TypeScript Static Methods and Properties

Summary: in this tutorial, you will learn about the TypeScript static properties and methods.

Static properties

Unlike an instance property, a static property is shared among all instances of a class.

To declare a static property, you use the `static` keyword. To access a static property, you use the `className.propertyName` syntax. For example:

```
class Employee {
    static headcount: number = 0;

    constructor(
        private firstName: string,
        private lastName: string,
        private jobTitle: string) {

        Employee.headcount++;
    }
}
```

In this example, the `headcount` is a static property that is initialized to zero. Its value is increased by 1 whenever a new object is created.

The following creates two `Employee` objects and shows the value of the `headcount` property. It returns two as expected.

```
let john = new Employee('John', 'Doe', 'Front-end Developer');
let jane = new Employee('Jane', 'Doe', 'Back-end Developer');

console.log(Employee.headcount); // 2
```

Static methods

Similar to the static property, a static method is also shared across instances of the class. To declare a static method, you use the `static` keyword before the method name. For example:

```
class Employee {
    private static headcount: number = 0;

    constructor(
        private firstName: string,
        private lastName: string,
        private jobTitle: string) {

        Employee.headcount++;
    }

    public static getHeadcount() {
        return Employee.headcount;
    }
}
```

In this example:

- First, change the access modifier of the `headcount` static property from `public` to `private` so that its value cannot be changed outside of the class without creating a new `Employee` object.
- Second, add the `getHeadcount()` static method that returns the value of the `headcount` static property.

To call a static method, you use the `className.staticMethod()` syntax. For example:

```
let john = new Employee('John', 'Doe', 'Front-end Developer');
let jane = new Employee('Jane', 'Doe', 'Back-end Developer');
```

```
console.log(Employee.getHeadcount); // 2
```

In practice, you will find a library that contains many static properties and methods like the `Math` object. It has `PI`, `E`, ... static properties and `abs()`, `round()`, etc., static methods.

Summary

- Static properties and methods are shared by all instances of a class.
- Use the `static` keyword before a property or a method to make it static.