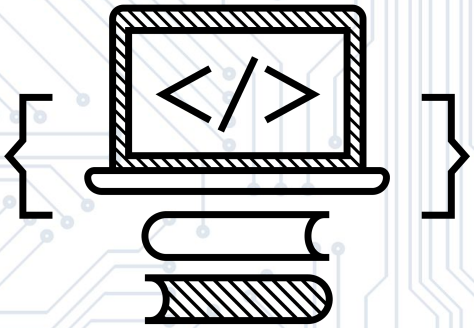
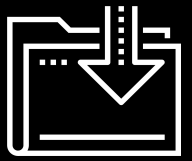


Jumping for JavaScript

Web Development Boot Camp

Lesson 3.2



Today’s Class

Objectives

In today's class, we'll cover:

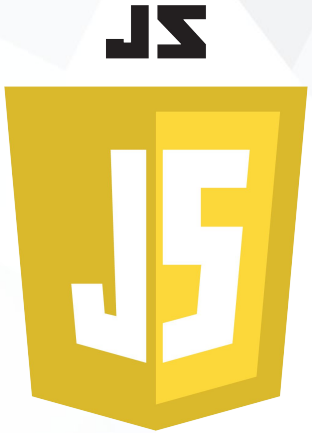
Array assignments

The concept of `for` loops

The art of pseudocoding

Building Rock-Paper-Scissors

Basics Recap



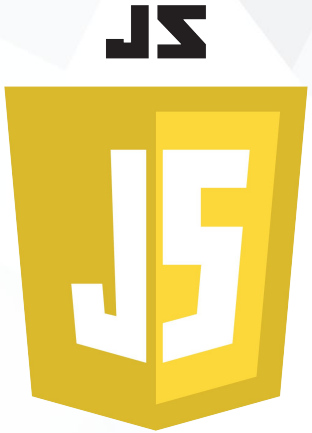
What is JavaScript?

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?

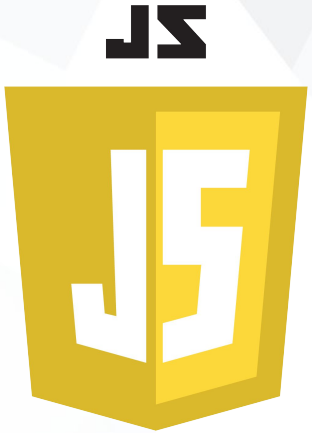
Variable Basics

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;
```



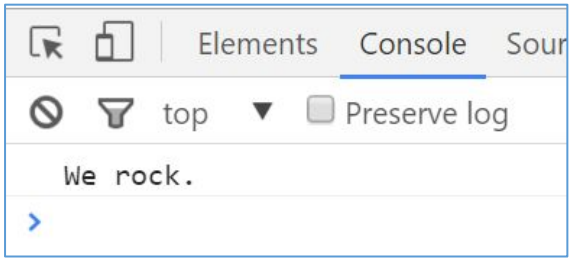
What is meant by `console.log`?

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.");
```



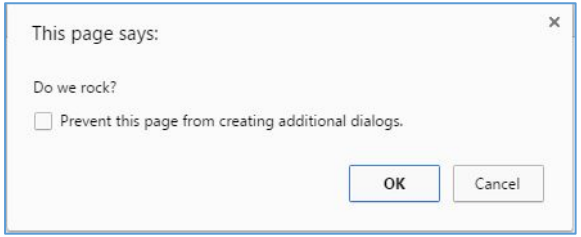

```
alert("We Rock.");
```

Console.log vs. JavaScript popup boxes

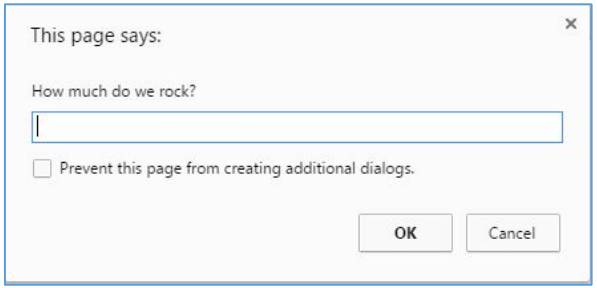
`confirm` displays a true/false popup.

