2020.12.01

1.页面跳转出现报错时:

(VM485:1 MiniProgramError \_vm.navTo is not a function Error: \_vm.navTo is not a function)

VM485:1微程序错误 \_虚拟机导航不是函数 错误：\_虚拟机导航不是函数

解决方法:找到methods:{}方法里中我们定义的.navTo方法,并将其打开.

2.定义一个分享的方法:写在methods里面

//分享

share(){

this.$refs.share.toggleMask();

},

3.定义一个分享的方法:写在methods 里

//收藏

toFavorite(){

this.favorite = !this.favorite;

},

4.绝对定位:position:filed(覆盖在原来文本流之上,可以用 left bottom right top)

5.弹性布局:display: flex;

<https://www.cnblogs.com/hellocd/p/10443237.html>

1. 相对定位:

**定义和用法**

position 属性规定元素的定位类型。

**说明**

这个属性定义建立元素布局所用的定位机制。任何元素都可以定位，不过绝对或固定元素会生成一个块级框，而不论该元素本身是什么类型。相对定位元素会相对于它在正常流中的默认位置偏移。



2020.12.03

前两天我的上级交给我一个任务，就是写一篇《上杭应急管理学习平台》后台的使用手册说明，任务看似简单，但是我仍然花了一天的时间才完成这项任务。我在编辑使用手册时也遇到了一些问题，通过领导的讲解和帮助，我大概了解了一些，虽然没有写过平台系统的使用手册说明，但是我再通过百度搜索，文案的参考，大概对使用说明手册的编写有了一定了解，再结合我们这个上杭应急管理学习平台项目后台的基本功能模块，自己尝试使用一遍，大概了解这个后台是怎么使用的，于是我就开始写使用手册说明，这个说明是要让用户看的懂，会使用的，因此，编写使用手册时一定要亲身使用过这个系统才能写出更好的使用手册，让用户达到一看就懂，一操作就能上手的效果。今天我又有了一定的小收获，很开心！

2020.12.08（9:50）

我在12月3日完成了《上杭应急管理学习平台》后台使用手册说明，上周五下午我的上级又交给我一个任务，这次的任务和上次大同小异，后台的使用手册完成了，现在就应该做桌面APP的使用手册说明了。因此，这次就是完成一篇《安全学习--上杭应急APP的下载安装使用说明手册》，有了上次写使用说明手册的经验，我觉得这次会比较容易一些。这么想着，我就开始了第二篇软件使用手册的编写。开始时，我还是有些疑惑的，于是我就请教我的上级，通过他的讲解，我明白了虽然这次的使用手册说明和上一次的不会有太大改变，但是，软件应用与后台管理还是有一定差别的，一是：后台是由专门有权限的管理员所管理的，权限管理方面、后台操作使用专门是针对少部分特定的人，因此不会太过复杂。而软件应用APP则是针对大多数不同身份的用户使用的，因此对不同的用户要有不同的说明，所以需要更加注重对象的分类，使用说明需要更加详细一些，让每个角色每个用户都能清楚的了解到自己具有哪些操作权限，让用户们理解起来更快，使用起来更加快捷方便！同样的，编写软件应用app的使用手册说明，也需要亲身体验过本款APP才能更好的写出使用手册，让用户达到一看就懂，一操作就能上手的效果。这次使用手册我花了一天半的时间，速度不算太快，也还需继续提升。

12月7日的上午由于公司又接到了另一个项目，时间比较紧急，需要赶一个系统开发进度表。它是一款不同于微信小程序但有与其类似的产品项目，是一款基于微信公众号的微信促销系统。我进公司以来由于只跟过一个项目的开发，而且我的技术并不是太理想，同时没有做过系统开发进度表，这对我来说又是一个新任务啦！我通过该项目详细的“系统整体框架解决方案”结合百度搜索类似做系统开发进度表的文献作参考，最后结合上级给我大体的讲解了该系统整体框架解决方案后，才开始着手。在上级的建议下，首先我把“系统整体框架解决方案”内的重点功能模块、重点业务逻辑（包括了身份改变的判断逻辑、活动逻辑等）打印出来，以便更容易查看理解。在做报表的时候，最重要的就是要能够很好的理解客户的各种需求、他需要我们实现的功能模块有哪些、理解系统框架的解决方案、理解所需业务的逻辑图等等。我在今天下班前把这份进度表完成了，并发给了我的上级。今天还加了半个小时的班把上一个app的使用手册说明完成了，也是充实满满的一天。

2020.12.08（16:30）

我的GitHub账号：①用户名 yuechu0325 ②电子邮箱 [wulifairyzhong@foxmail.com](mailto:wulifairyzhong@foxmail.com)

③密码 zzx2017036543

## **Git 与 SVN 区别**

Git 不仅仅是个版本控制系统，它也是个内容管理系统(CMS)，工作管理系统等。

如果你是一个具有使用 SVN 背景的人，你需要做一定的思想转换，来适应 Git 提供的一些概念和特征。

**Git 与 SVN 区别点：**

1、Git 是分布式的，SVN 不是：这是 Git 和其它非分布式的版本控制系统，例如 SVN，CVS 等，最核心的区别。

2、Git 把内容按元数据方式存储，而 SVN 是按文件：所有的资源控制系统都是把文件的元信息隐藏在一个类似 .svn、.cvs 等的文件夹里。

3、Git 分支和 SVN 的分支不同：分支在 SVN 中一点都不特别，其实它就是版本库中的另外一个目录。

4、Git 没有一个全局的版本号，而 SVN 有：目前为止这是跟 SVN 相比 Git 缺少的最大的一个特征。

5、Git 的内容完整性要优于 SVN：Git 的内容存储使用的是 SHA-1 哈希算法。这能确保代码内容的完整性，确保在遇到磁盘故障和网络问题时降低对版本库的破坏。

