一、移动端（pc）样式异常，兼容问题。

1.meta 标签设置缩放 属性（听说只在安卓机上生效）

2.js强行禁止

1. placeholder文字位置调整方法，position:relative;top:1px;left:1px;
2. js判断手机设备类型:

var u = navigator.userAgent;

var isiOS = !!u.match(/\(i[^;]+;( U;)? CPU.+Mac OS X/);

5.当弹性盒元素内部的子元素设置了flex属性时，其他子元素可能出现缩放，这是需要设置shrik属性来禁止缩放。

二、移动端（pc）js控制问题。

1、

方法一：参考链接 https://www.bbsmax.com/A/ZOJPXqG2Jv/

方法二:原文链接：（实测有效）https://blog.csdn.net/supertree\_l/article/details/81532099

document.body.addEventListener('touchmove', function (e) {

if(!flag){

e.preventDefault();//阻止默认的处理方式(阻止下拉滑动的效果)

}

},{passive: false}); //用来兼容ios和android

1. addEventListener的第三个参数。可以是对象形式。例如{passive: false} 表示是否禁止执行preventDefault()方法。
2. EventTarget.dispatchEvent方法在当前节点上触发指定事件，从而触发监听函数的执行。该方法返回一个布尔值，只要有一个监听函数调用了Event.preventDefault()，则返回值为false，否则为true。

例如：para.addEventListener('click', hello, false);

var event = new Event('click');

para.dispatchEvent(event);

三、window.open 在回调中会被谷歌浏览器拦截 有几种办法:ajax请求改成同步的

四、chrome浏览器中，输入框显示之前的输入记录。

解决办法：给form，或者是input 加autocomplete="off"属性；

五、移动端a标签点击时出现背景颜色

解决办法：.credit-card-item a:active{

-webkit-tap-highlight-color:rgba(0,0,0,0);

-webkit-tap-highlight-color: transparent;

outline:none;

background: none;

text-decoration: none;

}

六、margin-top加载了父元素上

解决办法：.info-list::after{

content:'';

display: table;

}

七、混合开发总结点：5 + APP

（1）混合开发的一些效果要在真机或模拟器上才能查看。像创建的子页面。

（2）侧向滑动的页面，不能添加子页面。

（3）双webview模式的上拉加载和下拉刷新，对DOM结构有固定的要求见文档。否则，会出现子页面滑动遮盖主页面的标题栏问题。

（4）更改下拉刷新文字位置

\*可以解决修改下拉刷新子页面默认top值后,下拉刷新提示框位置异常问题

根据实际需求在父页面给mui-content设置top属性，目前仅支持 双 webview 模式

.mui-bar-nav ~ .mui-content .mui-pull-top-pocket{

top: 180px !important;

}

八、小程序开发总结：

（1）wx.navigate跳转时，通过在路径的后面拼接参数的方式实现页面之间的传值。

（2）onLoad的生命周期函数中获取路由传递的参数。

（3）表单输入框的事件，通过e.detail.value取值。

（4）在app.js中，onlaunch生命周期是异步的，想获取里面的数据可以采用promise的形式。网上有

（5）在app.js中，跳转页面wx.navigateTo注意路径的写法。网上有

（6）本地路径的背景图片是无法显示的，必须把图片上传到服务器，使用网络地址。图片展示用本地地址没问题。

（7）事件触发的方法传参通过e.currentTarget.dataset获取。

（8）一定要配置合法request域名而且是https协议的。否则，在手机上预览时，页面没有数据。在开发者工具调试正常。

（9）滚动组件scroll-view无法横向滑动，要求其内部元素是inline-block类型，并且设置white-space:no-spance。在其父元素上定位时，无法滑动，必须将定位属性加在scroll-view上。

（10）inneraudiocontext音频api 无法获取currenttime,duration。需要先写onplay（callback）监听，callback中要包含一个settimeout，并且获取写duration或currenttime；再写onTimeUpdate（callback）,再在其回调函数中获取。

（11）调用暂停pause接口后，再调用seek接口设置进度，此时onTimeUpdate无法继续监听。需要seek接口后，再调play(),然后在延时函数中访问以下innercontext.plused属性，onTimeUpdate就开始重新工作了。

（12）上面所说的音频实例，只能播放线上url形式的资源，本地无法播放。音频资源必须正确，否则，也获取不到duration总时长，当前播放进度currenttime。

