git使用命令记录

1 生成密匙：

ssh-keygen-trsa几次回车，在.ssh/下的id_rsa.pub即为密匙文件；

```
ssh-keygen -t rsa -C "XXXX@huawei.com"
```

2 获取 git 分支：

A.获取仓库：

git clone git@10.110.143.100:mycloud/cc.git

B.查看所获取分支的路径：

git remote -v

C.切换到某个分支：

git checkout $branchname

```
例如 git checkout pgh
```

3 分支的新建与删除：

3.1 新建：

git branch $branchname（注意在哪个分支上执行就是基于哪个分支新建）

```
例如   git branch pgh     #新建 pgh 分支
```

git push origin $branchname（推到服务器仓库）

```
例如 git push origin pgh
```

3.2 删除：

git branch –D $branchname（删除本地的分支）

```
例如 git branch –D pgh
```

git branch –rd origin/$branchname（删除服务器仓库分支）

例如

git push origin :$branchname（注意冒号）

例如

git remote prune origin（同步远端已删除分支）

例如

4 修改内容查看及提交：

4.1 查看未提交的修改：

git status/git status.（查看修改的文件）

例如 `git status`

git diff/git diff.（查看修改的内容）

例如 `git diff`

4.2 查看已提交修改：

git log（查看提交信息）

例如 `git log`

git whatchanged（查看每个提交修改的文件）

例如 `git whatchanged`

git diff $2 $1（查看莫个"提交ID"$1的修改内容）

例如 未验证

4.3 还原被修改文件：

Git checkout –f */$fileanme

例如 `git checkout –f huojian.txt`

4.4 提交修改：

git add */$filename（将新建文件加入仓库）

例如 `git add test.txt`

git commit */$filename –m "***" (提交一个修改)

例如 `git commit test.txt -m "备注:一个修改"`

/git commit –a –m ""（提交当前所有修改，删除一个文件也可以）

例如 `git commit –a –m “备注:全部修改”`

Git push origin $branchname

例如 `git push origin master`

4.5 还原到某个提交 ID 前：

git reset“$提交ID”（注意避免冲突：如果本地有修改过即将还原的文件，可以先备份it，然后git checkout –f $it）

git push origin $branchname—force

git pull origin $branchname

4.6 提取同一仓库不同分支的修改：

git cherry-pick “$提交ID”

例如

4.7 解决冲突：

Git diff.

vi*/$冲突文件（注意去掉---->HEAD之类的冲突提示）

git add */$冲突文件

git commit -c "产生冲突的$提交ID"

git push origin $branchname

例如

4.8 清除当前所有修改：

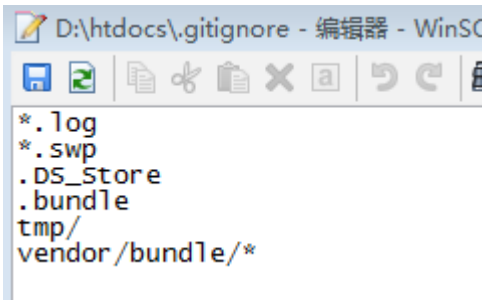
gitcheckout-f&&gitclean-df

例如

5 忽略一些不需要管理的文件/类型：

vim .gitignore

例如 在主目录中加入.gitignore 文件 然后在里面加入你不想管理的文件路径就 ok 了



6 .一些常见的异常处理：

6.1 分支在别处又提交，本地提交不了：

gitpullorigin$branchname然后再提交；

例如

6.2 新建仓库时，遇到漏提交文件：

find.-name".git*"|xargsrm-rf{}

例如

7 打 git patch：

patch -p1<$patchfile.patch（用过可以，但是不能新建.a/.so等库文件）或者git apply $patchfile

git branch 显示本地分支列表

git branch <branchname> 指向某个分支

git branch –d 删除分支

git checkout –b branchname 切换到新分支

git branch *号标识当前在哪个分支