1.解决页面使用overflow: scroll在ios上滑动卡顿的问题?

- 看是否能把body和html的height: 100%去除掉。
- 在滚动的容器中增加：-webkit-overflow-scrolling: touch或者给body增加：body {overflow-x: hidden}

2.ios页面橡皮弹回效果遮挡页面选项卡？

- 有时body和html的height: 100%去除掉问题可能就没有了
- 到达临界值的时候在阻止事件默认行为

```
var startY,endY;
//记录手指触摸的起点坐标
$('body').on('touchstart',function (e) {
     startY = e.touches[0].pageY;
});
$('body').on('touchmove',function (e) {
     endY = e.touches[0].pageY;  //记录手指触摸的移动中的坐标
     //手指下滑，页面到达顶端不能继续下滑
     if(endY>startY&& $(window).scrollTop()<=0){
         e.preventDefault();
     }
   //手指上滑，页面到达底部能继续上滑
     if(endY<startY&& $(window).scrollTop()+
         $(window).height()>=$('body')[0].scrollHeight){
         e.preventDefault();
     }
})
```

有时也会碰见弹窗出来后两个层的橡皮筋效果出现问题，我们可以在弹出弹出时给底层页面加上一个类名，类名禁止页面滑动这样下层的橡皮筋效果就会被禁止，就不会影响弹窗层。

3.IOS机型margin属性无效问题？

- 设置html body的高度为百分比时，margin-bottom在safari里失效
- 直接padding代替margin

4.Ios绑定点击事件不执行？

- 添加样式cursor ：pointer。点击后消除背景闪一下的css：-webkit-tap-highlight-color:transparent;

5.Ios键盘换行变为搜索？

- 首先，input 要放在 form里面。
- 这时 "换行" 已经变成 “前往”。
- 如果想变成 “搜索”，input 设置 type="search"。

6.Jq对a标签点击事件不生效？

出现这种情况的原因不明，有的朋友解释：我们平时都是点击的A标签中的文字了。 所以要想用JS模拟点击A标签事件，就得先往A标签中的文字添加能被JS捕获的元素，然后再用JS模拟点击该元素即可。但是我觉得不合理，虽然找不到原因但是解决办法还是有的。

- document.getElementById("abc ").click();
- $("#abc ")[0].click();

7.有时因为服务器或者别的原因导致页面上的图片没有找到？

```
<script type="text/javascript">
function nofind(){
    var img=event.srcElement;
    img.src="images/logoError.png";
    img.onerror=null; 控制不要一直跳动
}
</script>

<img src="images/logo.png" onerror="nofind();" />
```

8.transform属性影响position:fixed？

- 规范中有规定：如果元素的transform值不为none，则该元素会生成包含块和层叠上下文。CSS Transforms Module Level 1不只在手机上，电脑上也一样。除了fixed元素会受影响之外，z-index（层叠上下文）值也会受影响。绝对定位元素等和包含块有关的属性都会受到影响。当然如果transform元素的display值为inline时又会有所不同。最简单的解决方法就是transform元素内部不能有absolute、fixed元素.

9.ios对position: fixed不太友好，有时我们需要加点处理？

```
var u = navigator.userAgent, app = navigator.appVersion;
var isiOS = !!u.match(/\(i[^;]+;( U;)? CPU.+Mac OS X/); //ios终端
if (isiOS) {
    $('textarea').focus(function () {
        window.setTimeout('scrollBottom()', 500);
    });
}
function scrollBottom() {
    window.scrollTo(0, $('body').height());
}
```

10.jq validate插件验证问题、手机会出现断网的情况？

- input必须有name不然会出错;
- navigator.onLine可判断是否是脱机状态;

11.判断对象的长度？

- 用Object.keys，Object.keys方法返回的是一个数组，数组里面装的是对象的属性。

```
var person = {
    "name" : "zhangshan",
    "sex" : "man",
    "age" : "50",
    "height" : "180",
    "phone" : "1xxxxxxxxxx",
    "email" : "xxxxxxxxx@xxx.com"
};
var arr = Object.keys(person);
console.log(arr.length);
```

