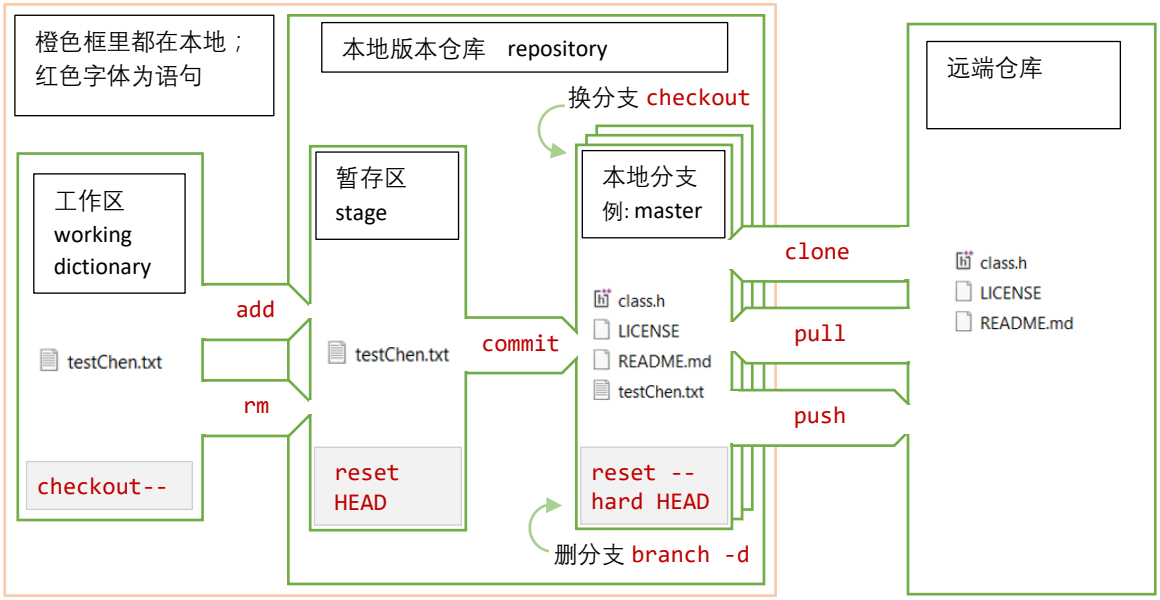


前提：

下文所有操作前先 cd 到工作区所在地址：

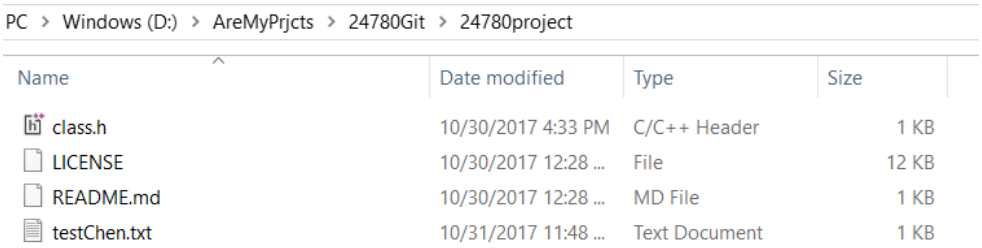
```
thuhu@Chens-Thinkpad MINGW64 ~
$ pwd
/c/Users/thuhu

thuhu@Chens-Thinkpad MINGW64 ~
$ cd ../../../

thuhu@Chens-Thinkpad MINGW64 /
$ cd d/AreMyPrjcts/24780Git/24780project/

thuhu@Chens-Thinkpad MINGW64 /d/AreMyPrjcts/24780Git/24780project (master)
```

我的工作区地址：D:\AreMyPrjcts\24780Git\24780project

工作区（working directory）：电脑中你自己指定的目录（下图）

直接点开文件修改后保存——对应——工作区的修改

$ git checkout -- testChen. txt 将工作区的修改全部取消（成为和暂存区一样的状态 P.S.如果没有暂存过，就和本地版本库一样；没有本地版本库，我还不知道，也许和上次从远端仓库 clone 下来后的状态一样？）