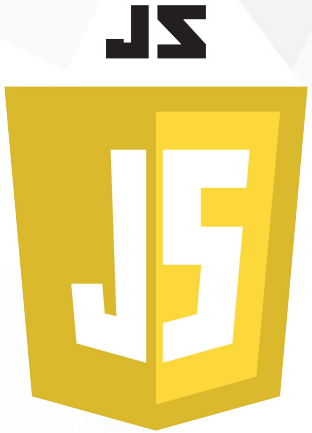
`prompt` displays a popup with a text-box input.

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




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



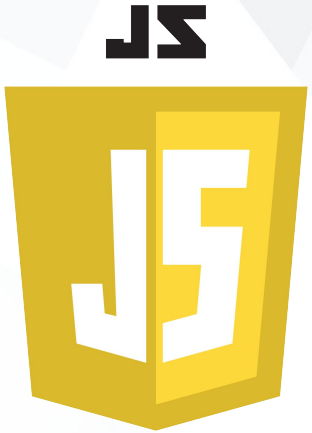


How do we **check conditions**?

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**?

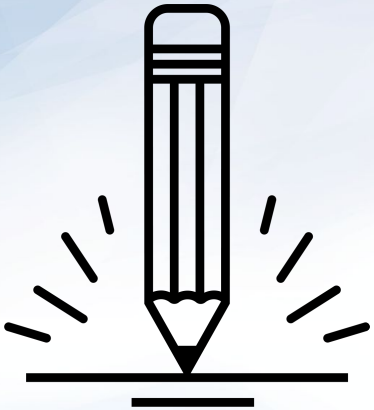
Basic Arrays

Arrays 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];
```



Activity:

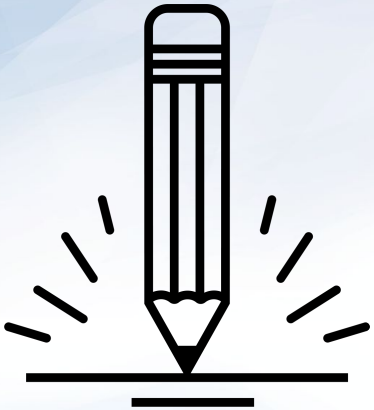
Basic JavaScript Dissection

`14-JSDissect/Unsolved/js-dissect-unsolved.html`

Activity: Basic JavaScript Dissection

Let's look at `14-JSDissect/Unsolved/js-dissect-unsolved.html`

See if you now better understand how it works.



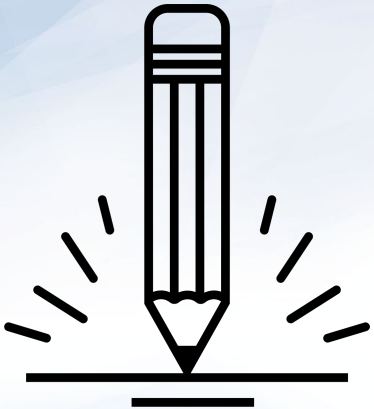
Code-Along Activity:
Array Logging

`15-CoolPeopleArray/Unsolved/array-logs-unsolved.html`

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.



Code-Along Activity:
Array Setting

`16-ArraySetting/Unsolved/array-setting-unsolved.html`

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.

Back to The Zoo Pen

Array name: zooAnimals

Coded in JavaScript using an array

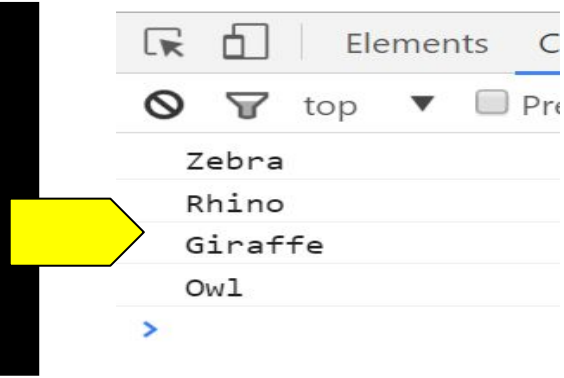
```
// Our array of zoo animals.
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];
```

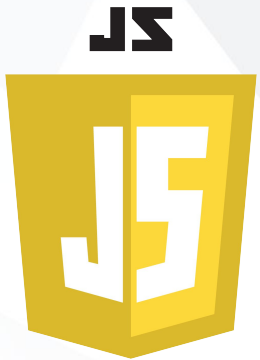
Back to The Zoo Pen

Array name: zooAnimals

Zebra	Rhino	Giraffe	Owl
Index 0	**Index 1**	**Index 2**	**Index 3**

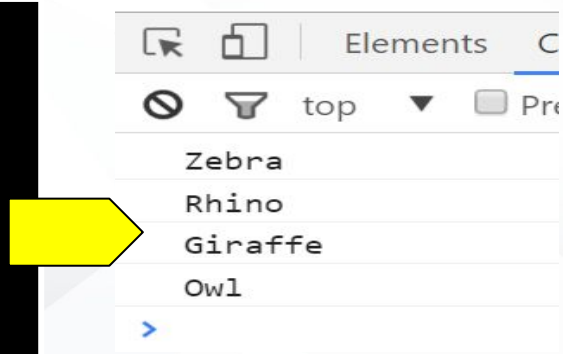
```
// Array of zoo animals.
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];

