jquery主要还是解决浏览器兼容性和简化dom操作。

JQuery是2006年8月诞生的一个开源项目，现在Jquery团队主要包括核心库、UI和插件开发等开发人员以及推广和网站设计人员。团队中3个核心成员是：Jhon Resig, Brandom Aaron和Jorn Zaeffefer. Jquery凭借简洁的语法和跨平台的兼容性，极大地简化了Javascript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax的操作。其独特而又优雅的代码风格改变了javascript程序员的设计思路和编写程序的方式。总之，无论是网页设计师、后台开发人员、业余爱好者还是项目管理者，都具有足够的理由去学习JQuery.

1. 选择器（找对象）

jQ对象都是一个**伪**数组HTML Collection，即使选择到只有一个元素，也是一个length为1的数组。jQ对象所使用到的所有数组的方法（push/each/map）都是经过封装的。

只要对象里面有length属性，且属性值大于或等于0，这个对象就叫伪数组。伪数组可以调用push方法，但是有一种情况例外，就是length不能被修改的伪数组，比如document.getElementsByClassName得到的伪数组的length是固定的，就不能调用push。

所有伪数组都有默认属性length，但是这个length有的是固定不变的，有的可以修改。如果用 arr[i]=... 这种赋值的方式，length不会自动改变，要手动地length++。但是如果用 [].push.apply(arr,[ ... ]) 的方法，length就会自动地时时更新。

var push = [].push;

push = {

apply: function (arr1, arr2){

for(var i = 0;i<arr2.length;i++){

arr1[arr1.length**++**] = arr2[i];

}

}

};

1. $(“”).each(function(){ }); 遍历jQ对象，this指向不同的**DOM元素**。
2. $.each(object, [callback]); 遍历任何对象。

$.each(arr, function ( i, **v** ) { //ES5中都是v,i

console.log( this );

console.log( i + ', ' + v );

});

1. map()遍历，返回回调函数的返回值

jQuery.map(arr|obj,callback)

$.map(o, function ( **v**, k ) {

console.log( k + ', ' + v );

return 'a';

});

1. 操作（方法）

|  |  |
| --- | --- |
| 样式操作 .css() | 节点操作  .append() .prepend() .after() .clone()  .html() .empty() .remove() |
| 类操作 .add/remove/  has/toggleClass() | 属性操作：  .attr() .val() .width()  .prop() .text() .scrollLeft()  .removeAttr() .offset() .scrollTop() |

JS是汽车，jQuery是带辅助驾驶系统的汽车

### jQuery的入口函数：

|  |  |  |
| --- | --- | --- |
| $(document).ready(function( ){}); | $(function( ){}); |  |

### DOM对象转换成jQuery对象：$(DOMElement)

jQuery对象转换成DOM对象：$(selector)[0]、$(selector).get(0)

### 过滤选择器：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| :odd | :even | :eq | :checkbox | :checked |
|  |  |  | 匹配所有复选框 | 匹配所有被选中的元素，不包括select中的option(option:selected) |

### 方法：

|  |  |  |
| --- | --- | --- |
| 符号(名称) | 说明 | 用法 |
| find(selector) | 查找指定元素的所有后代元素 | $(“#j\_wrap”).find(“li”).css(“color”, “red”);  选择id为j\_wrap的所有后代元素li |
| next() | 紧邻的下一个元素 |  |
| children() | 查找指定元素的直接子元素（亲儿子） | $(“#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”); |
| parent() | 查找父元素（亲的） | $(“#j\_liItem”).parent(“ul”).css(“color”, “red”); |
| eq(index) | 查找指定元素的第index个元素 | $(“li”).eq(2).css(“color”, “red”);  选择所有li元素中的第二个 |
| index() | 获取指定元素的索引号（从0开始的） | var index = $(this).index(); 不一定是在sibling中的索引，也可以是用$选定的伪数组中的索引 |
| click() |  | 不能给动态创建的元素绑定事件 |
| .each() |  | $(“”).each(callback) |
| .each( , ) | 此方法可用于例遍任何对象 | $.each(object, [callback])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，第二个为对应变量或内容。如果需要退出 each循环可使回调函数返回 false，其它返回值将被忽略。 |
| .hover(，) | 模仿悬停事件 | 当鼠标移动到一个匹配的元素上面时，会触发指定的第一个函数。  当鼠标移出这个元素时，会触发指定的第二个函数。 |
| is(selector) | is(“:visible”) | is() 根据选择器、元素或 jQuery 对象来检测匹配元素集合，如果这些元素中至少有一个元素匹配给定的参数，则返回 true。 |

