利用原生js来写全屏滚动

作者 :沉浮的磊子 时间 : 2014/12/02 分类：[js](http://fed.lvmama.com/category/js/" \o "查看js中的全部文章)

**2**

前段时间刚做了个垂直整屏滚动的项目，现在总结一下经验。  
起初考虑过用插件，但是仔细想想，不动头脑的用别人写的东西是没有进步的，于是决定自己写。然后想想自己的原生很烂，那就克服一下吧，试着用原生来写，抛弃jquery…..没办法，人任性起来，真可怕。。。

先看结构：

|  |  |
| --- | --- |
| 1 | <body> |
| 2 | <div class="wrap"> | |

|  |  |
| --- | --- |
| 3 | <div class="page page1"></div> |
| 4 | <div class="page page2"></div> |

|  |  |
| --- | --- |
| 5 | <div class="page page3"></div> |
| 6 | <div class="page page4"></div> |

|  |  |
| --- | --- |
| 7 | </div> |
| 8 | <ul class="menu"></ul> | |

|  |  |
| --- | --- |
| 9 | </body> |

几个核心的style

|  |  |
| --- | --- |
| 1 | body { position: relative; overflow: hidden;} |
| 2 | .wrap { position: absolute; left: 0; right: 0; top: 0;} | |

|  |  |
| --- | --- |
| 3 | .menu { position: fixed; top: 50%; right: 50px;} |

做垂直滚屏首先要明确以下几个要点，思路清楚了，做起来就不乱了。

1.需要获取浏览器可见高度;  
2.每一屏的高度等于浏览器的高度，超过在css中裁切。  
3.当窗口高度改变时，需重新获取可见高度，并赋给每屏的盒子;定位高度也要计调整。  
4.绑定鼠标滚轮事件。判断向上向下滚动  
5.滚动动画效果  
6.扩展菜单

—————————传说中的分隔符—————————————————

Exp:

**1.获取浏览器可见高度**

|  |  |
| --- | --- |
| 1 | document.documentElement.clientHeight |

这里我定义了一个方法

|  |  |  |
| --- | --- | --- |
| 1 | function changeDivHeight(){ | |
| 2 | h = document.documentElement.clientHeight; |

|  |  |
| --- | --- |
| 3 | //设置body高度 |
| 4 | body.style.height = h +"px"; | |

|  |  |
| --- | --- |
| 5 | } |

**2.每一屏的高度等于浏览器的高度，超过在css中裁切。**

|  |  |
| --- | --- |
| 1 | function changePageHeight() { |
| 2 | for (var i = 0; i < oPage.length; i++ ) { | |

|  |  |  |
| --- | --- | --- |
| 3 | oPage[i].style.height = h + "px"; | |
| 4 | } |

|  |  |
| --- | --- |
| 5 | } |

**3.当窗口高度改变时，需重新获取可见高度，并赋给每屏的盒子;定位高度也要计调整。**

|  |  |
| --- | --- |
| 1 | window.onresize = function(){ |
| 2 | //当窗口大小改变时触发 |

|  |  |  |
| --- | --- | --- |
| 3 | //需重新获取可见高度，并赋给每屏的盒子 | |
| 4 | changeDivHeight(); |

|  |  |  |
| --- | --- | --- |
| 5 | changePageHeight(); | |
| 6 |  |

|  |  |
| --- | --- |
| 7 | //窗口改变，top值也相应变化 |
| 8 | fullWrap.style.top = -(pageIndex \* h)  +"px"; | |

|  |  |
| --- | --- |
| 9 | } |

**4.绑定鼠标滚轮事件。判断向上向下滚动**

注册滚轮事件

|  |  |
| --- | --- |
| 1 | /\*注册事件\*/ |
| 2 | mouseScroll(document); | |

|  |  |
| --- | --- |
| 3 | function mouseScroll(elm){ |
| 4 | if(elm.addEventListener){ | |

|  |  |  |
| --- | --- | --- |
| 5 | elm.addEventListener('DOMMouseScroll',scrollFunc,false); | |
| 6 | }//W3C |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | window.onmousewheel=elm.onmousewheel=scrollFunc;//IE/Opera/Chrome | | |
| 8 | | }; |

