Javascript，万事万物皆对象；

但是呢，对象也是有区别滴，它分为普通对象和函数对象；

也就是object和function；

什么是原型对象？

在javascript中，每当定义一个对象的时候，对象都会包含一些自带的属性，其中，每一个函数对象都有一个属性：prototype属性，这个属性，就是指向函数的原型对象。

在JS里面，每一个对象，都有\_proto\_属性，但是只有函数对象，才有prototype属性；

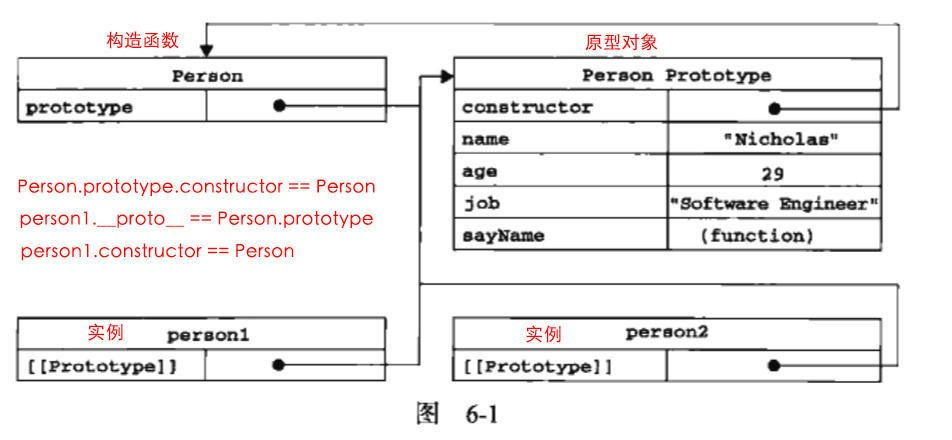
原型对象和普通对象，没啥区别，只要记住原型对象就是：

构造函数.prototype={}

也就是说，原型对象（person.prototype）就是构造函数（person）的一个实例

## \_proto\_

JS在创建对象的时候，都有一个叫做\_proto\_的内置属性，用于指向创建他的构造函数的原型对象；



通过这个连接图，我们可以看到

Person.prototype.constructor = Person；//构造函数的指针指向他自己

person1.\_proto\_ == Person.prototype；//实例对象的\_proto\_就是构造函数的原型对象

person1.constructor == Person;//实例对象的指针，是指向构造函数；

那么，这个连接链，它的重点是：他是存在于实例和构造函数的原型之间，而不是存在于实例和构造函数之间；

在javascript里面，创建对象有两种方法：

Var obj = {};字面量

Var obj = new Object();和上面完全相同

那么obj就是构造函数的一个实例，所以

obj.constructor ===Object;

obj.\_proto\_ ===Object.prototype;

也就是说，我们实例化的obj变量（实例化对象），他的指针，指向object这个构造函数

他的我\_proto\_指向Object的原型对象；

我们用来创建对象的构造函数又称之为构造器；

原型链：