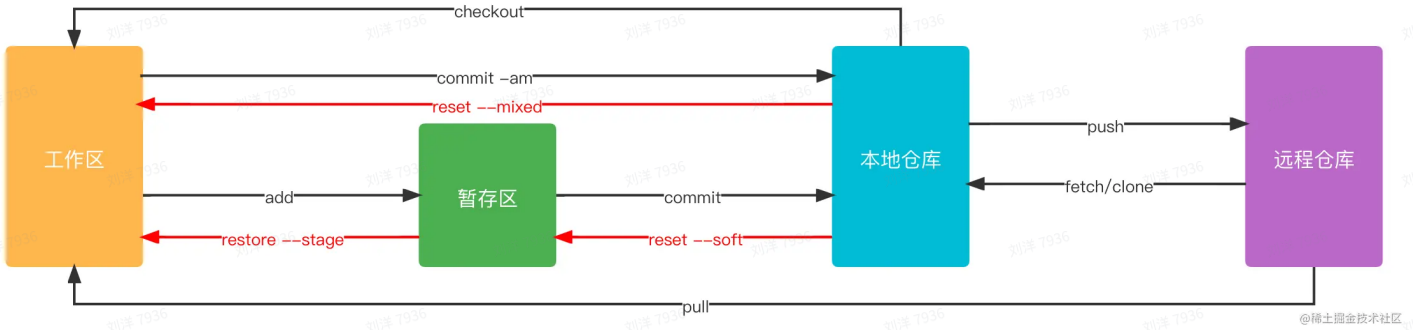
git常用命令

git工作区概念



常用命令

1. git init 初始化一个git仓库
2. git remote： 将本地仓库和远程仓库关联

```
//关联本地 git init 到远程仓库
git remote add origin <git url>

//移除与远程仓库关联
git remote remove <git url>

//查看所有关联别名
git remote

//修改推送源
git remote set-url origin <git url>
```

3. git status： 查看工作区状态

```
//查看当前工作区暂存
git status

//查看当前工作区暂存以概述形式
git status -s
```

4. git log/reflog：查看commit日志

```
//log 详细日志 reflog简单日志

//最近n次commit日志
git log -n

//以简要形式查看日志
git log --online
```

5. git branch：管理分支

```
//查看所有分支
git branch -v

//新建分支
git branch name

//删除分支
git branch -D name

//重命名分支
git branch -M name1 name2

//将本地分支与远程分支关联
git branch --set-upstream-to=origin/xxx

//将本地分支与远程分支取消关联
git branch --unset-upstream-to=origin/xxx
```

6. git dfif：对比差异

```
//当工作区有变动和暂存区都有变动，对比工作区和暂存区的代码差异
//当工作区有变动，暂存区无变动，对比工作区和本地仓库间的代码差异
git diff

//显示暂存区和本地仓库的代码差异
git diff --cached

//显示两个分支之间的差异
git diff branch1 branch2
```

7. git merge：合并代码

```
//将本地某分支合并至当前分支
git merge branch

//将远程分支合并至当前分支
git merge origin/branch branch
```

8. git clone：克隆仓库

```
//克隆仓库到本地
git clone <git url>

//克隆仓库到本地并将本地仓库的文件名称为myproject
git clone <git url> myproject

//克隆仓库到本地并切换到branch分支
git clone <git url> -b branch
```

9. git add：提交到暂存区

```
// 提交所有未追踪文件都暂存区
git add .

//提交filename到暂存区
git add filename

//提交js目录下的
git add ./js
```

10. git commit：提交暂存区代码到本地

```
//添加提交信息
git commit -m ''

//对上一次提交记录覆盖
git commit -m '' --amend

//跳过commit编写
git commit --amend --no-edit
```

```
9
10 //一次性提交 相当于git add + git commit
11 git commit -am ''
12
13 //跳过husky校验
14 git commit --no-verify -m ''
```

