JS JavaScript TUTORIAL

A Basic Guide to Enumerable Properties of an Object in JavaScript

Summary: in this tutorial, you will learn about JavaScript enumerable properties of an object.

Introduction to JavaScript enumerable properties

Enumerable properties are iterated using the `for...in` loop or `Objects.keys()` method.

In JavaScript, an object is an unordered list of key-value pairs. The key is usually a string or a symbol. The value can be a value of any primitive type (string, boolean, number, undefined, or null), an object, or a function.

The following example creates a new object using the object literal syntax:

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

The `person` object has two properties: `firstName` and `lastName` .

An object property has several internal attributes including `value` , `writable` , `enumerable` and `configurable` . See the Object properties for more details.

The `enumerable` attribute determines whether or not a property is accessible when the object's properties are enumerated using the `for...in` loop or `Object.keys()` method.

By default, all properties created via a simple assignment or via a property initializer are enumerable. For example:

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

person.age = 25;

for (const key in person) {
    console.log(key);
}
```

Output:

```
firstName
lastName
age
```

In this example:

- The `firstName` and `lastName` are enumerable properties because they are created via a property initializer.
- The `age` property is also enumerable because it is created via a simple assignment.

To change the internal `enumerable` attribute of a property, you use the `Object.defineProperty()` method. For example:

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

person.age = 25;

Object.defineProperty(person, 'ssn', {
    enumerable: false,
    value: '123-456-7890'
});

for (const key in person) {
    console.log(key);
}
```

Output:

```
firstName
lastName
age
```

In this example, the `ssn` property is created with the `enumerable` flag sets to `false`, therefore it does not show up in the `for...in` loop.

ES6 provides a method `propertyIsEnumerable()` that determines whether or not a property is enumerable. It returns `true` if the property is enumerable; otherwise `false`. For example:

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

person.age = 25;

Object.defineProperty(person, 'ssn', {
    enumerable: false,
    value: '123-456-7890'
});


console.log(person.propertyIsEnumerable('firstName')); // => true
console.log(person.propertyIsEnumerable('lastName')); // => true
console.log(person.propertyIsEnumerable('age')); // => true
console.log(person.propertyIsEnumerable('ssn')); // => false
```

Summary

- A property is enumerable if it has the `enumerable` attribute set to `true`. The `obj.propertyIsEnumerable()` determines whether or not a property is enumerable.
- A property created via a simple assignment or a property initializer is enumerable.