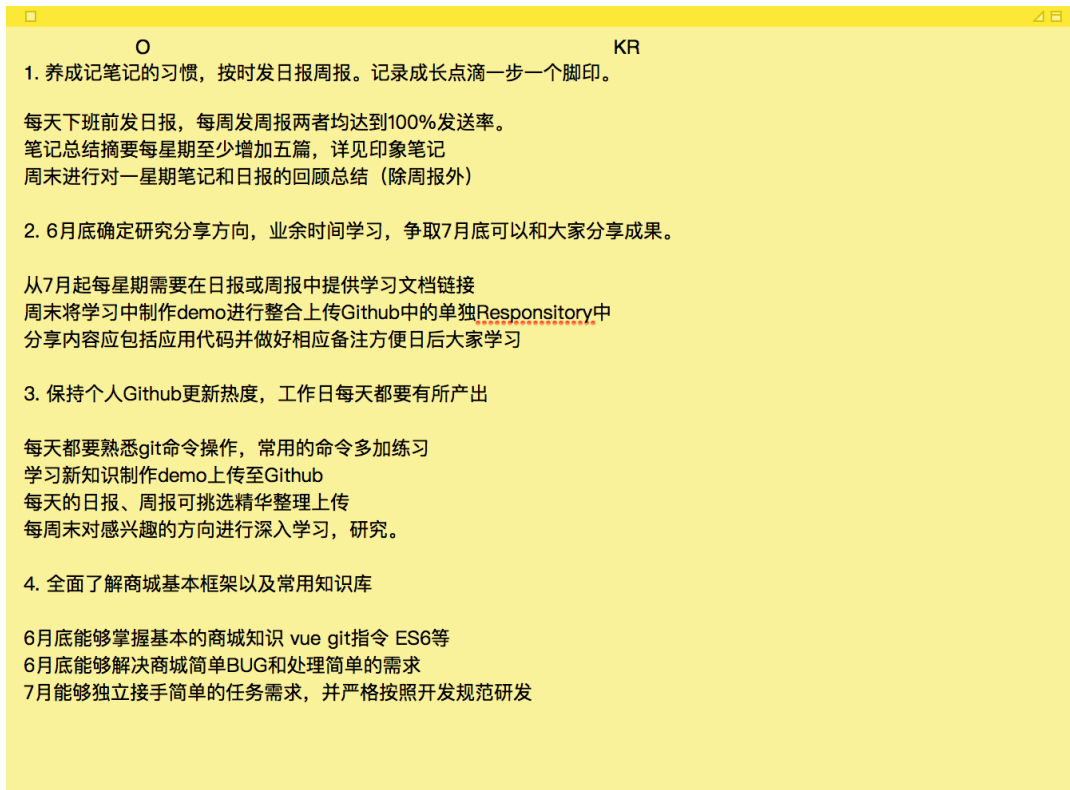


## 学习记录

### 1. 完成个人OKR制作



**O** **KR**

1. 养成记笔记的习惯，按时发日报周报。记录成长点滴一步一个脚印。

每天下班前发日报，每周发周报两者均达到100%发送率。  
笔记总结摘要每星期至少增加五篇，详见印象笔记  
周末进行对一星期笔记和日报的回顾总结（除周报外）

2. 6月底确定研究分享方向，业余时间学习，争取7月底可以和大家分享成果。

从7月起每星期需要在日报或周报中提供学习文档链接  
周末将学习中制作demo进行整合上传Github中的单独Repository中  
分享内容应包括应用代码并做好相应备注方便日后大家学习

3. 保持个人Github更新热度，工作日每天都要有所产出

每天都要熟悉git命令操作，常用的命令多加练习  
学习新知识制作demo上传至Github  
每天的日报、周报可挑选精华整理上传  
每周末对感兴趣的方向进行深入学习，研究。

