Collections in JavaScript

Our Goals

- Identify the types of collections in JavaScript
- Talk about the differences between them
- Identify the common methods and properties that are available on those collections

There are two types

- Arrays
- Objects

Both are very common. Arrays are like grocery lists, objects are like dictionaries

The Array

- Holds an ordered list, filled with any type of data
- It is zero-based - which causes a lot of problems

Why would you use Arrays?

- To keep an ordered list of anything
- If you need **iteration**

Creating Arrays

```
const emptyArray = [];

const randomNumbers = [ 12, 42, 1, 3, 92 ];

const rainbowColours = ['red', 'orange', 'yellow', 'green'];

const weirdInstruments = [
    "Badgermin",
    "The Great Stalacpipe Organ",
    "Stylophone",
    "Ondes Martenot",
    "Sharpischord",
    "Hydraulophone",
    "Pyrophone"
];
```

Accessing Elements

```
const weirdInstruments = [
    "Badgermin",
    "The Great Stalacpipe Organ",
    "Stylophone",
    "Ondes Martenot",
    "Sharpischord",
    "Hydraulophone",
    "Pyrophone"
];

weirdInstruments[0];
weirdInstruments[5];
weirdInstruments[ weirdInstruments.length - 1 ];
```

Accessing Elements

```
const rainbowColours = [
    'red',
    'orange',
    'yellow',
    'blue',
    'green'
];

const firstColour = rainbowColours[0];
const secondColour = rainbowColours[1];
const lastColour = rainbowColours[rainbowColours.length - 1];
```

Reassigning Values

```
const rainbowColours = [
    'red',
    'orange',
    'yellow',
    'blue',
    'green'
];

rainbowColours[0] = "Indigo";
rainbowColours[1] = "Purple";
rainbowColours[5] = "Violet";
```

Adding Elements

```
const fruits = ["Orange", "Banana", "Apple"];

// Adding to the end
fruits.push( "Kiwi Fruit" );

// Adding to the start
fruits.unshift( "Mango" );
```

Removing Elements

```
const fruits = ["Mango", "Orange", "Banana", "Apple", "Kiwi Fruit"];

// Removing the last element
fruits.pop();

// Removing the first element
fruits.shift();
```

Finding an item

```
const fruits = [
    "Mango",
    "Orange",
    "Banana",
    "Apple",
    "Kiwi Fruit"
];

const indexOfMango = fruits.indexOf( "Mango" ); // => 0
const indexOfBanana = fruits.indexOf( "Banana" ); // => 2
const indexOfKiwi = fruits.indexOf( "Kiwi Fruit" ); // => 4

const mango = fruits[ indexOfMango ]; // => "Mango"

fruits.includes( "Spaghetti" ); // => false
fruits.indexOf( "Spaghetti" ); // => -1
```

Removing an item by index

```
const fruits = [
    "Mango",
    "Orange",
    "Banana",
    "Apple",
    "Kiwi Fruit"
];

const bananaIndex = fruits.indexOf( "Banana" );
fruits.splice( bananaIndex, 1 );
```

Copying an array

```
const fruits = [
    "Mango",
    "Orange",
    "Banana",
    "Apple",
    "Kiwi Fruit"
];

const newFruits = fruits.slice();
// Copy everything

const twoFruits = fruits.slice( 1, 3 );
// Copy from index 1 to index 3
```

Looping through an array

```
const fruits = [
    "Mango",
    "Orange",
    "Banana",
    "Apple",
    "Kiwi Fruit"
];

for ( let i = 0; i < fruits.length; i++ ) {
  const fruit = fruits[ i ];
  console.log( fruit );
}
```

Review

```
const fruits = [ "Mango", "Orange" ];

const firstFruit = fruits[0];
// => "Mango"

const secondFruit = fruits[fruits.length - 1];
// => "Orange"

fruits.pop(); // Remove the last element
fruits.push("Orange"); // Add Orange to the end

fruits.shift(); // Remove the first element
fruits.unshift("Mango"); // Add Mango to the start

for (let i = 0; i < fruits.length; i++) {
    console.log( fruits[i] );
}
```

Have a crack at these exercises

Homework

- Array and Function Homework
- Array Documentation
- Speaking Javascript: Arrays
- Eloquent Javascript: Arrays
- Javascript.info's Description

Objects

Objects

- Contain an unordered list of definitions
  - i.e. "associate a name with a value"
  - or "create a mapping between a name and a value"
- In other languages, known as:
  - hash
  - dictionary
  - associative array
  - map
- It's not integer-based, it's word-based
- They are made up of key-value pairs

Why would you use Objects?

