04jq的ajax和json

今日内容：

- 复习昨天内容：事件和运动特效
- 常用方法的学习

1、常用方法

•get() ： 下标和length属性

```
<!DOCTYPE html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>无标题文档</title>

<script type="text/javascript" src="jquery-1.10.1.min.js"></script>
<script>

//get() : 就是把JQ转成原生JS

$(function(){

    //document.getElementById('div1').innerHTML

    //alert( $('#div1').get(0).innerHTML );

    /*for(var i=0;i<$('li').get().length;i++){
        $('li').get(i).style.background = 'red';
    }*/

    for(var i=0;i<$('li').length;i++){
        $('li').get(i).style.background = 'red';

        //$('li')[i].style.background = 'red';
    }

});



</script>
</head>

<body>
<div id="div1">div</div>
<ul>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
</ul>
</body>
</html>
```

•outerWidth() ： 针对隐藏元素和参数true

```

<!DOCTYPE html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>无标题文档</title>
<style>
#div1{ width:100px; height:100px; background:red; display:none;}
</style>
<script type="text/javascript" src="jquery-1.10.1.min.js"></script>
<script>

//outerWidth()

//offsetWidth : 是获取不到隐藏元素的值

$(function(){

    //alert( $('#div1').get(0).offsetWidth );

    alert( $('#div1').outerWidth() );

});



</script>
</head>

<body>
<div id="div1">div</div>

</body>
</html>
```

•text() ： 合体的特例

```
<!DOCTYPE html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>无标题文档</title>
<style>
#div1{ width:100px; height:100px; background:red; display:none;}
</style>
<script type="text/javascript" src="jquery-1.10.1.min.js"></script>
<script>

$(function(){

    //alert( $('div').html() );

    //alert( $('div').text() );  //会获取所有的内容(特例)

    $('div').text('<h3>h3</h3>');


});



</script>
</head>

<body>
<div>div1<span>span</span></div>
<div>div2</div>
<div>div3</div>

</body>
</html>
```

•remove() :detach()

```

<!DOCTYPE html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>无标题文档</title>
<style>
#div1{ width:100px; height:100px; background:red; display:none;}
</style>
<script type="text/javascript" src="jquery-1.10.1.min.js"></script>
<script>

//detach() : 跟remove方法一样，只不过会保留删除这个元素的操作行为

$(function(){

    $('div').click(function(){
        alert(123);
    });

    var oDiv = $('div').detach();

    $('body').append( oDiv );



});



</script>
</head>

<body>
<div>div1<span>span</span></div>


</body>
</html>
```

•parents()   closest()

```

<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="jquery-1.10.1.min.js"></script>
<script>

//parents() : 获取当前元素的所有祖先节点，参数就是筛选功能

//closest() : 获取最近的指定的祖先节点(包括当前元素自身),必须要写筛选的参数,只能找到一个元素

$(function(){

    //$('#div2').parents('.box').css('background','red');

    $('#div2').closest('.box').css('background','red');

});

</script>
```

```
</head>

<body class="box">
<div id="div1">aaa
    <div id="div2" class="box">bbb</div>
</div>
</body>
</html>
```

- siblings()
- nextAll() prevAll()
- parentsUntil()   nextUntil()   prevUntil()

```

<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="jquery-1.10.1.min.js"></script>
<script>

//siblings() : 找所有的兄弟节点，参数也是筛选功能

//nextAll() : 下面所有的兄弟节点，参数也是筛选功能

//prevAll() : 上面所有的兄弟节点

//Until() : 截止

$(function(){

    $('span').nextUntil('h2').css('background','red');

});

```

```
</script>
</head>

<body>
<div>div</div>
<span>span</span>
<p>p</p>
<h1>h1</h1>
<h2>h2</h2>
<em>em</em>
</body>
</html>
```

•add() slice()

```

<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="jquery-1.10.1.min.js"></script>
<script>



$(function(){

    /*var elem = $('div');

    var elem2 = elem.add('span');

    elem.css('color','red');
    elem2.css('background','yellow');*/

    $('li').slice(1,4).css('background','red');

});

```

```

</script>
</head>

<body>
<div>div</div>
<span>span</span>
<ul>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
</ul>

</body>
</html>
```

•serialize()　serializeArray()数据串连

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="jquery-1.10.1.min.js"></script>
<script>



$(function(){

    //console.log($('form').serialize());  //string : a=1&b=2&c=3

    console.log( $('form').serializeArray() );

    [
        { name : 'a' , value : '1' },
        { name : 'b' , value : '2' },
        { name : 'c' , value : '3' }
    ]