版本库的暂存区（stage）：用来暂时存放一些本地修改

add 语句, rm 语句, reset HEAD 语句——对应——暂存区的修改

$ git add testChen.txt 将工作区某文件添加（或修改）到暂存区

$ git rm testChen.txt 将暂存区某文件从暂存区删除，影响工作区

$ git diff testChen.txt 查看工作区与暂存区之间 testChen.txt 文件的差别

$ git reset HEAD testChen.txt 将暂存区清空，但不修改工作区内容，见下图（status 用法见备注）

```
thuhu@Chens-Thinkpad MINGW64 /d/AreMyPrjcts/2478OGit/2478Oproject (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   testChen.txt


thuhu@Chens-Thinkpad MINGW64 /d/AreMyPrjcts/2478OGit/2478Oproject (master)
$ git reset HEAD testChen.txt
Unstaged changes after reset:
M       testChen.txt

thuhu@Chens-Thinkpad MINGW64 /d/AreMyPrjcts/2478OGit/2478Oproject (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   testChen.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

此处，原本工作区与暂存区的内容一致（未提交到本地分支），执行 reset HEAD filename 后，暂存区清空，但工作区不变，因此工作区与暂存区不一致了。如果工作区也想清空，用上一段说到的 git checkout --filename。

版本库的本地分支（例如：master）：本地版本库的最终版

commit 语句， reset --hard HEAD 语句——对应——本地 “现在” 的分支的修改（初始时默认本地只有一个 master 分支，但可以自由加减分支）

$ git commit -m "Add some comments here" 将暂存区的内容同步（修改、添加、删除等各种）到本地分支，对本次同步加上评论，并删除暂存区的内容

$ git reset --hard HEAD^ 退回往上数第一个版本，HEAD^^往上数两个，HEAD~73往上数 73 个

退回版本后也可以回到修改后的版本，但需要找到版本号，没仔细看怎么搞

关于分支（branch）：用于本地版本库中的分叉与合并

$ git branch 查看本地分支，名字前带*的是现在工作的分支（HEAD 指向的分支）

$ git branch testbranch 以目前分支为基础，新建本地分支，名为 testbranch，若重名则不执行

$ git checkout testbranch 切换到 testbranch 分支

$ git checkout -b testbranchagain 新建本地分支，名为 testbranchagain，并切换到该分支

$ git merge testbranch 将 testbranch 的内容合并到现在指向的分支。如果没有充足，可以自动合并，如果版本出现冲突，会将两个版本的内容全部放到本地工作区的该文件中（下图），由我们手动选取，然后再执行 add, commit 操作完成合并步骤。在 commit 之前，不允许切换分支。

