Object properties and methods

Sure! Let's dive deep into **JavaScript Object Properties and Methods**.

🔷 What is an Object in JavaScript?

An object is a collection of **key-value pairs**, where:

- **Keys** are called **properties**
- **Values** can be any data type — including functions, which are then called **methods**

```
const person = {
  name: "Abhi",
  age: 22,
  greet: function () {
    console.log("Hello!");
  }
};
```

🔷 Object Properties

1. Creating Properties

You can add properties when defining the object:

```
const car = {
  brand: "Tesla",
  model: "Model S"
};
```

Or after creation:

```
car.color = "Red";
```

2. Accessing Properties

- **Dot notation**:

```
console.log(car.brand); // "Tesla"
```

- **Bracket notation** (useful for dynamic keys):

```
const key = "model";
console.log(car[key]); // "Model S"
```

3. Deleting Properties

```
delete car.color;
```

4. Checking Properties

- `in` operator:

```
"brand" in car; // true
```

- `hasOwnProperty()`:

```
car.hasOwnProperty("model"); // true
```

🔷 Object Methods

If a property's value is a **function**, it's called a **method**.

Example:

```
const user = {
  name: "Abhi",
  sayHello: function () {
    console.log(`Hello, I am ${this.name}`);
  }
};

user.sayHello(); // "Hello, I am Abhi"
```

`this` Keyword

Inside a method, `this` refers to the object itself.

🔷 Built-in Object Methods

`Object.keys(obj)`

Returns an array of keys:

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

`Object.values(obj)`

Returns an array of values:

```
Object.values(user); // ['Abhi', ƒ]
```

`Object.entries(obj)`

Returns an array of `[key, value]` pairs:

```
Object.entries(user);
// [['name', 'Abhi'], ['sayHello', ƒ]]
```

`Object.assign(target, source)`

Copies properties from `source` to `target`:

```
const newUser = Object.assign({}, user);
```

`Object.freeze(obj)`

Makes an object **immutable** (you can't change or add properties):

```
Object.freeze(user);
user.name = "New Name"; // won't work
```

`Object.seal(obj)`

Prevents adding/removing properties, but allows modification of existing ones.

🔷 Computed Properties (ES6)

You can define properties dynamically:

```
const prop = "age";
const person = {
  name: "Abhi",
  [prop]: 25
};
```

🔷 Short Syntax for Methods (ES6)

```
const person = {
  name: "Abhi",
  greet() {
    console.log("Hi!");
  }
};
```

🔷 Looping through Properties

Using `for...in`:

```
for (let key in person) {
  console.log(`${key}: ${person[key]}`);
}
```

✅ Summary

Concept	Example
Create property	`obj.name = "Abhi"`
Access property	`obj["name"]` or `obj.name`
Method	`obj.greet = function() {}`
Delete property	`delete obj.name`
Check property	`"name" in obj`, `obj.hasOwnProperty("name")`
Object keys/values	`Object.keys(obj)`, `Object.values(obj)`
Loop	`for (let key in obj)`
Freeze/seal	`Object.freeze(obj)`, `Object.seal(obj)`

If you'd like, I can give you **practice problems** or a **quiz** to reinforce these concepts. Want that?