

// jQuery.fn = jQuery.prototype \$.fn.extend是在jquery的原型上添加方法，\$.extend是添加jquery本身的方法

//\$fn.extend 将方法扩建到jquery原型上，所有的jquery对象都可以调用，相当于成员方法

```
$.fn.extend({
    add:function() {
        alert(111);
    }
});
var test = "<div id=\"test\">fn继承</div>";
$(test).appendTo('body');
$("#test").bind('click',function() {
    console.log(typeof this);
    console.log(typeof $("#test"));
    $("#test").add();
});
```

//\$extend 是jquery本身的扩展方法，相当于静态方法。

```
$.extend({
    add2:function() {
        alert(22222);
    }
});
var test2 = "<div id=\"test2\">继承</div>";
$(test2).appendTo('body');
$("#test2").click(function() {
    $.add2();
});
```

//\$extend 是也可以合并对象。

```
var foo = { name:"jack",age:12,state:true};
var bar = { name:"张三",age:15,type:1};
function co(str) {
    console.log(str);
}
```

```

};
co(foo);
co(bar);
$.extend(foo, bar);
co("---$.extend(foo, bar);----");
co(foo); //Object {name: "张三", age: 15, state: true, type: 1}
co(bar); //Object {name: "张三", age: 15, type: 1}

```

<http://t.mb5u.com/jquery/jquery.extend.html>

原文链接: <http://caibaojian.com/jquery-extend-and-jquery-fn-extend.html>

[jQuery](#)为开发插件提供了两个方法, 分别是:

[jQuery.fn.extend\(\)](#);

[jQuery.extend\(\)](#); [•](#)

jQuery.fn

```

jQuery.fn = jQuery.prototype = {
    init: function( selector, context ) { //...
//.....
};

```

原来 `jQuery.fn = jQuery.prototype`. 对prototype肯定不会陌生啦。

虽然 javascript 没有明确的类的概念, 但是用类来理解它, 会更方便。

jQuery便是一个封装得非常好的类, 比如我们用 语句 `$("#btn1")` 会生成一个 jQuery 类的实例。

jQuery.extend(object)

为jQuery类添加类方法, 可以理解为添加静态方法。如:

```

jQuery.extend({
min: function(a, b) { return a < b ? a : b; },
max: function(a, b) { return a > b ? a : b; }
});
jQuery.min(2, 3); // 2
jQuery.max(4, 5); // 5

```

Object jQuery.extend(target, object1, [objectN])

用一个或多个其他对象来扩展一个对象, 返回被扩展的对象

```
var settings = { validate: false, limit: 5, name: "foo" };
var options = { validate: true, name: "bar" };
jQuery.extend(settings, options);
结果: settings == { validate: true, limit: 5, name: "bar" }
```

jQuery.fn.extend(object);

对jQuery.prototype进行扩展，就是为jQuery类添加“成员函数”。jQuery类的实例可以使用这个“成员函数”。

比如我们要开发一个插件，做一个特殊的编辑框，当它被点击时，便alert 当前编辑框里的内容。可以这么做：

//code from <http://caibaojian.com/jquery-extend-and-jquery-fn-extend.html>

```
$.fn.extend({
    alertWhileClick:function() {
        $(this).click(function() {
            alert($(this).val());
        });
    }
});
```

\$("#input1").alertWhileClick(); // 页面上为：

\$("#input1") 为一个jQuery实例，当它调用成员方法 alertWhileClick后，便实现了扩展，每次被点击时它会先弹出目前编辑里的内容。

jQuery.extend() 的调用并不会把方法扩展到对象的实例上，引用它的方法也需要通过jQuery类来实现，如jQuery.init()，而 jQuery.fn.extend()的调用把方法扩展到了对象的prototype上，所以实例化一个jQuery对象的时候，它就具有了这些方法，这 是很重要的，在jQuery.js中到处体现这一点

jQuery.fn.extend = jQuery.prototype.extend

你可以拓展一个对象到jQuery的 prototype里去，这样的话就是插件机制了。

来源：[前端开发博客](#)