第十三课 class 类

学习目录

- **类基本介绍与用法**
- **类继承**
- **类成员访问修饰符**

一．类基本介绍与用法

以前我们学习过 es5 中构造函数来构建基于面向对象的组件，主要是因为在 js 中没有类的概念，需要使用构造函数来实现，不过随着 js 语法的发展，现在的 es6 已经可以使用 class 关键字来定义类了。当然在 ts 中我们也可以使用 class 关键字来定义类。

```
class Work {
    title: string;
    constructor(job: string) {
        this.title = job;
    }
    getTitleFn() {
        return "工作是：" + this.title;
    }
}
let w1 = new Work("前端工程师");
console.log(w1.getTitleFn());
```

二．类继承

在面向对象语言中，我们都可以使用继承来扩展现有的类，继承（Inheritance）也是面向对象几大特点之一，子类继承父类，子类可以除了拥有父类的所有属性和方法等特性之外，还可以扩展自己的属性和方法等更多特性。

```
class Book{
    saleFn(price: number = 0) {
        console.log(`本书价格为${price}元.`);
    }
}
class Fiction extends Book{
    getStarFn(num:number = 0) {
        console.log('小说评分为${num}分');
    }
}
let f1 = new Fiction();
f1.saleFn(188);
f1.getStarFn(100);
```

除了普通继承之外，class 类中还有构造函数 constructor 的概念，构造函数的继承需要子类调用 super()方法才行。

```
class Animal {
    name: string;
```

```
        constructor(theName: string) { this.name = theName; }
        move(distanceInMeters: number = 0) {
            console.log(`${this.name} moved ${distanceInMeters}m.`);
        }
    }
    class Cat extends Animal {
        constructor(name: string) { super(name); }
        move(distanceInMeters = 5) {
            console.log("running...");
            super.move(distanceInMeters);
        }
    }
    let c1: Animal = new Cat("goodCat");
    c1.move();
```

备注一下：在构造函数里访问 this 属性之前，我们必须定要调用 super()方法，这个和我们之前写 es5 构造函数的方法原理差不多，如果没有 super()方法调用，子类不可能继承得到父类构造函数的实例属性。

三．类成员访问修饰符

ts 中其实类的访问修饰符就是基于三个关键词 private、protected 和 public 的访问权

限问题，默认访问修饰符为 public。

public 访问修饰符

顾名思义就是任何地方都可以访问类的实例成员，因此 public 可以不用定义也可以显式定义。

```
class Work{
    public title: string;
    public constructor(job: string) {
    this.title = job;
}
    public getTitleFn() {
        return "工作是：" + this.title;
    }
}
```

private 访问修饰符

当类成员被标记为 private 时，这个成员就不能在声明它的类的外部访问了。

```
class Work{
    private title: string;
    constructor(job: string) { this.title = job; }
}
new Work("前端工程师").title;//错误
```

备注一下：假如有两个类的内部都定义了一个同名 private 属性成员，并不是表示这两个类可以相互兼容相等。

protected 访问修饰符

protected 修饰符的成员在派生子类中可以访问。

```
class Book{
    protected title:string;
    public getTitleFn(){
        console.log(this.title)
    }
}
class Fiction extends Book{
    public getBookTitle(){
        console.log(this.title);
    }
}
```

备注一下：constructor 构造函数也可以使用 protected 访问修饰符来定义，这样的话这个构造函数就不能在包含它的类外实例化了。

谢谢观看！如果觉得课程还不错的话，记得给个好评！

我是星星课堂老师：周小周