# JavaScript进阶

第3天课堂笔记（本课程共6天）

# 目录

[JavaScript进阶](#_Toc13613)

[目录](#_Toc29698)

[一、作业讲解](#_Toc10947)

[二、滚轮事件](#_Toc3439)

[三、滚滚屏原理](#_Toc28304)

[四、jQuery实现](#_Toc5617)

[4.1 拖拽](#_Toc19697)

[4.2 各种尺寸](#_Toc14394)

[4.3 净位置（到页面顶端的位置）](#_Toc15987)

[4.4 鼠标滚轮事件](#_Toc10430)

[4.5 窗口相关](#_Toc13164)

[4.6 缓冲](#_Toc13667)

# 一、作业讲解

拖拽模型，脑子中一下子要想到模型：

|  |
| --- |
| 1. op.onmousedown = function(event){ 2. var deltaX = event.clientX - op.offsetLeft; 3. var deltaY = event.clientY - op.offsetTop; 4. document.onmousemove = function(event){ 5. var x = event.clientX - deltaX; 6. var y = event.clientY - deltaY; 7. if(x < 0){ 8. x= 0; 9. }else if(x > odiv.clientWidth - op.clientWidth){ 10. x = odiv.clientWidth - op.clientWidth 11. } 12. op.style.left = x + "px"; 13. } 14. document.onmouseup = function(){ 15. document.onmousemove = null; 16. } 17. } |

如果想要得到鼠标在当前这个盒子中的位置：

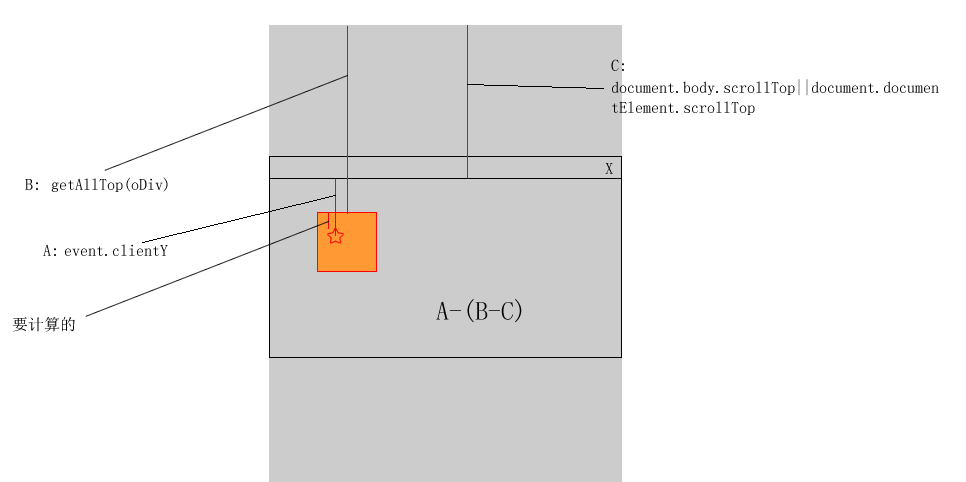
|  |
| --- |
| 1. event.offsetX; 2. event.offsetY; |

挺好用的，新版本火狐嗷嗷好用。如果为了怕一些版本浏览器不兼容，或者为了防止冒泡干扰：

周转：

|  |
| --- |
| 1. event.clientY - (getAllTop(oDiv) - scrollTop) |

原理图：



有一个小函数[isNaN](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/isNaN)，判断这个东西是不是NaN，

|  |
| --- |
| 1. isNaN() |

实验证明：

|  |
| --- |
| 1. isNaN(NaN) //true 2. isNaN(“NaN”) //true 3. isNaN(undefined) //true |

小算法：

|  |
| --- |
| 1. 给你一个数组，让你寻找里面第一个偶数，弹出来；如果没有偶数，弹框警报全是奇数。 2. var arr = [3,3,3,3,3,3,3,11]; 3. for(var i = 0 ; i < arr.length ; i++){ 4. if(arr[i] % 2 == 0){ 5. alert(arr[i]); 6. break; 7. } 8. } 9. if(i == arr.length){ 10. alert("全是奇数"); 11. } |

