正课:

1. \*\*\*面向对象:

继承, 多态

2. \*\*\*\*ES5

1. \*\*\*面向对象:

问题: 放在构造函数中的方法定义，会被反复创建副本，浪费内存！

解决: 继承

继承: 父对象的成员，子对象无需重复创建，就可直接使用

为什么: 代码重用, 节约内存！

何时: 只要多个子对象，需要相同的属性值或功能时

如何: js中的继承都是自动继承原型对象

原型对象:

什么是: 存储同一类型的所有子对象共有成员的父对象

何时: 只要多个子对象需要相同的属性值或功能时，都应该集中定义在原型对象中

如何使用: 2步:

1. 创建: 自动创建！买一赠一！

只要创建一个构造函数，都附赠一个空的原型对象

2. 继承: 自动继承！

只要使用构造函数创建一个子对象时，都会让子对象自动继承构造函数的原型对象——new的第2 步

如何向原型对象中添加共有成员:

类型名.prototype.新成员=值/function(){...}

成员的访问顺序:

优先使用自有成员

自己没有，才去父级原型对象中查找

自有属性和共有属性:

自有属性: 保存在对象本地，归当前对象独有的属性

共有属性: 保存在公共的父级原型对象中，所有子对象共享的属性

获取时: 没有差别: 子对象.属性名

修改时: 自有属性: 子对象.属性名=值

共有属性必须通过原型对象修改:

类型名.prototype.共有属性名=值

判断是否自有属性:

var bool=obj.hasOwnProperty("属性名")

内置类型的原型对象:

其实每种类型，都有两部分组成:

构造函数: 专门负责创建子对象

构造函数的原型对象: 专门负责保存所有子对象共有的属性和API

问题: 旧浏览器不支持新的API或浏览器中的API不够用！

解决: 为浏览器添加想要的API

如何: 2步:

1. 先判断当前浏览器是否支持该API:

其实就是判断当前浏览器的指定类型的原型对象中是否包含想用的API

如何: if(typeof 类型名.prototype.API !="function")

2. 向指定类型的原型对象中添加API

类型名.prototype.API=function(参数列表){

... ...

}