# Singleton Design Pattern

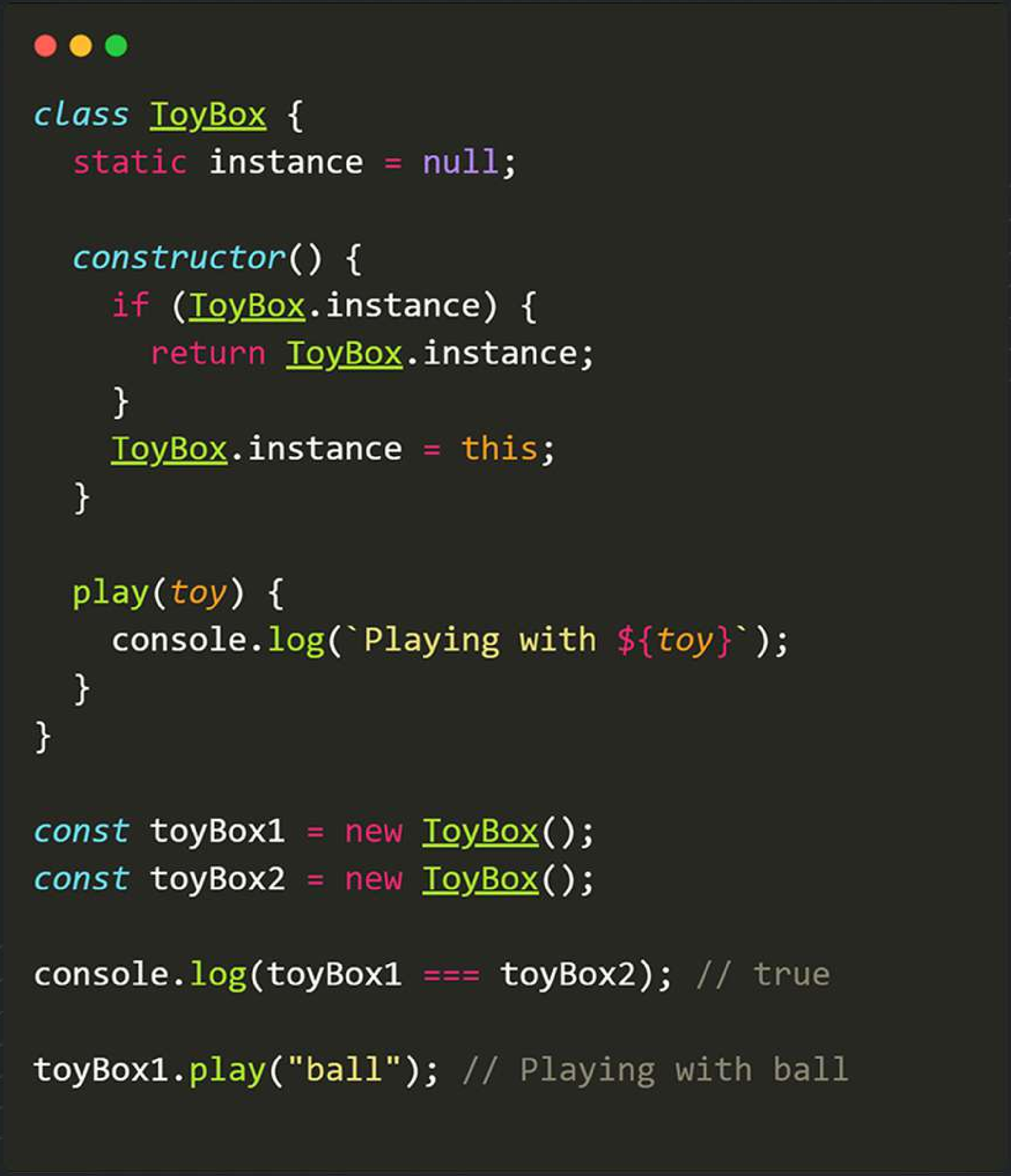
rofael-alfons

+Follow

Let's say you and your friends are playing a game and you all want to share the same toy. To make sure that only one of you can play with the toy at a time, you decide to take turns with it. This is similar to the singleton design pattern in computer programming.

