

Git 记录

远程分支

```
1
2 #查看远程分支列表
3 git branch -r
4
5 # 删除远程分支
6 git branch -r -d origin/feature/test
7 # or
8 git push origin --delete feature/test
9 # or
10 git push origin :feature/test
11
12 #查看远程分支列表
13 git branch -r
14
15 #更新远程分支列表
16
17 git remote update origin -p
18
19
20
21
```

比较差异

```
1 #####
2 #   commit 比较   #
3 #####
4 # 1.查看 commit 日志
5   git log
6 # 2. 比较
7   git diff c3408752 c8b6e845 --name-only
8 # 3. 比较并打tar包
9   git diff c3408752 c8b6e845 --name-only | xargs tar -czvf update.tar.gz
10
11 #####
12 #   branch 比较   #
13 #####
14
15 # 1. 查看本地分支
16   git branch
17 # 2. 查看远程分支
18   git branch -r
```

```
19 # 3. 比较本地分支与远程分支
20 git diff develop origin/feature/abilitymap --name-only
21
22 ## 查看当前分支改过的文件
23 git merge-base develop feature/abilitymap --name-only
24
```

```
1 git add 后撤销:
2 撤销所有add文件 git reset HEAD .
3 撤销单个add文件 git reset HEAD -filename
```

```
1 git commit 后撤销:
2 只回退commit的信息, 保留修改代码: git reset --soft head
3 彻底回退到上次commit版本, 不保留修改代码: git reset --hard head^
4 说明:
5 HEAD : 当前版本
6 HEAD^ : 上一个版本
7 --hard 参数会抛弃当前工作区的修改
8 --soft 参数的话会回退到之前的版本, 但是保留当前工作区的修改, 可以重新提交
```

```
1 撤销所有本地改动代码:
2 git checkout .
```

```
1 本地代码回退到与git远程仓库保持一致
2 git reset --hard 远程分支名
```

```
1 git push撤销
2 回滚此次push到服务器的代码:
3 git log查看commit的信息
4 git revert 以前commit的id
5 git push 此时本地回滚的代码到服务器就可以了
```

```
git merge 撤销
$ git checkout 【行merge操作时所在的分支】
$ git reset --hard 【merge前的版本号】
```

Git统计

查看username的代码量

```
1 git log --author="username" --pretty=tformat: --numstat | awk '{ add += $1; subs += $2; loc
  += $1 - $2 } END { printf "added lines: %s, removed lines: %s, total lines: %s\n", add,
  subs, loc }' -
```

统计每个人的增删行数

```
1 git log --format='%aN' | sort -u | while read name; do echo -en "$name\t"; git log --author="$name" --pretty=tformat: --numstat | awk '{ add += $1; subs += $2; loc += $1 - $2 } END { printf "added lines: %s, removed lines: %s, total lines: %s\n", add, subs, loc }' -; done
```

查看仓库提交者排名前 5

```
1 git log --pretty='%aN' | sort | uniq -c | sort -k1 -n -r | head -n 5
```

贡献者统计

```
1 git log --pretty='%aN' | sort -u | wc -l
```

提交次数统计

```
1 git log --oneline | wc -l
```