### 样式属性操作

1. 设置样式属性

：$(selector).css({“color”: “red”, “font-size”: “30px”});

var cssStys = {

"background": "pink",

"fontSize": "30px",

"color": "green"  
};  
$("li").css(cssStys);

1. 获取样式属性

var col = $("li").css("color");

### 类操作（增删改查）

1. 添加类样式

$(selector).addClass(“liItem”);

1. 移除类样式

$(selector).removeClass(“liItem”);

//不指定参数的话就移除所有类名

1. 判断有没有类样式

$(selector).hasClass(“liItem”); 返回true/false

1. 切换类样式

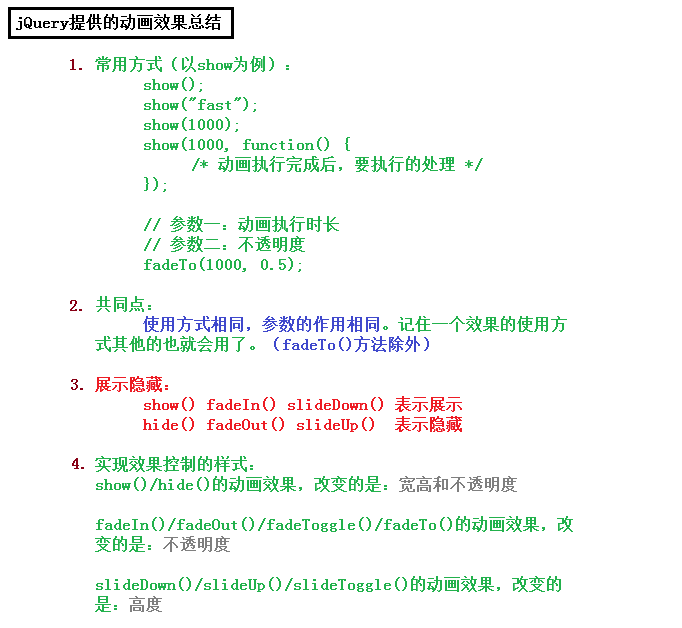
$(selector).toggleClass(“liItem”); 有则移除无则添加

### 动画

1. show( )、hide( )

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | | | |
| 数值类型 | 字符串类型 | 两个参数  $(selector).show (200, function() {}); | 无参数 |
| 毫秒 | 预设值：slow、normal、fast |  |
| 执行时长 | 600ms/400ms/200ms | 无动画 |

1. slideDown()、slideUp( )、slideToggle下拉动画
2. fadeIn( )、fadeOut( )、fadeToggle、fadeTo( 执行时间（必需），要达到的opacity值)



宽度、高度、不透明度这三个CSS属性的共性是：属性值只有一个，并且这个值是数值（去掉单位后）。

1. 自定义动画（所有能够执行动画的属性必须只有一个数字类型的值）

[http://www.w3school.com.cn/jquery/effect\_anima](http://www.w3school.com.cn/jquery/effect_animate.asp)[te.asp](http://www.w3school.com.cn/jquery/effect_animate.asp)

作用：执行一组CSS属性的自定义动画

$(selector).animate({attributes},speed,callback);

1. 停止动画

$(selector).stop(clearQueue,jumpToEnd);

## jQuery节点操作

### 动态创建元素

// $()函数的另外一个作用：动态创建元素

var $spanNode = $(“<span>我是一个span元素</span>”);

