关于 jQuery 的学习

作者：zuimengguchen
原文链接：https://ld246.com/article/1569070321243
来源网站：链滴

jQuery

刚开始接触jQuery时候我也在好奇，他是个什么东西，接触之后发现这个是个‘魔鬼’，用原生js代很多行才能实现的功能，jQuery几行就搞定了。卓然让我大吃一惊。！！！

关于jQuery

1. 众所周知jQeruy是一个库，关于js的库。最简单的理解就是jQuery就是一个js文件，只不过里面放许许多多别人写好的js原生代码，存放在里面。我们可以直接拿来使用，从而方便快捷的进行开发。juery的诞生初衷就是“write Less，Do More”，写更少的代码，做更多的事情。

2. 优点

轻量级。核心文件才几十kb，不会影响页面加载速度

链式编程（代码可以像一条锁链一样环环相扣的向后延伸）、隐式迭代（jQuery对象都是采用伪数组行存储，获取的话需要进行循环遍历，jQuery会默默进行循环遍历而不用你书写，直接用）（两大特）

对事跨浏览器兼容。基本兼容了现在主流的浏览器

样式、动画支持，大大简化了DOM操作

支持插件扩展开发。有着丰富的第三方的插件，例如：树形菜单、日期控件、轮播图等

免费、开源（免费很重要，开源就是指自己如果有想法可以进行添加新的功能）

jQuery的简单使用

1. jQuery入口函数

$为jQuery的顶级对象(我有钱我最大)。

在原生js中入口函数书写只有一种通过window对象调用。但是在jQuery入口函数有两种分别为:

$(function(){})//第一种
$(document).ready(function(){})//第二种方法

而且jQuery的入口函数和原生不太一样，原生入口函数会在页面元素都加在完成之后调用函数，但是jquery的入口函数会在 Dom结构渲染结束之后既可以执行内部代码。不必等到所有资源加载完成

2. jQuery对象和DOM对象

用原生 JS 获取来的对象就是 DOM 对象【document.getElement等方法】

jQuery 方法获取的元素就是 jQuery 对象【$('div')等】

jQuery 对象本质是： 利用$对DOM 对象包装后产生的对象（伪数组形式存储）

(ps:只有jQuery对象才能使用jQuery方法，DOM对象则只能使用原生的javaScript方法，不能混淆但两者对象可以相互转换)

3. jQuery对象和DOM对象相互转换

jQuery对象转换为DOM对象

原理因为jQuery获取对象之后都是以伪数组形式存储起来的所以可以通过伪数组方法来调用对象

第一种:
$(''div)[index];//index为索引号
第二种:
$('div') .get(index) //~~~~index 是索引号

jQuery 常用的Api

jQuery 选择器

$(“选择器”) // 里面选择器直接写 CSS 选择器即可，但是要加引号

$('#id')==》指定id元素

$('*')==》所有元素

$('.class')==》指定class元素

$('div')==》根据标签获取元素

$('div,p,li')==》获取多个

$('li.class')==>交集获取

$('ul>li')==>子代

$('ul li')==>后代

$('li').parent()父级

$('ul').children('li');子集【如果不加参数，获取所有的，如果添加指定的元素，按照指定的找】
筛选方法（重点）

$('ul').find('li')后代

$('li').siblings('li')兄弟

$('li'),nextAll();后面的

$('li').prevAll();前面的

//判断是否具有某个类名：$('div').hasClass('aaa')

$('div').eq(index);//指定索引方法【eq推荐用方法】

jQuery 样式操作

jQuery可以使用css方法来修改简单元素样式；也可以操作类，修改多个样式只不过和原生js方法不一样

操作css方法(三种)

```
//参数只写属性名，则返回属性值
$(this).css(''color'');
//参数是属性名，属性值，逗号分隔，是设置一组样式，属性必须加引号，值如果是数字可以不用跟位和引号
$(this).css(''color'', ''red'');
//参数可以是对象形式，方便设置多组样式。属性名和属性值用冒号隔开， 属性可以不用加引号，
$(this).css({ "color":"white","font-size":"20px"});
```

