**Git Order**

$ git config --global user.name "Your Name" ---配置name

$ git config --global user.email [email@example.com](mailto:email@example.com) ---配置email

mkdir 文件夹名 ---新建目录

cd ---进入目录

pwd ---显示当前目录print works directory

git init ---把该目录变成Git可以管理的仓库repository

ls –ah ---显示隐藏文件

把一个文件放入Git仓库分两步：

1. git add 文件名 ---告诉Git，把文件添加到仓库，每次都add一下，之后可以一起commit
2. git commit –m “说明” ---告诉Git，把文件提交到仓库（-m "xxx" 后接输入说明，比如git commit –m “wrote a readme file”）（可以一次提交多个被add的文件）

rm –rf 文件/文件名 ---删除文件、文件名

git rm 文件/文件名 ---删除版本库中文件 然后git commit

git status ---时刻掌握仓库当前的状态

git diff 文件名 ---哪些difference

git log ---看看历史记录

git log –-pretty=oneline ---历史记录单行显示

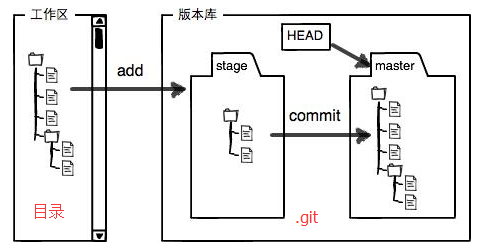
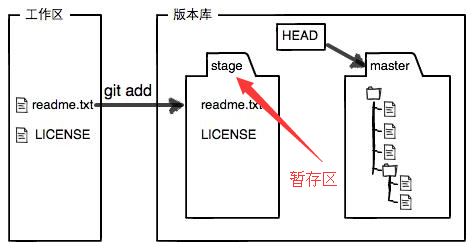
cat 文件名 ---显示文件中的内容

git reset –-hard HEAD^ ---回退到上一版本

git reset –-hard HEAD~i ---回退到往上i个版本

git reset –-hard commintid ---到commitid那个版本去

git reflog ---找到你的每次命令的commit id

git diff HEAD –- 文件名 ---查看工作区和版本库里面最新版本的不同之处

git checkout –- 文件名 ---丢弃工作区的修改（没有add进暂存区时），或者说把最近一次commit或add的返回到工作区

git reset HEAD 文件名 ---把已经放在暂存区的修改撤销掉（add进暂存区时）

git reset hard HEAD^ ---回退到上一个版本（add进了暂存区，且commit了本地库）

ssh-keygen ---创建ssh秘钥

git remote add origin [git@github.com:Fairme/XXX](mailto:git@github.com:Fairme/XXX)

---把本地仓库推送到github远程仓库

git push –u oringin master ---推送仓库（第一次）推送的是master分支，也就是主分支

git push origin master –f ---强制推送仓库（第二次省-u）

git clone [git@github.com:Fairme/xxx.git](mailto:git@github.com:Fairme/xxx.git)

---克隆github上的Fairme/xxx.git库到本地目录来

分支就是：新建一个分支，自己做一件事情，正常提交，直到你觉得ok了，别人可以在你的基础上继续了，才提交maser分支，以共同使用。分支就好比开了一个小黑屋工作。master是主分支

git branch 分支名 ---创建分支

git checkout 分支名 ---切换到<分支名>分支

git checkout –b 分支名 ---创建并切换到<分支名>分支下

git branch ---查看当前分支，\*表示当前分支

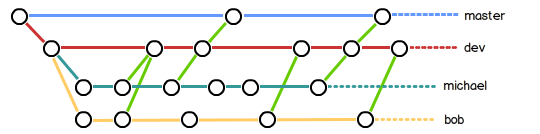
git merge 分支名 ---合并<分支名>分支到当前分支上

git branch –d 分支名 ---删除分支

master分支应该是非常稳定的，也就是仅仅用来发布新版本，平时不在上面干活。一般不要改动master分支。

那在哪干活呢？干活都在dev分支上，也就是说，dev分支是不稳定的，到某个时候，比如1.0版本发布时，再把dev分支合并到master上，在master分支发布1.0版本；

你和你的小伙伴们每个人都在dev分支上干活，每个人都有自己的分支，时不时地往dev分支上合并就可以了。



git merge --no-ff -m "merge with no-ff" dev ---这样子合并可以看到分支历史，禁用Fast forward，--no-ff是合并参数，用git log查看分支合并历史

git stash ---类似中断，把当前分支隐藏起来，做完另一件事继续 bug分支

git stash list ---查看隐藏的分支

git stash apply/pop ---恢复stash的内容并不删除/删除

git branch –D 分支 ---强制删除分支

git tag <name> ---新建一个标签，默认是HEAD

git tag –a <tagname> -m “xxx” ---指定标签信息

git tag –s <tagname> -m “XXX” ---可以用PGP签名标签

git tag ---查看所有标签

git push origin <tagname> ---推送一个本地标签

git push origin –tags ---推送全部未推送过的本地标签

git tag –d <tagname> ---删除一个本地标签

git push origin :refs/tags<tagname> ---删除一个远程标签

git config –-global alias.st status ---用git st 代替git status

git config --global alias.co checkout

git config --global alias.ci commit

git config --global alias.br branch

git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"