**No classes involved; Objects inherit from other objects.**

Use an empty temporary constructor function  F().  Set the prototype of  F() to be the parent object. Return a new instance of the temporary constructor.

[复制代码](javascript:void(0);)

function Object(o) {

function F() {}

F.prototype = o;

return new F();

}

// object to inherit from

var parent = {

name: "Papa"

};

// the new object

var child = Object(parent);

// testing

alert(child.name); // "Papa"

[复制代码](javascript:void(0);)

**Addition to ECMAScript 5**

In ECMAScript 5, the prototypal inheritance pattern becomes officially a part of the language. This pattern is implemented through the method Object.create().

var child = Object.create(parent);

Object.create()accepts an additional parameter, an object. The properties of the extra object will be added as own properties of the new child object being returned.

[复制代码](javascript:void(0);)

var child = Object.create(parent, {

age: { value: 2 } // ECMA5 descriptor

});

child.hasOwnProperty("age"); // true

[复制代码](javascript:void(0);)

**References*:***

*JavaScript Patterns -*by Stoyan Stefanov (O`Reilly)