### 添加元素

在元素的最后一个子元素后面追加元素：

**append()**（**重点**）

作用：在被选元素内部的最后一个子元素（或内容）后面插入内容（页面中存在或者创建出来的元素都可以）

如果是页面中存在的元素，那么调用append()后，会把这个元素从原先的位置移除，然后再插入到新的位置。

如果是给多个目标追加一个元素，那么append()方法的内部会复制多份这个元素，然后追加到多个目标里面去。（最好不要这么做）

常用参数：htmlString 或者 jQuery对象

// 在$(selector)中追加$node

$(selector).append($node);

// 在$(selector)中追加div元素，参数为htmlString

$(selector).append('<div></div>');

（了解）不常用操作：（用法跟append()方法基本一致）

1 appendTo()

作用：同append()，区别是：把$(selector)追加到node中去

$(selector).appendTo(node);

2 prepend()

作用：在元素的第一个子元素前面追加内容或节点

$(selector).prepend(node);

3 after()

作用：在被选元素之后，作为兄弟元素插入内容或节点

$(selector).after(content);

4 before()

作用：在被选元素之前，作为兄弟元素插入内容或节点

$(selector).before(content);

### html创建元素（推荐使用，重点）

作用：设置或返回所选元素的html内容（包括 HTML 标记）

设置内容的时候，如果是html标记，会动态创建元素，此时作用跟js里面的 innerHTML属性相同

// 动态创建元素

$(selector).html(‘<span>传智播客</span>’);

// 获取html内容

$(selector).html();

### 清空元素

// 清空指定元素的所有子元素（光杆司令）

// 没有参数

$(selector).empty();

$(selector).html(“”);

// “自杀” 把自己（包括所有内部元素）从文档中删除掉

$(selector).remove();

### 复制元素

作用：复制匹配的元素

// 复制$(selector)所匹配到的元素

// 返回值为复制的新元素

$(selector).clone();

总结：

推荐使用html(“<span></span>”)方法来创建元素或者html(“”)清空元素

## 操作form表单

### 属性操作

设置属性：

// 第一个参数表示：要设置的属性名称

// 第二个参数表示：改属性名称对应的值

$(selector).attr(“title”, “传智播客”);

获取属性：

// 参数为：要获取的属性的名称，改操作会返回指定属性对应的值

$(selector).attr(“title”);

此时，返回指定属性的值

移除属性：

// 参数为：要移除的属性的名称

$(selector).removeAttr(“title”);

注意：checked、selected、disabled要使用.prop()方法。

disabled属性写在行内的时候不论是true还是false都会禁用，要想启用，要用$(selector).removeProp(“disabled”);

prop方法通常用来影响DOM元素的**动态**状态，而不是改变的HTML属性。例如：input和button的disabled特性，以及checkbox的checked特性，option的selected。

细节参考：<http://api.jquery.com/prop/>

select点击可以选择不同的option，但是select没有点击事件，要用change事件。只要点击不同的option就会触发该事件。但是如果是用其他的按钮来改变了option，就不会触发。

### 值和内容

val()方法：

作用：设置或返回表单元素的值，例如：input, select, textarea的值

// 获取匹配元素的值，只匹配第一个元素

$(selector).val();

// 设置所有匹配到的元素的值

$(selector).val(“具体值”);

text() 方法:

作用：设置或获取匹配元素的文本内容

//获取操作不带参数（注意：这时候会把所有匹配到的元素内容拼接为一个字符串，不同于其他获取操作！）

$(selector).text();

//设置操作带参数，参数表示要设置的文本内容

$(selector).text(“我是内容”);

