**Shallow copy pattern**

[复制代码](javascript:void(0);)

function extend(parent, child) {

var i;

child = child || {};

for (i in parent) {

if (parent.hasOwnProperty(i)) {

child[i] = parent[i];

}

}

return child;

}

[复制代码](javascript:void(0);)

**Deep copy pattern**

[复制代码](javascript:void(0);)

function extendDeep(parent, child) {

var i,

toStr = Object.prototype.toString,

astr = "[object Array]";

child = child || {};

for (i in parent) {

if (parent.hasOwnProperty(i)) {

if (typeof parent[i] === "object") {

**child[i] = (toStr.call(parent[i]) === astr) ? [] : {};**

**extendDeep(parent[i], child[i]);**

} else {

child[i] = parent[i];

}

}

}

return child;

}

var dad = {

counts: [1, 2, 3],

reads: {

paper: true

}

};

var kid = extendDeep(dad);

kid.counts.push(4);

kid.counts.toString(); // "1,2,3,4"

dad.counts.toString(); // "1,2,3"

(dad.reads === kid.reads).toString(); // false

kid.reads.paper = false;

kid.reads.web = true;

dad.reads.paper; // true

[复制代码](javascript:void(0);)

Firebug (Firefox extensions are written in JavaScript) has a method called extend()that makes shallow copies  and  jQuery’s  extend() creates  a  deep  copy.  YUI3  offers  a  method  called Y.clone(), which creates a deep copy and also copies over functions by binding them to the child object.

**Advantage**

There are no prototypes involved in this pattern at all; it’s only about objects and their own properties.

**References*:***

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