设置类样式方法同样也是三种

```
//添加类
$(“div”).addClass(''current'');

//移除类
$(“div”).removeClass(''current'');

//切换类:是指若元素没有这个类则添加上去，若有这个类，则移除这个类
$(“div”).toggleClass(''current'');
```

jQuery 效果

显示隐藏效果

```
show([speed],[easing],[fn])//显示出来
hide([speed],[easing],[fn])//元素隐藏
toggle([speed],[easing],[fn])//在显示和隐藏之间切换
//参数说明
/*1.所有参数都可以省略，只不过无动画直接显示
2.speed：三种设定的速度之一（“slow” “normal” “fast”或者表示动画时长的毫秒数值）
3.easing：指的是切换效果默认是swing，可以更改为linear
4.fn：回调函数，在动画执行完后调用的函数，每个元素执行一次
*/
```

滑动效果

```
slideDown([speed],[easing],[fn])//显示出来slideUp([speed],[easing],[fn])//元素隐藏
slideToggle([speed],[easing],[fn])//在显示和隐藏
之间切换
//参数说明
/*1.所有参数都可以省略，只不过无动画直接显示
2.speed：三种设定的速度之一（“slow” “normal” “fast”或者表示动画时长的毫秒数值）
3.easing：指的是切换效果默认是swing，可以更改为linear
4.fn：回调函数，在动画执行完后调用的函数，每个元素执行一次
*/
```

事件切换

hover（[over],[out]）

（1）over：鼠标移到元素上要触发的函数（等价于mouseenter）

（2）out：鼠标移出元素要触发的函数（相当于mouseleave）

(3)如果只写一个函数，则鼠标经过和离开都会触发它

stop()动画停止函数

介绍之前首先要介绍一个概念，动画排队。如果一个动画被触发但是没有结束之前又触发了一个动画件就会造成多个动画或者效果排队执行。此时会造成效果紊乱

(1）stop() 方法用于停止动画或效果。

(2）注意： stop() 写到动画或者效果的前面， 相当于停止结束上一次的动画

淡入淡出效果

```
fadeIn([speed],[easing],[fn])//淡入fadeOut([speed],[easing],[fn])//淡淡出
fadeToggle([speed],[easing],[fn])//在淡入和淡出之间切换
fadeTo([speed],[easing],[fn])//透明到一定程度
//参数说明
/*1.所有参数都可以省略，只不过无动画直接显示
2.speed：三种设定的速度之一（“slow” “normal” “fast”或者表示动画时长的毫秒数值）
3.easing：指的是切换效果默认是swing，可以更改为linear
4.fn：回调函数，在动画执行完后调用的函数，每个元素执行一次
*/
```

自定义动画animate

```
/*使用语法为*/
animate（parmes，[speed],[easing],[fn]);
/*
参数说明：
（1）params：想要更改的样式，以对象形式传递，必须写。属性名可以不用带引号，如果复合属性需要采取驼峰的命名。比如fontSize。其余参数都可以省略
（2）speed：三种预定速度之一的字符串(“slow”，“normal”，or “fast”)或表示动画时长的毫数值(如：1000)。
```

（3）easing：(Optional) 用来指定切换效果，默认是“swing”，可用参数“linear”。
（4）fn：回调函数，在动画完成时执行的函数，每个元素执行一次。
*/

jQuery 属性操作

1. 获取元素固有属性(固有属性就是元素本身自带的属性)prop()

```
$( ‘div’ ).prop('属性名',属性值);
//只写属性名可以获取到属性值，两个都写是设置该属性属性值
```

2. 设置或者获取自定义属性值(自定义属性就是用户自己给元素添加的属性)attr()

```
//获取属性语法

$('div').attr(''属性'') // 类似原生 getAttribute()

//设置属性语法

$( " div" )attr(''属性'', ''属性值'') //类似原生 setAttribute()
```

jQuery 文本属性值

设置普通元素内容html()相当于原生js中的innerHTML()

```
//获取内容
$("div").html();
//设置内容
$("div").html("内容");
```

设置普通元素文本内容text()相当于原生js中的innerText()

```
//获取内容
$("div").text();
//设置内容
$("div").text("内容");
```