console.log(zooAnimals[0]);
console.log(zooAnimals[1]);
console.log(zooAnimals[2]);
console.log(zooAnimals[3]);
```





What's wrong here?

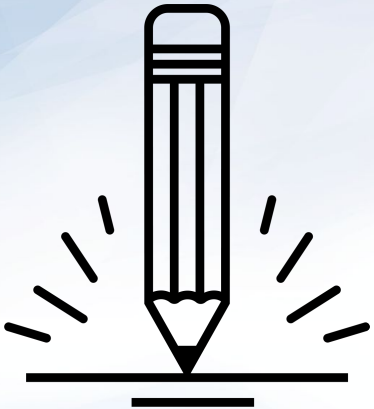
```
// Array of zoo animals.
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];

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]);
```



Code-Along Activity:

Dissecting `for` Loops

`17-MyFirstLoop/my-first-loop.html`

Activity: Dissecting for Loops

Let’s spend a few moments trying to dissect the code sent to you.

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

for Loops

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

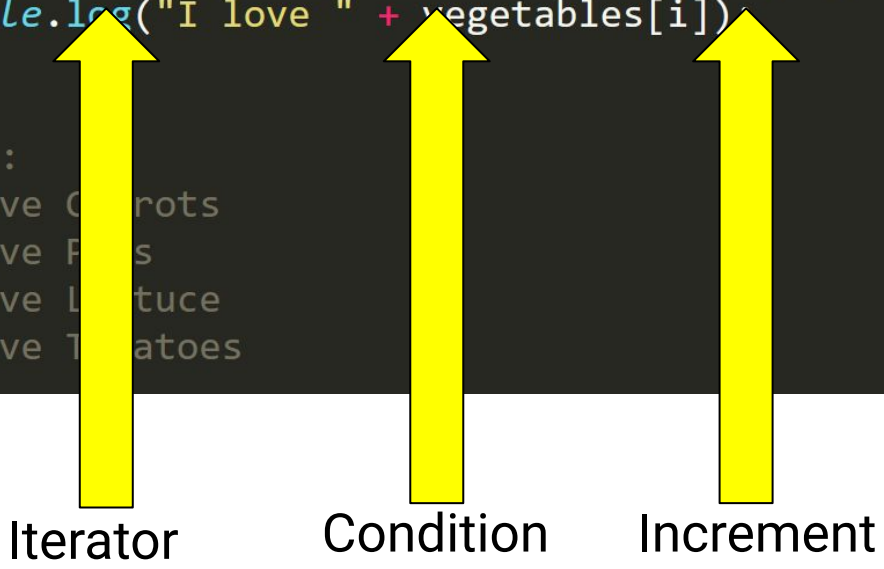
Each for loop is composed of a:

- 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
```



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.

