#### Angular读书笔记

##### 之JavaScript中的浅拷贝

摘要

JavaScript中的浅拷贝

正文

在典型的面向对象的语言中，如java，都存在类（class）的概念，类就是对象的模板，对象就是类的实例。但是在Javascript语言体系中，是不存在类（Class）的概念的，javascript中不是基于‘类的'，而是通过构造函数（constructor）和原型链（prototype chains）实现的。

原型链（prototype chains）

对象的属性和方法，有可能是定义在自身，也有可能是定义在它的原型对象。由于原型对象本身对于对象实例来说也是对象，它也有自己的原型，所以形成了一条原型链（prototype chain）。比如，a对象是b对象的原型，b对象是c对象的原型，以此类推。所有一切的对象的原型顶端，都是Object.prototype，即Object构造函数的prototype属性指向的那个对象。

JavaScript中的浅拷贝

1.带原型的浅拷贝

let mBasicInfo = Object.create(Object.getPrototypeOf(this.basicInfo), Object.getOwnPropertyDescriptors(this.basicInfo));

2.无原型的浅拷贝

let mBasicInfo: BasicInfoFull = <BasicInfoFull>JSON.parse(JSON.stringify(this.basicInfo));

□

p.s. JavaScript中完全意义上的深拷贝需要打开循环引用，算法复杂度高，故不常用.