01jq选择器

今日内容：

- 了解jq基础
- 了解常用选择器
- 了解常用函数

1、基础知识

●什么是JQ？

•一个优秀的JS库，大型开发必备

●JQ的好处？

•简化JS的复杂操作

•不再需要关心兼容性

•提供大量实用方法

- 如何学习JQ？
  - www.jquery.com
  - JQ只是辅助工具，要正确面对
  - 需要分阶段学习

目前这个阶段，主要学习如何来使用jQuery操作DOM，其实就是学习jQuery封装好的那些功能方法，这些方法叫做API（Application Programming Interface应用程序编程接口）。

这些API的共同特点是：几乎全都是方法。所以，在使用jQuery的API时，都是方法调用，也就是说要加小括号()，小括号里面是相应的参数，参数不同，功能不同。

兼容性：

Current Active Support

Desktop

- Chrome: (Current - 1) and Current
- Edge: (Current - 1) and Current
- Firefox: (Current - 1) and Current
- Internet Explorer: 9+
- Safari: (Current - 1) and Current
- Opera: Current

Mobile

- Stock browser on Android 4.0+
- Safari on iOS 7+

版本一：1.x版本

版本二：2.x版本

版本三：3.x版本

jquery 1.12.4版本api文档：

http://www.php100.com/manual/jquery/

同版本两个文件的区别：

min：压缩版，压缩过后，体积会更小

压缩指的是：把注释、空格、换行全部去掉，把变量名称尽可能的换成更加简短的字符。

压缩的主要目的：就是让文件变的更小。

平时开发过程中，这两个文件使用哪个都可以；但是，项目上线的时候，推荐使用压缩版。

jQuery入口函数与js入口函数的区别（理解）

js入口函数指的是：window.onload = function() {};

区别一：书写个数不同

Js入口函数只能出现一次，出现多次会存在事件覆盖的问题。

jQuery的入口函数，可以出现任意多次，并不会存在事件覆盖问题。

区别二：执行时机不同

Js入口函数是在所有的文件资源加载完成后，才执行。这些文件资源包括：页面文档、外部的js文件、外部的css文件、图片等。

jQuery的入口函数，是在文档加载完成后，就执行。文档加载完成指的是：DOM树加载完成后，就可以操作DOM了，不用等到所有的外部资源都加载完成。

文档加载的顺序：从上往下，边解析边执行。

2、选择器

基本选择器（重点）

符号(名称)	说明	用法
#	Id选择器	$(“#btnShow”).css(“color”, “red”); 选择id为btnShow的一个元素（返回值为jQuery对象，下同）
.	类选择器	$(“.liItem”).css(“color”, “red”); 选择含有类liItem的所有元素
element	标签选择器	$(“li”).css(“color”,“red”); 选择标签名为li的所有元素

```
<body>
<div id="div1" class="box">div</div>
<script>

//document.getElementById('div1');
//document.getElementsByTagName('div');
//getByClass(document,'box');

//$('#div1').css('background','red');
//$('div').css('background','red');
$('.box').css('background','red');

</script>
</body>
```

1.7.2 层级选择器（重点）、基本过滤选择器

符号(名称)	说明	用法
层级选择器		
空格	后代选择器	$(“#j_wrap li”).css(“color”, “red”); 选择id为j_wrap的元素的所有后代元素li
>	子代选择器	$(“#j_wrap > ul > li”).css(“color”, “red”); 选择id为j_wrap的元素的所有子元素ul的所有子元素li
基本过滤选择器		
:eq(index)	选择匹配到的元素中索引号为index的一个元素，index从0开始	$(“li:eq(2)”).css(“color”, ”red”); 选择li元素中索引号为2的一

		个元素
:odd	选择匹配到的元素中索引号为奇数的所有元素，index从0开始	$(“li:odd”).css(“color”,“red”); 选择li元素中索引号为奇数的所有元素
:even	选择匹配到的元素中索引号为偶数的所有元素，index从0开始	$(“li:odd”).css(“color”,“red”); 选择li元素中索引号为偶数的所有元素

1.7.3 筛选选择器（方法）（重点）

符号(名称)	说明	用法
find(selector)	查找指定元素的所有后代元素（子子孙孙）	$(“#j_wrap”).find(“li”).css(“color”, “red”); 选择id为j_wrap的所有后代元素li
children()	查找指定元素的直接子元素（亲儿子元素）	$(“#j_wrap”).children(“ul”).css(“color”, “red”); 选择id为j_wrap的所有子代元素ul
siblings()	查找所有兄弟元素（不包括自己）	$(“#j_liItem”).siblings().css(“color”, “red”); 选择id为j_liItem的所有兄弟元素
parent()	查找父元素（亲的）	$(“#j_liItem”).parent(“ul”).css(“color”, “red”); 选择id为j_liItem的父元素
eq(index)	查找指定元素的第index个元	$(“li”).eq(2).css(“color”, “red”);

