# **[移动端兼容性问题解决方案](http://www.cnblogs.com/luckyXcc/p/6237933.html)**

### **1. IOS移动端click事件300ms的延迟响应**

移动设备上的web网页是有300ms延迟的，玩玩会造成按钮点击延迟甚至是点击失效。这是由于区分单击事件和双击屏幕缩放的历史原因造成的,

2007年苹果发布首款iphone上IOS系统搭载的safari为了将适用于PC端上大屏幕的网页能比较好的展示在手机端上，使用了双击缩放(double tap to zoom)的方案，比如你在手机上用浏览器打开一个PC上的网页，你可能在看到页面内容虽然可以撑满整个屏幕，但是字体、图片都很小看不清，此时可以快速双击屏幕上的某一部分，你就能看清该部分放大后的内容，再次双击后能回到原始状态。

双击缩放是指用手指在屏幕上快速点击两次，iOS 自带的 Safari 浏览器会将网页缩放至原始比例。

原因就出在浏览器需要如何判断快速点击上，当用户在屏幕上单击某一个元素时候，例如跳转链接<a href="#"></a>，此处浏览器会先捕获该次单击，但浏览器不能决定用户是单纯要点击链接还是要双击该部分区域进行缩放操作，所以，捕获第一次单击后，浏览器会先Hold一段时间t，如果在t时间区间里用户未进行下一次点击，则浏览器会做单击跳转链接的处理，如果t时间里用户进行了第二次单击操作，则浏览器会禁止跳转，转而进行对该部分区域页面的缩放操作。那么这个时间区间t有多少呢？在IOS safari下，大概为300毫秒。这就是延迟的由来。造成的后果用户纯粹单击页面，页面需要过一段时间才响应，给用户慢体验感觉，对于web开发者来说是，页面js捕获click事件的回调函数处理，需要300ms后才生效，也就间接导致影响其他业务逻辑的处理。

解决方案：

* fastclick可以解决在手机上点击事件的300ms延迟
* zepto的touch模块，tap事件也是为了解决在click的延迟问题
* 触摸事件的响应顺序为 touchstart --> touchmove --> touchend --> click,也可以通过绑定ontouchstart事件，加快对事件的响应，解决300ms延迟问题

### **2.一些情况下对非可点击元素如(label,span)监听click事件，ios下不会触发，css增加cursor:pointer就搞定了。**

### **3. 三星手机遮罩层下的input、select、a等元素可以被点击和focus(点击穿透)**

问题发现于三星手机，这个在特定需求下才会有，因此如果没有类似问题的可以不看。首先需求是浮层操作，在三星上被遮罩的元素依然可以获取focus、click、change)，有两种解决方案，

1.是通过层显示以后加入对应的class名控制，截断显示层下方可获取焦点元素的事件获取

2.是通过将可获取焦点元素加入的disabled属性，也可以利用属性加dom锁定的方式（disabled的一种变换方式）

### **4. h5底部输入框被键盘遮挡问题**

h5页面有个很蛋疼的问题就是，当输入框在最底部，点击软键盘后输入框会被遮挡。可采用如下方式解决

[IMG_256](http://www.cnblogs.com/luckyXcc/p/javascript:void(0);)

IMG_257

var oHeight = $(document).height(); //浏览器当前的高度

$(window).resize(function(){

if($(document).height() < oHeight){

$("#footer").css("position","static");

}else{

$("#footer").css("position","absolute");

}

});

  关于Web移动端Fixed布局的解决方案，这篇文章也不错

http://efe.baidu.com/blog/mobile-fixed-layout/

IMG_258

[IMG_259](http://www.cnblogs.com/luckyXcc/p/javascript:void(0);)

### **5.不让 Android 手机识别邮箱**

<meta content="email=no" name="format-detection" />

### **6.禁止 iOS 识别长串数字为电话**

<meta content="telephone=no" name="format-detection" />

### **7.禁止 iOS 弹出各种操作窗口**

-webkit-touch-callout:none

### **8.消除 transition 闪屏**

-webkit-transform-style: preserve-3d; /\*设置内嵌的元素在 3D 空间如何呈现：保留 3D\*/

-webkit-backface-visibility: hidden; /\*(设置进行转换的元素的背面在面对用户时是否可见：隐藏)\*/

### **9.iOS 系统中文输入法输入英文时，字母之间可能会出现一个六分之一空格**

可以通过正则去掉      this.value = this.value.replace(/\u2006/g, '');

### **10.禁止ios和android用户选中文字**

-webkit-user-select:none

### **11.在ios和andriod中,audio元素和video元素在无法自动播放**

应对方案：触屏即播

$('html').one('touchstart',function(){

audio.play()

})

