JavaScript Object and Array Methods

An Introduction to JavaScript Objects

JavaScript Objects are collections of properties, where each property is defined as a key-value pair. Objects can hold any type of data, including numbers, strings, arrays, and even other objects.

Example:

```
let person = {
    firstName: "John",
    lastName: "Doe",
    age: 30,
    fullName: function() {
        return this.firstName + " " + this.lastName;
    }
};
```

Removing Properties from Objects

You can remove properties from objects using the delete operator. This permanently removes the specified property from the object.

Example:

```
delete person.age; // Removes the 'age' property from the 'person' object
```

The "this" Keyword in JavaScript Objects

The this keyword refers to the object from which it was called. It's commonly used inside methods to access or modify the object's properties.

Example:

```
let car = {
    brand: "Toyota",
    model: "Camry",
    getDetails: function() {
```

```
    return `Car: ${this.brand} ${this.model}`;
  }
};

console.log(car.getDetails()); // Output: Car: Toyota Camry
```

Linking Functions to Objects

You can link functions to objects by defining them as methods within the object.

Example:

```
let calculator = {
  add: function(a, b) {
    return a + b;
  },
  subtract: function(a, b) {
    return a - b;
  }
};

console.log(calculator.add(5, 3)); // Output: 8
```

Object Constructors

Object constructors are templates for creating multiple objects with the same properties and methods. They are defined using functions.

Example:

```
function Person(firstName, lastName, age) {
  this.firstName = firstName;
  this.lastName = lastName;
  this.age = age;
  this.fullName = function() {
    return this.firstName + " " + this.lastName;
  };
}

let person1 = new Person("Jane", "Doe", 25);
```

```
console.log(person1.fullName()); // Output: Jane Doe
```

Creating New Objects from Existing Ones

You can create new objects from existing ones using the Object.create() method. This method creates a new object, using an existing object as the prototype.

Example:

```
let animal = {
  type: "Invertebrates",
  displayType: function() {
    console.log(this.type);
  }
};

let animal1 = Object.create(animal);
animal1.type = "Fishes";
animal1.displayType(); // Output: Fishes
```

Object Methods

Object methods are functions that are properties of an object. These methods can manipulate the object's properties or perform actions based on the object's data.

Example:

```
let user = {
  name: "Alice",
  age: 27,
  greet: function() {
    return `Hello, my name is ${this.name}`;
  }
};

console.log(user.greet()); // Output: Hello, my name is Alice
```

Freezing Objects

The Object.freeze() method prevents an object from being modified. Once an object is frozen, you cannot add, remove, or change its properties.

Example:

```
let book = {
  title: "1984",
  author: "George Orwell"
};

Object.freeze(book);
book.title = "Animal Farm"; // This will not change the title
console.log(book.title); // Output: 1984
```

The map Method for JavaScript Arrays

The map() method creates a new array by applying a function to each element of the original array.

Example:

```
let numbers = [1, 2, 3, 4, 5];
let squares = numbers.map(function(number) {
  return number * number;
});

console.log(squares); // Output: [1, 4, 9, 16, 25]
```

The reduce and filter Methods for JavaScript Arrays

- The reduce() method executes a reducer function on each element of the array, resulting in a single output value.

Example:

```
let sum = numbers.reduce(function(total, number) {
  return total + number;
}, 0);
```

```
console.log(sum); // Output: 15
```

- The filter() method creates a new array with all elements that pass the test implemented by the provided function.

Example:

```
let evenNumbers = numbers.filter(function(number) {
  return number % 2 === 0;
});

console.log(evenNumbers); // Output: [2, 4]
```

The instanceOf Operator

The instanceof operator tests whether an object is an instance of a particular class or constructor function.

Example:

```
let date = new Date();
console.log(date instanceof Date); // Output: true
```