**https://blog.csdn.net/hardgirls/article/details/51722519**

**随着手机的普及，移动端的开发也成了一个重要的方向，但由于设备的不统一会造成一些兼容性问题，**

**1、安卓浏览器看背景图片，有些设备会模糊。**

用同等比例的图片在PC机上很清楚，但是手机上很模糊，原因是什么呢？

经过研究，是devicePixelRatio作怪，因为手机分辨率太小，如果按照分辨率来显示网页，这样字会非常小，所以苹果当初就把iPhone 4的960\*640分辨率，在网页里只显示了480\*320，这样devicePixelRatio＝2。现在android比较乱，有1.5的，有2的也有3的。

想让图片在手机里显示更为清晰，必须使用2x的背景图来代替img标签（一般情况都是用2倍）。例如一个div的宽高是100\*100，背景图必须得200\*200，然后background-size:contain;，这样显示出来的图片就比较清晰了。

代码可以如下：

background:url(../images/icon/all.png) no-repeat center center;

-webkit-background-size:50px 50px;

background-size: 50px 50px;display:inline-block; width:100%; height:50px;

或者指定 background-size:contain;都可以，大家试试！

**2、图片加载**

若您遇到图片加载很慢的问题，对这种情况，手机开发一般用canvas方法加载：

具体的canvas API 参见：http://javascript.ruanyifeng.com/htmlapi/canvas.html

下面举例说明一个canvas的例子：

<li><canvas></canvas></li>

js动态加载图片和li 总共举例17张图片！

vartotal=17;   
varzWin=$(window);   
varrender=function(){  
  varpadding=2;   
  varwinWidth=zWin.width();   
  varpicWidth=Math.floor((winWidth-padding\*3)/4);   
  vartmpl ='';  
  for(vari=1;i<=totla;i++){   
     varp=padding;   
     varimgSrc='img/'+i+'.jpg';  
     if(i%4==1){  
        p=0;  
     }  
  tmpl +='<li style="width:'+picWidth+'px;height:'+picWidth+'px;padding-left:'+p+'px;padding-top:'+padding+'px;"><canvas id="cvs\_'+i+'"></canvas></li>';  
  varimageObj = newImage();   
  imageObj.index = i;   
  imageObj.onload = function(){  
    varcvs =$('#cvs\_'+this.index)[0].getContext('2d');  
    cvs.width = this.width;  
    cvs.height=this.height;  
    cvs.drawImage(this,0,0);  
  }  
  imageObj.src=imgSrc;  
  }  
}  
render();

**3、假如手机网站不用兼容IE浏览器，一般我们会使用zeptojs**。

zeptojs内置Touch events方法，具体可以看http://zeptojs.com/#Touch events

看了一下zeptio新版的API，已经支持IE10以上浏览器，对zeptojs可以选择使用！

**4、防止手机中网页放大和缩小。**

这点是最基本的，最为手机网站开发者来说应该都知道的，就是设置meta中的viewport

还有就是，有些手机网站我们看到如下声明：

代码如下:

<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN" "http://www.wapforum.org/DTD/xhtml-mobile10.dtd">

设置了DTD的方式是XHTML的写法，假如我们页面运用的是html5，可以不用设置DTD,直接声明<!DOCTYPE html>。

使用viewport使页面禁止缩放。 通常把user-scalable设置为0来关闭用户对页面视图缩放的行为。

<meta name="viewport"content="user-scalable=0"/>

但是为了更好的兼容，我们会使用完整的viewport设置。

复制代码代码如下:

<meta name="viewport" content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=0" />

当然，user-scalable=0,有的人也写成user-scalable=no，都可以的。

**5、apple-mobile-web-app-capable**

apple-mobile-web-app-capable是设置Web应用是否以全屏模式运行。

语法：

<meta name="apple-mobile-web-app-capable"content="yes">

说明：

如果content设置为yes，Web应用会以全屏模式运行，反之，则不会。content的默认值是no，表示正常显示。你可以通过只读属性window.navigator.standalone来确定网页是否以全屏模式显示。

**6、format-detection**

format-detection 启动或禁用自动识别页面中的电话号码。

语法：

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

说明：

默认情况下，设备会自动识别任何可能是电话号码的字符串。设置telephone=no可以禁用这项功能。

**7、html5调用安卓或者ios的拨号功能**

html5提供了自动调用拨号的标签，只要在a标签的href中添加tel:就可以了。

如下：

<ahref="tel:4008106999,1034">400-810-6999 转 1034</a>

拨打手机直接如下

 <a href="tel:15677776767">点击拨打15677776767</a>  
**8、html5GPS定位功能**

具体请看：http://www.jb51.net/post/html5\_GPS\_getCurrentPosition

**9、上下拉动滚动条时卡顿、慢**

body {-webkit-overflow-scrolling: touch; overflow-scrolling: touch;}

Android3+和iOS5+支持CSS3的新属性为overflow-scrolling

**10、禁止复制、选中文本**

Element {-webkit-user-select:none;  
  -moz-user-select:none;  
  -khtml-user-select:none;  
   user-select:none;  
}