```
<<<<<<< HEAD
Chen, test. new ok, testbranch limited?
=======
Chen, test. hey this is testbranchagain
>>>>>>> testbranchagain
```

$ git branch - d testbranch 删除分支 testbranch。不可以在这个分支删除这个分支本身。如果被删除分支已经 commit 然而没有 merge 到别的分支，而确认真的要删除，那么使用大写 D 强制删除，即： $ git branch - D testbranch

$ git log - -graph查看之前各种分支 merge 的线路，

之前的 commit 语句是将暂存区的内容同步到 “现在” 的分支（HEAD 指向的分支），reset --hard HEAD 语句也是在 “现在” 的分支退回前一个版本。对工作区、暂存区的所有操作都一样使用，切换分区会导致工作区与暂存区的内容丢失，因此在 commit 之前使用 checkout 会被拒绝执行。

一般使用自建的本地分支工作，工作到可以使用后，合并到本地的 master 分支。这样可以保证本地的 master 分支始终是可以使用的版本。

——————————————

—— 分割线，以上操作一切修改都在本地，下面是对远程端的操作，请谨慎

—— 本地操作无需联网

—— 在与 GitHub 上的远程库沟通之前，需要添加 SSH key

——————————————

远程仓库（默认名叫 origin）：团队成员共享的仓库

clone 语句, git push 语句

$ git clone https://github.com/24780inoodlessquad/24780project 在本地建立一个以项目名为名的文件夹，并将远程仓库内容全部复制下来。如果本地有一个非空的同名文件夹，操作将不会执行。也可以使用 $ git clone git@github.com:24780inoodlessquad/24780project
$ git push origin master 将本地的master分支推送到远程仓库（origin）的master分支中去，也可以推送其他的分支到远程仓库的对应分支中，如果没有对应的关联分支，将会创建一个同名分支
$ git checkout -b Chen origin/ChenHu 将远程库的 ChenHu 分支同步到本地库，本地会新建一个名叫 Chen 的分支。如果本地已经有一个想要合并的版本库，可以使用 $ git branch --set-upstream-to=origin/ChenHu ChenHu，将本地叫做 ChenHu 的分支与远程库上 ChenHu 分支关联，再进入本地 ChenHu 分支执行$ git pull

$ git push origin -d ChenHu 将远端库的 ChenHu 分支删除。本地库与之关联的分支不会因远端的分支删除而消失，但如果需要删除本地分支，需要再执行本地的删除分支操作。

备注：

$ git status 查看所有区域之间的差别：

{

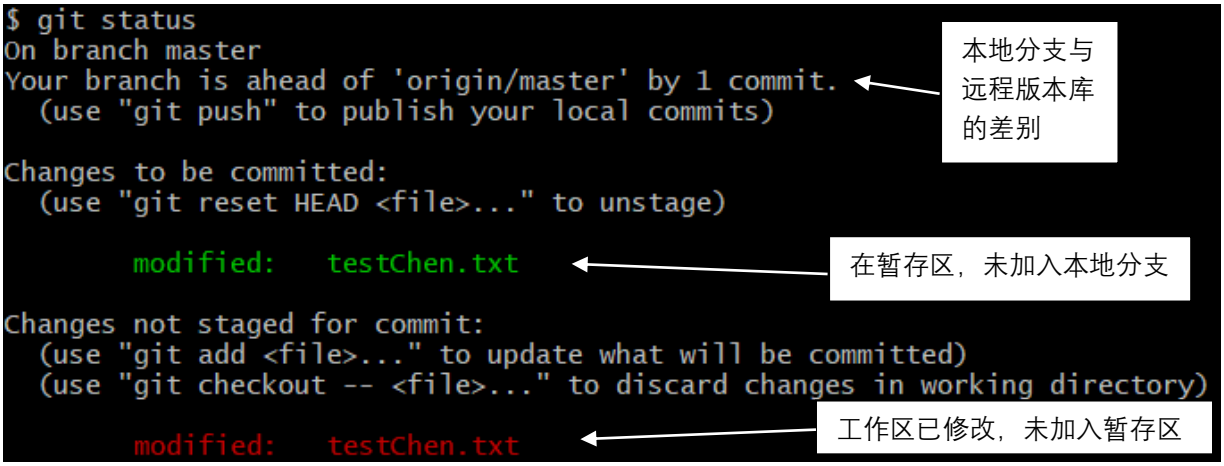
此前我曾顺序执行：

修改 testChen.txt，$ add，$ commit； //第一次修改被加入本地分支

再次修改 testChen.txt，$ add； //第二次修改在暂存区中

再再次修改 testChen.txt； //第三次修改在工作区

}


$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
Changes to be committed:
(use "git reset HEAD <file>..." to unstage)
modified: testChen.txt
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git checkout -- <file>..." to discard changes in working directory)
modified: testChen.txt
本地分支与远程版本库的差别
在暂存区，未加入本地分支
工作区已修改，未加入暂存区