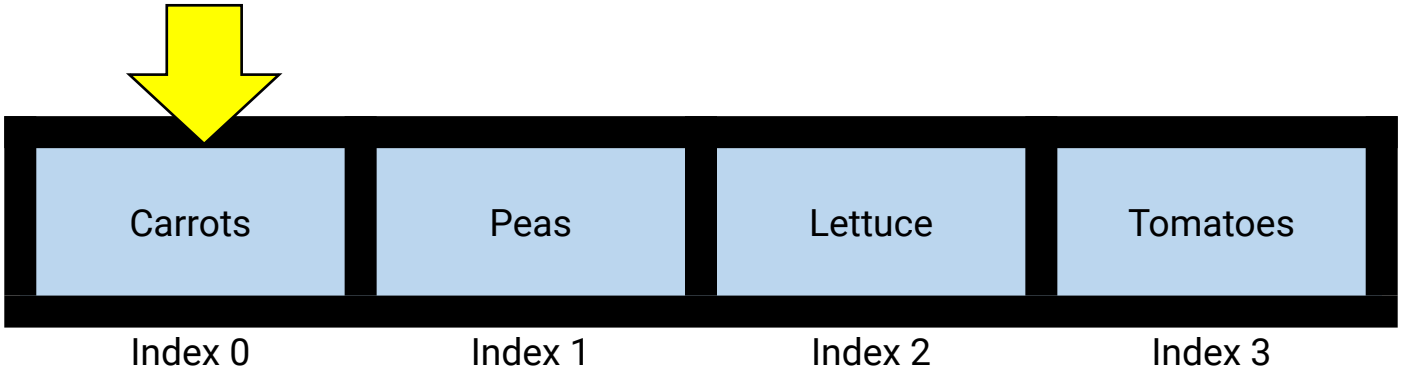
```
// Start with an Array.
var vegetables = ["Carrots", "Peas", "Lettuce", "Tomatoes"];

// 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]);
}
```

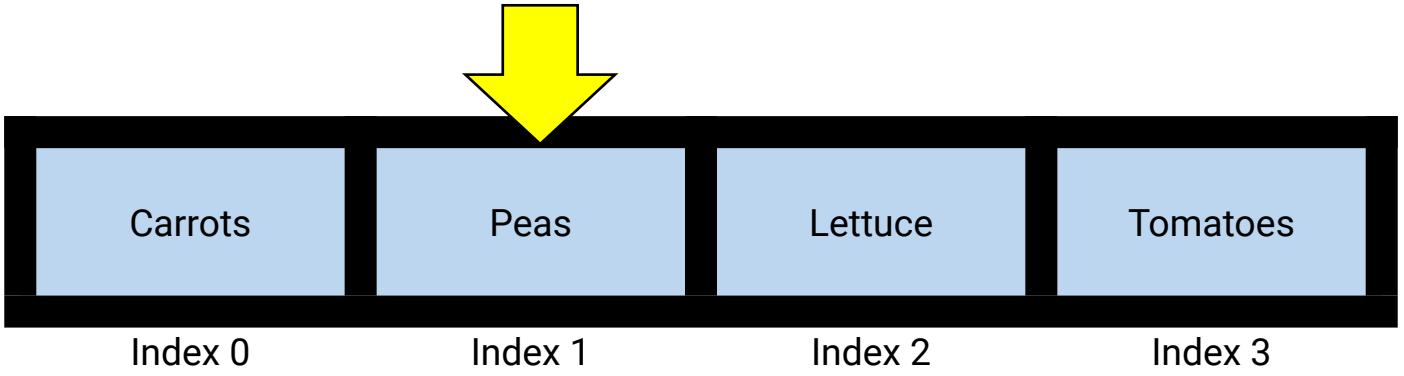
When i = 0 … `console.log`(“I love Carrots”)



