构造函数

多种构造对象的方法

1. 工厂方法
2. 直接构造
3. 构造函数

构造函数的工作过程

1. 隐式的创建
2. 把this指针初始化创建的对象
3. 构造函数本身的逻辑
4. 隐式的return

正常情况下，自定义的构造函数不应返回任何对象

如果构造函数返回原始变量类型，返回无效，函数会把this隐式返回；如果返回对象类型，隐式返回失效，被返回对象类型代替。

工厂方法VS构造函数

1.函数名:工厂方法首字母小写，构造函数首字母大写

2.调用时是否使用new关键字:工厂方法不使用，构造函数使用

3.返回值:工厂方法必须返回对象，构造函数建议不返回

4.对对象的引用:工厂方法，创造的谁用谁，构造函数，用this

JavaScript原型

* + 每个函数，都有一个内置属性,叫做prototype (原型)
  + prototype是对象
  + 如果一个对象是通过new产生的，那么这个对象将有缺省的属性，\_\_proto\_\_,这个属性指向函数的原型。

如果你用一个构造函数创建了一个对象，那么:

对象.\_\_ proto\_\_ .constructor就是构造函数的函数对象

对象instanceof 构造函数对象是true

原型:用来让所有构造函数生成的对象可以共享属性和方法

[javascript关于原型对象重写后的问题](https://blog.csdn.net/weixin_44539199/article/details/98192775?utm_medium=distribute.pc_relevant.none-task-blog-title-3&spm=1001.2101.3001.4242)