- Object.getOwnPropertyNames(obj).length

12.Object.keys与Object.getOwnPropertyNames他们的区别？

- Object.keys定义：返回一个对象可枚举属性的字符串数组；
- Object.getOwnPropertyNames定义：返回一个对象可枚举、不可枚举属性的名称；

定义：可枚举属性是指那些内部 “可枚举” 标志设置为 true 的属性，对于通过直接的赋值和属性初始化的属性，该标识值默认为即为 true，对于通过 Object.defineProperty 等定义的属性，该标识值默认为 false。

```
var obj = { "prop1": "v1" };
Object.defineProperty(obj, "prop2", { value: "v2", enumerable: false });
console.log(Object.keys(obj).length);            //output：1
console.log(Object.getOwnPropertyNames(obj).length);    //output：2

console.log(Object.keys(obj));           //output：Array[1] => [0: "prop1"]
console.log(Object.getOwnPropertyNames(obj));    //output：Array[2] => [0:
"prop1", 1: "prop2"]
```

```
var obj = { "prop1": "v1" };
Object.defineProperty(obj, "prop2", { value: "v2", enumerable: false});

console.log(obj.hasOwnProperty("prop1")); //output: true
console.log(obj.hasOwnProperty("prop2")); //output: true

console.log(obj.propertyIsEnumerable("prop1")); //output: true
console.log(obj.propertyIsEnumerable("prop2")); //output: false

console.log('prop1' in obj);    //output: true
console.log('prop2' in obj);    //output: true

for (var item in obj) {
    console.log(item);
}
//output：prop1

for (var item in Object.getOwnPropertyNames(obj)) {
    console.log(Object.getOwnPropertyNames(obj)[item]);
}
//ouput:[prop1,prop2]
```

13.移动开发不同手机弹出数字键盘问题？

- type="tel"： iOS和Android的键盘表现都差不多；
- type="number"： 优点是Android下实现的一个真正的数字键盘；

缺点一：iOS下不是九宫格键盘，输入不方便
缺点二：旧版Android（包括微信所用的X5内核）在输入框后面会有超级鸡肋的小尾巴，好在Android 4.4.4以后给去掉了。不过对于缺点二，我们可以用webkit私有的伪元素给fix掉：

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

- pattern属性: pattern用于验证表单输入的内容，通常HTML5的type属性，比如email、tel、number、data类、url等，已经自带了简单的数据格式验证功能了，加上pattern后，前端部分的验证更加简单高效了。
  显而易见，pattern的属性值要用正则表达式。

```
<input type="number" pattern="d">
<input type="number" pattern="[0-9]*">
```

14.Javascript：history.go()和history.back()的用法与区别？

go(-1): 返回上一页，原页面表单中的内容会丢失；

back(): 返回上一页，原页表表单中的内容会保留。

history.go(-1): 后退+刷新

history.back(): 后退
Chrome和ff浏览器后退页面，会刷新后退的页面，若有数据请求也会提交数据申请。类似于history.go(-1);
而safari（包括桌面版和ipad版）的后退按钮则不会刷新页面，也不会提交数据申请。类似于javascript：history.back();