### **jquery 表单验证插件**

## 尺寸位置操作

### 高度和宽度操作

高度操作height() ：

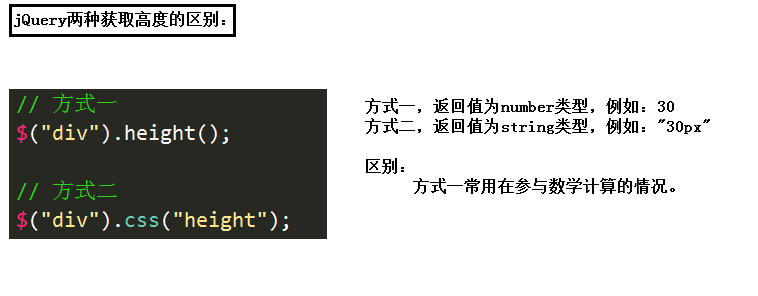
作用：设置或获取匹配元素的高度值

//带参数表示设置高度

$(selector).height(200);

//不带参数获取高度: $(selector).height();

css()获取高度和height获取高度的区别？



宽度操作

|  |  |
| --- | --- |
| width() | 设置或获取匹配元素的宽度值 |
| innerWidth() | 获取第一个匹配元素内部区域宽度（包括padding不包括border） |
| outerWidth() | 获取第一个匹配元素外部宽度（包括padding和border）。设置为 true 时，margin也计算在内。 |

### 坐标值操作

1. **offset()** Get the current coordinates of the first element, or set the coordinates of every element, in the set of matched elements, relative to the document. When positioning a new element on top of an existing one for global manipulation (in particular, for implementing drag-and-drop), .offset() is more useful.

offset() returns an object containing the properties top and left.

作用：获取或设置元素相对于文档左上角的位置

// 无参数表示获取，返回值为：{left:num, top:num}，值是相对于document的位置

$(selector).offset();

// 有参数表示设置，参数推荐使用数值类型

$(selector).offset({left:100, top: 150});

注意点：设置offset后，如果元素没有定位(默认值：static)，则被修改为relative

区别：position()获取匹配元素相对父元素的偏移。返回top/left。使用该方法时事实上是把该元素当绝对定位来处理，获取的是该元素相对于最近的一个拥有绝对或相对定位的父元素的偏移位置。offset()不管该元素如何定位，也不管其父元素如何定位，获取的都是该元素相对于当前视口的偏移。当执行拖拽操作时用这个方法最合适。

js中的offset属性指当前元素到它的offsetParent之间的距离。

1. scrollTop()

作用：获取或者设置元素垂直方向滚动的位置

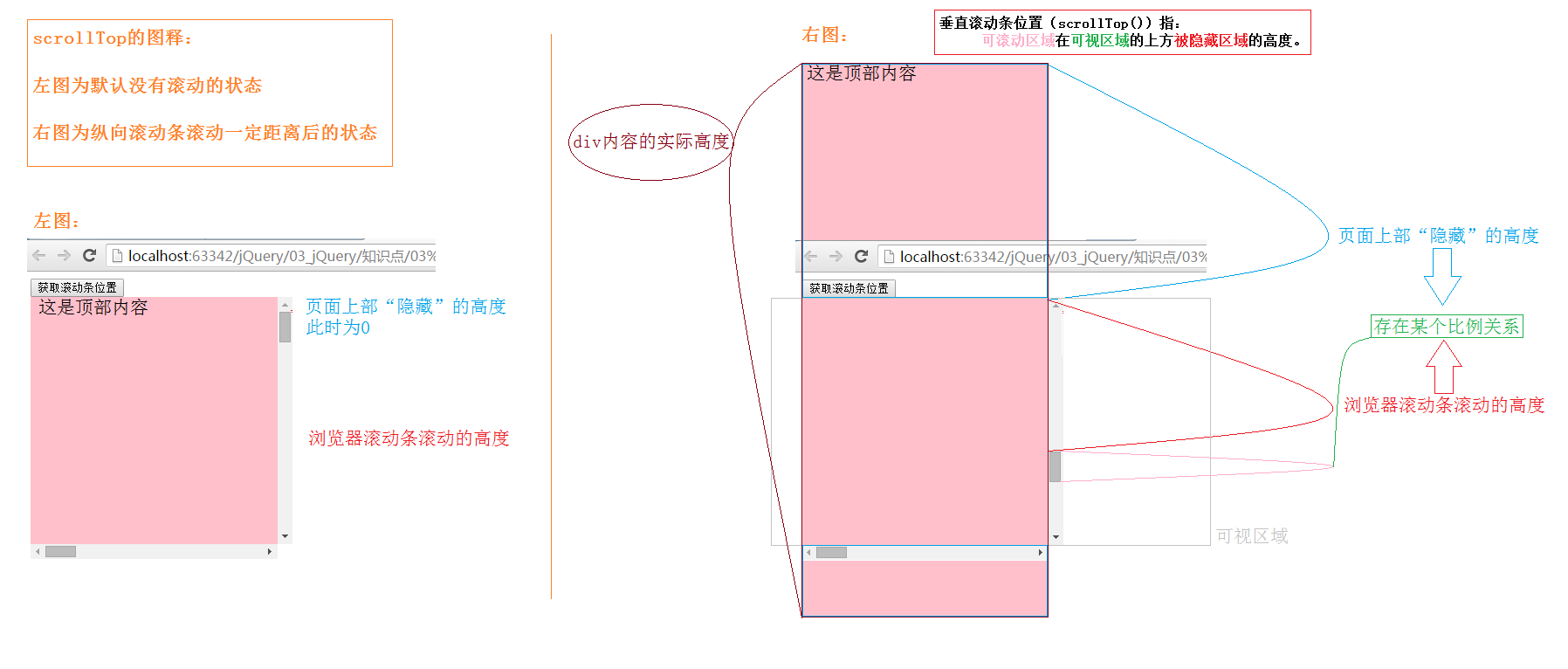
// 无参数表示获取偏移:$(selector).scrollTop();