获取表单的值val()

```
//获取表单的值
$("input").val();
//设置表单的值
$("input").val("内容");
```

总结：在设置或者获取元素内容时99使用html表单除外，获取表单直接使用val();

jQuery 元素操作

元素操作主要是对元素进行遍历，创建添加，和删除等操作。

遍历元素

虽然在jQuery中的隐式迭代，方便我们对同一类元素进行同样的操作。但是如果要给同一类元素添加同操作，就需要用到了遍历。活不多说直接上代码

```
//语法1
$("div").each(function(index, domEle) { })
/*1. each() 方法遍历匹配的每一个元素。主要用DOM处理。 each 每一个

2. 里面的回调函数有2个参数：  index 是每个元素的索引号；demEle 是每个DOM元素对象，不是jquery对象

3. 所以要想使用jquery方法，需要给这个dom元素转换为jquery对象  $(domEle)
*/
//语法2
$.each(object，function(index, element){ })
/*
1. $.each()方法可用于遍历任何对象。主要用于数据处理，比如数组，对象

2. 里面的函数有2个参数：  index 是每个元素的索引号；element  遍历内容
*/
```

创建元素

语法：$("   ");

######添加元素

```
$(element).appemd("内容")
//把内容放入匹配的元素最后面，功能和原生appendChild相似
$(element).perpeng("内容")
//把内容添加到元素的最前面
```

外部添加

```
$(element).after("内容")
//把内容放入目标元素后面
$(element).before("内容")
//把内容放入目标元素前面
```

区别：内部添加的元素，和目标元素时父子关系

外部添加的元素，和目标元素是兄弟关系

删除元素

```
element.remove()//删除匹配元素本身
element.empty()//删除匹配元素集合中的所有子节点
element.html("")//清空匹配元素的内容
```

jQuery 尺寸、位置操作

最主要的位置有三个：offset(),position(),scrollTop()/scrollLeft();

首先介绍下offset()方法用来设置获取元素偏移

1. offset()方法设置或返回被选元素相对于文档的偏移坐标跟父级元素没有关系
2. 该方法有两个属性left和top。因为offset()获取到的属性都是以对象格式储存的所以可以通过offset()top或者offset().left来获取具体的数值
3. 可以这是元素的偏移量offset({top:10;left:30});

position 获取元素偏移

1. position()方法用于返回被选元素相对于带有定位父级偏移坐标，如果父级都没有定位，则以文档准
2. 该方法有两个属性left，top position().top用于获取距离定位父级顶部的距离position()left用于获距离父级左侧的距离

特别注意：此属性只能获取

** scrollTop、scrollLeft()**设置或者获取被卷去的头部和左侧

1. scrollTop()方法设置或者返回被选元素被卷去的头部
2. 不跟参数是获取，参数为不带单位的数字则是设置被卷曲的头部

配合scroll事件(滚动事件使用,谁有滚动条就加给谁)

jQuery事件

jQuery事件虽然和原生js有些区别但是大致上相同。

注册事件

```
$("div").click(function(){
事件处理代码
})//和原生相比不需要再加上on
```

使用on()绑定事件

```
语法如下：
element.on("事件类型"，fn（）{事件处理程序代码})
/*
on()方法在jQuery中存在着三大优势
1.可以同时绑定多个事件的处理方法
$("div").on({
存储方式是以键值对方式存储停顿加，不要加;特别注意符号
click:function(){},
mouseover:function(){},
mouseout:function(){}
})
2. 可以进行事件委托
$("ul").on("click","li",function(){})
虽然以前还有bind(),live(),delegate()方法来处理事件绑定或者事件委派但是随着版本的迭代现在都使用on()
```

3. 可以给动态添加的元素绑定上事件
*/

off()解绑事件
off()方法可以移除通过on()绑定的事件处理程序
$("p").off()//解绑p元素所有事件处理程序
$("p").off("click")//解绑p元素上面的点击事件
$("ul").off("click","li");//解绑事件委托
ps如果有的事件只想触发一次那就可以使用one()来绑定事件

自动触发事件trigger()

有些事件希望自动触发, 比如轮播图自动播放功能跟点击右侧按钮一致。可以利用定时器自动触发右按钮点击事件，不必鼠标点击触发

element.click() // 第一种简写形式

element.trigger("type")//第二种自动触发模式

```
$("p").on("click", function () {

 alert("hi~");

});

$("p").trigger("click"); // 此时自动触发点击事件，不需要鼠标点击
```

element.triggerHandler(type) // 第三种自动触发模式

triggerHandler模式不会触发元素的默认行为，这是和前面两种的区别。
以上就是本人近期学习jQuery的心得，其实如果js原生学习不错用起来也是可行的。jQuery就像是糖。比起来原生更加好用而已。