# 面向对象编

JavaScript不区分类和实例的概念，而是通过原型来实现面向对象编程。

现有的对象。

var Student = {

name:’Robot’,

height:’1.6’,

run:function(){

console.log(this.name+’is running...’);

}

};

然后

var xiaoming={

name:’小明’

};

xiaoming.\_proto\_=Student;

xiaoming有自己的name属性，但没有定义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。

## 区别

JavaScript的原型链和java的class区别就在，他没有class的概念，所有的对象都是实例，所谓继承关系不过是把一个对象的原型指向另一个对象。

在JavaScript代码运行时期，你可以把xiaoming从student变成任何对象。

## 注意

不要直接使用obj.\_proto\_去改变一个对象的原型。Object.create()方法可以传入一个原型对象，并创建一基于该原型的新对象，但是新对象什么属性都没有，因此，我们可以编写一个函数来创建xiaoming。

