## 小程序问题统计

###### 在给小程序配置tabbar的时候，在所有的配置路径都没有写错的情况下，tabbar不显示。

如果将tabBar中list中的第一个路径参数改为和pages中第一个路径相同，tabbar就会正确显示了。

###### 微信小程序引入iconfont.wxss后报错渲染层网路层错误

在icon网站找到在线链接 @font-face 替换本地iconfont.wxss 修复

###### 微信小程序需要自定义导航栏

1、隐藏原生的navigationBar "window": {    "navigationStyle": "custom" }

2、通过胶囊按钮获取布局位置信息 wx.getMenuButtonBoundingClientRect();

3、完成自定义导航栏进行布局设置

###### 微信小程序rpx和px的关系

rpx（responsive pixel）: 可以根据屏幕宽度进行自适应。规定屏幕宽为750rpx。

如在 iPhone6 上，屏幕宽度为375px，共有750个物理像素，则375px = 750rpx = 750物理像素，1px = 2rpx = 2物理像素。

###### 导航栏要进行图片渐变混合展示，展示内容需要覆盖在导航栏的背景上并且默认背景色不是白色

导航栏图片渐变共同展示使用background的连写 图片要放在渐变前面

background: url('https://resource.boka.vc/MCP/staff/nav-bg.png'), linear-gradient(180deg, #9B90FF 0%, #6257FE 100%);

图片必须使用网络图片？

内容覆盖在导航背景图片上需要创建body使用margin-top距离顶部一个导航栏的高度，自身高度设定为100% - 导航栏高度 menuTop 为胶囊按钮top获取值

style="margin-top:{{ menuTop + 160}}px;height: calc(100% - {{ menuTop + 160}}px);"

在body的内容超长要进行滚动时 上面的高度设定会导致超出body高度的会展示白色背景

Body内的内容使用scroll-view并且背景色使用background linear-gradient 双色背景

background: linear-gradient(transparent 15.4%, rgba(235, 238, 242, 1) 0);

由于内容可能小于body的展示高度，所以上面的高度设定不能去除

###### 在wx:for 中使用slot只会被渲染第一次问题原因和解决方法

原因：指定的位置贴一块固定代码，是一个template引用

解决方法： 使用抽象组件实现

**支付宝小程序中使用ref问题 不止直接对组件声明ref并调用组件内部的数据和方法**

解决：

1. 在自定义组件内声明 ref函数 ref() { return this }
2. 在使用自定义组件的地方进行ref声明 <my-component ref=”ccc” />
3. 在使用自定义组件文件的js中保留对象

Ccc(ref) { this.$refs = Object.assign(this.$refs || {}, ref ) }

**支付宝小程序中全局缓存页面数据以达到跨页面传参**

解决：

1、定义class类

export default class PM {

    constructor() {

        this.$$cache = {};

    }

    add(pageModel) {

        let pagePath = this.\_getPageModelPath(pageModel);

        this.$$cache[pagePath] = pageModel;

    }

    get(pagePath) {

        return this.$$cache[pagePath];

    }

    delete(pageModel) {

        try {

            delete this.$$cache[this.\_getPageModelPath(pageModel)];

        } catch (e) {}

    }

    \_getPageModelPath(page) {

        return page.\_\_proto\_\_.route;

    }

}

1. 在app.js中进行引入并声明

This.pages = new pages();

3、在对应页面进行page缓存

const app = getApp(); app.pages.add(this);

4、可以在任意页面使用 app.pages.get(‘pages/a/a’)