

Javascript面向对象

● 工厂方式

- 面向对象中的封装函数
- 改成与系统对象类似写法
 - 首字母大写
 - New 关键字提取
 - This指向为新创建的对象
- 构造函数
 - 用来创建对象的函数，叫做构造函数
- 存在的问题
 - 对象的引用
 - 浪费内存

其实工厂方式就是封装函数

```
<script>
    function createPerson(name){
//      1. 原料
        var obj = new Object();
//      2. 加工
        obj.name = name;
        obj.showName = function(){
            console.log(this.name)
        }
//      3. 出场
        return obj;
    }
    var p1 = createPerson('小明');
    var p2 = createPerson('小强');
    p1.showName();
    p2.showName();
</script>
```

当new去调用一个函数，这个时候this就是创建出来的对象，而且函数返回值就是This（隐式返回）

new后面调用的函数叫构造函数

```
</script>  
// 当new去调用一个函数，这个时候this就是创建出来的对象，而且函数返回值就是This（隐式返回）  
// new后面调用的函数叫构造函数  
function CreatePerson(name){  
    this.name = name;  
    this.showName = function(){  
        console.log(this.name)  
    }  
}  
var p1 = new CreatePerson('小明');  
var p2 = new CreatePerson('小强');  
p1.showName();  
p2.showName();  
var arr = new Array();  
var date = new Date();  
console.log(arr)  
</script>
```