Javascript Object Methods

Summary: in this tutorial, you'll learn about JavaScript object methods and how to define methods for an object.

Introduction to the JavaScript object methods

An object is a collection of key/value pairs or properties. When the value is a function, the property becomes a method. Typically, you use methods to describe the object's behaviors.

For example, the following adds the `greet` method to the `person` object:

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

person.greet = function () {
    console.log('Hello!');
}

person.greet();
```

Output:

```
Hello!
```

In this example:

- First, use a function expression to define a function and assign it to the `greet` property of the `person` object.
- Then, call the method `greet()` method.

Besides using a function expression, you can define a function and assign it to an object like this:

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

function greet() {
    console.log('Hello, World!');
}

person.greet = greet;

person.greet();
```

In this example:

- First, define the `greet()` function as a regular function.
- Second, assign the function name to the `greet` property of the `person` object.
- Third, call the `greet()` method.

Object method shorthand

JavaScript allows you to define methods of an object using the object literal syntax as shown in the following example:

```
let person = {
    firstName: 'John',
    lastName: 'Doe',
    greet: function () {
        console.log('Hello, World!');
    }
};
```

ES6 provides you with the concise method syntax that allows you to define a method for an object:

```
let person = {
    firstName: 'John',
    lastName: 'Doe',
    greet() {
        console.log('Hello, World!');
    }
};

person.greet();
```

This syntax looks much cleaner and less verbose.

The this value

Typically, methods need to access other properties of the object.

For example, you may want to define a method that returns the full name of the person object by concatenating the first name and last name.

Inside a method, the `this` value references the object that invokes the method. Therefore, you can access a property using the `this` value as follows:

```
this.propertyName
```

The following example uses the `this` value in the `getFullName()` method:

```
let person = {
    firstName: 'John',
    lastName: 'Doe',
    greet: function () {
        console.log('Hello, World!');
    },
    getFullName: function () {
        return this.firstName + ' ' + this.lastName;
    }
};


console.log(person.getFullName());
```

Output

```
'John Doe'
```

Check out this tutorial for more information on `this` value.

Summary

- When a function is a property of an object, it becomes a method.