Day 8

Jumping for JS

The Coding Bootcamp | April 13, 2016

Today's Class

Objectives

In today's class we'll be covering:

- Array Assignments

- The Concept of For-Loops

- The Art of Pseudo-Coding

- 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, 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.

Please… Don't Pick Me.

What is a Variable?
(And how do we declare one?)

Basic Variables

- Variables are the <u>nouns</u> of programming.

- They are "things" (Numbers, Strings, Booleans, etc.)

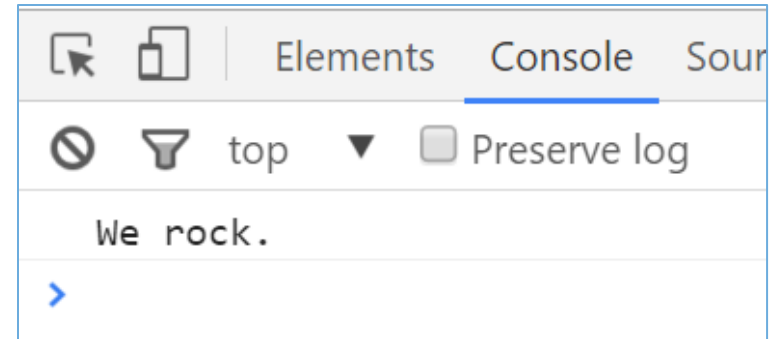- They are composed of <u>variable names</u> and <u>values</u>

```
var name = "Snow White";
var dwarfCount = 7;
var isSleeping = true;
```
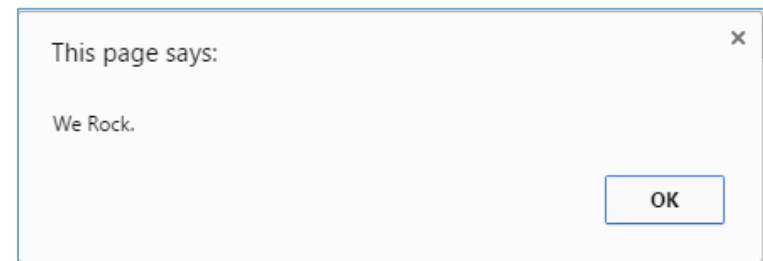
Please… Don't Pick Me.

What is meant by console.log?
(And how does it differ from an alert, prompt, or confirm?)

Basic Variables

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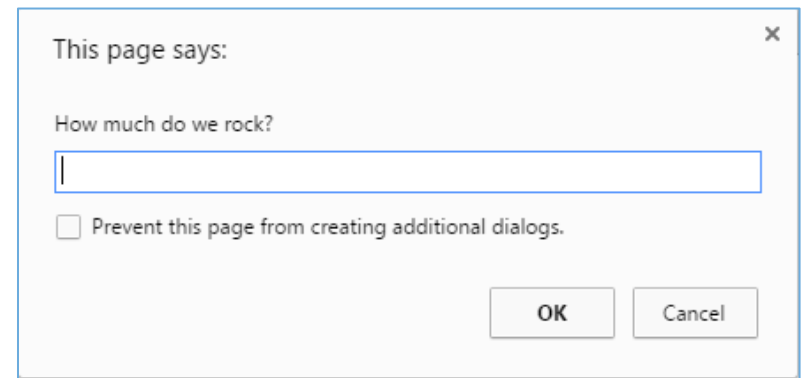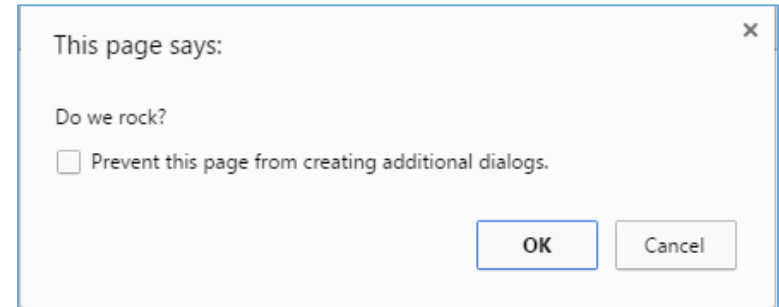
```
alert("We Rock.");
```

- **Console.log** displays discreetly to the debugger.

