配置SSH Key

git config --global user.name <GitHub账户名>

git config --global user.email <GitHub注册邮箱名>

初始化Git仓库 git init

查看所有文件的状态 git status

查看WORK区与STAGE区的具体差别，即未STAGE的修改 git diff [<file>]

查看STAGE区与HEAD的具体差别，即已STAGE的修改 git diff --cached [<file>]

查看WORK区与HEAD的具体差别，即未STAGE和已STAGE的全部修改 git diff HEAD [<file>]

查看提交历史 git log

查看命令历史 git reflog

更新与提交

把新文件、已修改的文件添加到STAGE区

git add <file>

把所有被跟踪的文件（被修改、被删除，不包括新建）提交到STAGE区

git add -u

git add -update

把所有被修改以及新建的文件（不包括被删除）提交到STAGE区

git add .

可以使用通配符：如\*.txt

把STAGE区提交到MASTER库

git commit -m <message>

把WORK区的全部修改提交到STAGE区与MASTER库

git commit -a -m <message>

把文件从WORK区与STAGE区删去

git rm <file>

把文件仅从STAGE区删去

git rm --cached <file>

重命名文件

git mv <old\_file> <new\_file>

撤销

覆盖上次提交

git commit --amend

取消STAGE

git reset [HEAD] <file>

把当前分支撤回到某个MASTER，STAGE区与WORK区不变

git reset --soft <commit>

把STAGE区和当前分支撤回到某个MASTER，WORK区不变

git reset [--mixed] <commit>

把WORK区、STAGE区和当前分支全部撤回到某个MASTER

git reset --hard <commit>

把WORK区的文件撤回到STAGE区的状态

git checkout -- <file>

把WORK区和STAGE区的文件撤回到某个MASTER

git checkout <commit> <file>

分支

查看所有分支

git branch

在HEAD处新建分支

git branch <branch>

在HEAD处新建分支并切换到它

git checkout -b <branch>

切换到某分支

git checkout <branch>

把某分支合并到当前分支

git merge <branch>

删除分支

git branch -d <branch>

强制删除分支

git branch -D <branch>

远程仓库

查看已经配置的远程仓库

git remote

添加远程仓库

git remote add <remote> <url>

远程仓库重命名

git remote rename <old\_remote> <new\_remote>

删除远程仓库

git remote rm <remote>

从远程仓库抓取所有新内容

git fetch <remote>

从远程仓库抓取并自动尝试合并到当前所在分支

git pull

把当前分支推送到默认远程仓库的对应分支

git push

把当前分支推送到远程仓库的对应分支

git push <remote>

与远程仓库同步：git pull 和 git push

下载 repository（仓库）： git clone

复制 cat

检查 SSH Key

ssh -T [git@github.com](mailto:git@github.com)

Fork 和 Pull request

git add .

git commit -m"v0.1"

git push

git clone SSH地址

Git仓库

git init

WORK区：根目录下除了.git文件夹之外的部分，即受Git管理的、实际WORK的区域。

STAGE区：暂时存放WORK区所有的更改。STAGE区即.git\index这个文件。

MASTER库：记录所有的MASTER与分支，HEAD指向当前分支。只要Git仓库还在，提交到MASTER库的数据基本是不会丢失的。

贮藏栈：用于保存/恢复WORK区的临时状态。

对象库：记录所有的对象（即所有数据内容。文件会被压缩，重复的文件不会再次放入）。STAGE区和MASTER库实际记录的其实是对象库中文件的索引，即所谓的“快照”。

在WORK区中增加/更改/删除文件。

STAGE，将文件快照放入STAGE区。

提交，将STAGE区的快照永久存入MASTER库。

git status

git diff

git log

git reset --hard HEAD^

git reset --hard (commit id)

git reflog

git checkout -- readme.txt

git reset HEAD readme.txt

git rm

git checkout -- test.txt

git remote add origin [git@github.com:chandlerriver/2023example.git](mailto:git@github.com:chandlerriver/2023example.git) 关联一个远程库

git push

git push -u origin master 第一次推送

git push origin master

git remote -v

git remote rm <name>

git remote rm origin 解除了本地和远程的绑定关系

git clone [git@github.com:chandlerriver/2022-Autumn-Atomic-Physics.git](mailto:git@github.com:chandlerriver/2022-Autumn-Atomic-Physics.git)

git checkout -b dev

git branch dev

git checkout dev

git branch

git merge dev

git branch -d dev

Git分支

查看分支：git branch

创建分支：git branch <name>

切换分支：git checkout <name>或者git switch <name>

创建+切换分支：git checkout -b <name>或者git switch -c <name>

合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>