});

</script>
</head>

<body>
<form>
    <input type="text" name="a" value="1">
    <input type="text" name="b" value="2">
    <input type="text" name="c" value="3">
</form>

</body>
</html>
```

•delegate() 事件委托：提高性能

undelegate()：取消事件委托

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="jquery-1.10.1.min.js"></script>
<script>



$(function(){

    /*$('li').on('click',function(){
        this.style.background = 'red';
    });*/

    $('ul').delegate('li','click',function(){

        this.style.background = 'red';

        $('ul').undelegate();

    });

});

</script>
</head>

<body>
<ul>
    <li>11111</li>
    <li>1111</li>
    <li>1111</li>
    <li>111</li>
    <li>1111</li>
</ul>

</body>
</html>
```

- ev.data   ev.target   ev.type

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="jquery-1.10.1.min.js"></script>
<script>



$(function(){

    $('#div1').on('click',{name:'hello'},function(ev){
        //alert(ev.data.name);

        //alert( ev.target );

        alert( ev.type );

    });


});

</script>
</head>

<body>
<div id="div1">aaaa</div>
</body>
</html>
```

●$下的常用方法

- •type()
- •trim()
- •inArray()
- •proxy()
- •noConflict()
- •parseJSON()

•makeArray()

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="jquery-1.10.1.min.js"></script>
<script>


//$().css()  $().html()  $().val()  : 只能给JQ对象用

//$.xxx()  $.yyy()  $.zzz()  : 不仅可以给JQ用，也可以给原生JS用 : 叫做工具方法




$(function(){

    //var a = null;

    //$.type() : 也是判断类型

    //alert( typeof a );

    //alert( $.type(a) );

    var str = '   hello  ';

    alert('('+$.trim(str)+')');

});

</script>
</head>

<body>

```

```
</body>
</html>
```

inArray()：找位置

proxy()：改变this指向

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="jquery-1.10.1.min.js"></script>
<script>


//$().css()  $().html()  $().val()  : 只能给JQ对象用

//$.xxx()  $.yyy()  $.zzz()  : 不仅可以给JQ用，也可以给原生JS用 : 叫做工具方法


//inArray() : 类似于 indexOf

$(function(){

    //var arr = ['a','b','c','d'];

    //alert(  $.inArray('b',arr)  );


    //proxy()  : 改变this指向的

    function show(n1,n2){
        alert(n1);
        alert(n2);
        alert(this);
    }

    //show();

    //$.proxy(show , document,3)(4);


    $(document).click( $.proxy(show,window,3,4)  );

});

</script>
</head>

<body>

</body>
</html>
```

noConflict()
•parseJSON()
•makeArray()

```

<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="jquery-1.10.1.min.js"></script>
<script>


//$().css()  $().html()  $().val()  : 只能给JQ对象用

//$.xxx()  $.yyy()  $.zzz()  : 不仅可以给JQ用，也可以给原生JS用 : 叫做工
```

```
具方法


//$ , jQuery

//noConflict() : 防止冲突的

/*var aa = $.noConflict();

var $ = 10;

aa(function(){

    aa('body').css('background','red');

});*/


//var str = '{"name":"hello"}';
//alert($.parseJSON( str ).name);


window.onload = function(){

    var aDiv = document.getElementsByTagName('div');  //类数组

    $.makeArray(aDiv).push();

};

</script>
</head>

<body>
<div></div>
<div></div>
<div></div>
</body>
</html>
```

2、ajax函数的运用

- ajax()： json形式的配置参数
  - –url success
  - –error contentType
  - –data type
  - –dataType cachetimeout
- 抽象出来的方法
  - –get()
  - –post()
  - –getJSON()
    - »支持jsonp的形式：指定?callback=?

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="jquery-1.10.1.min.js"></script>
<script>

/*$.ajax({
    url : 'xxx.php',
    data : 'name=hello&age=20',
    type : 'POST',
    success : function(data){
        alert(1);
    },
    error : function(){
        alert(2);
    }
});*/

$.get('xxx.php',function(){

});

$.post('xxx.php',function(){

});

$.getJSON('xxx.php?callback=?',function(data){
    data
});

随机({});

</script>
</head>

<body>
</body>
</html>
```

3、jq如何调用json数据

复习json：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <script src="jquery-1.10.1.min.js" type="text/javascript"
```

