Context: A Multiplayer RPG

From here starts your journey to learn Object Oriented Programming in JavaScript.
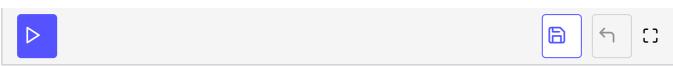
As a reminder, here's the code for our minimalist RPG taken from a previous chapter. It creates an object literal named `aurora` with four properties (`name`, `health`, `strength` and `xp`) and a `describe()` method.

```javascript
const aurora = {
  name: "Aurora",
  health: 150,
  strength: 25,
  xp: 0,

  // Return the character description
  describe() {
    return `${this.name} has ${this.health} health points, ${this
      .strength} as strength and ${this.xp} XP points`;
  }
};

// Aurora is harmed by an arrow
aurora.health -= 20;

// Aurora gains a strength necklace
aurora.strength += 10;

// Aurora learns a new skill
aurora.xp += 15;

console.log(aurora.describe());
```

To make the game more interesting, we'd like to have more characters in it. So here comes Glacius, Aurora's fellow.

```javascript
const glacius = {
  name: "Glacius",
  health: 130,
  strength: 30,
  xp: 0,

  // Return the character description
```

```
  // Return the character description
  describe() {
    return `${this.name} has ${this.health} health points, ${this
      .strength} as strength and ${this.xp} XP points`;
  }
};
```

Our two characters are strikingly similar. They share the same properties, with the only difference being some property values.

You should already be aware that code duplication is dangerous and should generally be avoided. We must find a way to share what's common to our characters.