**Public Static Members**

[复制代码](javascript:void(0);)

// constructor

var Gadget = function (price) {

this.price = price;

};

// a static method

Gadget.isShiny = function () {

// this always works

var msg = "you bet";

**// Checking if the static method is called by instance.**

**if (this instanceof Gadget) {**

// this only works if called non-statically

msg += ", it costs $" + this.price + '!';

**}**

return msg;

};

// a normal method added to the prototype

Gadget.prototype.setPrice = function (price) {

this.price = price;

};

// a normal method added to the prototype

Gadget.prototype.isShiny = function () {

return **Gadget.isShiny.call(this);**

};

// Attempting to call an instance method statically won’t work

typeof Gadget.setPrice; // "undefined"

[复制代码](javascript:void(0);)

Testing a static method call:

Gadget.isShiny(); // "you bet"

Testing an instance, nonstatic call:

var a = new Gadget('499.99');

a.isShiny(); // "you bet, it costs $499.99!"

Private Static Members

• Shared by all the objects created with the same constructor function

• Not accessible outside the constructor

[复制代码](javascript:void(0);)

// constructor

var Gadget = (function () {

// static variable/property

var counter = 0,

NewGadget;

// this will become the new constructor implementation

**NewGadget = function () {**

**counter += 1;**

**};**

// a privileged method

NewGadget.prototype.getLastId = function () {

return counter;

};

// overwrite the constructor

return **NewGadget**;

}()); // execute immediately

var iphone = new Gadget();

iphone.getLastId(); // 1

var ipod = new Gadget();

ipod.getLastId(); // 2

var ipad = new Gadget();

ipad.getLastId(); // 3

[复制代码](javascript:void(0);)

**References*:***

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