

## Jquery 内部是如何把函数方法添加给实例对象的？

给 Jquery 实例对象添加方法是很常用的，那该怎么做呢？

想给 Jquery 实例添加方法，其实就是在 Jquery 的原型上添加方法。在 Jquery 中提供了 `$.fn.entend()` 这样的方法来实现的。接下来探究一下关于 `$.fn.extend` 这个方法。

### `$.fn.extend` 基本用法

从字面理解嘛，这个拓展的是 `jQuery.fn` 的方法。`jQuery.fn` 是啥玩意儿呢？

从源码层面来看是：

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

所以 `jQuery.fn.extend` 拓展的是 `jQuery` 对象（原型的）的方法啊！

对象是啥？就是类的实例化嘛，例如 `$("#abc")`，`$(div)`

那就是说，`jQuery.fn.extend` 拓展的方法，你得用在 `jQuery` 对象上面才行说白了就是得这么用（假设 `xyz()` 是拓展的方法）：

```
$('selector').xyz();
```

调用方法如下：

```
(function($){  
    $.fn.extend({  
        speak: function(){  
            alert("how are you!");  
        }  
    });  
})(jQuery);  
$(document).ready(function(){  
    $("div").speak();  
})
```

以上是他的基本用法。其实很简单。与\$.fn.extend 很像的方法\$.extend()

Jquery 的扩展方法 extend 是我们在写插件的过程中常用的方法，该方法有一些重载原型，在此，我们一起去学习一下。

基本形式：

```
extend(dest,src1,src2,src3...);
```

它的含义是将 src1,src2,src3...合并到 dest 中,返回值为合并后的 dest,由此可以看出该方法合并后，是修改了 dest 的结构。如果想要得到合并的结果却又不想修改 dest 的结构，可以如下使用：

```
var newSrc=$.extend({},src1,src2,src3...)//也就是将"{}"作为 dest 参数。
```

这样就可以将 src1,src2,src3...进行合并，然后将合并结果返回给 newSrc 了。如下例：

```
var result=$.extend({},{name:"Tom",age:21},{name:"Jerry",sex:"Boy"})
```

那么合并后的结果

```
result={name:"Jerry",age:21,sex:"Boy"}
```

也就是说后面的参数如果和前面的参数存在相同的名称，那么后面的会覆盖前面的参数值。

## 二、省略 dest 参数

上述的 extend 方法原型中的 dest 参数是可以省略的，如果省略了，则该方法就只能有一个 src 参数，而且是将该 src 合并到调用 extend 方法的对象中去，如：

### 1、\$.extend(src)

该方法就是将 src 合并到 jquery 的全局对象中去，如：

```
$.extend({  
  hello:function(){alert('hello');}  
});
```

就是将 `hello` 方法合并到 `jquery` 的全局对象中。

两者调用方式不同：

`jQuery.extend()`, 一般由传入的全局函数来调用, 主要是用来拓展个全局函数, 如 `$.init()`, `$.ajax()`;

`jQuery.fn.extend()`, 一般由具体的实例对象来调用, 可以用来拓展个选择器, 例如 `$.fn.each()`;

两者的主要功能作用不同：

`jQuery.extend(object)`; 为扩展 `jQuery` 类本身, 为自身添加新的方法。

`jQuery.fn.extend(object)`; 给 `jQuery` 对象添加方法

大部分插件都是用 `jQuery.fn.extend()`

关于以上的内容在我们的就业班中都有讲过（`jquery` 中的第一堂课），大家也可以去看看视频。