JAVASCRIPT OBJECT METHODS COURSE

APRIL 2022

YONATAN BENEZRA

Just like String or array methods, Object methods can help us use, change, and retrieve data about objects in an easy manner.

In this lesson, we will learn about the following methods:

- Object.keys()
- Object.entries()
- Object.values()
- Object.is()

***Note:**
You can find a list of Object methods in the “object methods cheat sheet” lesson.
All of the functions we will discuss are built-in JavaScript functions.

Object.keys

This method returns an array of the given object’s property names in the same order as we get with a standard loop.

The Object.keys() method takes an object as an argument and returns an array of strings representing all the enumerable properties of a given object.

The Object keys() function returns an array of elements containing strings which correspond to the enumerable properties found directly upon the object.

The order of the properties within the array is the same as would be given manually by an object in the loop applied to the properties.

Syntax
Object.keys(obj)

Arguments
The obj parameter is required, and its properties are to be returned.

Return Value
The return value is an array of strings representing all the enumerable properties of a given object.

Example:

```
let obj = {
   name: 'Charizard',
   type: 'fire'
} ;
console.log(Object.keys(obj));
```

So, in the above example, we get an array of object keys.
We can do the same with an array. We can get the array keys as well.

```
// app.js

let companies= [
   'apple',
   'microsoft',
   'amazon',
   'alphabet',
   'alibaba'
];
console.log(Object.keys(companies));
```

Here we have defined an array. Next, we will get the keys to each element in the returned array.

Exercise:

Get the length of the object shown in the above example.

Object.keys() and Array.forEach()

JavaScript does not have an Object.forEach() method. Therefore, to iterate through all the keys, use the Array.forEach() function, as shown in the following example.

```
let data = {
  real_name: 'Donald Glover',
  character_name: 'Troy Barnes',
  series: 'Community'
};
Object.keys(data).forEach(item => {
```

```
  console.log(item);
});
```

We can use the Object.values() function if we want to receive the object's values.

Sorting Objects by keys

All methods that iterate over property keys do so in the same order:

- First, the array indices are sorted numerically.
- Then, all string keys (that are not indices) remain in the order in which they were created.
- Finally, all symbols remain in the order in which they were created.

In order to sort the Javascript Object Keys(), the following functions must be used:

- Array.forEach()
- Array.Sort()

Exercise:

Sort the following object using the keys, foreach, and sort methods.

Hint # 1:

