> git config --global user.name "FishC\_Service"

> git config --global user.email ["fishc\_service@126.com"](mailto:\"fishc_service@126.com\")

git --version 查看版本号

git init 即可初始化 Git

git add 命令用于把工作目录的文件放入暂存区域

git rm --cached 文件名 撤销暂存区内容

git commit 命令用于把暂存区域的文件提交到 Git 仓库

git reset 命令用于把 Git 仓库的文件还原到暂存区域

git checkout 命令用于把暂存区域的文件还原到工作目录

git commit -m "你干了啥"

git status 命令查看当前的状态

git log 命令查看历史提交

git reflog 命令

偷偷记录下了你每一次的操作（无耻小人），第一列就是每次执行完命令，HEAD 指向的版本 ID 号啦~

git diff 比较版本

如果希望比较最新提交的快照和当前目录的内容，输入 git diff HEAD 命令即可：  
如果希望比较最新提交的快照和暂存区域的文件，只需要执行 git diff --cached 命令：

git commit --amend 命令修改最后一次提交 类似于vim

如果需要修改提交说明，那么可以执行 git commit --amend -m "新的提交说明"

git rm 文件名 删除命令

执行 git reset --soft HEAD~ 命令将快照回滚到上一个位置，然后重新提交

有两种情况你们可能会遇到，比如我在工作目录中增加一个 test.py 文件，然后执行 git add test.py 命令将其添加到暂存区域，此时我修改 test.py 文件的内容，那么暂存区域和工作目录就是两个不同的 test.py 文件了：  
此时如果我执行 git rm test.py 命令，Git 会下意识地阻止我：  
因为两个不同内容的同名文件，谁知道你是不是搞清楚了都要删掉？还是提醒一下好，别等一下出错了又要赖机器……  
根据提示，执行 git rm -f test.py 命令就可以把两个都删除。  
还有另一种情况就是你可能只想删除暂存区域的文件（保留工作目录的），那么你可以执行 git rm --cached 文件名 命令实现目的。  
git mv game.py wordgame.py 重命名

git branch 分支名  创建分支

git branch -v 查看分支

没有任何提示说明分支创建成功（一般也不会失败啦，除非创建了同名的分支会提醒你一下），此时可以执行 git log --decorate 命令查看：  
如果希望以“精简版”的方式显示，可以加上一个 --oneline 选项（即 git log --decorate --oneline），这样就只用一行来显示一个快照记录。

git checkout 分支名 切换分支

git log --oneline --decorate --graph --all 命令：  
--graph 选项表示让 Git 绘制分支图，--all 表示显示所有分支

git merge 分支名 合并分支

git status

Git 会在有冲突的文件中加入一些标记，需要你手动去更改。

****注意：如果 Git 检测到同一行有不同的内容，还是会报冲突并让你自行决定谁去谁留的****

 git branch -d 分支名  删除分支

git remote -v 查看别名

git remote add shoufu <https://github.com/8682/-.git> 创建别名

git push shoufu master 推送项目

git clone 目标 下载项目

不需要太多权限管理的喜欢用git，有权限管理的不会用。之前公司用商业版的Perforce,原因很简单，几千万个submit的codechange,一旦崩溃，对公司的损失是巨大的，即使有备份，公司也不愿意去做第一个吃螃蟹的

有意思，居然有人踩我了我很纳闷那个踩我的有没有在大公司待过，知道的几个规模比较大的公司都没用git，很简单，谁闲的蛋疼把公司的商业密码以git的方式做分布式管理？像google的VCS甚至还是自己开发的一套工具，在规模几千人以上的团队中，一旦涉及到公司的商业机密，公司是不会把核心竞争力代码放出来的。无意挑起争论，但不要把git奉为神明，就是一个版本管理的工具而已，除了git外，还有很多做VCS的，比如hg,p4，公司用不用git跟公司的技术氛围和战略是相关的。

再补充点，小规模的互联网公司是喜欢用git的，原因简单，做服务型的互联网公司运营比技术更重要，传统服务的把代码完全转git的不多

版本许可证

Copyright (C) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT 许可证几乎是最宽松的版权约定，一旦你的程序采用，也就意味着别人只要在软件包含上边的版权声明，就可以对你的程序为所欲为了（包括“使用、复制、修改、合并、出版发行、散布、再授权和/或贩售软件及软件的副本”）。