对象调用链

我们看这个robot对象有名字，有身高，还会跑，有点像小明，干脆就根据它来“创建”小明得了！

于是我们把它改名为Student，然后创建出xiaoming：

**var** Student = {

name: 'Robot',

height: 1.2,

run: **function** () {

console.log(**this**.name + ' is running...');

}

};

**var** xiaoming = {

name: '小明'

};

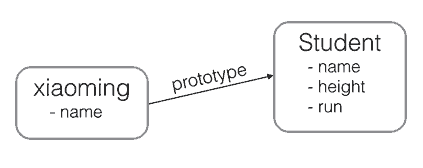
xiaoming.\_\_proto\_\_ = Student;

注意最后一行代码把xiaoming的原型指向了对象Student，看上去xiaoming仿佛是从Student继承下来的：

xiaoming.name; // '小明'

xiaoming.run(); // 小明 is running...

xiaoming有自己的name属性，但并没有定义run()方法。不过，由于小明是从Student继承而来，只要Student有run()方法，xiaoming也可以调用：



JavaScript的原型链和Java的Class区别就在，它没有“Class”的概念，所有对象都是实例，所谓继承关系不过是把一个对象的原型指向另一个对象而已。