

Git使用

1. 创建版本库

```
$ mkdir learngit//创建一个叫learngit的空目录
```

```
$ cd learngit
```

```
$ pwd//用于显示当前目录
```

2. 将目录变成Git可管理的仓库

```
$ git init
```

3. 将文件添加到版本库

- 将文件添加到对应目录下（如learngit目录下）
- `git add filename.kind` //如`git add readme.txt`，每次修改后如果不add到缓存区，就不会加入到commit中
- `git commit -m "对改动的说明"` //将文件提交到仓库，并加以说明，双引号不可忽略
- add可以加很多个文件，如`git add f1.txt f3.txt`
- commit可以一次提交很多个文件，只要在add下面commit就行。

4. 掌握工作区状态

使用 `git status` 命令，可看到文件是否被修改过；

使用 `git diff` 命令，可以查看修改内容

5. 查看过往记录

使用 `git log` 命令，可看到从最近到最远的提交日志，所以commit的时候，说明要写好

6. 返回文件的上一个版本

使用 `git reset` 命令，回到上一个版本，其中参数问题，在Git中，`HEAD` 表示当前版本，上一个版本为 `HEAD^`，上个版本为 `HEAD^^`，也可以用 `HEAD~10` 表示往上的10个版本。

完整例子：`git reset --hard HEAD^`

7. 重返未来

使用 `git reflog` 命令，查看命令历史，找到要返回的那个commit id,然后利用 `git reset --hard commit_id(such as 12345342)` 即可。

8. 撤销修改

- 直接丢弃工作区的修改
命令 `git checkout -- readme.txt` 将readrme.txt文件在工作区的修改全部撤销，让文件回到最近一次commit或者add时的状态。这个是在commit当前修改状态之前可以执行的撤销工作。
- 丢弃已经添加到暂存区的修改

第一步, `git reset HEAD file`

第二步, 重复上述那个checkout, 丢弃掉工作区的修改。

- 已经commit了的

按上面6

9. 删除文件

命令 `$ rm test.txt` 在文件管理器中删除掉文件test.txt

命令 `git rm test.txt`、`git commit` 在版本库中删除文件并提交改动

删除后, 可利用 `git checkout -- test.txt` 实现回退, 但是只能回退到最新版本, 会丢失到最近一次提交后修改的内容。

10. 本地库上传东西到远程库

- 初始化本地库 `init`, 然后 `add` 并 `commit`
- 命令 `git remote add 远程库名字 远程库https网址`, 关联远程库。
- `git pull 远程库名字 master` 后, `git push -u 远程库名字 master`。注: 如果是第一次push, 要使用格式-u, 反之不用, 直接 `git push 远程库名字 master`

11. 从远程库克隆到本地库

`clone`

12. 分支

查看分支: `git branch`

创建分支: `git branch <name>`

切换分支: `git checkout <name>`

创建+切换分支: `git checkout -b <name>`

合并某分支到当前分支: `git merge <name>`

删除分支: `git branch -d <name>`

小Tips

- <http://www.cnblogs.com/lulubai/p/6001334.html>