**自定义jquery 插件**

## 插件分类

|  |  |  |
| --- | --- | --- |
| 对象方法插件 | 附加到jQuery.fn对象上。  eg：appendTo()、addClass() | jQuery.fn.extend() |
| 封装全局函数插件 | 附加到jQuery对象本身上。  eg：jQuery.ajax()、jQuery.trim() | jQuery.extend() |
| 选择器插件 | 扩展选择器 |

## 插件基本特点

|  |
| --- |
| 1、命名推荐：jquery.插件名.js  2、在插件内部，this指向当前通过选择器获取的jquery对象，而不是DOM对象  3、可以通过this.each来遍历所有元素  4、所有的方法后函数插件都应当以分号结尾，否则压缩可能会出问题。  5、插件应该返回一个jquery对象，以保证插件可以链式操作。 |

## 插件中的闭包

|  |
| --- |
|  |

## 常见jquery插件代码结构

|  |
| --- |
| ;(function($){  //代码  })(jQuery); |

## 用jQuery.extend()扩展已有Object对象

|  |
| --- |
| var settimgs = {validate:false, limit:5, name:"foo"};  var options = {validate: false, name:'bar"};  var newOptions=jQuery.extend(settings, options);  结果为newOptions={validate:true, limit:5, name:"bar"}; |

## jQuery.extend()经常用于设置插件方法的一系列默认参数

|  |
| --- |
| function foo(options){  options = jQuery.extend({  name:"bar",  length:5,  dataType:"xml"  }, options);  } |

## 实例1：对象方法插件

|  |
| --- |
| ;(function($){  jQuery.fn.extend({  "color":function(value){  if(value==undefined){  return this.css("color");  }else{  return this.css("color", value);  }  },    "border":function(options){  options = $.extend({  odd: 'odd',  even:"even",  selected:"selected"  }, options);    return this.each(function(){  //插件代码  });  }  });  })(jQuery); |