- Give names to values - clearer than numerical indexes to array element
- Work with large amounts of data effectively
- Easily create highly structured data
- Encapsulation - grouping functionality
- GOTCHA WARNING: objects behave strangely when compared with ===

Creating Objects

```
const emptyObject = {};

const movie = {
    name: "Satantango",
    director: "Bela Tarr",
    duration: 432
};

const bookSeries = {
    name: "In Search of Lost Time",
    author: "Marcel Proust",
    books: [
        "Swann's Way",
        "The Guermantes Way",
        "Sodom and Gomorrah",
        "The Prisoner",
        "The Fugitive",
        "Time Regained"
    ]
};
```

Accessing Values

```
const movie = {
    name: "Satantango",
    director: "Bela Tarr",
    duration: 432
};

const movieName = movie.name;
const movieDirector = movie.director;
const movieDuration = movie.duration;
```

Accessing Values

```
const movie = {
    name: "Satantango",
    director: "Bela Tarr",
    duration: 432
};

const movieName = movie["name"];
const movieDirector = movie["director"];
const movieDuration = movie["duration"];

movie["gobbledygook"]; // => undefined
```

Reassigning Values

```
const movie = {
    name: "Satantango",
    director: "Bela Tarr",
    duration: 432
};

movie.name = "Sátántangó";
movie.director = "Béla Tarr";
movie.duration = 534;
```

Adding new values

```
const movie = {
    name: "Satantango",
    director: "Bela Tarr",
    duration: 432
};

movie.language = "Hungarian";
movie.rating = 21412523224616982;
movie.parts = 12;
```

Destroying Properties

```
const bestCamera = {
    brand: "Leica",
    model: "M3",
    year: 1955,
    memoryCard: "SD"
};

delete bestCamera.memoryCard;
```

Looping through Objects

```
const movie = {
    name: "Satantango",
    director: "Bela Tarr",
    duration: 432
};

for ( const key in movie ) {
    console.log( key + ": " + movie[key] );
}

const movieKeys = Object.keys( movie );
// => ["name", "director", "duration"]
```

Nested Objects

```
const explorer = {
    firstName: "Jacques",
    lastName: "Cousteau",
    birth: {
        day: 11,
        month: 6,
        year: 1910
    }
};

const birthDay = explorer.birth.day;
const birthMonth = explorer.birth.month;
const birthYear = explorer.birth.year;

const first = explorer.firstName;
const last = explorer.lastName;
```

Methods

```
const explorer = {
    firstName: "Jacques",
    lastName: "Cousteau",
    displayPurpose: function () {
        console.log( "Hello World" );
    }
};

explorer.displayPurpose();
```

Complex Data Structures

```
const marxFamily = [
    { name: "Groucho", birthYear: 1890 },
    { name: "Harpo", birthYear: 1888 },
    { name: "Chico", birthYear: 1887 },
    { name: "Zeppo", birthYear: 1901 },
    { name: "Gummo", birthYear: 1893 }
];

for ( let i = 0; i < marxFamily.length; i++ ) {
    const brother = marxFamily[ i ];
    console.log( brother.name, brother.birthYear );
}
```

Our first intro to "this"

```
const explorer = {
  firstName: "Jacques",
  lastName: "Cousteau",
  displayFullName: function () {
    const fullName = this.firstName + " " + this.lastName;
    return fullName;
  },
  talkTo: function (name) {
    const greeting = this.firstName + " says hi to " + name;
    return greeting;
  }
};

explorer.displayFullName();
```

Have a crack at these exercises

Homework

- Geometry Homework
- Read this: Sitepoint
- Read this: Speaking JavaScript
- Read this: Eloquent JavaScript
- Watch this: Code Academy