JS JavaScript TUTORIAL

JavaScript Static Properties

Summary: in this tutorial, you'll learn about the JavaScript static properties of a class and how to access the static properties in a static method, class constructor, and other instance methods.

Introduction to the JavaScript static properties

Like a static method, a static property is shared by all instances of a class. To define static property, you use the `static` keyword followed by the property name like this:

```
class Item {
  static count = 0;
}
```

To access a static property, you use the class name followed by the `.` operator and the static property name. For example:

```
console.log(Item.count); // 0
```

To access the static property in a static method, you use the class name followed by the `.` operator and the static property name. For example:

```
class Item {
  static count = 0;
  static getCount() {
    return Item.count;
  }
}

console.log(Item.getCount()); // 0
```

To access a static property in a class constructor or instance method, you use the following syntax:

```
className.staticPropertyName;
```

Or

```
this.constructor.staticPropertyName;
```

The following example increases the `count` static property in the class constructor:

```
class Item {
  constructor(name, quantity) {
    this.name = name;
    this.quantity = quantity;
    this.constructor.count++;
  }
  static count = 0;
  static getCount() {
    return Item.count;
  }
}
```

When you create a new instance of the `Item` class, the following statement increases the `count` static property by one:

```
this.constructor.count++;
```

For example:

```
// Item class ...

let pen = new Item("Pen", 5);
let notebook = new Item("notebook", 10);

console.log(Item.getCount()); // 2
```

This example creates two instances of the `Item` class, which calls the class constructor. Since the class constructor increases the `count` property by one each time it's called, the value of the `count` is two.

Put it all together.

```
class Item {
  constructor(name, quantity) {
    this.name = name;
    this.quantity = quantity;
    this.constructor.count++;
  }
  static count = 0;
  static getCount() {
    return Item.count;
  }
}

let pen = new Item('Pen', 5);
let notebook = new Item('notebook', 10);

console.log(Item.getCount()); // 2
```

Summary

- A static property of a class is shared by all instances of that class.
- Use the `static` keyword to define a static property.
- Use the `className.staticPropertyName` to access the static property in a static method.
- Use the `this.constructor.staticPropertyName` or `className.staticPropertyName` to access the static property in a constructor.