



# FEWD - ARRAYS

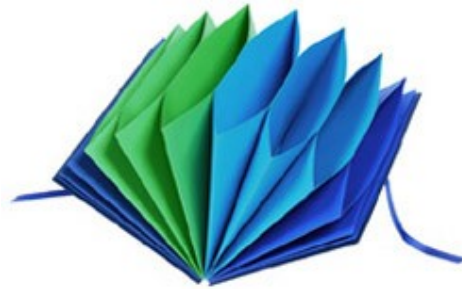
**DANIEL GILLESPIE**

@gillespium | #FEWD

# AGENDA

- Collection Of Data
- Manipulating Collections

# ARRAYS COLLECTIONS



# 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?

- List-like objects
- Store multiple values in a single variable
- Collect similar data together, and later be able to manipulate

# 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

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

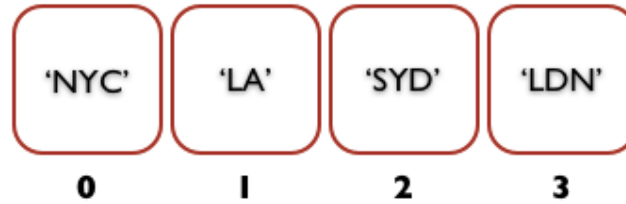
- Arrays are filled with elements: i.e. `myArr3 = [element, anotherElement];`
- Elements can be strings, numbers, or booleans.



# DECLARING ARRAYS

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

# ARRAYS INDEXING



# ARRAYS INDEXING

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

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

# ARRAYS INDEXING

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

```
myArr[1] = 'Stuff';
```

# ARRAYS INDEXING

We can overwrite all the elements of an array simply by giving the array new values or by setting an array equal to a different array.

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

# ARRAY LENGTH

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

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

# ARRAY LENGTH

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

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



# ARRAYS



# ITERATE OVER ARRAY

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

```
Array.forEach
```

# ARRAY LENGTH

```
var fruits=['Banana','Apple','Pear']  
  fruits.forEach(function(element,index){  
    console.log(element,index);  
  });
```

# MORE ON ARRAYS

For many more Array methods  
see: [https://developer.mozilla.org/en-US/docs/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/en-US/docs/JavaScript/Reference/Global_Objects/Array)



**CAROUSEL**

