Evan
Only let oneself become strong enough, good enough, can afford the life that you want to.

Jquery中find与each方法使用详解

标签： jquery find each

2017-02-11 13:24 138人阅读 评论(0) 收藏 编辑 删除

分类： JQuery (9)

目录(?) [+]

本文实例讲述了**jQuery**中find与each方法用法。分享给大家供大家参考。具体如下：

一、find()方法

jQuery选择器非常强大，利用css的命名规约，可以更快更方便的找出想要的元素。

图解：

```
<section>
    <input type="hidden" name="productPicture" value="${propertie.productPicture}">
    <input type="hidden" name="productPrice" value="${propertie.productPrice}">
    <input type="hidden" name="productStock" value="${propertie.productStock}">
    <input type="hidden" name="propertyName" value="${propertie.propertyName}">
    ${propertie.propertyName}
</section>
```

```
$('.type-button-detail').find('section').click(function(){
    var productPicture = $(this).find("input[name='productPicture']").val();
    var productPrice = $(this).find("input[name='productPrice']").val();
    var productStock = $(this).find("input[name='productStock']").val();
    var propertyName = $(this).find("input[name='propertyName']").val();
    alert(productPicture+"  "+productPrice+"  "+productStock+"  "+propertyName);

    var price=Number($('.ui-currency-sign.type-00').text());
    var index=$(this).index();
    var $type=$(this).text();
    var src=$('.main-image').attr('src');   //读取商品主图

    $(this).toggleClass('choosed').siblings().removeClass('choosed');
    if($(this).hasClass('choosed')){
        $('.display-choosed,.choose-product-type').html('已选择 ' +propertyName);
        $('.storage-count').find('span').html(productStock);
        $('.display-price').find('span').html(productPrice);
        $('.type-display-image').find('img').attr('src',productPicture);
    }else{
        $('.display-choosed,.choose-product-type').html('选择 属性分类');
        $('.storage-count').find('span').html(100);
        $('.display-price').find('span').html(price);
        $('.type-display-image').find('img').attr('src',src);
    }
})
```


比如：

```
[javascript]
<span style="font-size:18px;">

$("#id")
```

```
$("#"+"id")
$(this)
$(element)


</span>
```

等等，只要灵活运用，就能爆发出强大的可造型。

但是在实际使用中，仍然觉得有些不足。

如果想要在某个元素下寻找特定的元素，仅仅依靠上面这个方法，就必须对 $("#id")获取的元素进行遍历，获取其子元素。如此一来就显得格外的繁琐，代码量也非常庞大。

于是这就需要用到find()方法。

```
[javascript]
<span style="font-size:18px;">

$("#id").find("#child");
$("#id").find(".child");
$("#id").find("input[type='image']");


</span>
```

非常方便好用。

除了上面的find()方法之外，还有一种查找子元素的方法。

```
[javascript]
<span style="font-size:18px;">

$(".child",parent);


</span>
```

这种方法与find()方法的结果是一样的，也更加简洁。

我们举个例子：

```
[javascript]
<span style="font-size:18px;">

 function(table){
    $("tr",table).css("background-color","red");
 }


</span>
```

这种方法，方便代码的重用，更符合闭包的写法。

二、each()方法

each()方法能使DOM循环结构简洁，不容易出错。each()函数封装了十分强大的遍历功能，使用也很方便，它可以遍历一维数组、多维数组、DOM, JSON 等等
在javaScript开发过程中使用$each可以大大的减轻我们的工作量。