	素，index是索引号，从0开始	选择所有li元素中的第二个

3、jq的设计思想

- JQ写法
  - –方法函数化
  - –链式操作
  - –取值赋值合体
- JQ与JS关系
  - –可以共存，不能混用

```
<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>

//方法函数化:

/*window.onload = function(){};

$(function(){});

function $(){}

innerHTML = 123;

html(123)

function html(){}

onclick = function(){};
```

```
22
23 click(function(){})
24
25 function click(){}*/
26
27 /*window.onload = function(){
28     var oDiv = document.getElementById('div1');
29
30     oDiv.onclick = function(){
31         alert( this.innerHTML );
32     };
33
34 };*/
35
36 $(function(){
37
38     //var oDiv = $('#div1');
39
40     $('#div1').click(function(){
41
42         alert( $(this).html() );
43
44     });
45
46 });
47
48 $('ul').children().css('background','red');
49
50 </script>
51 </head>
52
53 <body>
54 <div id="div1">div</div>
55
56 </body>
```

js和jq的关系：可以共存，但是不要混用：

```
1 <script>
2
3
4
```

```
$(function(){

    $('#div1').click(function(){

        //alert( $(this).html() );  //jq的写法

        //alert( this.innerHTML );  //js的写法

        alert( $(this).innerHTML );  //错误的
        alert( this.html() );  //错误的



    });

});


</script>
```

链式写法：

```
<script>



$(function(){

    /*var oDiv = $('#div1');

    oDiv.html('hello');

    oDiv.css('background','red');

    oDiv.click(function(){
        alert(123);
    });*/

    $('#div1').html('hello').css('background','red').click(function(){
        alert(123);
    });
```

```
20
21 });
22
23
24
25 </script>
```

取值赋值合体：

```
1  <script>
2
3
4
5  $(function(){
6
7      //oDiv.innerHTML = 'hello';  //赋值
8
9      //alert( oDiv.innerHTML );  //取值
10
11     //$('#div1').html('hello');  //赋值
12
13     //alert( $('#div1').html() ); //取值
14
15     css('width','200px')
16     css('width')
17 });
18
19
20
21 </script>
```

```
1  <script>
2
3
4
5  $(function(){
6
7
8      //alert( $('li').html() );  //当一组元素的时候，取值是一组中的第一个
9
```

```
    $('li').html('hello');  //当一组元素的时候,赋值是一组中的所有元素

});



</script>
```

4、jq常用的方法

- $()下的常用方法
  - has()
  - not()
  - filter()
  - next()
  - prev()
  - find()
  - eq()
  - index()
  - attr()

attr（ ）：改变样式和值

```
<script>



$(function(){

    //alert($('div').attr('title'));

    $('div').attr('title','456');
    $('div').attr('class','box');

});



</script>
</head>

<body>
<div title="123">div</div>
</body>
```

filter（ ）：过滤含有。反义词not（ ）

```
<script>

//filter : 过滤
//not : filter的反义词

$(function(){

    //$('div').filter('.box').css('background','red');

    $('div').not('.box').css('background','red');

});



</script>
```

```
</head>

<body>
<div class="box">div1</div>
<div>div2</div>
</body>
```

has（ ）：含有子节点的节点

```
<script>

//has : 包含

$(function(){

    //$('div').has('span').css('background','red');

    //$('div').has('.box').css('background','red');
    $('div').filter('.box').css('background','red');

});



</script>
</head>

<body>
<div>div1<span class="box">span</span></div>
<div class="box">div2</div>
</body>
```

next（ ）：下一个兄弟节点

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


$(function(){

```

```
    $('div').next().css('background','red');

});



</script>
</head>

<body>
<div>div</div>
<span>span</span>
<p>p</p>
</body>
```

find ( ) :

```
<script>


$(function(){

    //$('div').find('h2').css('background','red');

    $('div').find('h2').eq(1).css('background','red');

});



</script>
</head>

<body>
<div>
    <h3>h3</h3>
    <h2>h2</h2>
    <h3>h3</h3>
    <h3>h3</h3>
    <h2>h2</h2>
    <h2>h2</h2>
</div>
```

```
<h2>h2</h2>

</body>
```

index ( ) ：

```
<script>


$(function(){

    alert( $('#h').index() );  //索引就是当前元素在所有兄弟节点中的位置

});



</script>
</head>

<body>
<div>
    <h3>h3</h3>
    <h2>h2</h2>
    <h3>h3</h3>
    <h3 id="h">h3</h3>
    <h2>h2</h2>
    <h2>h2</h2>
</div>

<h2>h2</h2>

</body>
```

选项卡：

```
<script type="text/javascript" src="jquery-1.10.1.min.js"></script>
<script>
/*window.onload = function(){
```

```
    var oDiv = document.getElementById('div1');
    var aInput = oDiv.getElementsByTagName('input');
    var aCon = oDiv.getElementsByTagName('div');

    for(var i=0;i<aInput.length;i++){

        aInput[i].index = i;

        aInput[i].onclick = function(){

            for(var i=0;i<aInput.length;i++){
                aInput[i].className = '';
                aCon[i].style.display = 'none';
            }

            this.className = 'active';
            aCon[this.index].style.display = 'block';