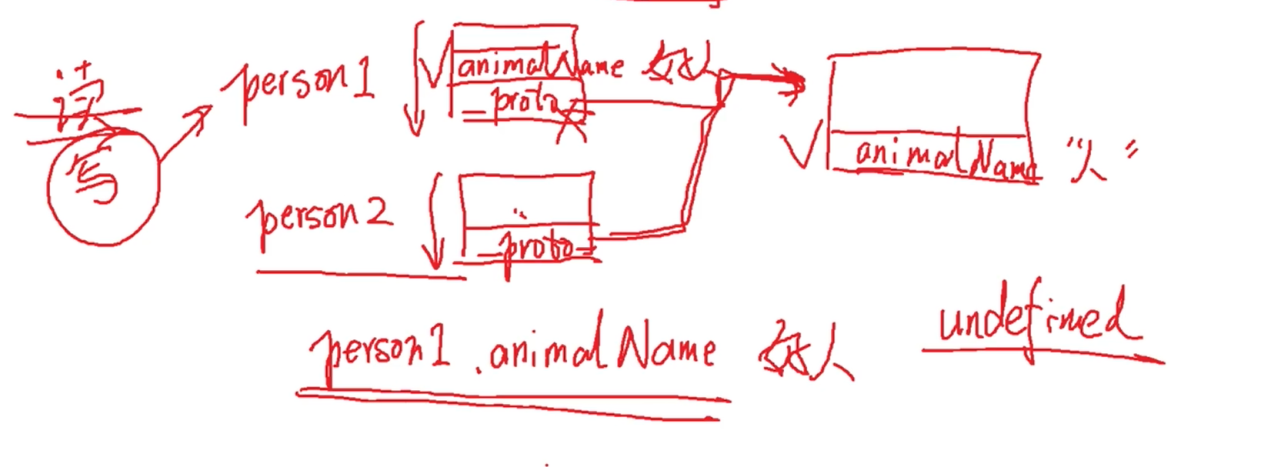
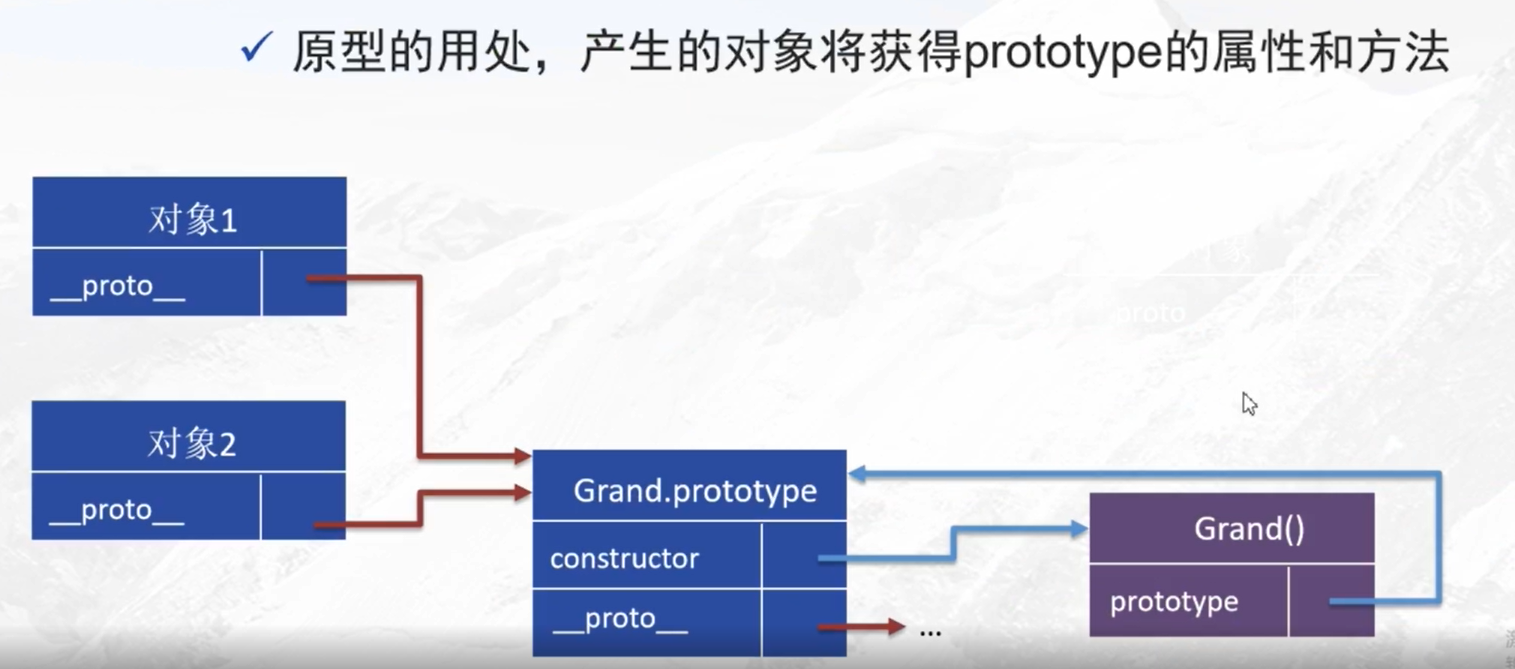
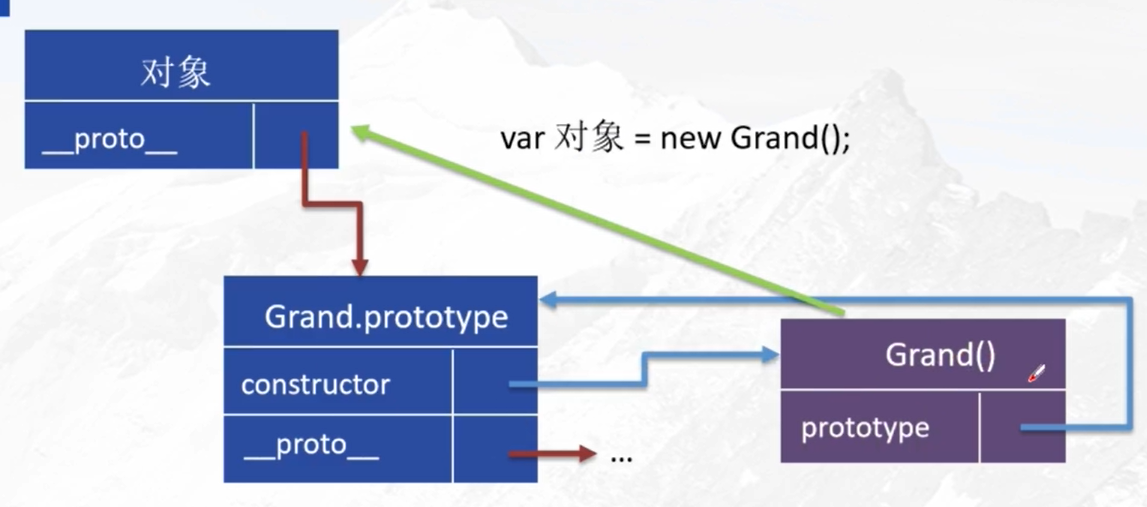
实例与构造函数

