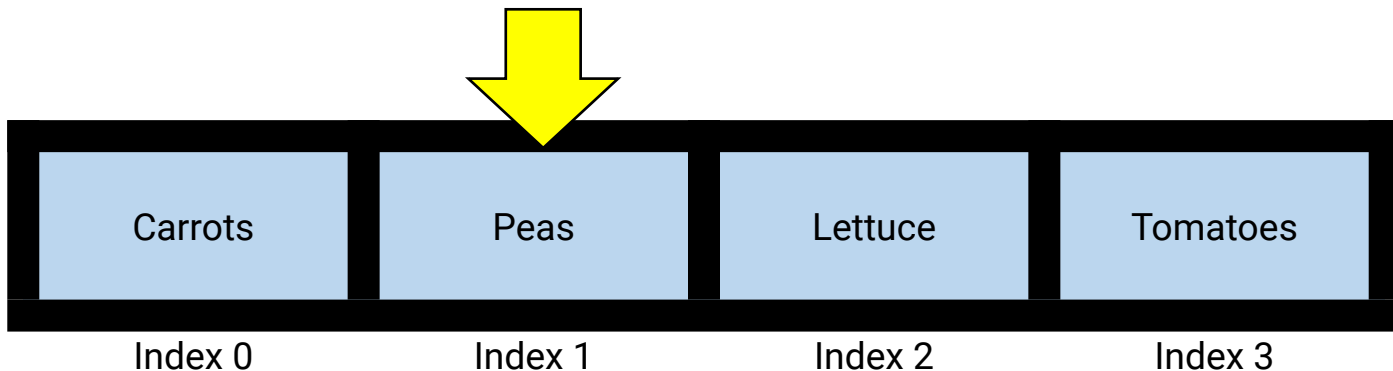
When  $i = 0$  ... `console.log("I love Carrots")`



# 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]);  
}
```

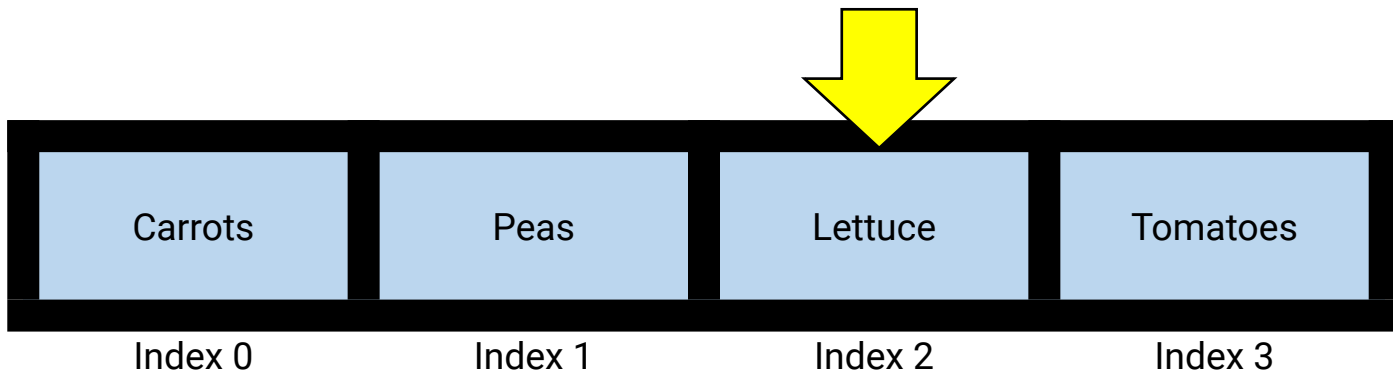
When  $i = 1$  ... `console.log("I love Peas")`



# Run That Loop

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

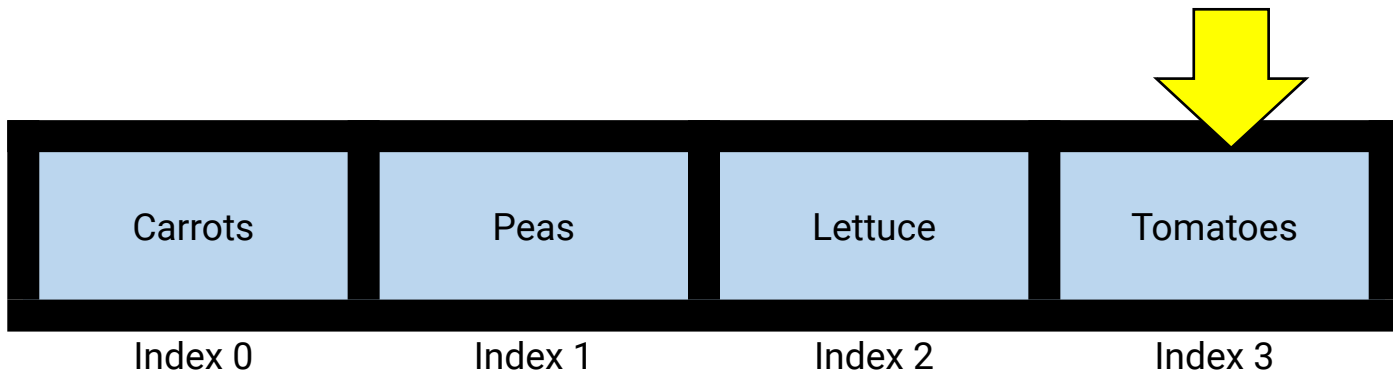
When  $i = 2$  ... `console.log("I love Lettuce")`



# 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]);  
}
```

