Previous Next

TypeScript - Abstract Class

TypeScript allows us to define an abstract class using keyword `abstract`. Abstract classes are mainly for inheritance where other classes may derive from them. We cannot create an instance of an abstract class.

An abstract class typically includes one or more abstract methods or property declarations. The class which extends the abstract class must define all the abstract methods.

The following abstract class declares one abstract method `find` and also includes a normal method `display`.

Example: Abstract Class

```
abstract class Person {
    name: string;

    constructor(name: string) {
        this.name = name;
    }

    display(): void{
        console.log(this.name);
    }

    abstract find(string): Person;
}

class Employee extends Person {
```

```
    empCode: number;

    constructor(name: string, code: number) {
        super(name); // must call super()
        this.empCode = code;
    }

    find(name:string): Person {
        // execute AJAX request to find an employee from a db
        return new Employee(name, 1);
    }
}

let emp: Person = new Employee("James", 100);
emp.display(); //James

let emp2: Person = emp.find('Steve');
```

In the above example, `Person` is an abstract class which includes one property and two methods, one of which is declared as abstract. The `find()` method is an abstract method and so must be defined in the derived class. The `Employee` class derives from the `Person` class and so it must define the `find()` method as abstract. The `Employee` class must implement all the abstract methods of the `Person` class, otherwise the compiler will show an error.

> **Note:**
>
> The class which implements an abstract class must call `super()` in the constructor.

The abstract class can also include an abstract property, as shown below.

Example: Abstract Class with Abstract Property

```
abstract class Person {
    abstract name: string;

    display(): void{
        console.log(this.name);
    }
}

class Employee extends Person {
    name: string;
    empCode: number;

    constructor(name: string, code: number) {
        super(); // must call super()

        this.empCode = code;
        this.name = name;
    }
}

let emp: Person = new Employee("James", 100);
emp.display(); //James
```

Previous Next