解决移动设备可选中页面文本(视产品需要而定)

**11、长时间按住页面出现闪退**

element {  
  -webkit-touch-callout:none;  
}

**12、iphone及ipad下输入框默认内阴影**

Element{  
  -webkit-appearance:none;  
}

**13、ios和android下触摸元素时出现半透明灰色遮罩**

Element {  
  -webkit-tap-highlight-color:rgba(255,255,255,0)  
}

设置alpha值为0就可以去除半透明灰色遮罩，备注：transparent的属性值在android下无效。

后面一篇文章有详细介绍，地址：http://www.jb51.net/post/phone\_web\_ysk

**14、active兼容处理 即 伪类 :active 失效**

方法一：body添加ontouchstart

<body ontouchstart="">

方法二：js给 document 绑定 touchstart 或 touchend 事件

<style>  
   a {  
     color:#000;  
   }  
   a:active {  
      color:#fff;  
    }  
    </style>  
    <a herf=foo >bar</a>  
 <script>  
    document.addEventListener('touchstart',function(){},false);  
</script>

**15、动画定义3D启用硬件加速**

Element {  
  -webkit-transform:translate3d(0,0,0)  
  transform: translate3d(0,0,0);  
}

注意：3D变形会消耗更多的内存与功耗

**16、Retina屏的1px边框**

Element{  
  border-width:thin;  
}

**17、webkit mask 兼容处理**

某些低端手机不支持css3 mask，可以选择性的降级处理。

比如可以使用js判断来引用不同class：

if('WebkitMask'indocument.documentElement.style){  
  alert('支持mask');  
}else{  
  alert('不支持mask');  
}

**18、旋转屏幕时，字体大小调整的问题**

html, body, form, fieldset, p, div, h1, h2, h3, h4, h5, h6{  
  -webkit-text-size-adjust:100%;  
}

**19、transition闪屏**

/设置内嵌的元素在3D 空间如何呈现：保留3D /  
-webkit-transform-style: preserve-3d;  
/ 设置进行转换的元素的背面在面对用户时是否可见：隐藏 /   
-webkit-backface-visibility:hidden;

**20、圆角bug**

某些Android手机圆角失效

background-clip: padding-box;

**21、顶部状态栏背景色**

<meta name="apple-mobile-web-app-status-bar-style"content="black"/>

说明：

除非你先使用apple-mobile-web-app-capable指定全屏模式，否则这个meta标签不会起任何作用。

如果content设置为default，则状态栏正常显示。如果设置为blank，则状态栏会有一个黑色的背景。如果设置为blank-translucent，则状态栏显示为黑色半透明。如果设置为default或blank，则页面显示在状态栏的下方，即状态栏占据上方部分，页面占据下方部分，二者没有遮挡对方或被遮挡。如果设置为blank-translucent，则页面会充满屏幕，其中页面顶部会被状态栏遮盖住（会覆盖页面20px高度，而iphone4和itouch4的Retina屏幕为40px）。默认值是default。

**22、设置缓存**

<meta http-equiv="Cache-Control"content="no-cache"/>

手机页面通常在第一次加载后会进行缓存，然后每次刷新会使用缓存而不是去重新向服务器发送请求。如果不希望使用缓存可以设置no-cache。

**23、桌面图标**

<link rel="apple-touch-icon"href="touch-icon-iphone.png"/>  
<link rel="apple-touch-icon"sizes="76x76"href="touch-icon-ipad.png"/>  
<link rel="apple-touch-icon"sizes="120x120"href="touch-icon-iphone-retina.png"/>  
<link rel="apple-touch-icon"sizes="152x152"href="touch-icon-ipad-retina.png"/>

iOS下针对不同设备定义不同的桌面图标。如果不定义则以当前屏幕截图作为图标。

上面的写法可能大家会觉得会有默认光泽，下面这种设置方法可以去掉光泽效果，还原设计图的效果！

   <link rel="apple-touch-icon-precomposed"href="touch-icon-iphone.png"/>

图片尺寸可以设定为5757（px）或者Retina可以定为114114（px），ipad尺寸为72\*72（px)