实际上涉及浏览器的窗口、卷动事件，就叫做BOM，不过这个词儿很少有人提了。

BOM = Browser Object Model ， 浏览器对象模型

所有关于浏览器的事儿，都是BOM技术，和DOM分不清：

比如，onscroll事件，是window对象的事件，window对象就是浏览器

但是页面卷动值，是document对象的属性scrollTop，document就是文档

BOM方法：

|  |
| --- |
| 1. window.alert() 2. window.setInterval(); |

窗口更改尺寸事件

|  |
| --- |
| 1. window.onresize = function(){ 2. } |

窗口的尺寸

|  |
| --- |
| 1. document.documentElement.clientWidth 2. document.documentElement.clientHeight |

# 二、滚轮事件

[onscroll](https://developer.mozilla.org/zh-CN/docs/Web/API/GlobalEventHandlers/onscroll) [元素的所有属性（包括事件属性）](https://developer.mozilla.org/en-US/docs/Glossary/Global_attribute)

滚轮就是鼠标上的滚轮，它滚动的时候触发事件。

注意，window.onscroll这是窗口的卷动事件，不管因为什么（滚轮、滚动条、键盘上下键）导致页面卷动了，就会触发window.onscroll事件。

滚轮事件就是鼠标滚动的时候触发，不一定引起页面的卷动。

|  |
| --- |
| 1. oDiv.onmousewheel = function(event){ 2. oDiv.innerHTML = event.wheelDelta; 3. } |

onmousewheel就是鼠标滚动事件，mouse鼠标，wheel就是轮子。

event参数最最重要的事就event.wheelDelta属性，表示滚动的方向。这是浏览器的规定：

鼠标往上滚， 120

鼠标往下滚， -120



如果滚动快速，可能出现120的整数倍，正负性是一样的。

onmousewheel事件的触发，是滚轮滚动一“咯噔”触发一次。

火狐浏览器不兼容，火狐使用自己的专用事件**DomMouseWheel**，并且这个事件只能通过标准的DOM2级的事件绑定方式添加。

|  |
| --- |
| 1. oDiv.addEventListener("**DOMMouseScroll**",function(event){ 2. oDiv.innerHTML = event.detail; 3. },false); |

**火狐添加的事件，是它自己的专门事件DOMMouseScroll，表示滚动方向的事件的属性叫做event.detail。**

detail就是细节的意思。反着的，滚动方向往上，-3；滚动方向往下，3。



兼容性的解决？

当我们给Chrome或者IE添加一个DOMMouseScroll事件的时候，会静默，不会报错。

添加监听好解决，就写两个监听就行了，事件处理函数，要写判断。

|  |
| --- |
| 1. function mousewheelhandler(event){ 2. event = event || window.event; //→ 解决IE6、7、8和高级浏览器的不兼容 3. //阻止默认事件 4. if(event.preventDefault){ //→ 解决IE6、7、8和高级浏览器的不兼容 5. event.preventDefault(); 6. }else{ 7. event.returnValue = false; 8. } 9. //Chrome、IE用的是event.wheelDelta; 10. if(event.wheelDelta){ //→ 解决火狐和其他浏览器的不兼容 11. var direction = event.wheelDelta > 0 ? 1 : -1; 12. }else if(event.detail){ 13. //火狐用的是event.detail; 14. var direction = event.detail > 0 ? -1 : 1; 15. } 16. oDiv.innerHTML = direction; 17. } |

这个事件处理程序中，已经屏蔽了3个事情的不兼容。

|  |
| --- |
| 1. //给IE、CHrome添加事件 2. oDiv.onmousewheel = mousewheelhandler; 3. //给火狐添加事件 4. oDiv.addEventListener("DOMMouseScroll",mousewheelhandler,false); |

