

# 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

```
var emptyArray = [];  
  
var randomNumbers = [ 12, 42, 1, 3, 92 ];  
  
var rainbowColours = [ 'red', 'orange', 'yellow', 'green' ];  
  
var weirdInstruments = [  
    "Badgermin",  
    "The Great Stalacpipe Organ",  
    "Stylophone",  
    "Ondes Martenot",  
    "Sharpischord",  
    "Hydraulophone",  
    "Pyrophone"  
];
```

# Accessing Elements

```
var weirdInstruments = [  
    "Badgermin",  
    "The Great Stalacpipe Organ",  
    "Stylophone",  
    "Ondes Martenot",  
    "Sharpischord",  
    "Hydraulophone",  
    "Pyrophone"  
];  
  
weirdInstruments[0];  
weirdInstruments[5];  
weirdInstruments[ weirdInstruments.length - 1 ];
```

# Accessing Elements

```
var rainbowColours = [  
    'red',  
    'orange',  
    'yellow',  
    'blue',  
    'green'  
];  
  
var firstColour = rainbowColours[0];  
var secondColour = rainbowColours[1];  
var lastColour = rainbowColours[rainbowColours.length - 1];
```



# Reassigning Values

```
var rainbowColours = [  
    'red',  
    'orange',  
    'yellow',  
    'blue',  
    'green'  
];  
  
rainbowColours[0] = "Indigo";  
rainbowColours[1] = "Purple";  
rainbowColours[5] = "Violet";
```

# Adding Elements

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

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

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

# Removing Elements

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

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

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

# Finding an item

```
var fruits = [  
    "Mango",  
    "Orange",  
    "Banana",  
    "Apple",  
    "Kiwi Fruit"  
];  
  
var indexOfMango = fruits.indexOf( "Mango" ); // => 0  
var indexOfBanana = fruits.indexOf( "Banana" ); // => 2  
var indexOfKiwi = fruits.indexOf( "Kiwi Fruit" ); // => 4  
  
var mango = fruits[ indexOfMango ]; // => "Mango"  
  
fruits.includes( "Spaghetti" ); // => false  
fruits.indexOf( "Spaghetti" ); // => -1
```

# Removing an item by index

```
var fruits = [  
    "Mango",  
    "Orange",  
    "Banana",  
    "Apple",  
    "Kiwi Fruit"  
];  
  
var bananaIndex = fruits.indexOf( "Banana" );  
fruits.splice( bananaIndex, 1 );
```

# Copying an array

```
var fruits = [  
    "Mango",  
    "Orange",  
    "Banana",  
    "Apple",  
    "Kiwi Fruit"  
];  
  
var newFruits = fruits.slice();  
// Copy everything  
  
var twoFruits = fruits.slice( 1, 3 );  
// Copy from index 1 to index 3
```

# Looping through an array

```
var fruits = [  
    "Mango",  
    "Orange",  
    "Banana",  
    "Apple",  
    "Kiwi Fruit"  
];  
  
for ( var i = 0; i < fruits.length; i++ ) {  
    var fruit = fruits[ i ];  
    console.log( fruit );  
}
```

# Review

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

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

var 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 (var 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

- A dictionary
- Contains an unordered list of definitions
- It's not integer-based, it's word-based
- They are made up of key-value pairs

# Why would you use Objects?

- Encapsulation - grouping functionality
- Give names to values
- Work with large amounts of effectively

# Creating Objects

```
var emptyObject = {};  
  
var movie = {  
  name: "Satantango",  
  director: "Bela Tarr",  
  duration: 432  
};  
  
var bookSeries = {  
  name: "In Search of Lost Time",  
  author: "Marcel Proust",  
  books: [  
    "Swann's Way",  
    "In the Shadow of Young Girls in Flower",  
    "The Guermentes Way",  
    "Sodom and Gomorrah",  
    "The Prisoner",  
    "The Fugitive",  
    "Time Regained"  
  ]  
};
```

# Accessing Values

```
var movie = {  
    name: "Satantango",  
    director: "Bela Tarr",  
    duration: 432  
};  
  
var movieName = movie.name;  
var movieDirector = movie.director;  
var movieDuration = movie.duration;
```

# Accessing Values

```
var movie = {  
  name: "Satantango",  
  director: "Bela Tarr",  
  duration: 432  
};  
  
var movieName = movie["name"];  
var movieDirector = movie["director"];  
var movieDuration = movie["duration"];  
  
movie["gobbledygook"]; // => undefined
```



# Reassigning Values

```
var movie = {  
  name: "Satantango",  
  director: "Bela Tarr",  
  duration: 432  
};  
  
movie.name = "Sátántangó";  
movie.director = "Béla Tarr";  
movie.duration = 534;
```

# Adding new values

```
var movie = {  
  name: "Satantango",  
  director: "Bela Tarr",  
  duration: 432  
};  
  
movie.language = "Hungarian";  
movie.rating = 21412523224616982;  
movie.parts = 12;
```

# Destroying Properties

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

# Looping through Objects

```
var movie = {  
  name: "Satantango",  
  director: "Bela Tarr",  
  duration: 432  
};  
  
for ( var key in movie ) {  
  console.log( key + ": " + movie[key] );  
}  
  
var movieKeys = Object.keys( movie );  
// => ["name", "director", "duration"]
```

# Nested Objects

```
var explorer = {  
  firstName: "Jacques",  
  lastName: "Cousteau",  
  birth: {  
    day: 11,  
    month: 6,  
    year: 1910  
  }  
};  
  
var birthDay = explorer.birth.day;  
var birthMonth = explorer.birth.month;  
var birthYear = explorer.birth.year;  
  
var jacques = explorer.firstName;  
var cousteau = explorer.lastName;
```

# Methods

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

# Complex Data Structures

```
var marxFamily = [  
  { name: "Groucho", birthYear: 1890 },  
  { name: "Harpo", birthYear: 1888 },  
  { name: "Chico", birthYear: 1887 },  
  { name: "Zeppo", birthYear: 1901 },  
  { name: "Gummo", birthYear: 1893 }  
];  
  
for ( var i = 0; i < marxFamily.length; i++ ) {  
  var brother = marxFamily[ i ];  
  console.log( brother.name, brother.birthYear );  
}
```

# Our first intro to This

```
var explorer = {
  firstName: "Jacques",
  lastName: "Cousteau",
  displayFullName: function () {
    var fullName = this.firstName + " " + this.lastName;
    return fullName;
  },
  talkTo: function (name) {
    var 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](#)