

23. 类 class 实现

学习要点：

1. 类 class
2. getter 和 setter

本节课我们来开始学习 ES6 新增的类 class 的实现方法。

一. 类 class

1. 在 ES6 之前，JavaScript 不能像其它语言(PHP, Java)等有完整的类支持；
2. 我们采用了原型链实现了面向对象的功能，但从 ES6 开始，提供了真正的类语法；
3. 当然，虽然说是真正的类语法，而本质上内部实现和原型链还是一样的；
4. 对于面向对象和类的基础概念，这里不再赘述，学到这里不可能没这个基础的；
5. 首先，我们创建一个基本的类，并创建构造函数(构造方法)，具体如下：

```
//创建一个类
class Person {

    //构造函数(构造方法)
    constructor(name) {
        //this.name 是类的属性
        //name 是构造参数赋值给属性
        this.name = name;
    }

    //普通方法
    run() {
        console.log('类的方法输出！' + this.name);
    }
}

//实例化一个 Person 对象
let p = new Person('Mr. Lee');
//执行 run() 方法
p.run();
//输出对象的属性
console.log(p.name);
//判断 p 是否是 Person 对象
console.log(p instanceof Person);
//判断类的类型：function
console.log(typeof Person);
```

6. 除了上面的 `class Person` 这种常规类的写法外，ES6 还支持表达式写法；

```
//No.1
let Per = class Person {};
//此时 new Person 会报错
let p = new Per('Mr.Lee');

//No.2
let Person = class {};

//No.3
let p = new class {}('Mr.Lee');
p.run();
```

二. getter 和 setter

1. 根据面向对象的三大定律中成员属性，我们需要对它进行封装，变成私有属性；
2. 而目前的 `this.name`，基本是对外公开的，可以在类外取值和赋值；
3. 当我们假设类的属性是私有的，那么需要通过 `get()` 和 `set()` 方法实现；

```
//创建一个类
class Person {
    #name; //提案，浏览器暂时不支持

    //构造函数(构造方法)
    constructor(name) {
        this.#name = name; //私有属性，类外无法访问
    }

    get name() {
        return this.#name;
    }

    set name(value) {
        this.#name = value;
    }
}

let p = new Person('Mr.Lee');
p.name = 'Mr.Wang';
console.log(p.name);
```