15.Meta基础知识

```
<meta name="viewport"content="width=device-width,initial-scale=1.0,minimum-scale=1.0,maximum-scale=1.0,user-scalable=no" />
// width    设置viewport宽度，为一个正整数，或字符串‘device-width’
// height   设置viewport高度，一般设置了宽度，会自动解析出高度，可以不用设置
// initial-scale    默认缩放比例，为一个数字，可以带小数
// minimum-scale    允许用户最小缩放比例，为一个数字，可以带小数
// maximum-scale    允许用户最大缩放比例，为一个数字，可以带小数
// user-scalable    是否允许手动缩放
<!-- 设置缩放 -->
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no, minimal-ui" />
<!-- 可隐藏地址栏，仅针对IOS的Safari（注：IOS7.0版本以后，safari上已看不到效果） -->
<meta name="apple-mobile-web-app-capable" content="yes" />
<!-- 仅针对IOS的Safari顶端状态条的样式（可选default/black/black-translucent ） -->
<meta name="apple-mobile-web-app-status-bar-style" content="black" />
<!-- IOS中禁用将数字识别为电话号码/忽略Android平台中对邮箱地址的识别 -->
<meta name="format-detection"content="telephone=no, email=no" />
其他meta标签
```

```
<!-- 启用360浏览器的极速模式(webkit) -->
<meta name="renderer" content="webkit">
<!-- 避免IE使用兼容模式 -->
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<!-- 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">
<!-- uc强制竖屏 -->
<meta name="screen-orientation" content="portrait">
<!-- QQ强制竖屏 -->
<meta name="x5-orientation" content="portrait">
<!-- UC强制全屏 -->
<meta name="full-screen" content="yes">
<!-- QQ强制全屏 -->
<meta name="x5-fullscreen" content="true">
<!-- UC应用模式 -->
<meta name="browsermode" content="application">
<!-- QQ应用模式 -->
<meta name="x5-page-mode" content="app">
<!-- windows phone 点击无高光 -->
<meta name="msapplication-tap-highlight" content="no">
```

16.移动端如何定义字体font-family？

@ 宋体 SimSun
@ 黑体 SimHei
@ 微信雅黑 Microsoft Yahei
@ 微软正黑体 Microsoft JhengHei
@ 新宋体 NSimSun
@ 新细明体 MingLiU
@ 细明体 MingLiU
@ 标楷体 DFKai-SB
@ 仿宋 FangSong
@ 楷体 KaiTi
@ 仿宋*GB2312 FangSong_GB2312*
@ *楷体*GB2312 KaiTi_GB2312
@ 说明：中文字体多数使用宋体、雅黑，英文用Helvetica

body { font-family: Microsoft Yahei,SimSun,Helvetica; }

17.打电话发短信写邮件怎么实现？

```
// 一、打电话
<a href="tel:0755-10086">打电话给:0755-10086</a>

//  二、发短信，winphone系统无效
<a href="sms:10086">发短信给: 10086</a>

// 三、写邮件
<a href="mailto:863139978@qq.com">点击我发邮件</a>
//2.收件地址后添加?cc=开头，可添加抄送地址（Android存在兼容问题）
<a href="mailto:863139978@qq.com?cc=zhangqian0406@yeah.net">点击我发邮件</a>
//3.跟着抄送地址后，写上&bcc=,可添加密件抄送地址（Android存在兼容问题）
```

```
<a href="mailto:863139978@qq.com?
cc=zhangqian0406@yeah.net&bcc=384900096@qq.com">点击我发邮件</a>
//4.包含多个收件人、抄送、密件抄送人，用分号(;)隔开多个邮件人的地址
<a href="mailto:863139978@qq.com;384900096@qq.com">点击我发邮件</a>
//5.包含主题，用?subject=
<a href="mailto:863139978@qq.com?subject=邮件主题">点击我发邮件</a>
//6.包含内容，用?body=;如内容包含文本，使用%0A给文本换行
<a href="mailto:863139978@qq.com?body=邮件主题内容%0A腾讯诚信%0A期待您的到来">点击我发
邮件</a>
//7.内容包含链接，含http(s)://等的文本自动转化为链接
<a href="mailto:863139978@qq.com?body=http://www.baidu.com">点击我发邮件</a>
//8.内容包含图片（PC不支持）
<a href="mailto:863139978@qq.com?body=<img src='images/1.jpg' />">点击我发邮件</a>
//9.完整示例
<a href="mailto:863139978@qq.com;384900096@qq.com?
cc=zhangqian0406@yeah.net&bcc=993233461@qq.com&subject=[邮件主题]&body=腾讯诚邀您参
与%0A%0Ahttp://www.baidu.com%0A%0A<img src='images/1.jpg' />">点击我发邮件</a>
```