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]);
}
```

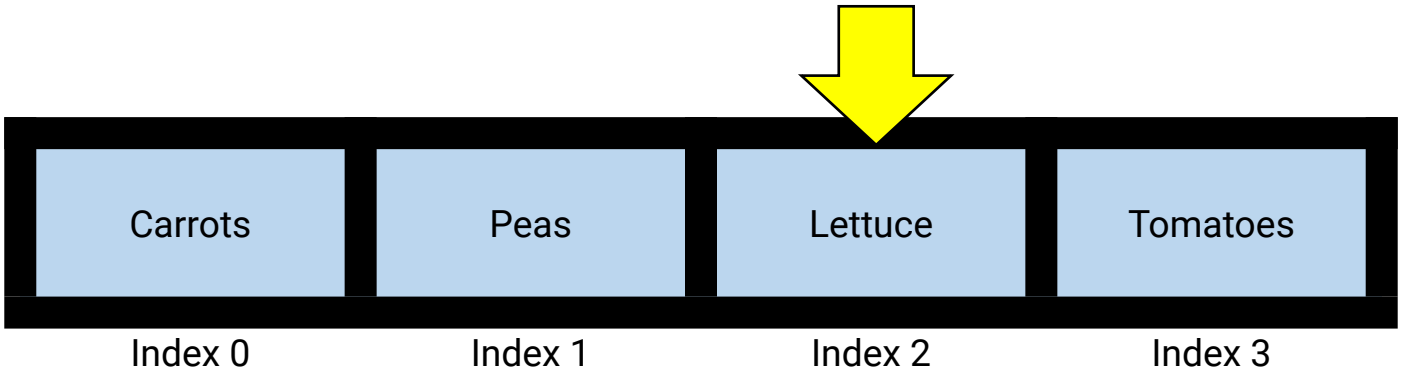
When i = 1 … `console.log`(“I love Peas”)



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]);
}
```

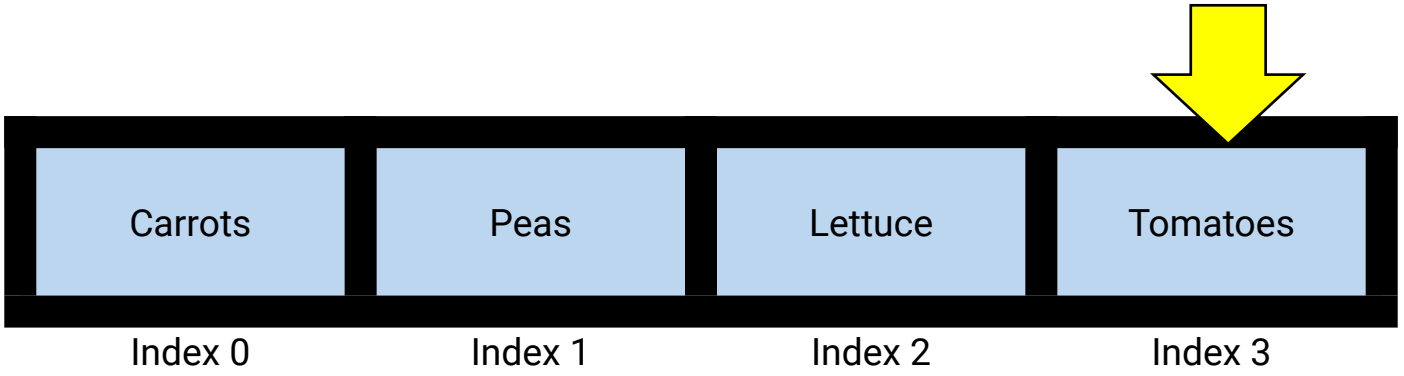
When i = 2 … `console.log`(“I love Lettuce”)