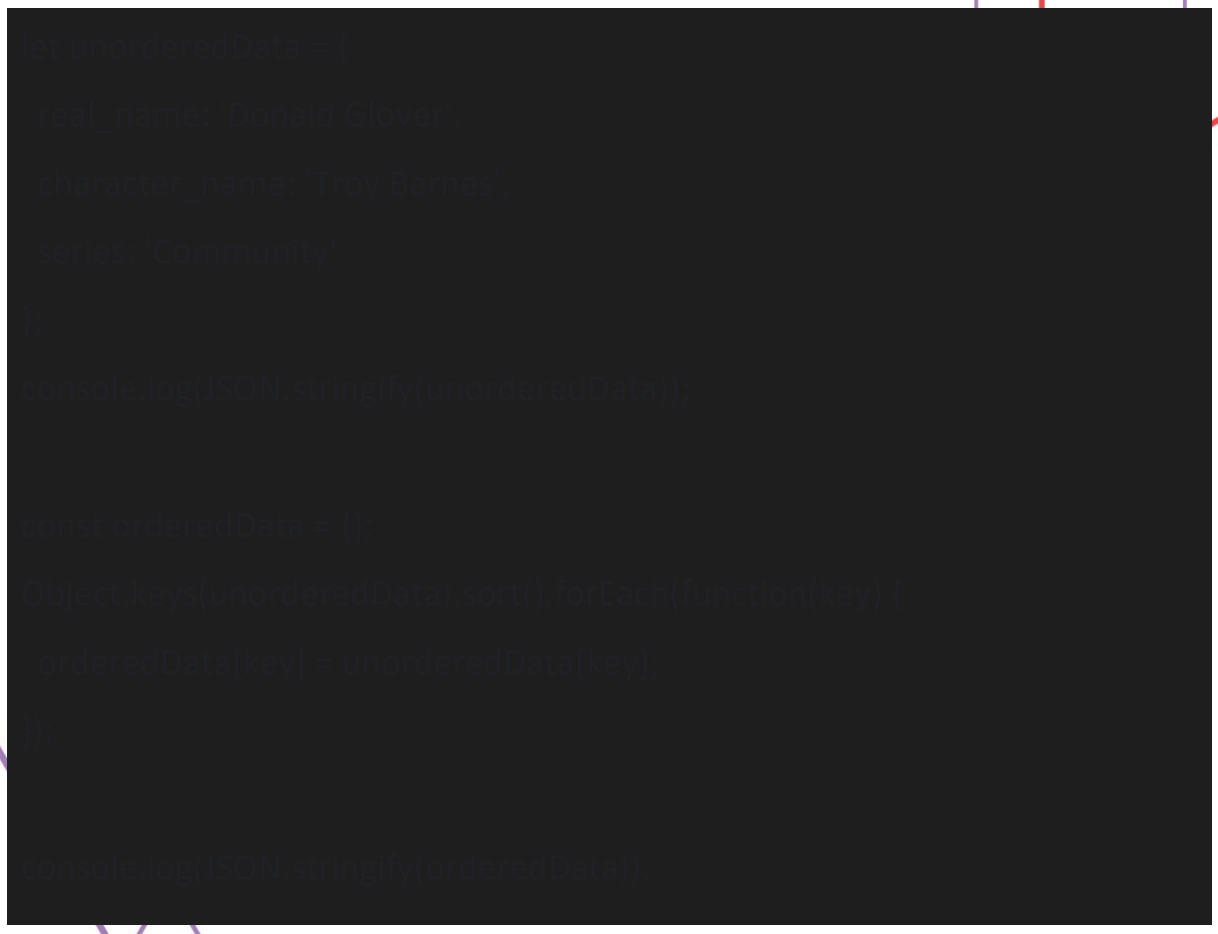
```
const orderedData = {};
```

Hint # 2:

```
Object.keys(unorderedData).sort().forEach(function(key) {});
```

Hint # 3:

```
Object.keys(unorderedData).sort().forEach(function(key) {
  orderedData[key] = unorderedData[key];
});
```



Object.entries()

The JavaScript Object.entries() method returns an array of key-value pairs of object properties.

The syntax of the entries() method is:
Object.entries(obj)

The entries() method, being a static method, is called using the Object class name.

Entries() Parameters

The entries() method takes in:

- obj - The object whose enumerable string-keyed property key and value pairs are to be returned.

Return value from entries()

- Returns an array of the given object's own enumerable string-keyed property [key, value] pairs.

***Note:**
The ordering of the properties is the same as they would be if you were to loop over them manually using for...in loop.

Example: Using Object.entries()

```
const obj = { name: "Yonatan", age: 25, location: "Israel" };
console.log(Object.entries(obj)); // [ [ 'name', 'Yonatan' ], [ 'age', 25 ], [
'location', 'Israel' ] ]

// Array-like objects
const obj1 = { 0: "A", 1: "B", 2: "C" };
console.log(Object.entries(obj1)); // [ [ '0', 'A' ], [ '1', 'B' ], [ '2', 'C' ] ]

// random key ordering
const obj2 = { 42: "a", 22: "b", 71: "c" };
// [ [ '22', 'b' ], [ '42', 'a' ], [ '71', 'c' ] ] -> arranged in numerical order of keys
console.log(Object.entries(obj2));

// string -> from ES2015+, non-objects are coerced to object
const string = "code";
console.log(Object.entries(string)); // [ [ '0', 'c' ], [ '1', 'o' ], [ '2', 'd' ], [ '3', 'e' ] ]

// primitive types have no properties
console.log(Object.entries(55)); // []
```

```
// Iterating through key-value of objects
for (const [key, value] of Object.entries(obj)) {
  console.log(`${key}: ${value}`);
}
```

Output

```
[ [ 'name', 'Yonatan' ], [ 'age', 25 ], [ 'location', 'Israel' ] ]
[ [ '0', 'A' ], [ '1', 'B' ], [ '2', 'C' ] ]
[ [ '22', 'b' ], [ '42', 'a' ], [ '71', 'c' ] ]
[ [ '0', 'c' ], [ '1', 'o' ], [ '2', 'd' ], [ '3', 'e' ] ]
[]
name: Yonatan
age: 25
location: Israel
```

Object.values()

The Object.values() accepts an object and returns its own enumerable property's values as an array.

Example:

```
const person = {
    firstName: 'John',
    lastName: 'Doe',
    age: 25
};

const profile = Object.values(person);

console.log(profile); // [ 'John', 'Doe', 25 ]
```

Object.in()

Exercise:

Create a function that determines which variables have the same value using Object.in().

Learn how to do this at the following link:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/is