下面提一下each的几种常用的用法

each处理一维数组

```
[javascript]
<span style="font-size:18px;">

var arr1 = [ "aaa", "bbb", "ccc" ];
$.each(arr1, function(i,val){
    alert(i);
    alert(val);
});


</span>
```

alert(i)将输出0，1，2
alert(val)将输出aaa，bbb，ccc

each处理二维数组

```
[javascript]
<span style="font-size:18px;">
    var arr2 = [['a', 'aa', 'aaa'], ['b', 'bb', 'bbb'], ['c', 'cc', 'ccc']]
      $.each(arr, function(i, item){
         alert(i);
         alert(item);
      });


</span>
```

arr2为一个二维数组，item相当于取这二维数组中的每一个数组。
item[0]相对于取每一个一维数组里的第一个值
alert(i)将输出为0，1，2，因为这二维数组含有3个数组元素
alert(item)将输出为 ['a', 'aa', 'aaa'] , ['b', 'bb', 'bbb'] , ['c', 'cc', 'ccc']

对此二位数组的处理稍作变更之后

```
[javascript]
<span style="font-size:18px;">

     var arr = [['a', 'aa', 'aaa'], ['b', 'bb', 'bbb'], ['c', 'cc', 'ccc']]
         $.each(arr, function(i, item){
              $.each(item,function(j,val){
                    alert(j);
                    alert(val);
```

```
        });
    });


</span>
```

alert(j)将输出为0，1，2，0，1，2，0，1，2

alert(val)将输出为a，aa，aaa，b，bb，bbb，c，cc，ccc

each处理json数据，这个each就有更厉害了，能循环每一个属性

```
[javascript]
<span style="font-size:18px;">


var obj = { one:1, two:2, three:3};
  each(obj, function(key, val) {
        alert(key);
        alert(val);
  });


</span>
```

这里alert(key)将输出one two three
alert(val)将输出one，1，two，2，three,3
这边为何key不是数字而是属性呢，因为json格式内是一组无序的属性-值，既然无序，又何来数字呢。
而这个val等同于obj[key]

ecah处理dom元素，此处以一个input表单元素作为例子。
如果你dom中有一段这样的代码

```
[javascript]
<span style="font-size:18px;">

    <input name="aaa" type="hidden" value="111" />
    <input name="bbb" type="hidden" value="222" />
    <input name="ccc" type="hidden" value="333" />
    <input name="ddd" type="hidden"  value="444"/>


</span>
```

然后你使用each如下

```
[javascript]
<span style="font-size:18px;">

$.each($("input:hidden"), function(i,val){
    alert(val);
    alert(i);
    alert(val.name);
```

```
    alert(val.value);
});


</span>
```

那么，alert(val)将输出[object HTMLInputElement]，因为它是一个表单元素。
alert(i)将输出为0，1，2，3
alert(val.name);将输出aaa,bbb,ccc,ddd，如果使用this.name将输出同样的结果
alert(val.value)； 将输出111,222,333,444，如果使用this.value将输出同样的结果

如果将以上面一段代码改变成如下的形式

[javascript]

```
<span style="font-size:18px;">

    $("input:hidden").each(function(i,val){
        alert(i);
        alert(val.name);
        alert(val.value);
    });


</span>
```

可以看到，输出的结果是一样的，至于两种写法究竟区别在哪，我也还不知。此改变运用到上面几段数组的操作也会输出同样的结果。

这样，几个例子的实际结果已经得到答案。接着再继续往下研究，总不能知其然不知其所以然。

从以上的例子中可知jQuery和jQuery对象都实现了该方法，对于jQuery对象，只是把each方法简单的进行了委托：把jQuery对象作为第一个参数传递给jQuery的each方法。

看下jQuery中的each实现（网络摘抄）

[javascript]

```
<span style="font-size:18px;">

    function (object, callback, args) {
    //该方法有三个参数:进行操作的对象obj，进行操作的函数fn，函数的参数args
    var name, i = 0,length = object.length;
    if (args) {
    if (length == undefined) {
    for (name in object) {
    if (callback.apply(object[name], args) === false) {
    break;
    }
    }
    } else {
    for (; i < length;) {
    if (callback.apply(object[i++], args) === false) {
    break;
    }
    }
    }
    } else {
    if (length == undefined) {
    for (name in object) {
    if (callback.call(object[name], name, object[name]) === false) {
    break;
```

```
    }
    }
    } else {
    for (var value = object[0]; i < length && callback.call(value, i, value) !== false; value = object[++i]) {}
    /*object[0]取得jQuery对象中的第一个DOM元素，通过for循环，
    得到遍历整个jQuery对象中对应的每个DOM元素，通过 callback.call( value,i,value);
    将callback的this对象指向value对象，并且传递两个参数,i表示索引值，value表示DOM元素；
    其中callback是类似于 function(index, elem) { ... } 的方法。
    所以就得到 $("...").each(function(index, elem){ ... });
    */
    }
    }
    return object;
    }


</span>
```

