



FEWD - ARRAYS

AGENDA

- Review/Present
- What is an Array?
- Declaring arrays
- Accessing data in arrays
- Practice!

HW REVIEW AND GENERAL QUESTIONS

- Functions (named, anonymous)
- Using input fields and jQuery to get values from a user
- Async JavaScript and callback functions

ARRAYS

What if we had a collection of images that we wanted to display to the screen one at a time?

How could we store all the images?

ARRAYS

What is an array?

- A data type, just like a string, or number
- An array is a collection of values
- The values in an array can be any data type
- An array is good for storing and accessing a collection of data (like a collection of images)

DECLARING ARRAYS

There are 3 ways to declare an array.

DECLARING ARRAYS

```
var myArr = new Array();
```

- declaring an empty array using the Array constructor.

DECLARING ARRAYS

```
var myArr = [ ];
```

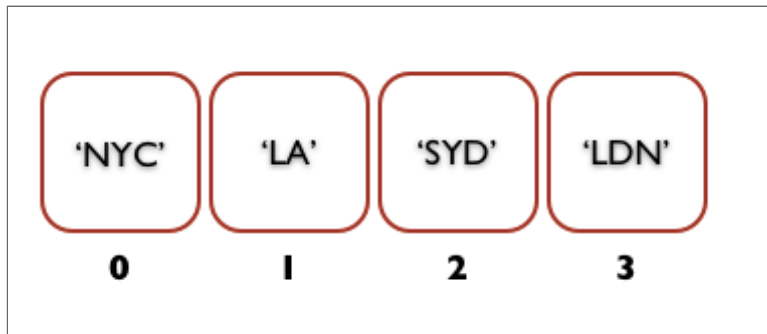
- declaring an empty array using literal notation.

DECLARING ARRAYS

```
var myArr = ['Hello', 54.3, true];
```

- Arrays are filled with values: i.e. `myArr3 = [element, anotherElement];`
- Values can contain strings, numbers, booleans, and more.

ARRAYS INDEXING



ACCESSING VALUES INSIDE ARRAYS

Use square brackets with the specified index

ARRAYS INDEXING

Array elements can be fetched by their index number (starts from 0).

```
var myArr = ['Hello', 54.3, true];  
  
console.log(myArr[0]); //prints Hello  
console.log(myArr[1]); //prints 54.3  
console.log(myArr[2]); //prints true
```

ARRAYS INDEXING

We can insert new values into any space in the array using the positions index.

```
var myArr = [];  
myArr[0] = 'Stuff';  
myArr[1] = 'Another string'
```

myArr looks like this: ['Stuff', 'Another String']

ARRAYS INDEXING

When you assign a variable to be an array, that variable can be re-assigned just like any other variable. You can either change one value of the array, or change the whole array

ARRAY RE-ASSIGNMENT

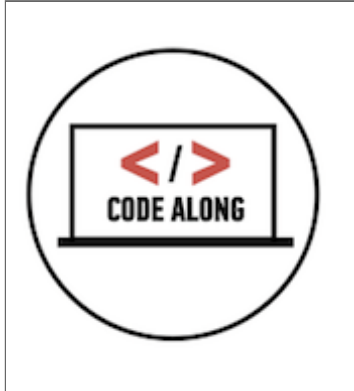
```
var fruits = ['Apples', 'Oranges', 'Pears', 'Bananas'];  
var myArr=[1,2,3];  
myArr = fruits;  
  
console.log(myArr); //prints Apples, Oranges, Pears, Bananas
```

```
var fruits = ['Apples', 'Oranges', 'Pears', 'Bananas'];  
  
fruits[1] = 'Blueberries'  
  
console.log(fruits); //prints Apples, Blueberries, Pears, Bananas
```

ARRAY LENGTH

What if I would like to know how long my array is (how many elements)?

```
var fruits = ['Apples', 'Oranges', 'Pears', 'Bananas'];  
  
console.log(fruits.length); //prints 4
```

ARRAYS

ITERATE OVER ARRAY

Allows you to run code using each element from the array as a value

Syntax:

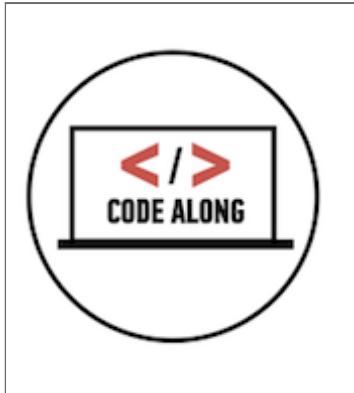
```
var myArr = ['string one', 'string two', 'string three', 'string four'];
```

Native `Array.forEach` function:

```
myArr.forEach(function(element, index) {  
    console.log(element)  
});
```

OR with jQuery:

```
$.each(myArr, function(index, element) {  
    console.log(element);  
});
```

ITERATING OVER ARRAYS

MORE ON ARRAYS

For many more Array methods

see: **https://developer.mozilla.org/en-US/docs/JavaScript/Reference/Global_Objects/Array**



CAROUSEL