When  $i = 3$  ... `console.log("I love Tomatoes")`





## **Activity:** for Loop Zoo

**Suggested Time:**  
15 minutes



# Activity: for Loop Zoo

01

Spend a few moments rewriting the code below using a for loop.

02

If you need help, use the code from the previous example as a guide.

03

Then try to explain to the person next to you how your code works.

```
// Array of zoo animals.  
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];  
  
console.log(zooAnimals[0];  
console.log(zooAnimals[1];  
console.log(zooAnimals[2];  
console.log(zooAnimals[3];
```

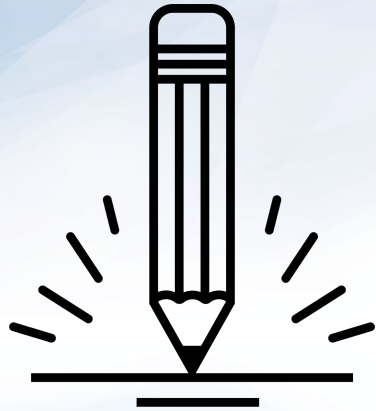
**Suggested Time:** 15 minutes





A close-up, high-angle shot of a computer keyboard. The central focus is a large, white, rectangular key with rounded corners. On this key, there is a dark blue icon of a coffee cup with three wavy lines above it representing steam. Below the icon, the word "Break" is printed in a dark blue, serif font. The key is set against a light-colored, textured keyboard surface. Surrounding the main key are other keys, including one with a double quote symbol to the left and one with a dash/slash symbol to the right, all of which are slightly out of focus.

Break



## **Activity:** Another Loop

**Suggested Time:**  
15 minutes



## Activity: Another Loop

---

Starting from scratch, create a for loop that prints the following lines:

I am 0

I am 1

I am 2

I am 3

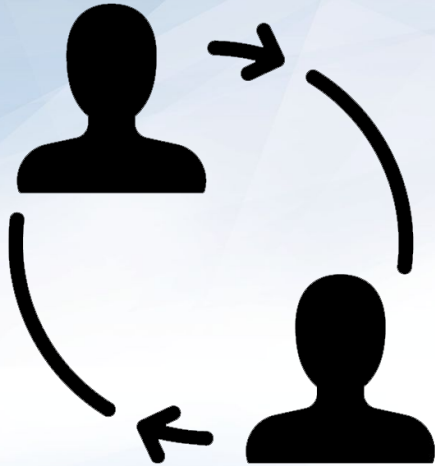
I am 4



**This time, don't use an array!**

**Suggested Time:** 15 minutes





## **Partner Activity:** Loop with Conditions

**Suggested Time:**  
10 minutes



# Partner Activity: Loop with Conditions



Starting from scratch, write code that loops through the following array:

```
// This is our starting myFarm array.  
var myFarm = ["chickens", "pigs", "cows", "horses", "ostriches"];
```



Use `console.log` to display the name of each animal on the farm.



Using the `.charAt()` method (research it), check if the first letter in the animal's name begins with a "c" or "o". If it does, create an alert saying: "Starts with c or an o!"

Suggested Time: 10 minutes





## **Activity:** Random Numbers

**Suggested Time:**  
7 minutes



# Activity: Random Numbers

---



Research how to use `Math.random()` to generate a whole number between 1 and 10.



Open `21-RandomNumbers/Unsolved` and modify the code so that it logs random *whole numbers* from 1 to 10 inclusive.