11. git push：提交到远程仓库

```
1 //强制提交
2 git push -f
3
4 //提交到远程仓库
5 git push origin branch
6
7 //提交到远程仓库建立跟踪
8 git push --set-stream origin banch
```

12. git pull：拉取远程仓库代码并合并

```
1 //相当于git fetch + git merge
2 git pull origin branch
3
4 //使用rebase模式合并
5 git pull --rebase
```

13. git checkout：切换分支

```
1 //切换到本地已有分支
2 git checkout branch
3
4 //切换到远程已有分支
5 git checkout origin/branch
6
7 //基于远程已有分支新建分支
8 git checkout origin/branch -b branch
9
10 //新建分支并切换
11 git checkout -b branch
12
13 //撤销工作区文件更改【谨慎操作】
```

```
git checkout .
git checkout -- <file>
```

14. git reset：撤销操作

```
//撤销暂存区更改
git reset .
git reset --cached .

// 撤销commmit 取消commit 并修改文件内容
git reset --hard <commit_sha>

//撤销commit 取消commit操作 不取消add操作 不改变文件内容
git reset --soft <commit_sha>

//撤销commit 取消commit\add操作 不改变文件内容
git reset --mixed <commmit_sha>
```

15. git cherry-pick：合并指定commit

```
//将commit_sha的变动合并至当前分支
git cherry-pick <commit_sha>

//将多次 commit 变动合并至当前分支
git cherry-pick commit-sha1 commit-sha2

//将 commit-sha1 到 commit-sha5 中间所有变动合并至当前分支，中间使用..
git cherry-pick commit-sha1..commit-sha5
```

16. git stash：缓存代码

```
//把本地的改动缓存起来
git stash

//缓存代码时添加备注，便于查找
git stash save "message"

//查看缓存记录
git stash list

//取出上一次缓存的代码，并删除这次缓存
```

```
git stash pop

//取出 index 为2缓存代码，并删除这次缓存，index 为对应 git stash list 所列出来的
git stash pop stash@{2}

//取出上一次缓存的代码，但不删除这次缓存
stash apply

//取出 index 为2缓存代码，但不删除缓存
git stash apply stash@{2}

//清除某次的缓存
git stash drop stash@{n}

//清除所有缓存
git stash clear
```

17. git revert：取消某次commit内容

```
//取消某次 commit 内容，但是保留 commit 记录
git revert <commit-sha>
```

18. git rebase 简介commit记录https://www.jianshu.com/p/4a8f4af4e803
19. git tag：打标签https://www.cnblogs.com/zhilu/p/14082019.html

```
//添加标签
git tag -a v0.0.1 -m ''

//查看所有标签
git tag

//查看标签的详细
git show/log v0.0.1

//删除标签
git tag -d v0.0.1

//推送标签至远程仓库
git push origin v0.0.1
git push origin --tags

//查看远程所有标签
git ls-remote --tags origin
```

```
19
20 //切换标签
21 git checkout tagname
```

20. 常见操作
    a. --abort 取消所有动作，并恢复之前
    b. --quit 退出 但是会合并不冲突的代码
    c. --continue 解决完冲突之后继续命令操作
    d. --skip 跳过某个动作

常见问题

1. 线上发布可以基于branch、tag；tag更符合实际
2. 账号上云代码合并问题
    a. Q：开发的时候就已经错误了，导致了合并的错误，开发拉取分支是从dev分支拉取的，dev分支有很多正在开发的新功能，从dev合并到master，会出现很多错误，并且不能确保线上正常运行
    b. A：线上的分支的代码不动 撤销commit到拉取的分支的commit记录 git reset --mixed <commit_sha> ==> git stash save 缓存代码 ==> git checkout master git checkout -b branch 切换到master新建分支 ==> git stash apply 弹出缓存代码 ==> git commit -am ' ' 提交缓存代码 git push origin 发布分支 ==> 合并
    c. 基于master拉取分支
3. git merge！！！
    a. 自己的分支只能合并master
    b. 如果合并了其他的分支，git reset --hard. git push -f

借鉴学习：https://gist.github.com/guweigang/9848271