**HTML5中的data-\*属性和jQuery中的.data()方法使用**

2016年02月19日 15:56:17 [SuperMan-zhang](https://me.csdn.net/fly_zxy" \t "_blank) 阅读数：32730

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/fly\_zxy/article/details/50687691

**HTML5中的data-\*属性**

我们往往会根据需要在HTML标记上添加自定义的属性来存储和操作数据，我们自定义的属性名字也千奇百怪，五花八门。我们可以通过原生的getAttribute()或jQuery中的.attr()来获取我们自定义的属性。但是前端技术在向着规范化前进。HTML5标准规定，自定义的属性都已data—\*开头，这样就区分开了固有属性和自定义属性。HTML代码如下：

<**div** id="testDiv" data-cname="张三" data-e-name="zhangsan" data-myName="my name is zs.">测试在元素上存储一个key-value</**div**>

1. $(document).ready(**function**(){
2. **var** el = document.getElementById("testDiv");
4. console.log(el.dataset.cname);//=>张三
5. el.dataset.cname = "ZS";//设置值为"ZS"
6. console.log(el.dataset.cname);//=>"ZS"
8. console.log("jQuery data:", $("#testDiv").data("eName") );
9. $("#testDiv").data("eName", "abcefg");
10. console.log("jQuery data:", $("#testDiv").data("eName") );
12. console.log("遍历testDiv上的自有属性");
13. $.each(el.dataset, **function**(key, value){
14. console.log(key+":"+value);
15. });
17. //遍历testDiv上的自有属性
18. //cname:ZS
19. //eName:zhangsan
20. //myname:my name is zs.
21. <span style="white-space:pre"> </**span**>
22. });

在获取自定义属性的值时，使用dataset属性，然后通过自定义属性(这时去掉data-前缀)的名称去获取值。dataset是JS5规范中新增加的属性，用于和data-\*配合使用。  
通过each打印的结果，我们需要注意：

1，如果data-后面包含了“-”，例如 date-e-name，"-"会去掉，并转换成为骆驼峰式的命名 eName。这是因为JSON Object 的key不能包含”-“。

2，如果data-后面的单词有大写，例如 myName，将会转存成小写 myname。这里就不理解为什么会转换成小写了，JSON Object的key是区分大小写的。求高人指定？？

3，如果你想删除一个data-\*属性，请使用 delete dataset.cname或者dataset.cname=null;

data-\*属性看起来很好，但是不幸的是只有在Chrome 8+ Firefox(Gecko) 6.0+ Internet Explorer 11+ Opera 11.10+ Safari 6+的浏览器中实现了此标准。如果你的脚步需要兼容各主流浏览器的不同版本建议还是使用原生的getAttribute()或jQuery中的.attr()。

**jQuery中的.data()**

jQuery中的.data()方法作用：在jQuery对象对应的DOM元素上获取或存放key-value对。我们可以通过html5的data-\*属性在元素上存取数据，在jQuery也可以通过data()来实现。data()方法有多个重载，传递一个参数是获取value值，传递两个参数是存储或覆盖已存在的值，传递一个JSON Object是在元素上存储或覆盖已存在的多个key-value对。如果通过data() 在元素上存储key-value，value可以是一个对象，而不是像data-\*只能存储一个字符串的value。下面给出一个综合示例：

1. <**script** id="jquery\_183" type="text/javascript" class="library" src="//cdn.bootcss.com/jquery/2.2.0/jquery.js"></**script**>
2. <!DOCTYPE html>
3. <**html**>
4. <**head**>
6. </**head**>
7. <**body**>
8. <**div** id="testDiv" data-cname="张三" data-e-name="zhangsan" data-myName="my name is zs.">
9. 测试在元素上存储一个key-value
10. </**div**>
11. </**body**>
12. </**html**>

1. $(**function**(){
3. **function** **printVlaue**(){
5. //通过data()在testDiv元素上存储 {ename:zhangsan} 健/值对。
6. $("#testDiv").data("ename", "zhangsan");
7. //修改data()通过 data-\* 属性存储的值
8. $("#testDiv").data("cname", "我是张三");
10. console.log( "获取通过data()存储在testDiv元素上的ename值:", $("#testDiv").data("ename") );
11. console.log( "获取通过 data-\* 存储在testDiv元素上的cname值:", $("#testDiv").data("cname") );
12. //$("#testDiv").jQueryremoveAttr("data-"+"abc");
14. console.log( "获取ename值:", $("#testDiv").data("ename") );
15. console.log( "获取cname值:", $("#testDiv").data("cname") )
17. console.log("删除testDiv元素上 ename 和 cname 键/值对。");
18. $("#testDiv").removeData("ename");
19. $("#testDiv").removeData("cname");
21. console.log("输出 eanem 和 canme 对应的值，看是否已经删除");
22. console.log( $("#testDiv").data("ename") );
23. console.log( $("#testDiv").data("cname") );

