面向对象的总结目录

创建对象的3种方法，优缺点

{}字面量

优点：方便

缺点：创建相同 属性和方法时，会出现大量重复代码

工厂模式

优点：解决了创造相似的东西

缺点：共同的方法还是不重复

构造函数模式——对new的理解

优点：方法可以继承，节省内存

缺点：

对象的静态和动态的属性和方法

Object.hasOwnProperty(proto)

Object.isPrototypeOf(obj, prototype)

Object.setPrototypeOf()

Object.assign()

Object.is()

Object.create()

Object.keys()

Object.values()

Object.entries()

Object.definePropoty(obj,name,{

value

writable

configure

enumerable

})

instanceof运算符

in运算符

面向对象的特性的理解

抽象

封装

继承

多态

继承的实现，优缺点

子.prototype=父.prototype

浅拷贝

子.prototype=Object.assign(父.prototype)

子.prototype=Object.assign(子.prototype，父.prototype)

注意：Draglimite.prototype.constructor=Draglimite;

深拷贝（for in）

原型链——对原型和原型链的理解

子.prototype=new 父

子.prototype.\_\_proto=父.prototype

子.prototype=Object.create(父.prototype)

es6：class 子 extent 父

原型和原型链的理解

为什么存在原型：

特征：

构造函数

原型：构造函数.property。。。constructor

实例：new 构造函数。。。\_\_propo\_\_

实例.\_\_propo\_\_=构造函数.property

构造函数.property.constructor=构造函数

对new的理解

new fn或new fn()或new fn(参)

arguments

var Object={}，this=Object

执行代码

return this

自定义事件的封装

阻止默认事件、阻止冒泡

事件对象

触发

绑定

取消绑定

