

🚩 Current Skill Adding and Modifying Object Properties

Adding & Modifying Object Properties

In order to **add a new property to an object** you assign a new value to a property with the **assignment operator (=)**.

For example, we can add a numerical data type to the gimli object as the new age property. Both the dot and bracket notation can be used to add a new object property.

We can also **add a new method to the object** by using the same process. We can then call the newly created method the same way we did before.

```
// Initializing a gimli object
```

```
var gimli = {  
  name: "Gimli",  
  race: "dwarf",  
  weapon: "axe",  
  greet: function() {  
    return `Hi, my name is ${this.name}!`;  
  },  
};
```

```
// Adding a new age property to gimli using the dot notation
```

```
gimli.age = 139;
```

```
// Adding new age property to gimli using the dot notation
```

```
gimli["age"] = 139;
```

```
// Adding a new fight method to gimli
```

```
gimli.fight = function() {  
  return `Gimli attacks with an ${this.weapon}.`;  
}
```

```
console.log(gimli)
```

```
// prints {name: "Gimli", race: "dwarf", weapon: "axe", age: 139, greet: f, fight: f}
```

```
// Calling the newly created method fight
```

```
console.log(gimli.fight());
```

```
// prints "Gimli attacks with an axe."
```

Adding & Modifying Object Properties

Using the same methods we've mentioned in the previous slide, an object's property can be **modified** by assigning a new value to an existing property.

What you should keep in mind at this point is that through the assignment operator = we can also modify the properties and methods of a JavaScript object.

```
// Updating weapon from axe to battle axe
```

```
gimli.weapon = "epic battle axe";
```

```
// Calling the previously created method fight AGAIN
```

```
console.log(gimli.fight());
```

```
// prints "Gimli attacks with an epic battle axe."
```

< Previous

next >