- **Alert** displays a pop-up message to the user.

Basic Variables

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

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

This page says: ×

Do we rock?

☐ Prevent this page from creating additional dialogs.

OK   Cancel

This page says: ×

How much do we rock?

[   ]

☐ Prevent this page from creating additional dialogs.

OK   Cancel

- **Confirm** displays a True/False popup.

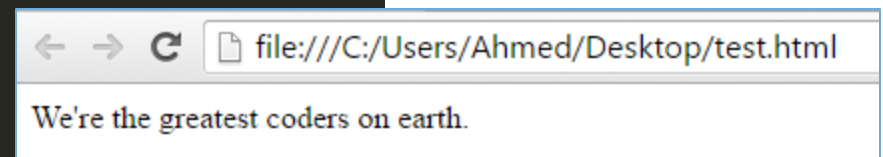- **Alert** displays a prompt with a text-box input.

Please… Don't Pick Me.

How do we "write" text to the HTML itself?

Writing to HTML

- We can use JavaScript to directly write to the HTML page itself using **document.write( ).**

- Later we will go over *much* more advanced approaches for writing HTML using JavaScript and jQuery.

```
1  <!DOCTYPE html>
2  <html lang="en-us">
3    <head>
4      <meta charset="UTF-8">
5      <title>Document Write</title>
6    </head>
7    <body>
8
9      <script type="text/javascript">
10
11       document.write("We're the greatest coders on earth.");
12
13     </script>
14
15   </body>
16 </html>
```

Test.html (chrome)

file:///C:/Users/Ahmed/Desktop/test.html

We're the greatest coders on earth.

Test.html (sublime)

How do we check conditions?

If/Else Statements

- 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.

```javascript
// 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 {
  document.write("You don't like sushi or ginger tea.");
}
```

Please… Don't Pick Me.

What is an array?

Basic Arrays

- Arrays are a type of variable that are <u>collections</u>.

- These collections can be made up of <u>strings</u>, <u>numbers</u>, <u>booleans</u>, other <u>arrays</u>, <u>objects</u>, anything.

- Each <u>element</u> of the array is marked by an <u>index</u>. Indexes always start with 0.

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

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

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

Code Dissection: Basic JS

- Re-examine the file sent to you during yesterday's class.

- See if you can better understand how it works – after having gone through today's class.

- Prepare to share once the time is up.

Code Creation: Array Logging (If Needed)

- 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 6 times._

- Be prepared to share once time is up.

Code Creation: Array Setting

- Follow the instructions in the file provided to convert each item in the array to lower case.

