

递给构造器，而它将返回一个构造完全的对象。

原型

Prototypal

在一个纯粹的原型模式中，我们会摒弃类，转而专注于对象。**基于原型的继承相比基于类的继承在概念上更为简单**：一个新对象可以继承一个旧对象的属性。也许你对此感到陌生，但它真的很容易理解。你通过构造一个有用的对象开始，接着可以构造更多和那个对象类似的对象。这就可以完全避免把一个应用拆解成一系列嵌套抽象类的分类过程。

让我们先用对象字面量去构造一个有用的对象：

```
var myMammal = {
  name : 'Herb the Mammal',
  get_name : function () {
    return this.name;
  },
  says : function () {
    return this.saying || '';
  }
};
```

51 一旦有了一个想要的对象，我们就可以利用第 3 章中介绍过的 `Object.create` 方法构造出更多的实例来。接下来我们可以定制新的实例：

```
var myCat = Object.create(myMammal);
myCat.name = 'Henrietta';
myCat.saying = 'meow';
myCat.purr = function (n) {
  var i, s = '';
  for (i = 0; i < n; i += 1) {
    if (s) {
      s += '-';
    }
    s += 'r';
  }
  return s;
};
myCat.get_name = function () {
  return this.says + ' ' + this.name + ' ' + this.says;
};
```

这是一种“差异化继承 (*differential inheritance*)”^{译注 1}。通过定制一个新的对象，我们**指明它与所基于的基本对象的区别**。

译注 1：关于差异化继承的更多内容请参见 https://developer.mozilla.org/en/Differential_inheritance_in_JavaScript。