18.移动端touch事件（区分webkit和winphone）？

- // 以下支持webkit
  touchstart——当手指触碰屏幕时候发生。不管当前有多少只手指
  touchmove——当手指在屏幕上滑动时连续触发。通常我们再滑屏页面，会调用event的preventDefault()可以阻止默认情况的发生：阻止页面滚动
  touchend——当手指离开屏幕时触发
  touchcancel——系统停止跟踪触摸时候会触发。例如在触摸过程中突然页面alert()一个提示框，此时会触发该事件，这个事件比较少用;

- //TouchEvent说明：
  touches：屏幕上所有手指的信息
  targetTouches：手指在目标区域的手指信息
  changedTouches：最近一次触发该事件的手指信息
  touchend时，touches与targetTouches信息会被删除，changedTouches保存的最后一次的信息，最好用于计算手指信息

- //参数信息(changedTouches[0])
  clientX、clientY在显示区的坐标
  target：当前元素

- //事件响应顺序
  ontouchstart > ontouchmove > ontouchend > onclick

19.点击元素产生背景或边框怎么去掉

```
//ios用户点击一个链接，会出现一个半透明灰色遮罩，如果想要禁用，可设置-webkit-tap-highlight-color的alpha值为0去除灰色半透明遮罩；
//android用户点击一个链接，会出现一个边框或者半透明灰色遮罩，不同生产商定义出来额效果不一样，可设置-webkit-tap-highlight-color的alpha值为0去除部分机器自带的效果；
//winphone系统,点击标签产生的灰色半透明背景，能通过设置<meta name="msapplication-tap-highlight" content="no">去掉；
//特殊说明：有些机型去除不了，如小米2。对于按钮类还有个办法，不使用a或者input标签，直接用div标签
a,button,input,textarea {
    -webkit-tap-highlight-color: rgba(0,0,0,0);
    -webkit-user-modify:read-write-plaintext-only; //-webkit-user-modify有个副作用，就是输入法不再能够输入多个字符
}
// 也可以
* { -webkit-tap-highlight-color: rgba(0,0,0,0); }
//winphone下
<meta name="msapplication-tap-highlight" content="no">
```

20.美化表单元素

- //一、使用appearance改变webkit浏览器的默认外观
  input,select { -webkit-appearance:none; appearance: none; }
- //二、winphone下，使用伪元素改变表单元素默认外观
  //1.禁用select默认箭头，::-ms-expand修改表单控件下拉箭头，设置隐藏并使用背景图片来修饰
  select::-ms-expand { display:none; }
- //2.禁用radio和checkbox默认样式，::-ms-check修改表单复选框或单选框默认图标，设置隐藏并使用背景图片来修饰
  input[type=radio]::-ms-check,
  input[type=checkbox]::-ms-check { display:none; }
- //3.禁用pc端表单输入框默认清除按钮，::-ms-clear修改清除按钮，设置隐藏并使用背景图片来修饰
  input[type=text]::-ms-clear,
  input[type=tel]::-ms-clear,
  input[type=number]::-ms-clear { display:none; }

21.移动端字体单位font-size选择px还是rem？

- // 如需适配多种移动设备，建议使用rem。以下为参考值：
  html { font-size: 62.5%; } //10*16 = 62.5%
- //设置12px字体 这里注意在rem前要加上对应的px值，解决不支持rem的浏览器的兼容问题，做到优雅降级
  body { font-size:12px; font-size:1.2rem; }

22.input标签添加上disable属性在ios端字体颜色不兼容的问题？

```
input[disabled],input:disabled,input.disabled{
    color: #3e3e3e;
    -webkit-text-fill-color: #3e3e3e;
    -webkit-opacity:1;
    opacity: 1;
}
```

23.IOS 的光标大小问题

IE：不管该行有没有文字，光标高度与font-size一致。
FF：该行有文字时，光标高度与font-size一致。该行无文字时，光标高度与input的height一致。
Chrome：该行无文字时，光标高度与line-height一致；该行有文字时，光标高度从input顶部到文字底部(这两种情况都是在有设定line-height的时候)，如果没有line-height，则是与font-size一致。
IOS中情况和Chrome 相似。:

设置字体大小和行高一致，然后通过 padding 撑开大小
只给IE浏览器设置 line-height
-ms-line-height:40px;