[Try..catch](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Statements/try...catch)

try{……}catch(ex){}

try表示尝试，里面的语句可能有错误，catch(ex)参数ex就是try语句段里面出现的错误。在catch里面进行了一个捕获，可以把错误控制在这个catch语句块里，而不向浏览器抛出。

|  |
| --- |
| 1. try{ 2. console.log("123"); 3. var a = b; //这行语句有错误，b没有被定义 4. console.log("哈哈"); 5. }catch(err){ 7. } 8. console.log("456"); |



程序的执行逻辑是这样的： 运行try语句块里面的语句，如果遇见了错误的语句，不向浏览器抛出错误。而是执行catch语句段里的语句。try语句里面没有执行完的语句，不执行了。catch语句块执行完毕之后，执行后面的语句。

# 三、滚滚屏原理

我们现在用原生JS讲解原理，页面上动画、尤其是动画序列、delay进场用jQuery来做更合适。

火车法：

HTML布局：

|  |
| --- |
| 1. <div class="contanier"> 2. <div class="page page0"><h1>0</h1></div> 3. <div class="page page1"><h1>1</h1></div> 4. <div class="page page2"><h1>2</h1></div> 5. <div class="page page3"><h1>3</h1></div> 6. <div class="page page4"><h1>4</h1></div> 7. </div> |

CSS从小到大都是height:100%; 撑高为浏览器屏幕高度

|  |
| --- |
| 1. html,body{ 2. width: 100%; 3. height: 100%; 4. overflow: hidden; 5. } 6. .contanier{ 7. width: 100%; 8. height: 100%; 9. position: absolute; 10. top: -300%; 11. left: 0; 12. } 13. .page{ 14. width: 100%; 15. height: 100%; 16. background: yellowgreen; 17. } |

# 四、jQuery实现

我们要把这几天学习的东西，如何用jQuery实现

## 4.1 拖拽

插件来完成，jQuery牛逼之处不仅仅在于API强大、社区强大……还有更重要的原因，插件多。

**实现拖拽要用的就是jQueryUI这个插件，这是一个官方插件**，用来实现：

Draggable ： 拖拽

Droppable ： 拖放

Resizable ：改变尺寸

Selectable ： 可选择

Sortable ：可排序

插件的意思是，我这个插件包依赖你的jQuery：

<script type="text/javascript" src="js/jquery-1.12.3.min.js"></script>

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

一条语句能实现拖拽：

|  |
| --- |
| 1. $("div").draggable(); |

重要的参数，参数都以JSON来配置

|  |
| --- |
| 1. $("div").draggable(JSON); |

文档： <http://api.jqueryui.com/>

|  |
| --- |
| 1. $("p").draggable({ 2. axis: "x", //约束在某个轴上 3. containment: "parent", //在父盒子中拖拽 4. grid : [100], //步长 5. drag : function(event, ui){ //事件，值就是ui.position.top 6. console.log(ui.position.left,ui.position.top); 7. } 8. }); |

jQueryUI还能做什么？改变尺寸、可选择等等。

## 4.2 各种尺寸

jQuery中的尺寸：

|  |
| --- |
| 1. $("div").width(); //计算后的内容宽度 2. $("div").innerWidth(); //计算后的算上padding的宽度 3. $("div").outerWidth(); //算上padding和border 4. $("div").outerWidth(true);//算上padding和border和margin |

## 4.3 净位置（到页面顶端的位置）

就是我们昨天封装的getAllTop，jQuery中也替我们封装了。叫做offset()这个方法返回一个带有top、left、bottom、right属性的对象。

