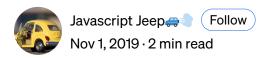
How to Use Object.keys in JavaScript





Consider an object:

```
var user = {
  name: "Jagathish",
```

1 of 4 7/23/21, 3:42 PM

```
age: 20
}
```

In the user object, the name and age are the keys of the object. Keys are also referred to as object "properties". We can access the property value by <code>obj.propertyName</code> or <code>obj[propertyName]</code>.

The <code>object.keys()</code> method returns an array of strings of a given object's own property/key names. The following is what we get for our <code>user</code> object:

```
Object.keys(user); // ["name", "user"]
```

Let's look at another example:

```
var user = {
  name : "Jagathish",
  age : 20,
  getAge() {
    return this.age;
  }
}
Object.keys(user); // ["name", "age", "getAge"]
```

The key names are returned for all properties, whether it's a function or primitive variable type. The order of key names in the array will be the same as they were in the object.

Syntax

```
Object.keys(obj)
```

Parameter: obj

2 of 4 7/23/21, 3:42 PM

The only parameter the <code>object.keys()</code> function takes is an object itself.

- The object of which the enumerable's own properties are to be returned.
- If we pass an empty object, then it returns an empty array.
- If we don't pass any argument (which is equivalent to passing undefined) or if we pass null, then it throws an error.

Return value: Array of strings

An array of strings that represent all the enumerable properties of the given object.

When we pass a non-object except undefined, it will be coerced to an object.

```
Object.keys(123) // []
Object.keys(123.34) // []
Object.keys("hi") // ["0" , "1"]
```

Please make a donation <u>here</u>. 80% of your donation is donated to someone needs food ①. Thanks in advance.

3 of 4 7/23/21, 3:42 PM