// 有参数表示设置偏移，参数为数值类型:$(selector).scrollTop(100);

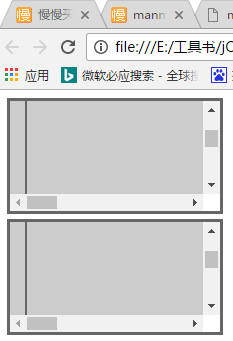
scrollLeft()

作用：获取或者设置元素水平方向滚动的位置

$(selector).scrollLeft(100);

对scrollTop的理解：

**垂直滚动条位置是可滚动区域在可视区域上方的被隐藏区域的高度。**

**如果滚动条在最上方没有滚动或者当前元素没有出现滚动条，那么这个距离为0。**

也就是说如果整个html过长而出现滚动条，那么这个滚动条属于window，设置的话用 $(window).scrollTop(200),如果某个元素的内容过多出现滚动条的话，这个滚动条就属于该元素，必须先取到该元素才能设置它的scrollTop。

|  |  |  |
| --- | --- | --- |
| **jQuery重点内容** | | |
| **名称** | 说明 | 用法 |
| **css()** | 设置或获取样式属性 | 设置：$( selector).css(“color”, “red”);  获取：$( selector).css(“color”); |
| **addClass()** | 设置样式 | $( selector).addClass(“liItem”); |
| **removeClass()** | 移除样式 | $( selector).removeClass(“liItem”); |
| **show()** | 显示 | $( selector).show(1000);  $( selector).show(1000, function() {}); |
| **stop()** | 停止动画 | $(selector).stop(); |
| **html()** | 设置或获取html内容  不传参数是获取，传参数是设置，但是不能写成赋值的形式 | 设置：（动态创建元素）$(“div”).html(“”);  $(“div”).html(“<p>我是一个p元素</p>”);  获取：$(“div”).html(); |
| **empty()** | 清空节点 |  |
| **text()** | 获取或者设置文本内容 | 获取：.text()  设置：.text(“这是文本内容”); |
| **append()** | 追加元素 | 为div元素追加子元素：  $(“div”).append(“<p>传智播客</p>”); |
| **val()** | 获取或者设置表单元素的值 | 获取：.val()  设置：.val(“这是表单的值”); |
| **attr()** | 获取或者设置元素的属性 | 用法同：css()这个方法  获取：attr(“href”)  设置：attr(“href”, “http://web.itcast.cn”);  类似js中的getAttribute() 和 setAttribute() |
| **prop()** | prop通常用来影响DOM元素的动态状态，而不改变HTML属性。 | checked、selected、disabled要使用.prop() |