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

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

- Be prepared to share once time is up.
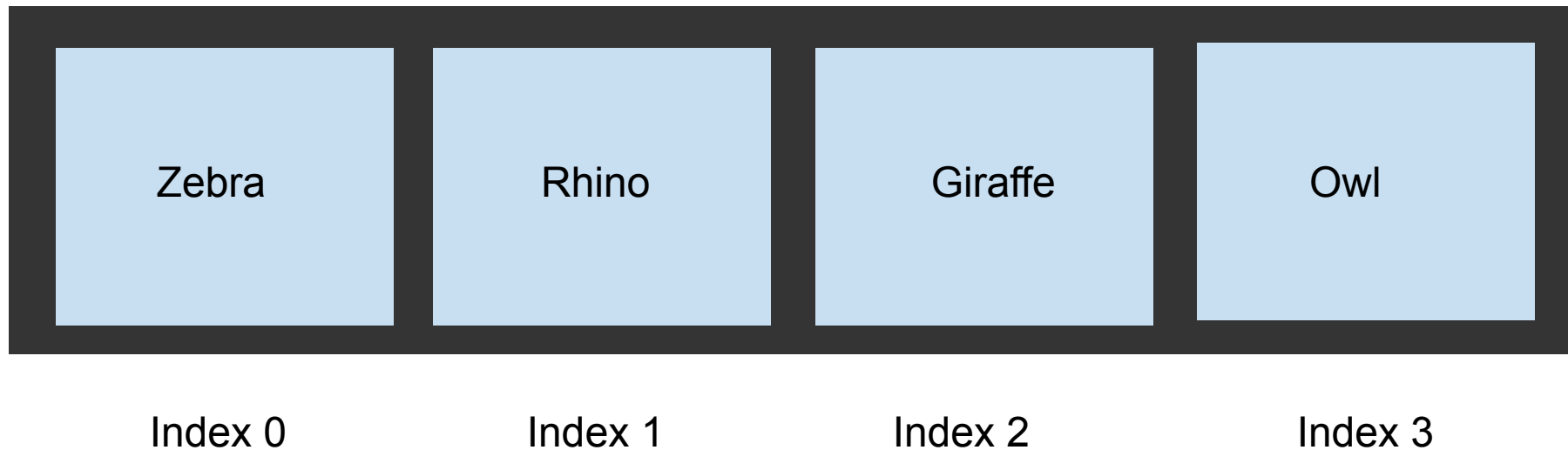
For Loops

Back to The Zoo Pen

Array Name: zooAnimals

|			
Zebra	Rhino	Giraffe	Owl
Index 0	Index 1	Index 2	Index 3
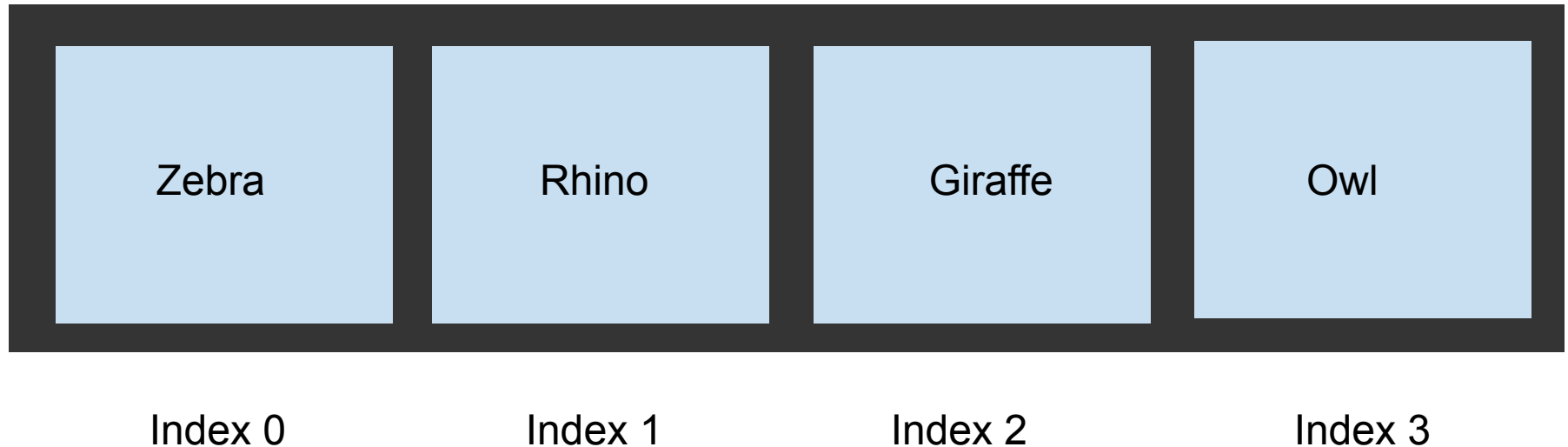
```
// Our array of zoo animals.
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];
```

Back to The Zoo Pen (Logging)