1.在构造函数原型上的属性和方法，都是共享的

2.写实例:直接在实例上增加的方法或者属性，是属于实例本身的，不会影响构造函数的原型

3.读实例:首先在实例本身上寻找改方法或者属性，如果找不到去原型上找，如果还找不到，返回undefined

Delete 删除一个不存在的东西不会报错



原型链

构造原型链的步骤

1.从Object开始，每一层底层的构造函数构造出实例，作为上一层构造函数的原型对象

2.构造好后，通过对象的\_proto\_，可以一直追溯到object\null，这个链表称为原型链

原型链的性质

1. 挂在原型链对象上的属性和方法,能够被所有的派生实例共享
2. 如果访问一个示例的属性或者方法,先从对象本身找起,如果发现没有,会沿着原型链由近至远寻找，如果找不到返回undefined
3. Document是一个构造函数,原型链:Document ->Node -> EventTarget – >Object
4. HTMLDivElement -> HTML Element -> Element -> Node -> EventTarget -> Object

面向对象标准

封装

1.的数据或者方法赋予标准的访问权限，没有权限的时候不可以访 问

2.支持:对象所有的属性和方法都是公开的，可以使用立即执行函数和闭包把一部分隐藏起来

继承

一种实体之间的Is-A的关系:

在任何B能够被正常调用的地方，A都可以，我们说A is-a B

A继承自B

所有B能够干的事情，A都可以，B给A赋能

JS支持的:以原型链方式实现

多态

A继承自B，如果A修改了一个B已经存在的属性或者方法，A的实例在任何地方按照名字调用修改后的属性和方法，应该能够成功

A对B进行扩展，赋能

覆盖：JS支持

重载：JS不支持

借用构造函数：base.call(this,…)

组合继承：借用构造函数+原型链继承

拷贝

浅拷贝是只拷贝值，如果拷贝对象是数组或者其他对象，那么只拷贝引用

深拷贝，如果拷贝对象是数据或者其他对象，完整拷贝这些数组和对象

a.hasOwnProperty(i):

hasOwnProperty表示是否有自己的属性。这个方法会查找一个对象是否有某个属性，但是不会去查找它的原型链。

[Object.defineProperty](https://blog.csdn.net/qq_39148344/article/details/89082741)

