



ES6中的类(class)和继承



jCodeLife

0.366

2020.09.03 15:35:07 字数 576 阅读 88

编辑文章

提起面向对象编程，任何语言都必然会提及类和继承，js也是一样，但在悠悠长长的发展过程中，js一直是使用函数来模拟实现类的，跟其他语言对比，可读性就会差很多。比如java，java里面有专门的关键字class来声明一个类，js没有；java有extends继承一个类，js没有；java有静态关键字static声明自己私有的东西子类不能用的等等，js也没有；一直到ES6出现，才开启了class及相关关键字的使用。让js的类使用起来更像其他语言的类，可读性更高。

```
1 class Person {
2   constructor(name) {
3     this.name = name;
4   }
5   show() {
6     console.log(this.name);
7   }
8   static fn() {
9     console.log("私有方法fn")
10  }
11 }
12 let p = new Person('alice');
13 typeof(Person)//function
14 Person.fn();//私有方法fn
15 console.log(p.name);//alice
16 p.show();//alice
17 p.fn();//Uncaught TypeError: p.fn is not a function
```

由于就是一些语法规则，比较简单，这里小记几点即可：

1. ES6 的类，完全可以看作构造函数的另一种写法，就是一个语法糖；本质还是一个function
2. 构造函数的prototype属性，在 ES6 的“类”上面继续存在。即：ES6类上的所有方法（除了用static定义的）都定义在类的prototype属性上面
3. 使用static定义的属于属性或者方法叫静态属性和静态方法，只能通过类名调用，不会被子类继承
4. 由于实例方法都是定义在构造函数prototype中，所以使用Object.assign()可以方便的向一个类中添加多个方法

```
1 class A{
2   constructor(){ }
3 }
4 Object.assign(A.prototype, {
5   xx(){},
6   xxx(){ }
7 });
```

5. 通过new生成对象时，自动调用constructor方法，如果没有显式定义该方法，系统会隐式的添加一个空的constructor方法；子类构造函数必须加上super()执行父类的构造constructor函数
6. 在“类”的内部可以使用get和set关键字，在读取或者设置该属性时可以拦截到，其实这点es5也是一样
7. 除了class 类名{}的声明方式，还可以使用表达式的形式：const A= class{...}，但比较少用
8. class内部默认是严格模式，且class声明的变量不会有预编译变量提升的东西
9. 使用extends实现继承，格式：

```
1 class A {}
2
3 class B extends A {
4   constructor() {
5     super();
6   }
7 }
```



jCodeLife

总资产30 (约2.65元)

一篇搞清AJAX及实现手动封装
阅读 70

webpack有什么作用？
阅读 2

Vue3为什么引入composition API
阅读 30

推荐阅读

如何理解JavaScript中的对象？
阅读 183

从事7年前端开发，有些经验想对转行学习前端的伙伴说说！
阅读 136

12个前端开发必备开发的工具
阅读 486

vue 书本翻页的效果
阅读 290

javascript 近乎神话般的概念：闭包
阅读 695

写下你的评论...

评论 0

赞 3

...

"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



jCodeLife  书山有路勤为径，学海无涯苦作舟
总资产30 (约2.65元) 共写了13.6W字 获得329个赞 共25个粉丝



写下你的评论...


全部评论 0 只看作者 关闭评论

按时间倒序 按时间正序

被以下专题收入，发现更多相似内容

 投稿管理

+ 收入我的专题

 前段开发那些事儿