Array Name: zooAnimals

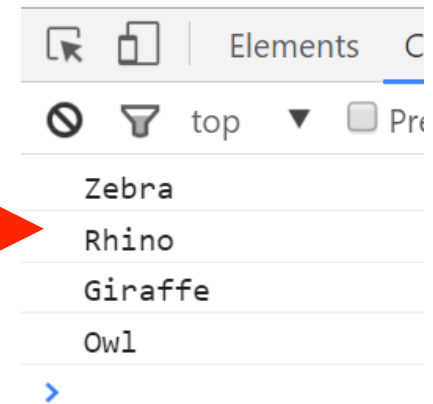|			
Zebra	Rhino	Giraffe	Owl
Index 0	Index 1	Index 2	Index 3

```javascript
// 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]);
```

Elements   C

top   ▼   ☐ Pre

Zebra
Rhino
Giraffe
Owl

>

Please… Don't Pick Me.

```javascript
// 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]);
```

Elements C

top ▼ Pr
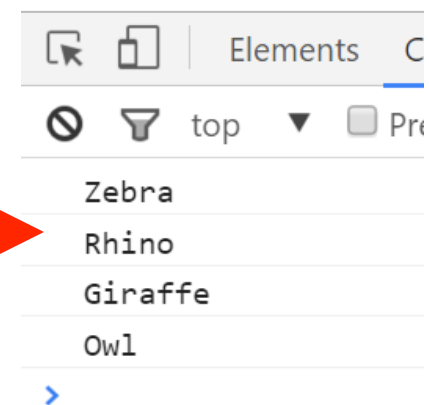
Zebra
Rhino
Giraffe
Owl

What's wrong here?

Don't Repeat Yourself (DRY)

```javascript
// 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]);
```
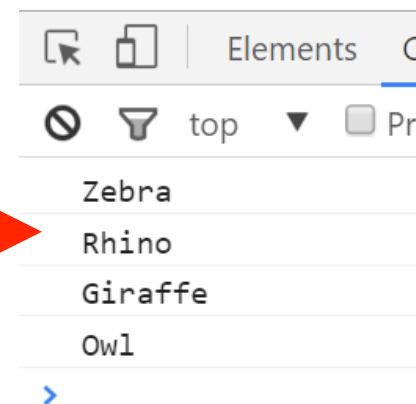

Repeated Code!
Let's be more efficient

Code Creation: For Loop Dissection

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

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

- Feel free to do research if you are stumped. As a hint, look into the phrase: "For-Loop".

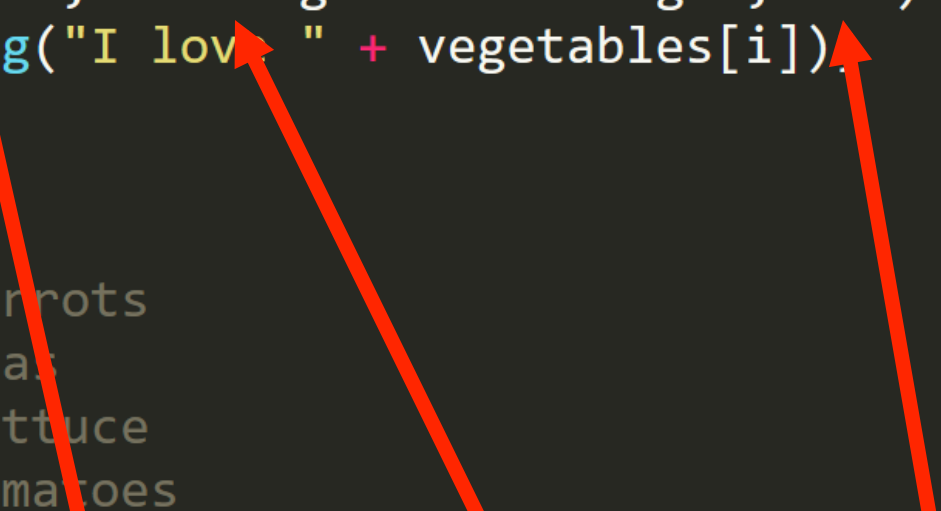- Be prepared to share when time is up.

