

帮助信息：

网页帮助：git merge --help/-h

git config --global user.name "yangsh"

git config --global user.email "yshbioinfo@126.com"

git 使用规则：管理和监控本地目录：

git init .

增加文件或目录

git add tmp/

git commit . -m 'add tmp'

git add .

git push 本地仓库送到server仓库

git pull server仓库拉到本地目录

git status  查看本地的修改

修改之后 git push 推送到服务器

修改后放弃修改，git checkout .

查看log文件：

git log -2

git reflog

git log --pretty=oneline

版本回滚：

git re**set** --hard HEAD^

git re**set** --hard 3628164

远程仓库：

本地仓库关联一个远程库，使用命令:

git remote add origin git@github.com:yangshGit/learngit.git

关联后，使用命令git push -u origin master第一次推送master分支的所有内容

从远程仓库克隆：

git clone git@github.com:yangshGit/learngit.git

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>

合并dev分支到当前工作分支：git merge dev

保留分支关系合并：git merge dev --no-ff -m "message"

查看git的图形化分支合并图：git log --graph --pretty=oneline --abbrev-commit

git log --oneline --graph

多人协作命令：

查看远程库信息，使用git remote -v；

本地新建的分支如果不推送到远程，对其他人就是不可见的；

从本地推送分支，使用git push origin branch-name，如果推送失败，先用git pull抓取远程的新提交；

在本地创建和远程分支对应的分支，使用git checkout -b branch-name origin/branch-name，本地和远程分支的名称最好一致；

建立本地分支和远程分支的关联，使用git branch --set-upstream branch-name origin/branch-name；

从远程抓取分支，使用git pull，如果有冲突，要先处理冲突

标签管理：

给指定commit id打标签：git tag v0.9 6224937

git tag -a v0.1 -m "version 0.1 released" 3628164

删除标签：git tag -d v0.1

推送所有标签到远程：git push origin --tags

* 命令git push origin <tagname>可以推送一个本地标签；
* 命令git push origin --tags可以推送全部未推送过的本地标签；
* 命令git tag -d <tagname>可以删除一个本地标签；
* 命令git push origin :refs/tags/<tagname>可以删除一个远程标签。

配置文件：

配置Git的时候，加上--global是针对当前用户起作用的；

当前用户的Git配置文件放在用户主目录下的一个隐藏文件.gitconfig中

cat ~/.gitconfig

如果不加，那只针对当前的仓库起作用：

Git配置文件都放在.git/config

列出所有配置信息：git config -l

别名配置：

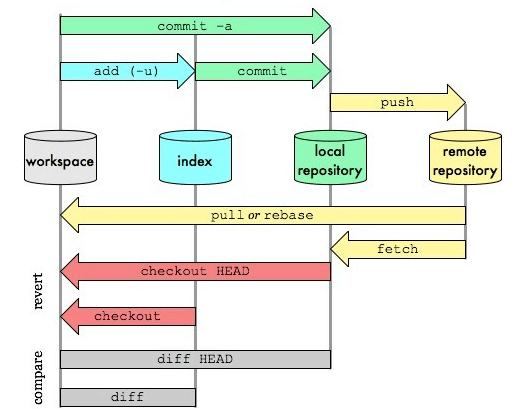
git config --global alias.logt 'log --graph --pretty=oneline'

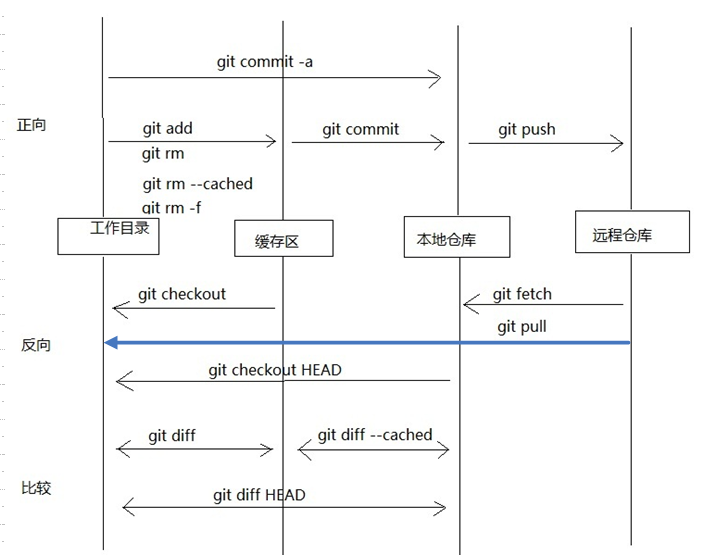
git config --global alias.merge "merge --no-ff"

—可以保留分支信息和commit，然后合并

忽略文件：

* 忽略某些文件时，需要编写.gitignore；
* .gitignore文件本身要放到版本库里，并且可以对.gitignore做版本管理！





想用云端覆盖本地，

|  |  |
| --- | --- |
| 1  2 | git fetch --all  git reset --hard origin/master |

[Git](http://lib.csdn.net/base/git) fetch 只是下载远程的库的内容，不做任何的合并 git reset 把HEAD指向刚刚下载的最新的版本