|  |
| --- |
| 1. var t = $("p").offset().top; |

净位置如同我们昨天学习的getAllTop()，但是他比我们牛逼许多，十年磨一剑。

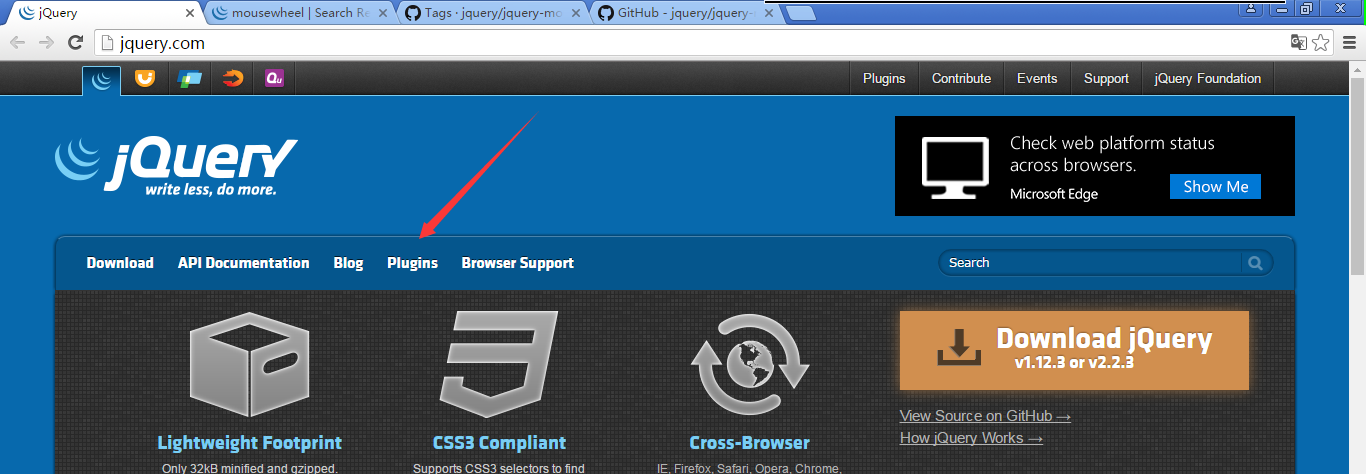
IE8的border问题都解决了。

先打点调用offset()方法，然后是top属性

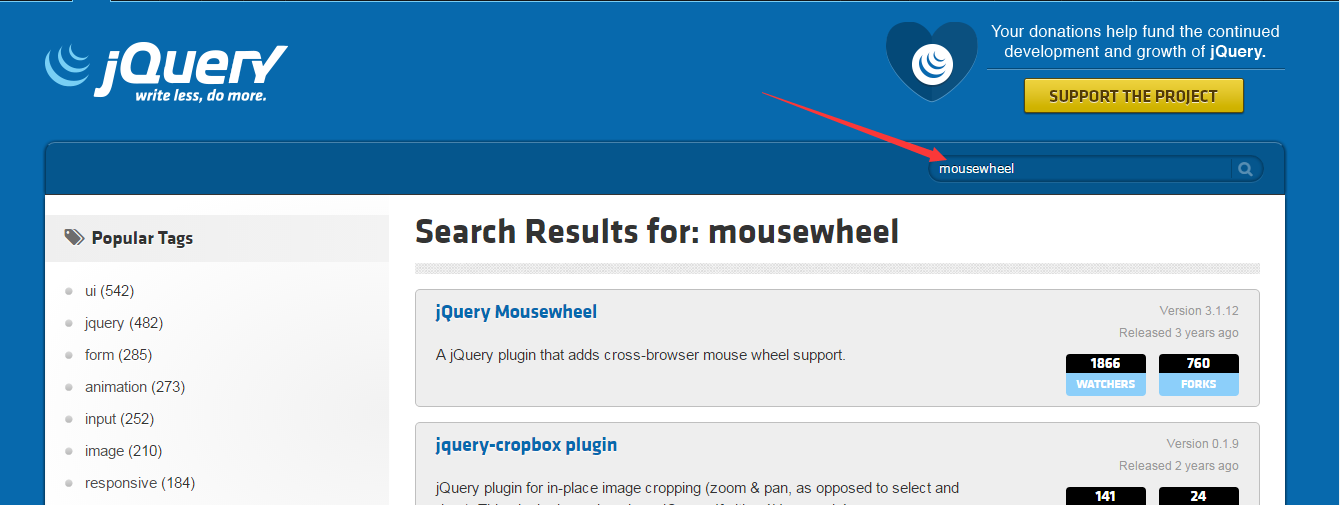
## 4.4 鼠标滚轮事件

jQuery天生没有mousewheel事件，必须依靠插件。

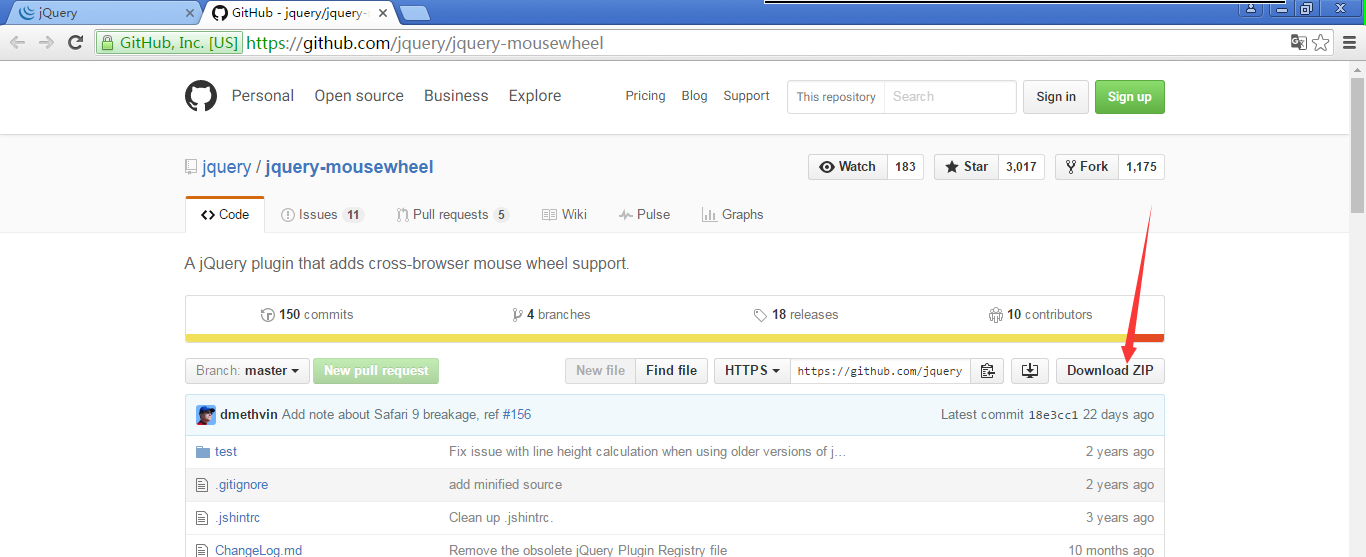
插件官网：https://github.com/jquery/jquery-mousewheel



搜索：



下载：



没有火狐的兼容问题了，方向也都是对的。event.deltaY属性，鼠标滚轮往下滚-1，往上滚1。

|  |
| --- |
| 1. $("div").mousewheel(function(event){ 2. if(event.preventDefault){ 3. event.preventDefault(); 4. }else{ 5. event.returnValue = false; 6. } 7. $("div").html(**event.deltaY**); 8. }); |

## 4.5 窗口相关

窗口的卷动事件：

|  |
| --- |
| 1. $(window).scroll(function(){ 2. console.log($(window).scrollTop()); 3. }); |

还记得么？$()里面只有三个东西不加引号：$(this)、$(window)、$(document)。

原生的：

window.onscroll = function(){

console.log(document.body.scrollTop || document.documentElement.scrollTop)

}

jQuery简单多了！

窗口卷动值：

|  |
| --- |
| 1. $(window).scrollTop(); //方法！！不是属性 |

窗口的宽度：

|  |
| --- |
| 1. $(window).width(); 2. $(window).height(); |

窗口冲锋：

|  |
| --- |
| 1. $(“html,body”).animate({ //对象变成了$(“html,body”) 2. “scrollTop” : 2000 //变成了属性 3. },1000); |

## 4.6 缓冲

就是我们说动运动公式，就是t、b、c、d

需要用插件 ， jQuery eaise.js