**24、启动画面**

<link rel="apple-touch-startup-image"href="start.png"/>

iOS下页面启动加载时显示的画面图片，避免加载时的白屏。

可以通过madia来指定不同的大小：

<!--iPhone-->  
<link href="apple-touch-startup-image-320x460.png"media="(device-width: 320px)" rel="apple-touch-startup-image"/>  
<!-- iPhone Retina -->  
<link href="apple-touch-startup-image-640x920.png"media="(device-width: 320px) and (-webkit-device-pixel-ratio: 2)" rel="apple-touch-startup-image"/>  
<!-- iPhone 5-->  
<link rel="apple-touch-startup-image"media="(device-width: 320px) and (device-height: 568px) and (-webkit-device-pixel-ratio: 2)" href="apple-touch-startup-image-640x1096.png">  
<!-- iPad portrait-->  
<link href="apple-touch-startup-image-768x1004.png"media="(device-width: 768px) and (orientation: portrait)" rel="apple-touch-startup-image"/>  
<!-- iPad landscape-->  
<link href="apple-touch-startup-image-748x1024.png"media="(device-width: 768px) and (orientation: landscape)" rel="apple-touch-startup-image"/>  
<!-- iPad Retina portrait-->  
<link href="apple-touch-startup-image-1536x2008.png"media="(device-width: 1536px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 2)" rel="apple-touch-startup-image"/>  
<!-- iPad Retina landscape-->  
<link href="apple-touch-startup-image-1496x2048.png"media="(device-width: 1536px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 2)"rel="apple-touch-startup-image"/>

**25、浏览器私有及其它meta**

以下属性在项目中没有应用过，可以写一个demo测试以下！

QQ浏览器私有

全屏模式

<meta name="x5-fullscreen"content="true">

强制竖屏

<meta name="x5-orientation"content="portrait">

强制横屏

<meta name="x5-orientation"content="landscape">

应用模式

<meta name="x5-page-mode"content="app">

UC浏览器私有

全屏模式

<meta name="full-screen"content="yes">

强制竖屏

<meta name="screen-orientation"content="portrait">

强制横屏

<meta name="screen-orientation"content="landscape">

应用模式

<meta name="browsermode"content="application">

其它

针对手持设备优化，主要是针对一些老的不识别viewport的浏览器，比如黑莓

<meta name="HandheldFriendly"content="true">

微软的老式浏览器

<meta name="MobileOptimized"content="320">

windows phone 点击无高光

<meta name="msapplication-tap-highlight"content="no">

**26、 IOS中input键盘事件keyup、keydown、keypress支持不是很好**

问题是这样的，用input search做模糊搜索的时候，在键盘里面输入关键词，会通过ajax后台查询，然后返回数据，然后再对返回的数据进行关键词标红。用input监听键盘keyup事件，在安卓手机浏览器中是可以的，但是在ios手机浏览器中变红很慢，用输入法输入之后，并未立刻相应keyup事件，只有在通过删除之后才能相应！

解决办法：

可以用html5的oninput事件去代替keyup

<input type="text"id="testInput">  
<script type="text/javascript">  
  document.getElementById('testInput').addEventListener('input',function(e){  
    varvalue = e.target.value;  
  });  
</script>

然后就达到类似keyup的效果！

**27、h5网站input 设置为type=number的问题**

h5网页input 的type设置为number一般会产生三个问题，一个问题是maxlength属性不好用了。另外一个是form提交的时候，默认给取整了。三是部分安卓手机出现样式问题。

问题一解决，我目前用的是js。如下

<input type="number"oninput="checkTextLength(this ,10)">  
   
functioncheckTextLength(obj, length) {    
      if(obj.value.length > length)  {       
        obj.value = obj.value.substr(0, length);    
      }   
}

问题二，是因为form提交默认做了表单验证，step默认是1,要设置step属性，假如保留2位小数，写法如下：

<input type="number"step="0.01"/>

关于step，我在这里做简单的介绍，input 中type=number，一般会自动生成一个上下箭头，点击上箭头默认增加一个step，点击下箭头默认会减少一个step。number中默认step是1。也就是step=0.01,可以允许输入2位小数，并且点击上下箭头分别增加0.01和减少0.01。

假如step和min一起使用，那么数值必须在min和max之间。

看下面的例子：

<input type="number"step="3.1"min="1"/>

输入框可以输入哪些数字？

首先，最小值是1，那么可以输入1.0，第二个是可以输入（1+3.1）那就是4.1,以此类推，每次点击上下箭头都会增加或者减少3.1，输入其他数字无效。这就是step的简单介绍。

问题三，去除input默认样式

input[type=number] {  
  -moz-appearance:textfield;  
}  
input[type=number]::-webkit-inner-spin-button,  
input[type=number]::-webkit-outer-spin-button {  
  -webkit-appearance:none;  
  margin:0;  
}