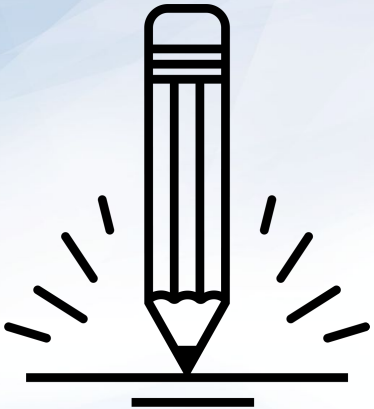


Run That Loop

```
// 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”)





Breakout Activity:
for Loop Zoo

`18-ZooLoop/Unsolved/zoo-loop-unsolved.html`

Activity: for Loop Zoo

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

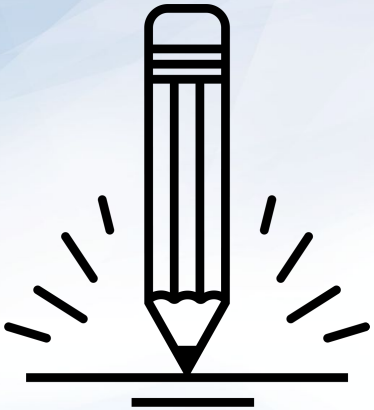
02 If you need help, use the code from the previous example as a guide.

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


Break



Code-Along Activity:
Another Loop

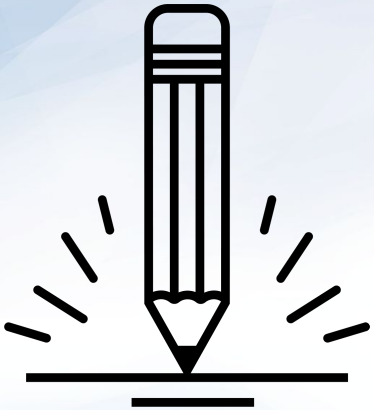
`03-JavaScript/01-Activities/19-AnotherLoop`

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!



Code-Along Activity:
Loop with Conditions

`03-JavaScript/01-Activities/20-HardLoop`

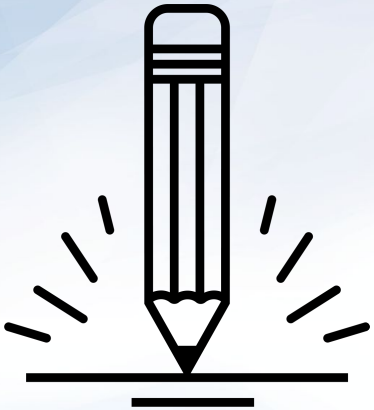
Activity: Loop with Conditions

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!"



Code-Along Activity:

Random Numbers

`03-JavaScript/01-Activities/21-RandomNumbers/Unsolved/random-numbers-unsolved.html`

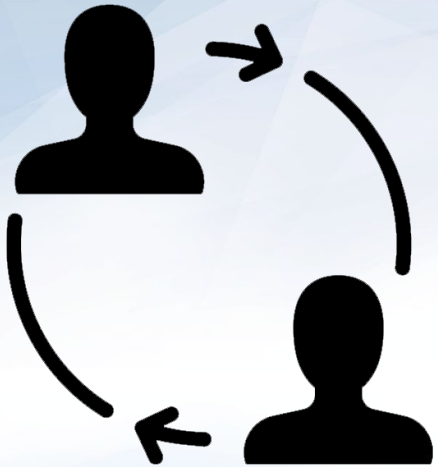
Activity: Random Numbers

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 is logs random *whole numbers* from 1 to 10 inclusive.

Rock-Paper-Scissors with a Partner!

Play five rounds.



Partner Activity:

Pseudocode Rock-Paper-Scissors (RPS)

`03-JavaScript/01-Activities/22-PseudoCode`

Breakout Activity: Pseudocode RPS

In your breakout room, 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.

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:

Coding RPS

`03-JavaScript/01-Activities/23-RPS-Coded`

Group Challenge: Coding RPS

In your breakout room, begin the process of coding out the Rock-Paper-Scissors game.

Do as much as you can on your own. We will be rotating through the breakout rooms regularly during the activity, so ping us on Slack in **#01-mw-live** if you get totally stuck in the second half of the activity, but keep trying to brainstorm with your team on what the solution might be!

Questions?