JAVASCRIPT

HOW TO LOOP THROUGH OBJECT



@satyam.js

In this article we will learn how to loop through object.

It is very common to iterate over an object in JavaScript.

There is more than one way to do this, and in this lesson we will discuss all the ways.

Methods that can be used to loop through objects in JavaScript:

- For loop
- for ... in
- Object.keys method
- Object.values method
- Object.entries method

Problem

We must understand the problem first before solving it. We want to iterate on the object right?

Now imagine that there is an object like this:

```
const myObject = {
   name: "John",
   age: 30,
}
```

And we want to iterate over the elements to give us this

```
output:
name John
age 30
```

How do you solve that problem?

As we said before, there is more than one solution

How to loop through an object with for...in loop?

We can use **for** .. **in** loop to iterate over the object keys.

```
// Object
const myObject = {
   name: "John",
   age: 30,
}

// Print keys
for(const key in myObject)
{   console.log(key);
}
Output

name
age
```

Well, now we have half the solution, which is to print the keys, but what about the values?

```
// Object
const myObject = {
    name: "John",
    age: 30,
}

// Print keys & values
for(const key in myObject) {
    console.log(key,
}yObject[key]);
Output

name John
age 30
```

How to loop through an object with Object.keys() method?

We can use **Object.keys()** method to iterate over the object keys as well.

It takes the object that you want to loop over as an argument and returns an array of keys.

```
// Object
const myObject = {
    name: "John",
    age: 30,
}

// Keys
const keys = Object.keys(myObject);

// Print Keys
console.log(keys);
Output

[ 'name', 'age']
```

As you can see, it returned an array with all the keys, and this means that we can iterate over the array in any way.

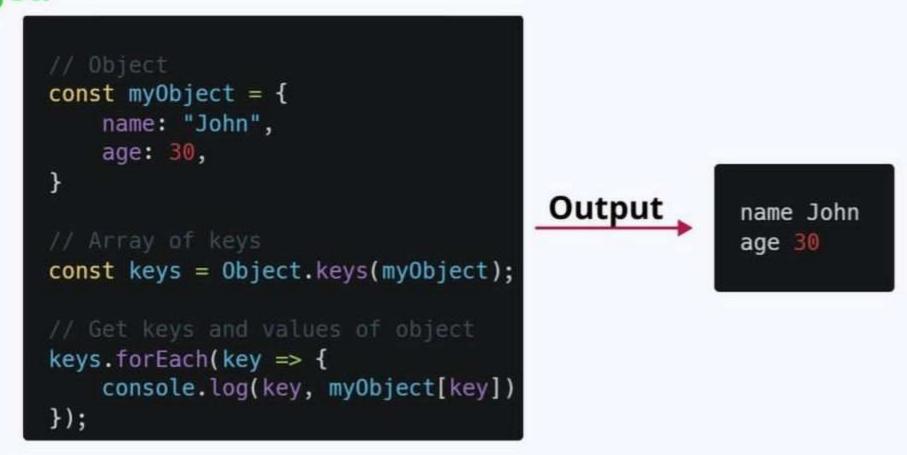
```
// Object
const myObject = {
    name: "John",
    age: 30,
}

// Array of keys
const keys = Object.keys(myObject);

// Iterate over keys with `forEach`
keys.forEach(key => {
    console.log(key);
});
Output

name
age
```

Well, **now half of the problem is solved**, because we made iterations on the keys, **but what about the values?**As long as we can return the keys, **the values will be easy to get**.



How to loop through an object with Object.values() method?

We can use **Object.values() method** to itrate over object values.

The Object.values() method returns an array of a given object's own enumerable string-keyed property values.

```
// Object
const myObject = {
    name: "John",
    age: 30,
}

// Array of values
const values = Object.values(myObject);

// Iterate over values
values.forEach((value) => {
    console.log(value)
});
Output

John
30
```

How to loop through an object with Object.entries() method?

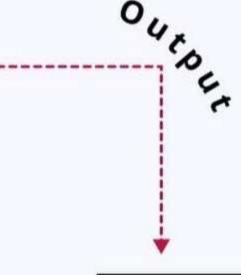
We can use **Object.entries()** to iterate over the object keys and values.

The Object.entries() method returns an array of a given object's own enumerable string-keyed property key-value pairs.

```
// Object
const myObject = {
    name: "John",
    age: 30,
}

// Array of keys and values
const entries = Object.entries(myObject);

// Iterate
for (let i = 0; i < entries.length; i++) {
    for (let n = 0; n < entries[i].length; n++) {
        console.log(entries[i][n]);
    }
}</pre>
```



name John age