Enter the For-Loop

- For loops are <u>critical</u> in programming.

- We use for loops to run <u>repeated blocks of code</u> over a set period.

- Each for loop is composed of a:
  - Variable declaration or counter (iterator)
  - A loop condition
  - An iteration (addition)

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

Enter the For-Loop

```javascript
// 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.*

Enter the For-Loop

```javascript
// 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
```

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

Enter the For-Loop

```javascript
// 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
```

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

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]);
}
```
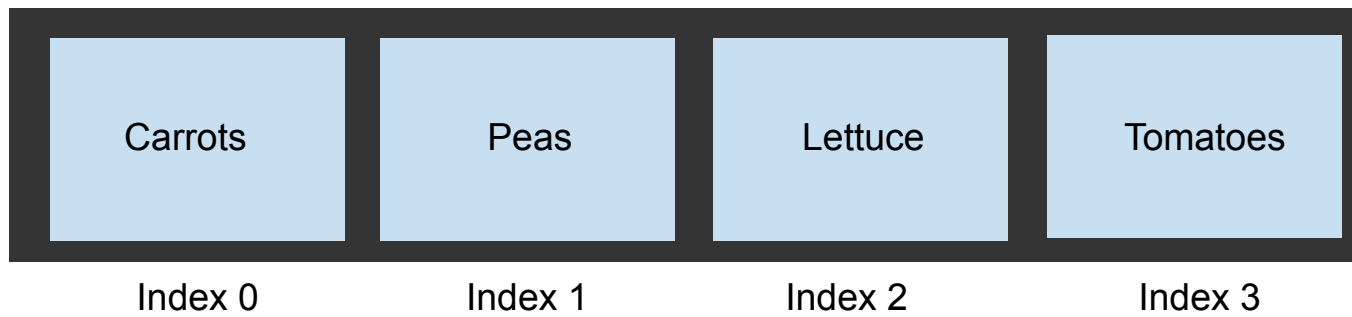
When i = 0 … console.log("I love Carrots")

| Carrots | Peas | Lettuce | Tomatoes |
| Index 0 | Index 1 | Index 2 | Index 3 |

Run That Loop

```javascript
// 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]);
}
```
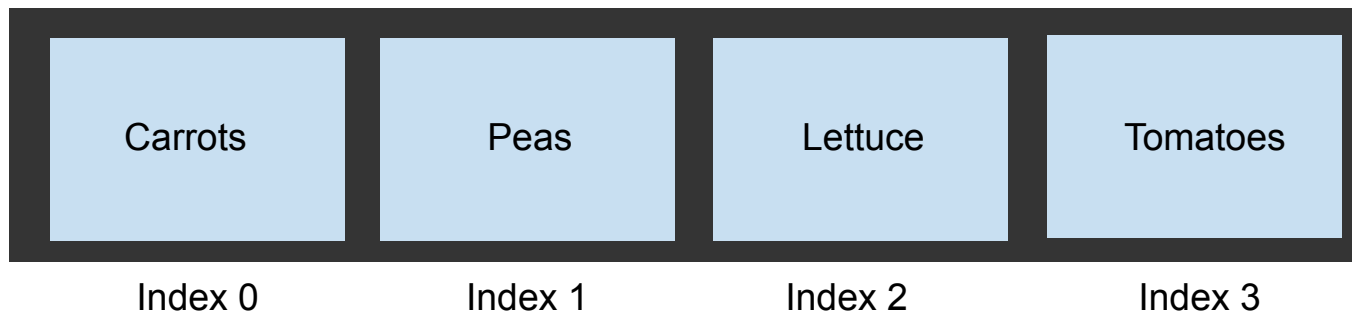
When i = 1 … console.log("I love Peas")

