Arrays and Loops

Advanced JavaScript

What is an Array?

var instructors = [“John”, “Caitlin”, “Emilia”]

- Ordered collection of related data types.
- Organized by index.
  - Indexing begins at 0

Accessing items in an array

var instructors = [“John”, “Caitlin”, “Erica”]

If we want to access “John” from the array, we can use this syntax:

instructors[0] = John

Turn and Talk

Can you think of some examples of when you might want to use an array?

.length

var instructors = [“John”, “Caitlin”, “Erica”]

Let’s say we need to find out how many items are in our array. We can use the .length method

instructors.length = 3

.pop()

var instructors = [“John”, “Caitlin”, “Erica”]

Looks like our instructors are no longer accurate! We should remove Erica from the array.

instructors.pop() = [“John”, “Caitlin”]

.push()

var instructors = [“John”, “Caitlin”]

Now, let’s add Emilia to our instructors array

instructors.push(“Emilia”) = [“John”, “Caitlin”, “Emilia”]

.shift()

var fruits = [“banana”, “orange”, “apple”, “mango”]

I hate bananas, so I want to remove it from my array. We know the .pop() method, but that removes the last item.

fruits.shift() = [“orange”, “apple”, “mango”]

.unshift()

var fruits = [“orange”, “apple”, “mango”]

I like strawberries, so I want to add that into the beginning of my fruits array instead of banana.

fruits.unshift(“strawberry”) = [“strawberry”, “orange”, “apple”, “mango”]

.splice()

var veggies = [“carrot”, “celery”, “broccoli”, “cucumber”]

Now let’s say we want to remove celery from this array. How can we select a specific item and remove it?

veggies.splice(1, 1) = [“carrot”, “broccoli”, “cucumber”]

** The first number represents the index value that we want to remove, the second is how many items we are removing*

.join()

var veggies = [“carrot”, “celery”, “broccoli”, “cucumber”]

What if we want to take the items in our array and turn them into a string?

veggies.join(“ and ”) = carrot and celery and broccoli and cucumber

.indexOf()

var veggies = [“carrot”, “celery”, “broccoli”, “cucumber”]

Now we may want to know the index of broccoli in the array

veggies.indexOf(“broccoli”) = 2

Remember, indexing begins at 0

.reverse()

var veggies = [“carrot”, “celery”, “broccoli”, “cucumber”]

What if we decide we want to reverse the order of the items in our array?

veggies.reverse() = [“cucumber”, “celery”, “broccoli”, “cucumber”]

Bonus! Multidimensional arrays

No need to worry too much about this yet, but you can also put arrays inside of arrays!

var foods = [[“cake”, “donuts”, “pie”], [“apple”, “pear”, “peach”]]

Foods[0] = [“cake”, “donuts”, “pie”]

Foods[0][2] = Pie

You do:

- Create an index.html and script.js file
- In your script.js file, create an array and store it as a variable
- Open up your index.html file in your browser and open your console
- Use some of the tools we just learned to manipulate arrays!

What is an Loop?

Now we have these arrays, but what can we really do with it if we can’t access each individual item in the array?

[“John”, “Caitlin”, “Emilia”] === “John”

This would return false, despite the fact that one of the items in the array matches the condition that we are checking

For Loops

```
for(var i = 0; i < 10; i++){
        console.log(i);
}
```

For Loops

```
for (var i=0; i<10; i++) {
    console.log(i);
}
```

There are three parts of the For loop:

1. Tell the loop what number to start at (typically in relation to an array)
2. Tell the loop how many iterations to run for
3. Tell the loop how much to increment after each iteration

For Loops

BEWARE!

When you're working with loops, be sure that the conditions of your loop will eventually return false. Otherwise, you will create an "infinite loop" and crash your browser.

While Loops

```
var i = 0;
while (i < 10) {
    console.log(i);
    i++;
}
```

The while loop only uses one condition that causes it to continue looping, and you need to increment var i separately.

For vs. While loops

- Which loop do you prefer?
- Can you think of some pros and cons of each loop syntax?

You do:

- Create an array of 5 values and store it as a variable
- Loop through all the values in the array and print out each value in the console
  - Try using one **for** loop and one **while** loop to produce the same output!

Real World Application

Can you think of some examples where a loop would be useful?

Loops and Conditional Statements

Using loops in conjunction with conditional statements can be incredibly useful

You already know how to use conditional statements, so this will just be combining two things you already know!

Loops and Conditional Statements

```
var instructors = [“Caitlin”, “John”, “Emilia”]

for (var i=0; i<instructors.length; i++) {
    if (instructors[i] == “John”) {
        console.log(instructors[i] + “ is one cool dude!”)
    } else {
        console.log(instructors[i]);
    }
}
```

What do you think this will return?

You do:

- Create an array of 5 values and store it as a variable
- Loop through all the values in the array using a **for** or **while** loop
  - Inside the loop, write a conditional statement that will print out “I love JavaScript” at the 3rd index value
  - For all other index values in the loop, print out the array’s value at that index

FIZZBUZZ!

Write a program where:

For the numbers 1 to 100, print "**Fizz**" if the number is divisible by 3, print "**Buzz**" if it's divisible by 5, print "**FizzBuzz**" if it's divisible by both, and otherwise just print the number.

HINT: Use the modulus operator to find the remainder (%)

30 % 5 == 0 would evaluate to “true” because 30 is divisible by 5