（13）video 组件，updatetime 事件执行的前提是，绑定onplay事件。没有onplay事件updatetime就无法执行。Updatetime事件中可以获取currenttime和duration

九、react技术点总结

（1）生命周期的执行顺序：willMount(可以在这里改变state的里的数据，不会引起二次 渲染这也是他跟didmount的区别之一)---render---didMount

（2）函数的参数，e事件对象这个形参验放到最后。

（3）有些场景下，当改变状态的时候，并不希望它实时的更新页面的显示。例如表单输入完成时采取更新，输入过程中的数据是不希望更新页面的变化，这时可以用非受控组件

（4）React 元素本质就是对象（object），可以被当作属性或函数的参数传递。

十、mysql数据库总结

1、命令行操作相关

（1）进入安装目录下的bin,可以修改密码等操作。命令行以管理员身份运行，否则会提示错误。

（2）net start mysql 启动数据库服务。

（1）进入安装目录下的bin,可以修改密码等操作。命令行以管理员身份运行，否则会提示错误。

（2）net start mysql 启动数据库服务。

drop user XXX;删除已存在的用户，默认删除的是'XXX'@'%'这个用户，如果还有其他的用户如'XXX'@'localhost'等，不会一起被删除。如果要删除'XXX'@'localhost'，使用drop删除时需要加上host即 drop user 'XXX'@'localhost'

delete from user where user='XXX' and host='localhost';其中XXX为用户名，localhost为主机名。

区别：drop不仅会将user表中的数据删除，还会删除其他权限表的内容。而delete只删除user表中的内容，所以使用delete删除用户后需要执行FLUSH PRIVILEGES;刷新权限，否则下次使用create语句创建用户时会报错。

（3）mysql -h远程主机地址 -u 用户名 -p 密码 -P端口。登录数据库（非默认端口启动一定得加-P参数）。

（4）set password for user@host = password('123')。修改用户密码（超级用户才有权限）。

（5）grant rights on database.\* to user@host identified by "pass"。对用户授权。

（6）quit 或 exit 退出登录

（7）net stop mysql 停止数据库服务

十一、layUI前端框架总结

（1）layUI自带jQuery,无需单独引入jquery。否则，会因冲突报错。

（2）

十二、诸葛io使用流程

（1）注册用户，登录。根据诸葛io的技术标准配置埋点事件。

（2）按照要求引入对应的js-sdk。并进行相关配置。

（3）autoTrack: false, //启用全埋点采集（选填，默认false）

singlePage: false, //是否是单页面应用（SPA），启用autoTrack后生效（选填，默认false）

debug:true //实时调试的时候开启，方便检查埋点效果。调试结束后应立即关闭，防止产生大量无用数据。

（4）对用户，及用户数的统计，前提是用户有相关操作才行。

十三、vue使用注意事项

1、v-for 循环组件时，更改循环数据时，组件数据不刷新。

原因：实际上新数据已经引起了组件的view层的数据更新，但是组件内部的周期函数没有被重新执行。所以，数据通信层面的数据无法动态改变。

解决：改变组件上面的key，组件就不会被复用，周期函数都会重新执行。

1. npm常用指令
2. npm view 包名 versions : 查看npm服务器上该包的所有版本号。
3. npm view 包名 version : 查看npm 服务器上该包的最新版本号。
4. npm ls 包名 : 查看本地局部安装的包版本号。
5. npm ls 包名 -g : 查看全局安装的包版本号。

十五、git常用指令

1、git stash save “message”

十六、前端性能优化

1、网络请求

2、JavaScript缓存

**以Apache为例，可在Apache的配置文件httpd.conf中设置Expires。**

（1）浏览器缓存

浏览器缓存是根据第一次请求时，响应数据对请求头的设置来确定的。通常分为两种，强缓存和协商缓存，并且缓存策略是通过http header设置的。

从位置角度来说，缓存的位置有四种。Service Worker、memory cache（内存缓存）、disk cache（磁盘缓存）、push cache。其中，大部分缓存都来自硬盘缓存。

1. http强缓存和协商缓存

强缓存：服务器返回的响应头中设置Cache-Control和Expiress表示启用强缓存。

协商缓存：通过设置响应头中的Etag和Last-Modified启用协商缓存

1. 浏览器渲染解析机制优化

生成dom树—生成css规则树—生成渲染树—根据渲染树进行布局（回流、自动重排）

1. 更改dom几何属性，触发重排，会更新完整的渲染流水线。开销最大

Css的transform动画可以避开重排和重绘阶段开销最小