26. };
28. $("<button>获取值</button>").appendTo("body").bind("click", printVlaue);;
29. console.log("debug use.");;
31. });
32. /\*
33. 获取通过data()存储在testDiv元素上的ename值: zhangsan
34. 获取通过 data-\* 存储在testDiv元素上的cname值: 我是张三
35. 获取ename值: zhangsan
36. 获取cname值: 我是张三
37. 删除testDiv元素上 ename 和 cname 键/值对。
38. 输出 eanem 和 canme 对应的值，看是否已经删除
39. undefined
40. 张三
41. \*/

从上例中我们可以看出，.data()可以获取通过data-\*属性存储的值。但是通过.data()修改data-\*属性的值不会显示在HTML标记中，我们只是通过.data()在元素存储了一个新值。.removeData()也不能删除通过data-\*存储的数据。由于jQuery是一个通用的JS框架,最浏览器兼容性做的非常好，大多数情况，我们应该通过.data()在元素上存取数据(有一个好处是可以存储一个对象)。如果我们必须通过data-\*来存取和修改数据,建议使用JS5的dataset属性，这个属性的兼容问题，我们可以通过写jQuery插件来解决。这是我的插件代码，如下：

1. <**script** id="jquery\_183" type="text/javascript" class="library"
2. src="//cdn.bootcss.com/jquery/2.2.0/jquery.js"></**script**>
3. <!DOCTYPE html>
4. <**html**>
5. <**head**>
7. </**head**>
8. <**body**>
9. <**div** id="testDiv" data-cname="张三" data-e-nAme="zhangsan" data-myName="my name is zs.">
10. 测试在元素上存储一个key-value
11. </**div**>
12. </**body**>
13. </**html**>
14. $(document).ready(**function**(){
15. **function** **printValue**(){
17. **var** $testDiv = $("#testDiv").udDataAttr();
19. console.log("udDataAttr get()=>", $testDiv.get("cname") );
20. $testDiv.set("cname", "set 张三");
21. console.log("udDataAttr get()=>", $testDiv.get("cname") );
22. console.log("打印所有已data-\*开头的属性和值=>", $testDiv.dataset());
24. console.log("udDataAttr get e-name=>", $testDiv.get("e-name") );


28. //var el = document.getElementById("testDiv");
29. //console.log("el.dataset.eName=>", el.dataset.eName);
30. };
32. $("<button>获取值</button>").appendTo("body").bind("click", printValue);
34. console.log("debug use.");
35. });
37. (**function**($, window){
38. //user-defined Attribute
39. **var** plugName = "udDataAttr";
41. $.fn[plugName] = **function**(){
42. **var** $self = $(**this**),
43. obj = ($self.length && $self[0].dataset) || **null**;
45. **return** {
46. get: **function**(name){
47. **return** obj===**null**? $self.attr("data-"+name) : obj[ **this**.\_format(name) ];
48. }
49. ,set: **function**(name, value){
50. **if**(obj === **null**){
51. $self.attr("data-"+name, value);
52. }**else**{
53. obj[ **this**.\_format(name) ] = value;
54. }
55. **return** $self;
56. }
57. ,dataset: **function**(){
58. **var** newObj = {};
59. **if**(obj === **null**){
60. $.each(obj, **function**(key, value){
61. newObj[key] = value;
62. });
63. }**else**{
64. newObj = $self.data(); //获取所有以data-\*开头的属性
65. }
67. **return** newObj;
68. }
69. /\*
70. \* 将name转换成dataset可识别的格式。
71. \* 例如：e-name 转换成 eName
72. \* 总感觉 \_format()的实现不是最优的,求高手给出实现
73. \*/
74. ,\_format: **function**(name){
75. console.log("\_format old name=>",name);
76. name = name.toLowerCase();
77. **if**(name.indexOf('-')>-1){
78. **var** array = name.split('-');
79. **for**(**var** i=1,len=array.length; i<len; i++){
80. **var** v = array[i];
81. array[i] = v.substr(0,1).toUpperCase() + v.substr(1,v.length);
82. name=array.join("");
83. }
84. }
85. console.log("\_format() new name=>",name);
86. **return** name;
87. }
88. };
90. };

93. }(jQuery, window));