事件绑定

|  |  |  |
| --- | --- | --- |
| 1 | var scrollFunc = function (e) { | |
| 2 | if(!play) { |

|  |  |  |
| --- | --- | --- |
| 3 | return false; | |
| 4 | } |

|  |  |  |
| --- | --- | --- |
| 5 | var e = e || window.event; | |
| 6 |  |

|  |  |  |
| --- | --- | --- |
| 7 | //声明变量 一会用来存滚动值 | |
| 8 | var t = null; |

|  |  |
| --- | --- |
| 9 | } |

值得一提的是 wheelDelta和detail，它们是干什么用的？  
wheelDelta支持IE/Opera/Chrome  
而讨厌的Firefox 则需要detail来获取滚动值，并且两个值还不一样  
wheelDelta向下滚动值为-120 向上为:120  
detail向下滚动值为3,向上为-3  
在这里，我们把滚动的值存在变量t里。  
这里我可给页面建立索引，声明变量pageIndex,通过滚动来改变索引

so:可以这样判断了

IE/Opera/Chrome 下 向上滚，t 为正时 则pageIndex–; 向下滚 t为负时 则pageIndex ++ ;  
firefox下 向上滚，t为负时 则pageIndex –; 向下滚 t为正时 则pageIndex ++;

|  |  |  |
| --- | --- | --- |
| 01 | if (e.wheelDelta) { | |
| 02 |  |

|  |  |
| --- | --- |
| 03 | //IE/Opera/Chrome |
| 04 | t = e.wheelDelta; |

|  |  |
| --- | --- |
| 05 |  |
| 06 | //IE/Opera/Chrome 下 向上滚，t 为正时 则pageIndex - 1; 向下滚 t为负时 则pageIndex + 1; | |

|  |  |  |
| --- | --- | --- |
| 07 | t > 0 ? pageIndex-- : pageIndex++; | |
| 08 | } else if (e.detail) { |

|  |  |
| --- | --- |
| 09 |  |
| 10 | //Firefox | |

|  |  |  |
| --- | --- | --- |
| 11 | t = e.detail; | |
| 12 |  |

|  |  |  |
| --- | --- | --- |
| 13 | //firefox下 向上滚，t为负时 则pageIndex - 1; 向下滚 t为正时 则pageIndex + 1; | |
| 14 | t < 0 ? pageIndex-- : pageIndex++; |

|  |  |
| --- | --- |
| 15 | } |

那么，这里我们又引出了一个问题：当pageIndex 自加 大于总页面时 以及 自减 小于零时怎么办？

当大于总页面时就不让滚动了

所以，这里需要做个判断：

|  |  |
| --- | --- |
| 1 | pageIndex = pageIndex >= oPage.length ? oPage.length : pageIndex; |

当小于0时，也不让滚动

|  |  |
| --- | --- |
| 1 | pageIndex = pageIndex < 0 ?  0 : pageIndex; |

**5.滚动动画效果**

好了，到现在都没看到滚动效果，是不是不蛋定啦~~  
少年，hold住：  
分析下，动画前咱们得 存储动前的高度，到时才好计算应该滚动多少，有个初始值

|  |  |
| --- | --- |
| 1 | function foldTop(){ |
| 2 | oldTop = pageIndex \* h; //老的高度 | |

|  |  |  |
| --- | --- | --- |
| 3 | top1 = -oldTop; //暂存一下 应该滚动的高度。稍后动画需要加减其值 | |
| 4 | } |

那算了初使值 ，接下来就要算 滚动后的值啦

设置动画滚动高度 在这里就是索引变化后，需要滚动到的值

|  |  |
| --- | --- |
| 1 | aniTop = pageIndex \* h; |

这里考虑到一个性能问题，万一哪个多动症少年对着某一个按钮按着激动的根本停不下来，那怎么办，不断的去计算吗？我们对于这种流氓行为当然说no~~

|  |  |  |
| --- | --- | --- |
| 1 | //如果当前值和老的索引值一样，不处理（这里的可能性就是在导航事件上，重复点同一个按钮,优化性能） | |
| 2 | if (aniTop == oldTop) return false; |

有了这句，下面的代码就不执行啦，就乖乖定在这吧，您就按吧~！

现在需要遍历每个box,或每个菜单，因为它们的数量都相等的，遍谁都一样，然后再利用正则去除每个box和li的active状态

|  |  |
| --- | --- |
| 1 | for (var i = 0; i < oPage.length; i++) { |
| 2 | oPage[i].className = oPage[i].className == "" ? "" : oPage[i].className.replace(/\sactive|active\s|\sactive\s/i,""); | |

|  |  |  |
| --- | --- | --- |
| 3 | node[i].className = node[i].className == "" ? "" : node[i].className.replace(/\sactive|active\s|\sactive\s/i,""); | |
| 4 | node[i].className = ""; |

