## Javascript继承的两种经典模式

第一种：原型继承

使用原型继承来创建一个Pet对象，和一个继承它的Cat对象。Javascript教程中常常能见到这种经典的继承模式：

var Pet = function(name, legs) {

this.name = name;

this.legs = legs;

};

Pet.prototype.getDetails = function() {

return this.name + ' has ' + this.legs + ' legs';

};

var Cat = function(name) {

Pet.call(this, name, 4);

};

Cat.prototype = new Pet();

Cat.prototype.action = function() {

return 'Catch a bird';

};

var petCat = new Cat('Felix');

var details = petCat.getDetails();

var action = petCat.action();

console.log(details);//Felix has 4 legs

console.log(action);// Catch a bird

petCat.name = 'Sylvester';

petCat.legs = 7;

details = petCat.getDetails();

console.log(details);//Sylvester has 7 legs

第二种：函数继承

另一种选项无需使用prototype或new，而是利用javascript的“函数继承”特性来吸收和增强对象实例；

var pet = function(name, legs) {

var that = {

name: name,

getDetails: function() {

return that.name + ' has ' + legs + ' legs';

}

};

return that;

}

var cat = function(name,num) {

var that = pet(name, num);

that.action = function() {

return 'Catch a bird';

}

return that;

}

var petCat2 = cat('renlei',4);

var details = petCat2.getDetails();

var action = petCat2.action();

console.log(details);//renlei has 4 legs

console.log(action);//Catch a bird

petCat2.name = 'Sylvester';

petCat2.legs = 7;

details = petCat2.getDetails();

console.log(details);//Sylvester has 4 legs