

⚡ Current Skill Creating an Object

Creating an Object

There are two ways to construct an object in JavaScript:

- The **object literal**, which uses curly brackets: {}
- The **object constructor**, which uses the **new** keyword

Both of these approaches will create an empty object. Using object literals is the more common and preferred method because it is less prone to inconsistencies and unexpected results.

```
// Initializing an object literal with curly brackets
```

```
var objectLiteral = {};
```

```
// Initializing an object constructor with new Object
```

```
var objectConstructor = new Object();
```

Creating an Object



Let's create an example object contained in the variable `gimli` to describe a character.

Our new object is `gimli` and it has three properties. Each property consists of a **name: value** pair, also known as key: value pair. `Weapon` is one of the property names which is linked to the property value `"axe"`, a string. It has one method (`greet`) and its value consists of the function's contents.

PS: Within `greet`, you may notice the `"this"` keyword. When using `this` inside of an object, it refers to the current object, in this case `gimli`.



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

```
var gimli = {  
  name: "Gimli",  
  race: "dwarf",  
  weapon: "axe",  
  greet: function() {
```



```
return `Hi, my name is ${this.name}!`;
```

```
},
```

```
};
```

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

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

```
// PS: This output may render differently depending on what console you are using, but
```

[< Previous](#)

[next >](#)