Carrots	Peas	Lettuce	Tomatoes
Index 0	Index 1	Index 2	Index 3

Run That Loop

```javascript
// 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]);
}
```
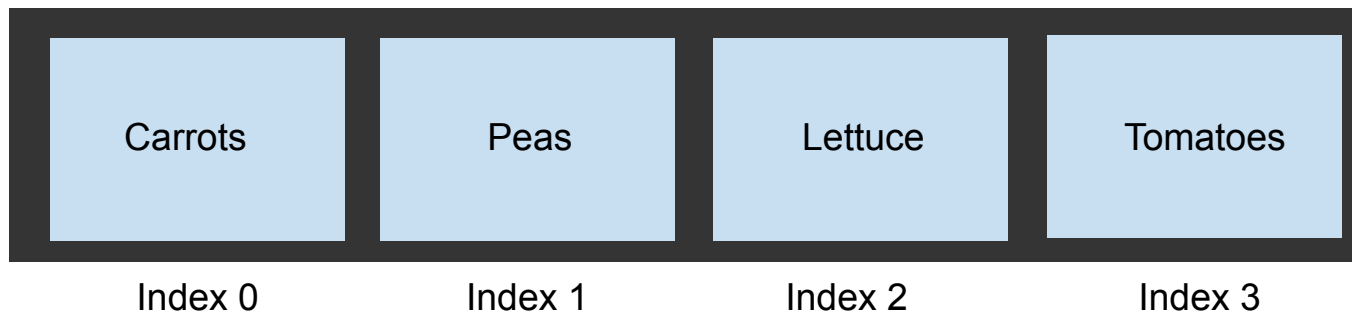
When i = 2 … console.log("I love Lettuce")

Carrots	Peas	Lettuce	Tomatoes
Index 0	Index 1	Index 2	Index 3

Run That Loop

```javascript
// 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]);
}
```
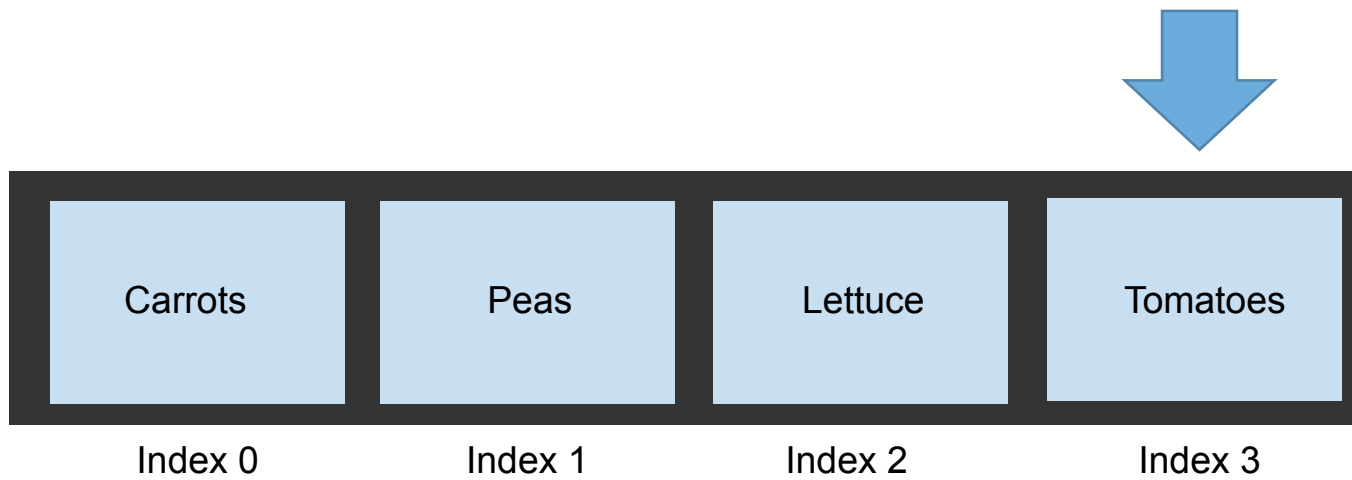
When i = 3 ... console.log("I love Tomatoes")

Carrots	Peas	Lettuce	Tomatoes
Index 0	Index 1	Index 2	Index 3

Code Creation: For-Loop Zoo

- Spend a few moments, re-writing the code below using a for-loop.