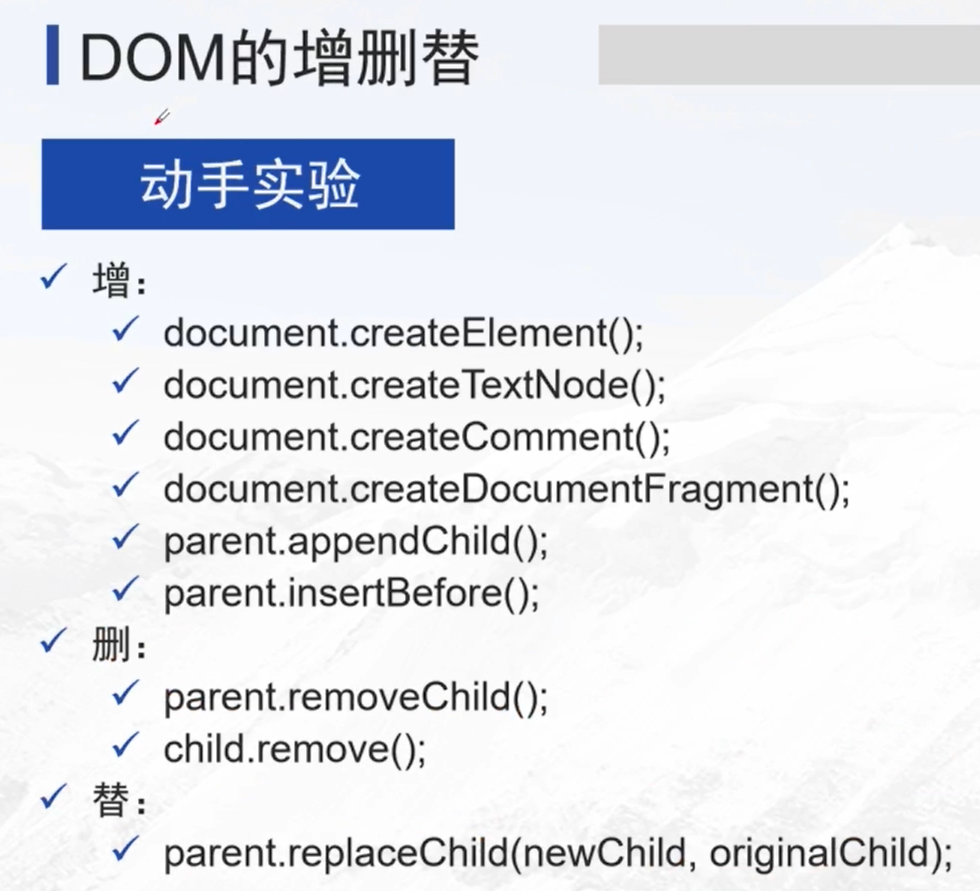
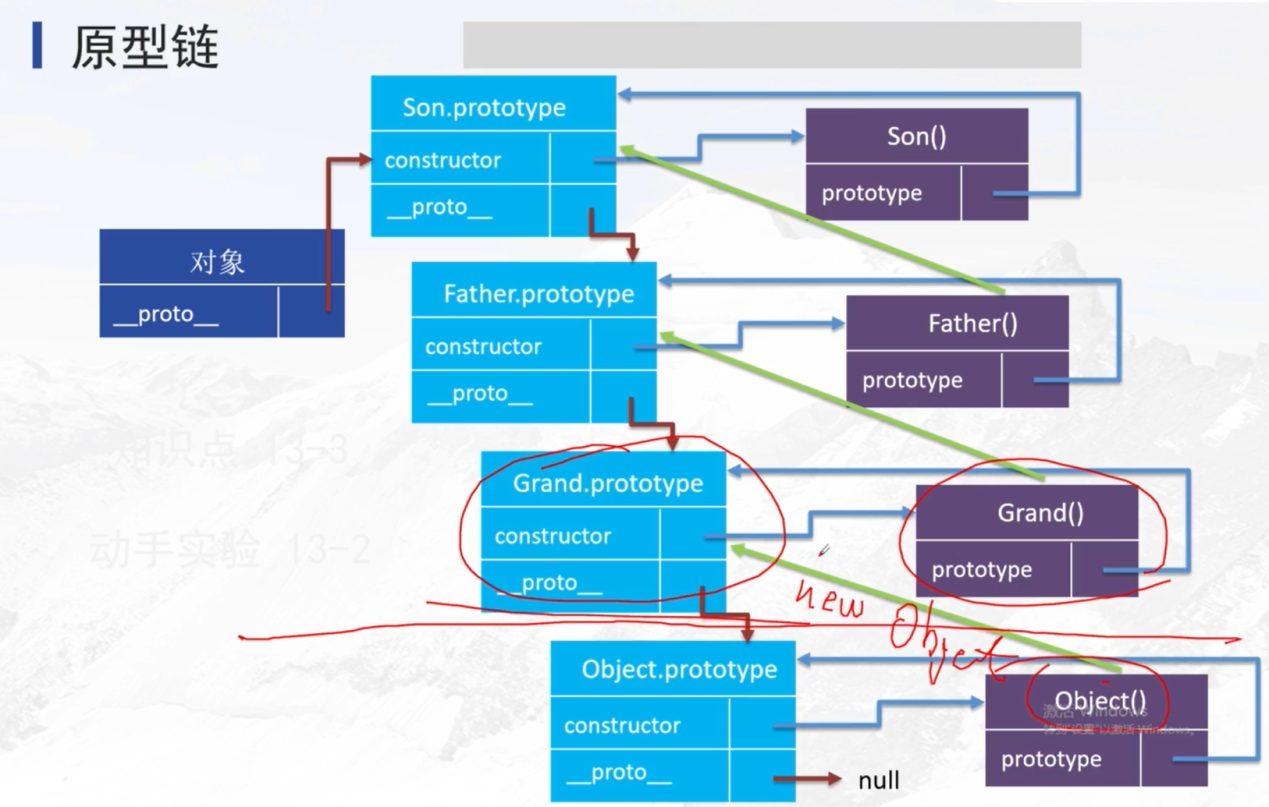
Object.defineProperty() 方法会直接在一个对象上定义一个新属性，或者修改一个对象的现有属性， 并返回这个对象。

原型链继承会用到

Object.defineProperty(Cat.prototype, 'constructor', {

enumerable: false

});



正则表达式

正则表达式允许程序员定义一系列的规则，这种规则称为模式(pattern），并且可以检测输入的字符串是否和这种模式相匹配

有限自动状态机

[输入框事件监听(一)：keydown、keyup、input](https://blog.csdn.net/yiifaa/article/details/52372022)