第十二课 工具函数

学习目录

- **数组对象工具函数**
- **其他工具函数**

一．数组对象工具函数

1.$.inArray(value,arr,[index])

判断元素是否在数组中。

第一个参数 value:用于在数组中查找是否存在的某个值。

第二个参数 arr:待查找的数组。

第三个参数是可选参数，用来指定起始下标，默认为 0。

如果查找的值在数组中就返回这个值的下标，如果不在就返回-1。

```
var arr = [ 1, "星星课堂", 2, 300 ];

$.inArray(300, arr);  //3
```

2.$.each(object,callback)

遍历数组、对象、jquery 获取的对象集合等任意对象。

第一个参数 object:可以是任何对象 ,比如说 jquery 获取到的元素对象集合、数组或者对象。

第二个参数 callback:每一个遍历的元素要执行的回调函数。

```
var arr = [100,200,300];

$.each(arr, function(index,item){
```

```
  console.log("数组下标" + index + "数组元素" + item);
});
```

3.$.makeArray(object)

可以把拥有 length 属性的类数组对象转成数组。

第一个参数 object:拥有 length 属性的类数组对象。

```
var $div = document.getElementsByTagName("div");
var divArr = $.makeArray($div);
console.log(divArr);
```

4.$.extend([deep],target,[objN])

用于把一个或多个对象的内容合并到目标对象，也是拷贝对象。

第一个参数 deep:是否开启深拷贝。

第二个参数 target:目标对象。

第三个参数 objN:准备拷贝给目标对象的一个或多个对象。

```
var obj1= {
    num: 100,
    data: {aaa: 200, bbb: 300},
};
var obj2 = {
    num: 600,
    otherNum: 900
```

```
};
$.extend(obj1, obj2);//obj2 合并到 obj1 中
$.extend(true,obj1,obj2);//obj2 合并到 obj1 中
```

如果第一个参数不传，则不进行深度拷贝，第一个对象上没有的属性会从第二个对象上拷贝得到，第一个对象与第二个对象属性相同的属性值会完全拷贝到第一个对象上，第一个对象上的同名属性值会被覆盖。

如果第一个参数为 true，且多个对象的某个同名属性也都是对象，则该属性对象的属性会进行合并，而不会覆盖。

如果第一个参数为 false，则不进行对象的拷贝。

另外一种用法是在组件化开发中合并配置参数与默认参数的时候，可以在不改变默认参数的去情况下得到一个新的设置参数用来设置组件数据。

```
var defaults = { flag: false, num: 5, title: "星星课堂" };
var options = { flag: true, title: "xingxingclassroom" };
var settings = $.extend({}, defaults, options); //合并默认对象和参数对象，不修改默认对象
console.log(settings);//{"flag":true,"num":5,"title":"xingxingclassroom"}
```

二．其他工具函数

1.$.trim(str)

删除字符串开始和末尾的所有换行符，空格(包括连续的空格)和制表符。

console.log($.trim(" I like xingxingclassroom! "));//"I like xingxingclassroom!"

2.$.param(object,[traditional])

新建对象的序列化表示形式，序列化的值相当于构建一个 url 查询字符串。

var params = { pageCount:1, pageSize:10};

var query = $.param(params);

console.log(query);//pageCount=1&pageSize=10

也可以序列化数组，但是不建议这么用，第二个参数是以传统方式形式浅层序列化。

var params = { pageCount:1, pageSize:10,arr:[1,2,3]};

var query = $.param(params,true);

console.log(query);//pageCount=1&pageSize=10&arr=1&arr=2&arr=3

3.data([key],[value])

给目标元素附加数据，或者从被选元素获取数据，这个方法在存储较多数据时比较好用。

第一个参数 key:存储的数据名。

第二个参数 value:将要存储的任意数据。

$(".div1").data("abc", { num: 100, title: "xingxingclassroom" });

$(".div1").data("abc").num//100;

$(".div1").data("abc").title//xingxingclassroom;

4.index([selector|element])

从匹配元素中搜索指定元素，并返回相应元素的索引值，从 0 开始计数。

如果未找到元素，index() 将返回 -1。

```
$(".btn").click(function(){
    console.log($(this).index());
});
```

如果参数是一组 DOM 元素或者 jQuery 对象，那么返回值就是传递的元素相对于原先集合的位置。

```
$('div').index(document.getElementById('.div1'));
```

5.each(callback)

遍历 jquery 获取的对象集合。

第二个参数 callback:每一个遍历的元素要执行的回调函数。

```
$("div").each(function(){
    console.log($(this).html())
});
```

谢谢观看！

我是星星课堂老师：周小周