```
charset="utf-8"></script>
        <script type="text/javascript">
            $(function(){
                var json={
                    'name':'张三',
                    'sex':'女',
                    'tel':164656656
                }

                alert(json.name);
            });
        </script>
    </head>
    <body>

    </body>
</html>
```

遍历json：for in 循环

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <script src="jquery-1.10.1.min.js" type="text/javascript"
charset="utf-8"></script>
        <script type="text/javascript">
            $(function(){
                var json={
                    'name':'张三',
                    'sex':'女',
                    'tel':164656656
                }

                for(var i in json){
                    alert(i+': '+json[i]);
                }
            });
        </script>
    </head>
    <body>

    </body>
</html>
```

上面的数据的json形式比较简单。json形式一般有两种：

第一种：

```
var json={'name':'张三','sex':'女','tel':164656656}
```

第二种：数组和json的结合

```
                      var json = {
                             '01':
                             [
                             {id:1,name:'张晓','语文':80,'英语',75},
                             {id:2,name:'李成','语文':88,'英语',75},
                             {id:3,name:'罗明','语文':90,'英语',75}
                             ]
                             },
                             '02':
                             [
                             {id:1,name:'李梅','语文':80,'英语',75},
                             {id:2,name:'刘德华','语文':88,'英语',75},
                             {id:3,name:'王祖贤','语文':90,'英语',75}
                             ]

                             }
```

对于第一种类型的json我们采用for in 循环就很简单的遍历了json。但是第二种就没有那么简单。

对于第一种数据的使用：

```html
<!DOCTYPE html>
<html>

    <head>
        <meta charset="UTF-8">
        <title></title>
        <script src="jquery-1.10.1.min.js" type="text/javascript" charset="utf-8"></script>
        <script type="text/javascript">
            $(function() {
                    var json = {
                        'name': '张三',
                        'sex': '女',
                        'tel': 164656656
                    }

                $('#btn').click(function(){
                    var str='';
                    str=str+'姓名是：'+json.name+'<br>';
                    str=str+'性别是：'+json.sex+'<br>';
                    str=str+'电话是：'+json.tel;

                    $('div').html(str);
                });
            });

        </script>
    </head>

    <body>
            <input type="button" id="btn" value="点击" />
            <div></div>
    </body>
```

```

</html>
```

each遍历数组和json：

```
<script type="text/javascript">
            $(function() {
                    var anObject = {
                        one: 1,
                        two: 2,
                        three: 3
                    }; //对json数组each
                    $.each(anObject, function(name, value) {
                        alert(name);
                        alert(value);
                    });
                    var anArray = ['4', '5', '6'];
                    $.each(anArray, function(n, value) {
                        alert(n);
                        alert(value);
                    });
                }
            );
        </script>
```

再看一个例子：

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="UTF-8">
        <title></title>
        <script src="jquery-1.10.1.min.js" type="text/javascript"
charset="utf-8"></script>
```

```
        <script>
            var arr = [{
                name: "john",
                lang: "js"
            }, {
                name: "nailwl",
                lang: "jquery"
            }, {
                name: "吴磊",
                lang: "ext"
            }];
            $.each(arr, function(index, content) {
                alert("the man's no. is: " + index + ",and " +
content.name + " is learning " + content.lang);
            });
        </script>
    </head>

    <body>

    </body>

</html>
```

jquery 对象的 $().each() 方法，此方法可用于例遍任何对象
回调函数拥有两个参数：
第一个为对象的成员或数组的索引例遍数组，同时使用元素索引和内容

```
$.each( [0,1,2], function(index, content){
  alert( "item #" + index + " its value is: " + content );
});
```

第二种数据形式的遍历：

```html
<!DOCTYPE html>
<html>

    <head>
        <meta charset="UTF-8">
        <title></title>
        <script src="jquery-1.10.1.min.js" type="text/javascript"
charset="utf-8"></script>
        <script type="text/javascript">
            $(function() {


                        var ss = {
                            'a':
                            [
                            {id:1,name:'张晓','语文':80,'英语':75},
                            {id:2,name:'李成','语文':88,'英语':75},
                            {id:3,name:'罗明','语文':90,'英语':75}
                            ]
                            ,
                            'b':
                            [
                            {id:1,name:'李梅','语文':80,'英语':75},
                            {id:2,name:'刘德华','语文':88,'英语':75},
                            {id:3,name:'王祖贤','语文':90,'英语':75}
                            ]

                            }


                        //var a=ss.a
                    $.each(ss,function(index,value){
                        $.each(value,function(n,content){
                            alert('学号是'+content.id+'名字
是:'+content.name);
                        }
                        )
                    }



```

```
                );

            });
        </script>
    </head>

    <body>

    </body>

</html>
```