        };
    }

};*/


$(function(){

    $('#div1').find('input').click(function(){

        $('#div1').find('input').attr('class','');
        $('#div1').find('div').css('display','none');

        $(this).attr('class','active');

        $('#div1').find('div').eq( $(this).index()
).css('display','block');

    });

});
</script>
</head>

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

```
    <input class="active" type="button" value="1" />
    <input type="button" value="2" />
    <input type="button" value="3" />
    <div style="display:block">111111</div>
    <div>222222</div>
    <div>333333</div>
</div>
</body>
```

选择器分类：

- 基本选择器
- 层次选择器
- 过滤选择器
  - 简单过滤选择器
  - 内容过滤选择器
  - 可见性过滤选择器
  - 属性过滤选择器
  - 子元素过滤选择器
  - 表单对象属性过滤选择器
- 表单选择器

1、基本选择器

$("#myELement")　选择id值等于myElement的元素，id值不能重复在文档中只能有一个id值是myElement所以得到的是唯一的元素

$("div")　　选择所有的div标签元素，返回div元素数组

$(".myClass")　　选择使用myClass类的css的所有元素

$("*")　　选择文档中的所有的元素，可以运用多种的选择方式进行联合选择：例如$("#myELement,div,.myclass")

2、层次选择器

$("form input")　　选择所有的form元素中的input元素

$("#main > *")　　选择id值为main的所有的子元素

$("label + input")　选择所有的label元素的下一个input元素节点，经测试选择器返回的是label标签后面直接跟一个input标签的所有input标签元素

$("#prev ~ div")　　同胞选择器，该选择器返回的为id为prev的标签元素的所有的属于同一个父元素的div标签

3、过滤选择器

基本过滤选择器：

$("tr:first")　　选择所有tr元素的第一个

$("tr:last")　　选择所有tr元素的最后一个

$("input:not(:checked) + span")

过滤掉：checked的选择器的所有的input元素

$("tr:even")　　选择所有的tr元素的第0，2，4... ...个元素（注意：因为所选择的多个元素时为数组，所以序号是从0开始）

$("tr:odd")　　选择所有的tr元素的第1，3，5... ...个元素

$("td:eq(2)")　　选择所有的td元素中序号为2的那个td元素

$("td:gt(4)")　　选择td元素中序号大于4的所有td元素

$("td:lt(4)")　　选择td元素中序号小于4的所有的td元素

$(":header")　　选择h1、h2、h3之类的

$("div:animated")　选择正在执行动画效果的元素

内容过滤选择器：

$("div:contains('John')") 选择所有div中含有John文本的元素

$("td:empty")　选择所有的为空（也不包括文本节点）的td元素的数组

$("div:has(p)")　选择所有含有p标签的div元素

$("td:parent")　选择所有的以td为父节点的元素数组

可视化过滤选择器：

$("div:hidden")　选择所有的被hidden的div元素

$("div:visible")　选择所有的可视化的div元素

属性过滤选择器：

$("div[id]")　选择所有含有id属性的div元素

$("input[name='newsletter']")　选择所有的name属性等于'newsletter'的input元素

$("input[name!='newsletter']") 选择所有的name属性不等于'newsletter'的input元素

$("input[name^='news']")　选择所有的name属性以'news'开头的input元素

$("input[name$='news']")　选择所有的name属性以'news'结尾的input元素

$("input[name*='man']")　选择所有的name属性包含'news'的input元素

$("input[id][name$='man']")　可以使用多个属性进行联合选择，该选择器是得到所有的含有id属性并且那么属性以man结尾的元素

子元素过滤选择器：

$("ul li:nth-child(2)"),$("ul li:nth-child(odd)"),$("ul li:nth-child(3n + 1)")

$("div span:first-child") 返回所有的div元素的第一个子节点的数组

$("div span:last-child") 返回所有的div元素的最后一个节点的数组

$("div button:only-child") 返回所有的div中只有唯一一个子节点的所有子节点的数组

4、表单选择器

$(":input") 选择所有的表单输入元素，包括input, textarea, select 和 button

$(":text") 选择所有的text input元素

$(":password") 选择所有的password input元素

$(":radio") 选择所有的radio input元素

$(":checkbox") 选择所有的checkbox input元素

$(":submit") 选择所有的submit input元素

$(":image") 选择所有的image input元素

$(":reset") 选择所有的reset input元素

$(":button") 选择所有的button input元素

$(":file") 选择所有的file input元素

$(":hidden") 选择所有类型为hidden的input元素或表单的隐藏域

表单元素过滤选择器：

$(":enabled") 选择所有的可操作的表单元素

$(":disabled") 选择所有的不可操作的表单元素

$(":checked") 选择所有的被checked的表单元素

$("select option:selected") 选择所有的select 的子元素中被selected的元素