jquery会自动根据传入的元素进行判断，然后在根据判断结果采取apply还是call方法的处理。在fn的实现中，可以直接采用this指针引用数组或是对象的子元素。

1.obj对象是数组

each方法会对数组中子元素的逐个进行fn函数调用，直至调用某个子元素返回的结果为false为止，也就是说，我们可以在提供的fn函数进行处理，使之满足一定条件后就退出each方法调用。当each方法提供了arg参数时，fn函数调用传入的参数为arg，否则为：子元素索引，子元素本身

2.obj 对象不是数组

该方法同1的最大区别是：fn方法会被逐次不考虑返回值的进行进行。换句话说，obj对象的所有属性都会被fn方法进行调用，即使fn函数返回false。调用传入的参数同1类似。

三 eq() 定义和用法

eq() 方法将匹配元素集缩减值指定 index 上的一个。

语法

```
.eq(index)
```

参数	描述
index	整数，指示元素的位置（最小为 0）。 如果是负数，则从集合中的最后一个元素往回计数。

详细说明

如果给定表示 DOM 元素集合的 jquery 对象，.eq() 方法会用集合中的一个元素构造一个新的 jQuery 对象。所使用的 index 参数标示集合中元素的位置。

请看下面这个简单的列表：

```
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li>list item 3</li>
  <li>list item 4</li>
```

```
  <li>list item 5</li>
</ul>
```

例子 **1**

我们可以把该方法应用到这个列表项目集：

```
$('li').eq(2).css('background-color', 'red');
```

这个调用的结果是为项目 3 设置了红色背景。请注意，index 是基于零的，并且是在 jQuery 对象中引用元素的位置，而不是在 DOM 树中。

综合的例子

[javascript]

```
<tbody id="ScoreTab">
    <tr>
        <td>陈晶/chenjing1</td>
        <td>
        <input type="radio" name="grade0" value="5">5分
        <input type="radio" name="grade0" value="4">4分
        <input type="radio" name="grade0" value="3">3分
        <input type="radio" name="grade0" value="2">2分
        <input type="radio" name="grade0" value="1">1分
        <span style="display:none">e74dde12-68a3-36d6-8cac-a778186053fd</span>
        </td>
    </tr>
    <tr>
        <td>张硕/zhangshuo</td>
        <td>
        <input type="radio" name="grade1" value="5">5分
        <input type="radio" name="grade1" value="4">4分
        <input type="radio" name="grade1" value="3">3分
        <input type="radio" name="grade1" value="2">2分
        <input type="radio" name="grade1" value="1">1分
        <span style="display:none">a456efa0-af65-31ff-9646-26a8800e77cb</span>
        </td>
    </tr>
</tbody>
```

代码

[javascript]

```
<span style="font-size:18px;">

    function getScore(opt)
    {
        var x=0;
        //获取 <tbody id="ScoreTab">下面的所有tr
        $("#ScoreTab").find("tr").each(function(){
            //获取tr下面的所有子元素
            var arrtd = $(this).children();
            var psn = arrtd.eq(0).text();
            //获取第二个子元素下面 input元素
            var scoreInput = arrtd.eq(1).find("input");
            //获取单选框选中的值
            var scoreNum = $("input[name='grade"+x+"']:checked").val();
            var psn = arrtd.eq(1).find("span").text();
            x++;

        })
    }
```