Suggested Time: 7 minutes



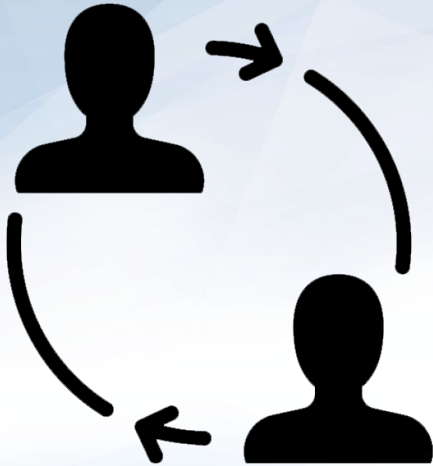
# Rock-Paper-Scissors with a Partner!

---

Play five rounds.







## **Partner Activity:**

Pseudocode

Rock-Paper-Scissors (RPS)

**Suggested Time:**  
8 minutes



# Partner Activity: Pseudocode RPS

---



With a partner, spend a few moments outlining all the steps and conditions that go into a single game of Rock-Paper-Scissors.



Try to break it down into steps that you could “code out.”



Think of basic elements like loops, if-then statements, arrays, alerts, etc.



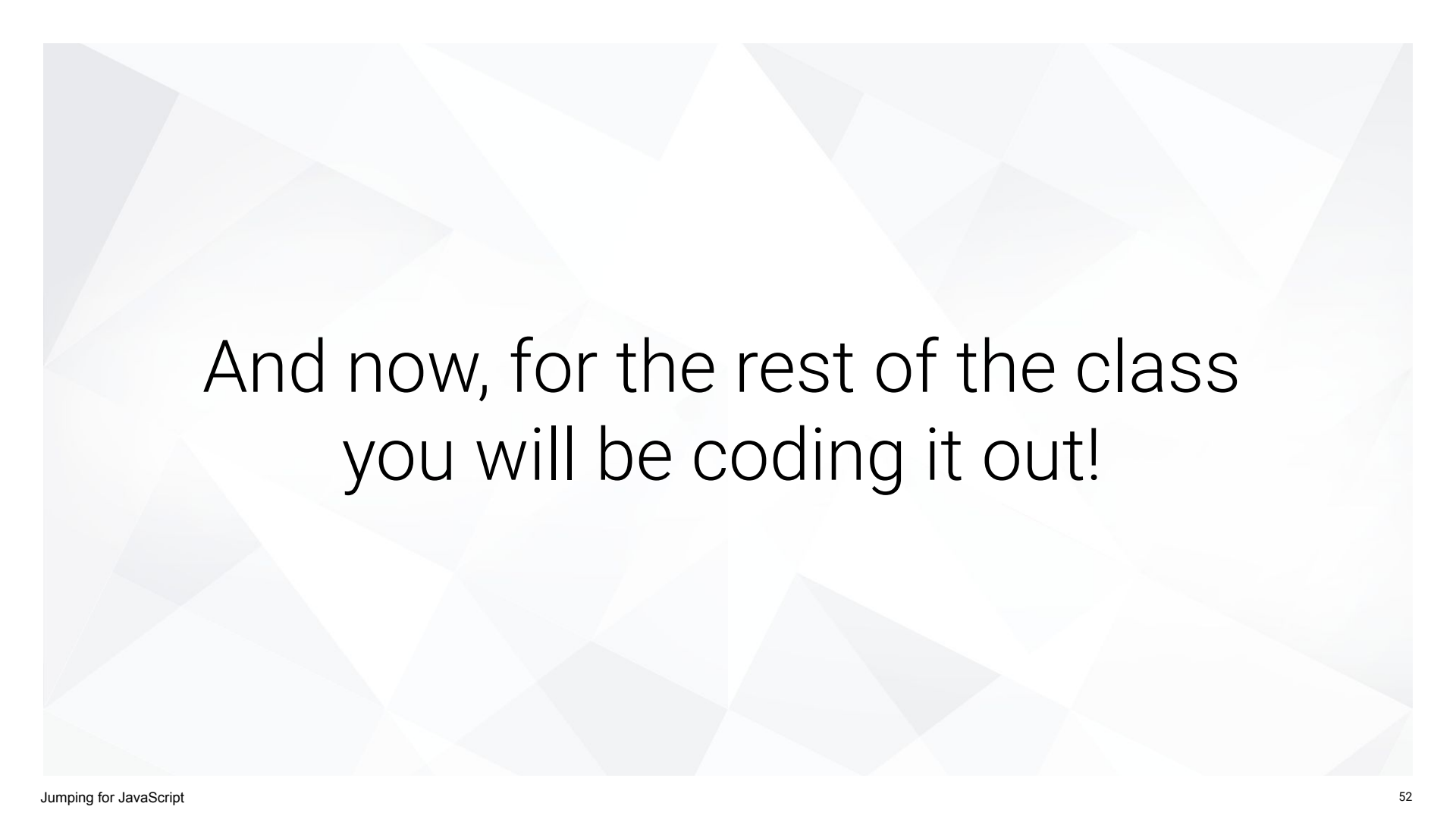
Be prepared to share your outlined approach.