4. 全面了解商城基本框架以及常用知识库

6月底能够掌握基本的商城知识 vue git指令 ES6等  
6月底能够解决商城简单BUG和处理简单的需求  
7月能够独立接手简单的任务需求，并严格按照开发规范研发

### 2. 学习item2的快捷设置及使用方法

快捷键：

- 查找: **Cmd+f**
- 命令历史: **Cmd+Shift+H**
- 命令回放: **Cmd+Alt+B**
- iTerm2 Expose: **Cmd+ Alt+E**
- **Ctrl+a**: 光标移动到行首
- **Ctrl+e**: 光标移动到行末
- **Ctrl+f**: 前移一个字符
- **Ctrl+b**: 后退一个字符
- **Ctrl+l**: 清屏
- **Ctrl+p**: 显示历史命令
- **Ctrl+r**: 倒转查找
- **Ctrl+d**: 删除当前字符
- **Ctrl+h**: 删除前一字符

- Ctrl+w: 删除光标之前的字符
- Ctrl+k: 删除光标之后整行
- Cmd + d 垂直分割
- Cmd + shift + d 水平分割

3. git 命令中进行版本回退命令 以及返回新版本命令

git log 查看版本号

git log --pretty=oneline 更直观清楚查看版本号

git reset commit id 版本切换跳转

git revert commit id 版本回退（不同于git reset，他不会擦除回退之后的commit id,而是当做一次commit产生一个新的操作记录）

git reflog 查看命令记录

4. git中出现的四种文件状态

### git库所在的文件夹中的文件大致有4种状态

- **Untracked:** 未跟踪, 此文件在文件夹中, 但并没有加入到git库, 不参与版本控制. 通过 `git add` 状态变为 `Staged`.
- **Unmodify:** 文件已经入库, 未修改, 即版本库中的文件快照内容与文件夹中完全一致. 这种类型的文件有两种去处, 如果它被修改, 而变为 `Modified`. 如果使用 `git rm` 移出版本库, 则成为 `Untracked` 文件
- **Modified:** 文件已修改, 仅仅是修改, 并没有进行其他的操作. 这个文件也有两个去处, 通过 `git add` 可进入暂存 `staged` 状态, 使用 `git checkout` 则丢弃修改过, 返回到 `unmodify` 状态, 这个 `git checkout` 即从库中取出文件, 覆盖当前修改
- **Staged:** 暂存状态. 执行 `git commit` 则将修改同步到库中, 这时库中的文件和本地文件又变为一致, 文件为 `Unmodify` 状态. 执行 `git reset HEAD filename` 取消暂存, 文件状态为 `Modified`

5. webpack 中的dependencies 和 devDependencies的区别

前者是整个项目最基础能正常运行的依赖, 而后者是开发人员对整个项目所需要的全部依赖

npm install 默认装两者

npm install --production 只安装dependencies

npm install package-name --dev 只安装devDependencies

已解决

1. git 如何将缓冲区和工作区相互传递?

git status 查看文件状态

git reset HEAD <file> 将file文件从缓冲区拿回工作区

git checkout . 已经再缓存区保有原文件，撤回工作区的错误操作

git checkout -- <file>已经再缓存区保有原文件，撤回工作区的错误操作

2. git 中的modified状态是 add之后有追踪关系对文件进行更改但未提交 这是可以进行撤销操作 也就是git checkout . 但是没有追踪关系也就是缓冲区没有该文件将会提示下面的错误

```
meicais-MacBook-Pro-2:Git yunshan$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    add/add6的副本 2.txt

nothing added to commit but untracked files present (use "git add" to track)
meicais-MacBook-Pro-2:Git yunshan$ git checkout .
meicais-MacBook-Pro-2:Git yunshan$ git checkout -- add/add6的副本 2.txt
error: pathspec 'add/add6的副本' did not match any file(s) known to git.
error: pathspec '2.txt' did not match any file(s) known to git.
meicais-MacBook-Pro-2:Git yunshan$
```

待解决