|  |  |
| --- | --- |
| 5 | } |

好了，代码执行到这，也是要进入播放的了，但是想了一下，如果这个动画还没执行完，有的性格急的，滚的太快，或按的太快，那么，动画就易乱。所以这里我们还是限制一下的好。  
道理就是：此时动画执行，如果没执行完，是不可以再增加索引值的

|  |  |
| --- | --- |
| 1 | play = false; |

因为我们动画开头就有个判断，所以动画过程中是必需指定不给继续播放其他的索引动画的。

|  |  |
| --- | --- |
| 1 | if(!play) { |
| 2 | return false; | |

|  |  |
| --- | --- |
| 3 | } |

动画要执行的结果就是 最外层的盒子fullWrap 的top值.  
之前暂存的top1 是索引值变化前的fullWarp top值,在这里我们跟据索引值变化情况不断的去加或减它,让fullWrap动起来。  
上面说到跟据索引值变化情况，倒底是什么情况呢？情况就是索引增大或减小  
当索引值减小，盒子肯定是向下滚动。因为老的top值要小于新的top值;  
相反，索引值增加，则盒子是向上滚动的，因为，老的top值大于新的top值

|  |  |
| --- | --- |
| 1 | //向上 |
| 2 | if (oldTop <= aniTop) { | |

|  |  |
| --- | --- |
| 3 | top1 -= 30; //每次以30相素的减 |
| 4 | (top1 < -aniTop) && (top1 = -aniTop); //当top1减到比新的top还要低时，就定死 top1 = -aniTop | |

|  |  |
| --- | --- |
| 5 | } else { //向上 |
| 6 | top1 += 30; | |

|  |  |  |
| --- | --- | --- |
| 7 | (top1 > -aniTop) && (top1 = -aniTop); //当top1加到比新的top还要大时，就定死 top1 = -aniTop | |
| 8 | } |

|  |  |
| --- | --- |
| 9 | fullWrap.style.top = top1  +"px"; //移动外层fullWrap |

当top1 = -aniTop 时就定死啦，不让加减啦，所以这里就表示动画执行结束，可以写回调方法了

**回调** 当高度滚到 之前计算的滚动值，则停止定时器，并且给新索引值对应的li加上active 状态

|  |  |
| --- | --- |
| 1 | if (top1 == -aniTop) { |
| 2 | clearInterval(timer); | |

|  |  |
| --- | --- |
| 3 | play = true; |
| 4 | oPage[pageIndex].className = oPage[pageIndex].className == "" ?  "active" : oPage[pageIndex].className + " active"; | |

|  |  |  |
| --- | --- | --- |
| 5 | node[pageIndex].className = node[pageIndex].className == "" ? "active" : node[pageIndex].className + " active"; | |
| 6 | }; |

看看，上面是不是少了啥,没有设定滚动一屏的时间。

so,要把上面的全包含在一个定时器 timer里,设定时间为10s

|  |  |  |
| --- | --- | --- |
| 1 | timer = setInterval(function (){ | |
| 2 | //....上面的动画代码 |

|  |  |
| --- | --- |
| 3 | },10); |

**6.扩展菜单**

声明node = [];

遍历每页，然后创建追加节点，定义索引，每个节点邦定点击事件。

|  |  |
| --- | --- |
| 01 | for (var i = 0; i < oPage.length; i++) { |
| 02 | node[i] = document.createElement("li"); | |

|  |  |  |
| --- | --- | --- |
| 03 | oMenu.appendChild(node[i]); | |
| 04 | node[i].index = i; |

|  |  |
| --- | --- |
| 05 |  |
| 06 | node[i].onclick = function () { | |

|  |  |  |
| --- | --- | --- |
| 07 | if(!play) return false; //检查是否正在执行动画 | |
| 08 | foldTop(); //存储老的高度 |

|  |  |  |
| --- | --- | --- |
| 09 | pageIndex = this.index; //变更索引为点击的值 | |
| 10 | animat ();  //执行动画 |

|  |  |  |
| --- | --- | --- |
| 11 | } | |
| 12 | }; |

好了，附上

[demo](http://fed.lvmama.com/wp-content/uploads/2014/12/js-pagescroll.html)

[实例](http://www.lvmama.com/trip/lvyoubao)…

其实这个全屏滚动功能 用jquery来写，快速又高效，实际开发时建议用jquery,节省时间，但对原生js掌握不好的同学，建议多练习，这样在使用jquery时更能了解其原理。  
可能有点见解不对，且代码冗余，待请大神指教，Thanks