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

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

```javascript
// 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 Creation: Another Loop (Time Permitting)

- 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 Creation: Hard Loop (Time Permitting)

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

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

- And console.logs the name of each animal on the farm.

- Then 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!"
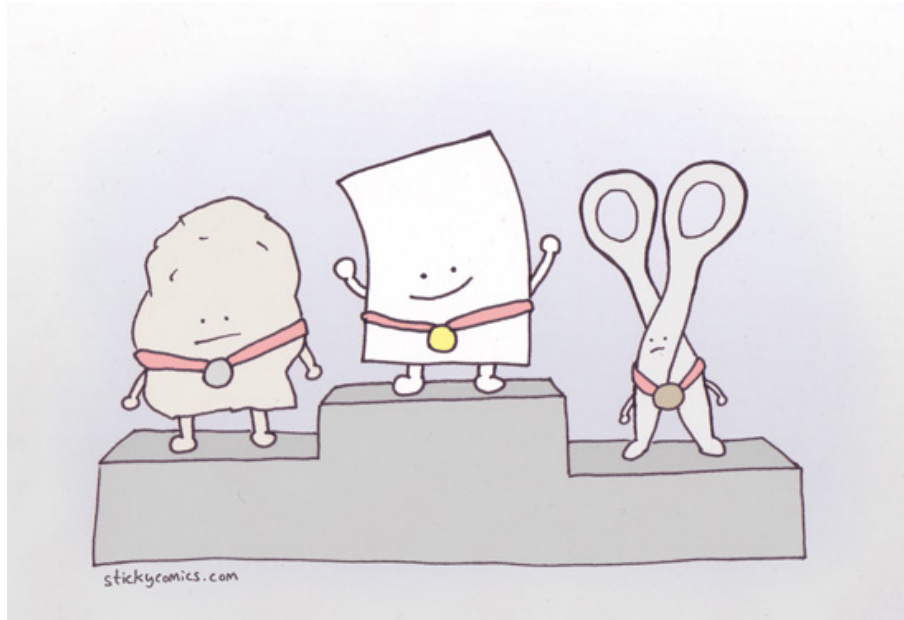
Instructor: Demo
(EventsExample.html | 11-Events)

Rock Paper Scissors

Rest of Class!

I'ma Beat You…

Play Rock Paper Scissors with the Person Next to You!

Play 5 Rounds

> YOUR TURN!!

Code Creation: Pseudocode

- 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.

You just pseudocoded!

Basically a Coder!

Now… for the rest of the class YOU will be coding it out.

Basically a Coder!

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

Demo Final Solution

Instructor: Demo
(rps-9.html | 9-RPS-Coded)

> YOUR TURN!!   Activity: 9-RPS-Coded | Suggested Time: 1 hour 10 min

Code Creation: Coding out RPS

- In groups of 4, 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.

- **Note:** Don't use "document.write" as it will delete the contents of your page including your JavaScript. Use "document.querySelector" or "document.getElementById", alongside either "innerHTML" or "textcontent", to write to the DOM.

- **Note:** Don't worry. We know this will be very challenging. We also know that you won't know where to start. In fact, we haven't shown you EVERYTHING you need yet. But that's okay. Accepting the confusion is a HUGE first step in becoming a coder.

- **Note to Instructor/TAs:** Use the files in RPS-Coded to help guide students through the process. Feel free to present relevant code on the projector.

Recap Activity

Time Permitting

Demo Questions

Let's fill in the Missing Code (Together)
(Recap_UNSOLVED | 10-Recap)

Questions