GENERAL ASSEMBLY

FRONT-END WEB DEVELOPMENT

JavaScript - Arrays and Loops

Emerson Taymor

co-founder, philosophie

ARRAYS AND LOOPS

- Creating Arrays
- Array Methods
- Iterating Over Arrays Using Loops

WHAT IS AN ARRAY?

An array is a list-like object that can store a set of data

The size of an array is not fixed, items can be added or removed at any point

The data types of an arrays elements are not fixed either.

Arrays have several built-in methods and functions

```
//declaring an empty array using the Array constructor.
var myArr = new Array();

//declaring an empty array using literal notation.
var myArr = [ ];

//Arrays are filled with elements: i.e. myArr3 = [element, anotherElement];
//Elements can be strings, numbers, or boolean.
myArr = ['Hello', , 54.3, true];

//If you leave a blank spot in an array it creates a blank shelf space (undefined) placeholder.
```

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

//Array elements can be fetched by their index number (starts from 0).
console.log(myArr[0]); //prints Hello
console.log(myArr[1]); //prints undefined
console.log(myArr[2]); //prints 54.3
console.log(myArr[3]); //prints true

//We can insert new values into any space in the array using the positions index.
myArr[1] = 'Stuff';
```

ASSOCIATIVE ARRAYS

Associative arrays allow us to define the index as a string.

We can use associative arrays as a “Key: Definition” list

They are nice because it can be easier to remember a string than a number and also allow us to match things together (favoriteColor: orange, favoriteFood: BBQ)

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

//Associative arrays allow us to define the index as a string.

myArr['Greeting'] = myArr[0];

//Now we can refer to that specific position by using its association.

console.log(myArr['Greeting']); //prints Hello
```

```
//We can overwrite all the elements of an array simply by giving the array new values. Or passing one array into another.

var fruits = ['Apples', 'Oranges', 'Pears', 'Bananas'];

myArr = fruits;

console.log(myArr); //prints Apples, Oranges, Pears, Bananas
```

```
myArr = ['Apples', 'Oranges', 'Pears', 'Bananas'];

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

console.log(myArr.length); //prints 4

//To get the last elements index position I can subtract one (remember indexes start with zero instead of one).

var pos = myArr.length - 1;
console.log(myArr[pos]); //prints Bananas
```

```
myArr = ['Apples', 'Oranges', 'Pears', 'Bananas'];
//We can insert on to the end of an Array simply by using the push method.

myArr.push('Strawberries'); // you can push multiple items onto the end by coma separating if you wish.

console.log(myArr); //prints Apples, Oranges, Pears, Bananas, Strawberries
```

```
myArr = ['Apples', 'Oranges', 'Pears', 'Bananas', ' Strawberries',];

//you can pull the last element off the end using the pop method.
myArr.pop();

console.log(myArr);
//prints Apples, Oranges, Pears, Bananas.
//Notice Strawberries is now missing.
```

```
myArr.splice(2, 0, 'Tiger'); //This goes to index position 2 and after it removes 0 (none) and adds new value of 'Tiger'.

console.log(myArr); //prints Apples, Oranges, Tiger, Pears, Bananas where previously was Apples, Oranges, Pears, Bananas. Tiger has been inserted After Oranges and the items that followed have been bumped forward 1 index.
```

For many more Array methods see:

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

ITERATIONS

Loops allow us to traverse over data so we can get more than one piece of information at a time. We will literally iterate over each type of content until we are satisfied.

We can use loops with arrays to print or modify all (or some) of the information inside an array

There are two main types of loops: “for” and “while”

```
//A for loop repeats until a specified condition evaluates to false.

//SYNTAX: for ([initialExpression]; [condition]; [incrementExpression])
{statement}

var vegetables = ['Broccoli','Peas','Carrots'];

for (var i = 0; i < vegetables.length; i++) {
      console.log(vegetables[i]);
}

//prints Broccoli, Peas, Carrots
```

```
//A while statement executes its statements as long as a specified condition evaluates to true.

//SYNTAX: do {statement} while (condition);

var cars = ['Corvette','Mustang','Porsche'];
var i = 0;

do {
     console.log(cars[i]);
     i += 1;
}
while (i < cars.length);    //prints Corvette, Mustang, Porsche
```

```
//SYNTAX: while (condition){statement}

var fish = ['Snapper', 'Tuna', 'Salmon'];
var i = 0;

while (i < fish.length) {
      console.log(fish[i]);
      i += 1;
}

//prints Snapper, Tuna, Salmon
```