Suggested Time: 8 minutes

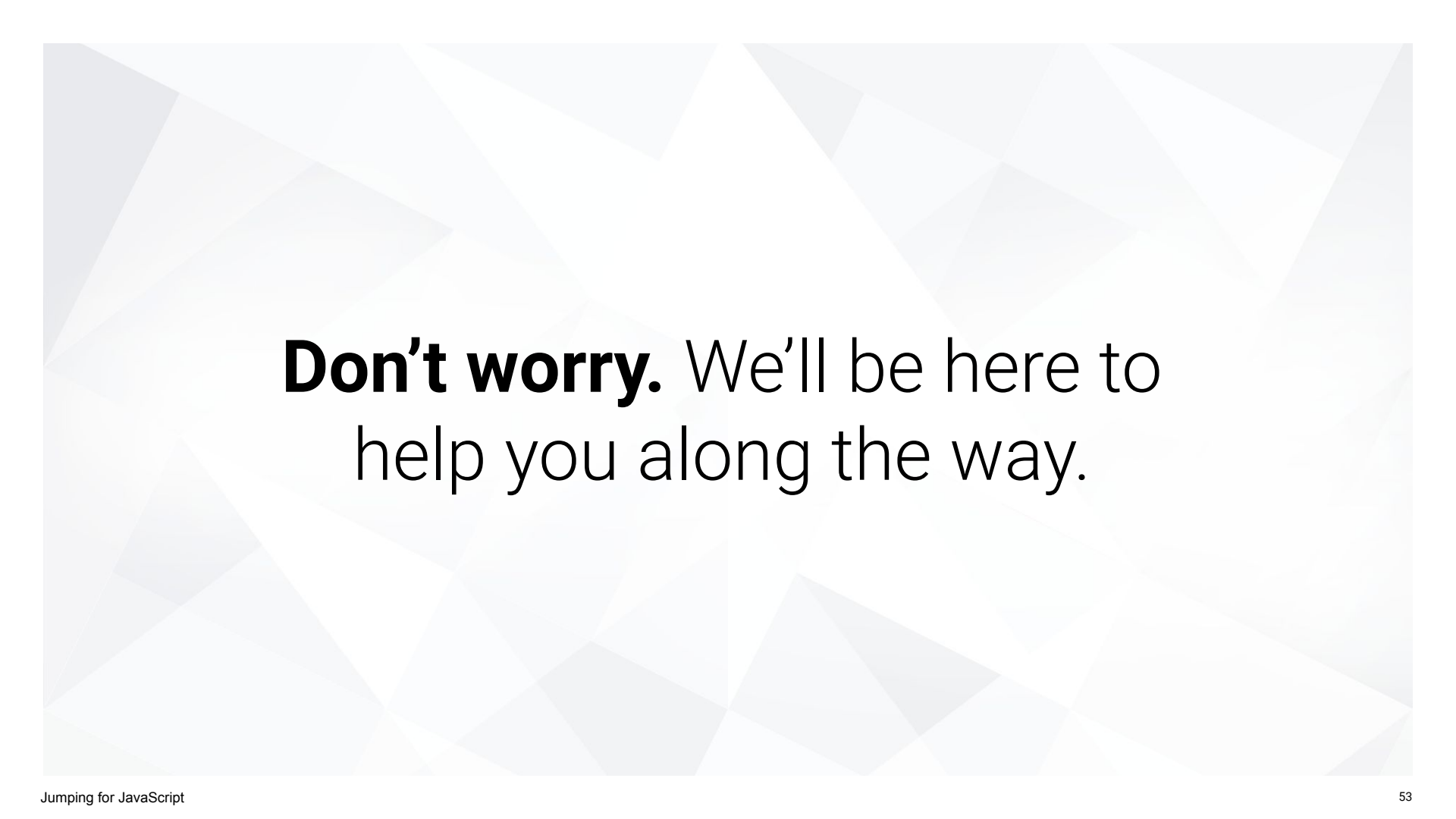


You just **pseudocoded!**





And now, for the rest of the class  
you will be coding it out!