### attr()与css()比较

css()只能设置style里面的样式，比如宽、高、背景

attr()设置的是在DOM树中和style平级的样式，比如class, href，但是改变不了控制台里面对象的数据

### attr( )与prop( )比较（正选反选）

attr( )设置的是HTML的行内样式，在页面上的checkbox里面√不会出现，打印对象的时候，checked的值是不会改变的。

prop( )不改变HTML属性，但会影响DOM元素的动态状态，所以页面上的checkbox里面√会时隐时现，打印对象的时候，checked的值true/false会交替出现。

:checked 匹配所有选中的被选中元素(复选框、单选框等，也包括select中的option，即使option是selected)

## jQuery事件机制

指JQ对JS操作DOM事件的封装，包括：事件绑定、事件解绑、事件触发。

### 事件绑定

发展历程：简单事件绑定 → bind事件绑定 → delegate 事件绑定 → **on**

1. 简单事件绑定：

click() dbclick() blur() mouseenter() mouseleave() change() focus() keydown()

1. bind：不能给新创建出来的元素绑定事件（不推荐）
2. delegate：给动态生成的元素绑定事件的时候必须用代理，将事件绑定在父元素上，子元素只负责触发事件。
3. **on**（最现代的方式，兼容zepto，Jquery1.7版本之后，用on统一了所有事件处理方法，强烈推荐）

Jquery内部所有的事件都是用on处理的，包括click()也是用on，即使使用click()方法，内部也是调用了on().

JavaScript：btn.onclick = function() {}; 这种绑定事件的方式后面的函数会覆盖前面的，如果不想被覆盖，就要添加事件监听器。

jQuery：$btn.click(function( ) {}); 最好用$( ).on.(“click”, function( ){ }); 这两种绑定事件的方式是叠加式，而不是覆盖式。

绑定：$(selector).on(events,[selector],[data],handler);

第一个参数表示绑定事件的类型，第二个参数表示触发事件的元素，第三个表示事件处理函数。

// 解绑匹配元素的所有click事件

$(selector).off(“click”);

// 解绑所有代理的click事件，元素本身的事件不会被解绑

$(selector).off( “click”, “\*\*” );

$(“button”).click(function(){

$(“input”).trigger(“select”);

//点击按钮触发input的select事件

});

$(selector).triggerHandler(“focus”);

triggerHandler与trigger类似，但是有以下三个区别：

1. 不触发浏览器的默认行为
2. 不会产生事件冒泡
3. 只触发jQuery对象集合中第一个元素的事件处理函数
4. 这个方法返回的是事件处理函数的返回值，而不是具有可链性的jQuery对象。此外，如果最开始的jQuery对象集合为空，则这个方法返回undefined.

|  |  |  |  |
| --- | --- | --- | --- |
| jQuery事件对象介绍 | | | |
| event.data | 传递给事件处理程序的额外数据 | event.pageX | 鼠标相对于页面左边的位置 |
| event.currentTarget | 等同于this | event.stopPropagation()； | 阻止事件冒泡 |
| event.target | 触发事件源，子元素也可以触发父元素的事件，若阻止冒泡就是this | event.preventDefault(); | 阻止默认行为 |
| event.keyCode | 键盘按键代码 | return false; | 同时阻止冒泡和默认行为 |
| event.type | 事件类型：click，dbclick… | event.which | 鼠标的按键类型：左1 中2 右3 |

[select中指定option选中触发事件](http://blog.sina.com.cn/s/blog_5dc351100101l53p.html" \t "http://cn.bing.com/_blank)不能用Onclick，要用onchange.

function func(){

//如何知道被选中的option标签是哪个？

var vs = $('select option:selected').val();

}