rofael-alfons

+Follow



```javascript
class ToyBox {
  static instance = null;

  constructor() {
    if (ToyBox.instance) {
      return ToyBox.instance;
    }
    ToyBox.instance = this;
  }

  play(toy) {
    console.log(`Playing with ${toy}`);
  }
}

const toyBox1 = new ToyBox();
const toyBox2 = new ToyBox();

console.log(toyBox1 === toyBox2); // true

toyBox1.play("ball"); // Playing with ball
```

rofael-alfons

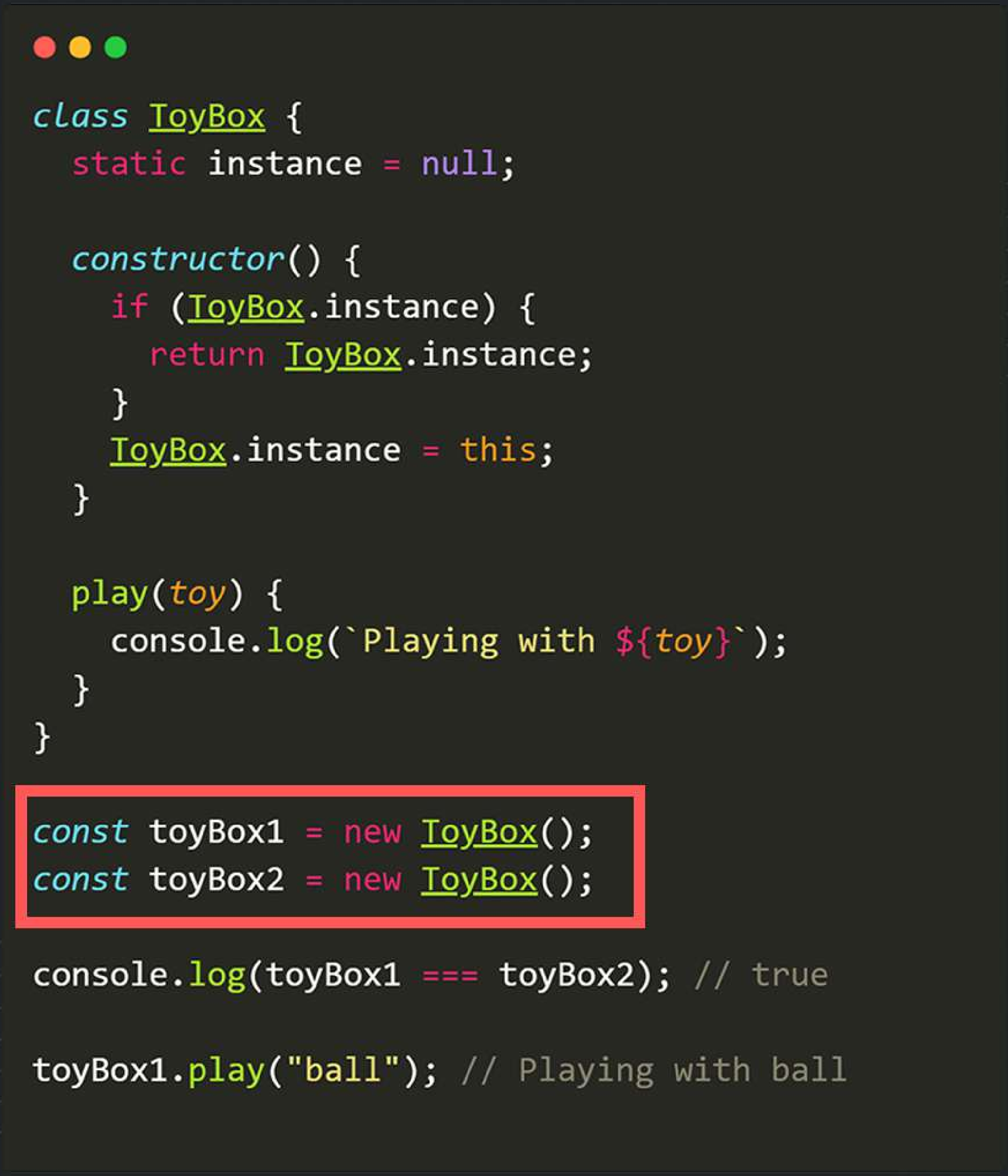
+Follow

In programming, the singleton pattern is used to make sure that a class has only one instance. This is useful when we want to share data or resources between different parts of our code.

In the previous example, the ToyBox class is a singleton. We use the instance property to keep track of the single instance of the class. When we create a new ToyBox object, the constructor checks if an instance already exists. If it does, it returns the existing instance. If not, it creates a new instance and sets it as the instance property.

rofael-alfons

+Follow



```
class ToyBox {
  static instance = null;

  constructor() {
    if (ToyBox.instance) {
      return ToyBox.instance;
    }
    ToyBox.instance = this;
  }

  play(toy) {
    console.log(`Playing with ${toy}`);
  }
}

const toyBox1 = new ToyBox();
const toyBox2 = new ToyBox();

console.log(toyBox1 === toyBox2); // true

toyBox1.play("ball"); // Playing with ball
```

rofael-alfons

+Follow

Now, every time we create a new ToyBox object, we'll always get the same instance. This means that we can use it to share data or resources between different parts of our code.

Just like in the game, we can only play with the toy one at a time. This is similar to how we can only have one instance of the ToyBox class in our code.

rofael-alfons

+Follow

Thanks for reading

Did you find it useful ?

rofael-alfons

+Follow