原型链只是考察点：

1.创建对象有几种方法

2.原型、构造函数、实例、原型链

3.instanceof的原理

4.new运算符

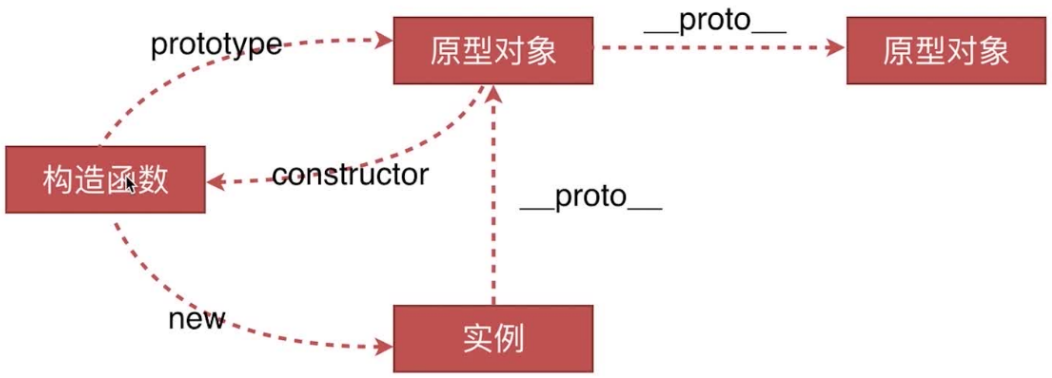
1.字面量方法

2.显示的构造函数创建

3.object.create（）

只有函数才有prototype，实例对象只有_proto_

原型链类 原型、构造函数、实例、原型链



原型链类 instanceof

instanceof的原理：
实例对象._proto_ === 构造函数.prototype
也就是他们是否在一条原型链上，是否找到同一个原型对象

判断函数FuncA是否是实例objA的构造函数的方法：
1.instanceof：
objA instanceof FuncA === true
原理：obj._proto_ [_proto_加n个]=== FuncA.prototype
缺陷：这个方法不够准确，因为只要在同一条原型链上都会返回true
eg: obj._proto_._proto_ ===FuncA.prototype._proto_===Object.prototype
2.constructor
obj._proto_.constructor === FuncA一定准确

new构造函数的原理：
1.生成一个空对象

2.将空对象的_proto_指向构造函数的prototype
3.执行构造函数，this上下文指向空对象
4.构造函数如果return了对象，放弃掉空对象；反之，返回前面的那个空对象