**Don't worry.** We'll be here to  
help you along the way.



# Group Challenge (4 people):

## Coding RPS

**Suggested Time:**  
60 minutes



# Group Challenge: Coding RPS

---



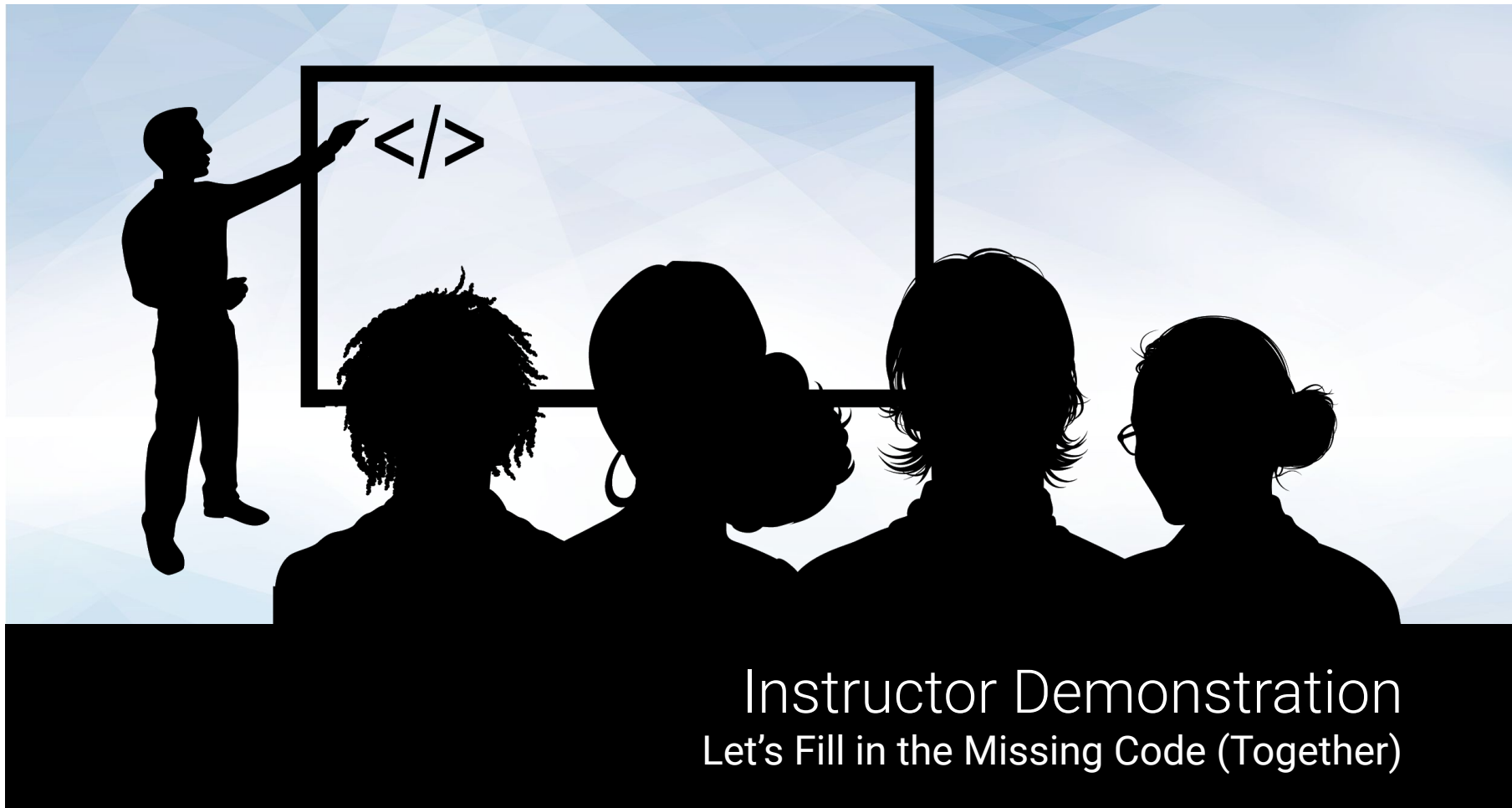
In groups of four, begin the process of coding out the Rock-Paper-Scissors game.



Do as much as you can on your own, but don't be afraid to ask for help if you feel your team is struggling.

Suggested Time: 60 minutes





# Instructor Demonstration

## Let's Fill in the Missing Code (Together)





# Questions?