### **12.ios下取消input在输入的时候英文首字母的默认大写**

<input autocapitalize="off" autocorrect="off" />

### **13.android下取消输入语音按钮**

input::-webkit-input-speech-button {display: none}

### **14  CSS动画页面闪白,动画卡顿**

[IMG_260](http://www.cnblogs.com/luckyXcc/p/javascript:void(0);)

IMG_261

解决方法:  
1.尽可能地使用合成属性transform和opacity来设计CSS3动画，不使用position的left和top来定位  
2.开启硬件加速

-webkit-transform: translate3d(0, 0, 0);

-moz-transform: translate3d(0, 0, 0);

-ms-transform: translate3d(0, 0, 0);

transform: translate3d(0, 0, 0);

IMG_262

[IMG_263](http://www.cnblogs.com/luckyXcc/p/javascript:void(0);)

### **16.fixed定位缺陷**

* ios下fixed元素容易定位出错，软键盘弹出时，影响fixed元素定位
* android下fixed表现要比iOS更好，软键盘弹出时，不会影响fixed元素定位
* ios4下不支持position:fixed
* 解决方案： 可用iScroll插件解决这个问题

### **17.阻止旋转屏幕时自动调整字体大小**

html, body, form, fieldset, p, div, h1, h2, h3, h4, h5, h6 {-webkit-text-size-adjust:none;}

## **18 Input 的placeholder会出现文本位置偏上的情况**

input 的placeholder会出现文本位置偏上的情况：PC端设置line-height等于height能够对齐，而移动端仍然是偏上，解决是设置line-height：normal

## **19 往返缓存问题**

点击浏览器的回退，有时候不会自动执行js，特别是在mobilesafari中。这与**往返缓存(bfcache)**有关系。

解决方法 ：window.onunload = function(){};

### **20. calc的兼容性处理**

CSS3中的calc变量在iOS6浏览器中必须加-webkit-前缀，目前的FF浏览器已经无需-moz-前缀。  
Android浏览器目前仍然不支持calc，所以要在之前增加一个保守尺寸：

div {

width: 95%;

width: -webkit-calc(100% - 50px);

width: calc(100% - 50px);

}

### **21 iOS6下伪类:hover**

除了<a>之外的元素无效；在Android下则有效。类似

div#topFloatBar\_l:hover #topFloatBar\_menu { display:block; }

这样的导航显示在iOS6点击没有点击效果，只能通过增加点击侦听器给元素增减class来控制子元素。

### **22 在移动端修改难看的点击的高亮效果，iOS和安卓下都有效：**

\* {-webkit-tap-highlight-color:rgba(0,0,0,0);}

不过这个方法在现在的安卓浏览器下，只能去掉那个橙色的背景色，点击产生的高亮边框还是没有去掉，有待解决！

一个CSS3的属性，加上后，所关联的元素的事件监听都会失效，等于让元素变得“看得见，点不着”。IE到11才开始支持，其他浏览器的当前版本基本都支持。详细介绍见这里：[https://developer.mozilla.org/zh-CN/docs/Web/CSS/pointer-events](https://developer.mozilla.org/zh-CN/docs/Web/CSS/pointer-events" \t "http://www.cnblogs.com/luckyXcc/p/_blank)

pointer-events: none;

### **23. Zepto点透的解决方案**

zepto的tap是通过兼听绑定在document上的touch事件来完成tap事件的模拟的,及tap事件是冒泡到document上触发的,在点击完成时的tap事件(touchstart\touchend)需要冒泡到document上才会触发，而在冒泡到document之前，用户手的接触屏幕(touchstart)和离开屏幕(touchend)是会触发click事件的,因为click事件有延迟触发(这就是为什么移动端不用click而用tap的原因)(大概是300ms,为了实现safari的双击事件的设计)，所以在执行完tap事件之后，弹出来的选择组件马上就隐藏了，此时click事件还在延迟的300ms之中，当300ms到来的时候，click到的其实不是完成而是**隐藏之后的下方的元素，如果正下方的元素绑定的有click事件此时便会触发，如果没有绑定click事件的话就当没click，但是正下方的是input输入框(或者select选择框或者单选复选框)，点击默认聚焦而弹出输入键盘，也就出现了上面的点透现象。**

引入fastclick.js，在页面中加入如下js代码

1 window.addEventListener( "load", function() {

2 FastClick.attach( document.body );

3 }, false );

或者有zepto或者jQuery的js里面加上

1 $(function() {

2 FastClick.attach(document.body);

3 });

当然require的话就这样：

1 var FastClick = require('fastclick');

2 FastClick.attach(document.body, options);

方案二：用touchend代替tap事件并阻止掉touchend的默认行为preventDefault()

1 $("#cbFinish").on("touchend", function (event) {

2 //很多处理比如隐藏什么的

3 event.preventDefault();

4 });

方案三：延迟一定的时间(300ms+)来处理事件

1 $("#cbFinish").on("tap", function (event) {

2 setTimeout(function(){

3 //很多处理比如隐藏什么的

4 },320);

5 });