**28、ios 设置input 按钮样式会被默认样式覆盖**

解决方式如下：

input,  
textarea {  
  border: 0;   
  -webkit-appearance: none;   
}

设置默认样式为none

**29、IOS键盘字母输入，默认首字母大写**

解决方案，设置如下属性

<input type="text"autocapitalize="off"/>

**30、select 下拉选择设置右对齐**

设置如下：

select option {  
direction: rtl;  
}

**31、通过transform进行skew变形，rotate旋转会造成出现锯齿现象**

可以设置如下：

-webkit-transform: rotate(-4deg) skew(10deg) translateZ(0);  
 transform: rotate(-4deg) skew(10deg) translateZ(0);  
 outline: 1px solid rgba(255,255,255,0)

**32、移动端点击300ms延迟**

300ms尚可接受，不过因为300ms产生的问题，我们必须要解决。300ms导致用户体验并不是很好，解决这个问题，我们一般在移动端用tap事件来取代click事件。

推荐两个js，一个是fastclick，一个是tap.js

关于300ms延迟，具体请看：http://thx.github.io/mobile/300ms-click-delay/

**33、移动端点透问题**

案例如下：

<div id="haorooms">点头事件测试</div>  
   
<a href="www.baidu.net">www.baidu.com</a>

div是绝对定位的蒙层,并且z-index高于a。而a标签是页面中的一个链接，我们给div绑定tap事件：

$('#haorooms').on('tap',function(){  
$('#haorooms').hide();  
});

我们点击蒙层时 div正常消失，但是当我们在a标签上点击蒙层时，发现a链接被触发，这就是所谓的点透事件。

原因：

touchstart 早于 touchend 早于click。 亦即click的触发是有延迟的，这个时间大概在300ms左右，也就是说我们tap触发之后蒙层隐藏， 此时 click还没有触发，300ms之后由于蒙层隐藏，我们的click触发到了下面的a链接上。  
解决：

（1）尽量都使用touch事件来替换click事件。例如用touchend事件(推荐)。  
（2）用fastclick，https://github.com/ftlabs/fastclick  
（3）用preventDefault阻止a标签的click  
（4）延迟一定的时间(300ms+)来处理事件 （不推荐）  
（5）以上一般都能解决，实在不行就换成click事件。  
下面介绍一下touchend事件，如下：

$("#haorooms").on("touchend",function(event) {  
   event.preventDefault();  
 });

**34、消除 IE10 里面的那个叉号**

input:-ms-clear{display:none;}

**35、关于 iOS 与 OS X 端字体的优化(横竖屏会出现字体加粗不一致等)**

iOS 浏览器横屏时会重置字体大小，设置 text-size-adjust 为 none 可以解决 iOS 上的问题，但桌面版 Safari 的字体缩放功能会失效，因此最佳方案是将 text-size-adjust 为 100% 。

-webkit-text-size-adjust:100%;  
-ms-text-size-adjust:100%;  
text-size-adjust:100%;

**36、关于 iOS 系统中，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,'');

**37、移动端 HTML5 audio autoplay 失效问题**

这个不是 BUG，由于自动播放网页中的音频或视频，会给用户带来一些困扰或者不必要的流量消耗，所以苹果系统和安卓系统通常都会禁止自动播放和使用 JS 的触发播放，必须由用户来触发才可以播放。

解决方法思路：先通过用户 touchstart 触碰，触发播放并暂停（音频开始加载，后面用 JS 再操作就没问题了）。

解决代码：

document.addEventListener('touchstart',function() {  
  document.getElementsByTagName('audio')[0].play();  
  document.getElementsByTagName('audio')[0].pause();  
});

**38、移动端 HTML5 input date 不支持 placeholder 问题**

这个我感觉没有什么好的解决方案，用如下方法

复制代码代码如下:

<input placeholder="Date" class="textbox-n" type="text" onfocus="(this.type='date')"  id="date">

有的浏览器可能要点击两遍！

**39、部分机型存在type为search的input，自带close按钮样式修改方法**

有些机型的搜索input控件会自带close按钮（一个伪元素），而通常为了兼容所有浏览器，我们会自己实现一个，此时去掉原生close按钮的方法为

#Search::-webkit-search-cancel-button{  
  display:none;   
}

如果想使用原生close按钮，又想使其符合设计风格，可以对这个伪元素的样式进行修改。

**40、唤起select的option展开**

zepto方式:

$(sltElement).trrgger("mousedown");

原生js方式:

functionshowDropdown(sltElement) {  
  varevent;  
  event = document.createEvent('MouseEvents');  
  event.initMouseEvent('mousedown',true,true, window);  
  sltElement.dispatchEvent(event);  
};