# Git基本操作

1. 基本单词释义
   1. Repository

仓库

* 1. branch

分支

* 1. merge

合并

* 1. passphrase

密码

* 1. stage

暂存区

1. 文件
   1. .gitignore

忽略的文件，push的时候不会推上去

* 1. [README.md](http://readme.md/)

用于记录一些日志及更新的东西吧？？？！

1. Git区域

git 区域可以分为工作区间，暂存区，本地仓库区和远程仓库

* 1. 工作区（work directory)

一般是我们项目的根目录，通过git add .可以把代码推到暂存区中

* 1. 暂存区(stage)

通过git commit -m [注释] ，把暂存中的内容推到本地仓库中

* 1. 本地仓库区(repository)

通过git push,把本地仓库区中的内容推到远程仓库中

* 1. 远程仓库

1. 基本操作
   1. 初始化基本操作
      1. 创建一个仓库

git init
在源码目录中生成一个.git的目录

* + 1. 注册并查看用户信息

git config --global --list
git config --global [user.name](http://user.name/) [Name]
git config --global [user.email](http://user.email/) [Email]

* + 1. 查看当前代码库的状态

git status

* + 1. 查看版本信息

git log

* + 1. 查看git 版本号

git --version

* 1. Repository仓库操作
     1. 查看远程仓库的名称

git remote -v

* + 1. 把远程仓库的内容拉到本地

git clone [URL]

* + - 1. 把远程仓库指定的分支拉到本地

git clone -b [branch] [url]

* + - 1. 把远程仓库其他的分支拉到本地

查看远程仓库，本地创建和远程仓库一样的分支
​git branch [branch] origin/[branch]
​git checkout [branch]

* 1. 文件操作
     1. 修改文件
        1. 把文件放到暂存区

git add <file> (一般使用git add .)

* + - 1. 把文件还原到工作区间

git checkout -- <file> (一般使用git checkout -- .)

* + 1. 删除文件
       1. 手动删除文件

手动删除文件
git add .

* + - 1. 自动删除文件

git rm <file>git

* + 1. 重命名文件
       1. 手动重命名文件

1，git add <file>
2，git rm <file>

* + - 1. 自动重命名文件

git mv [oldname] [newname]

* + 1. 移动文件
       1. 手动移动文件git

git add .

* + - 1. 自动移动文件

git rm <file> [directory]

* + 1. 移动文件并重命名

git rm <file> [direction]/<file>

* + 1. 查看文件前后的变化
       1. git log --pretty=oneline <file>

返回提交的Id，通过git show <Id>查看操作

* + - 1. git log -p <file>
    1. 文件的还原
       1. 在没有使用git add之前 (还没有到暂存区）

git checkout -- <file> //回到了最后一次（commit）的状态

* + - 1. 到了暂存区

git reset HEAD <file> //把暂存区的内容撤销到工作区
git checkout --<file>

* + - 1. 到了本地仓库

进行版本操作

* 1. 版本操作
     1. 回到上一个版本

git reset --hard HEAD^ //一个^代表一个版本，两个^表示回到前两个版本

* + 1. 回到指定的版本

git reset --hard <ID> //<ID> 可以通过git log获取到

* + 1. 将指定文件回到上一个版本

git checkout <ID> -- <file>

* 1. 标签管理
     1. 创建标签

git tag [tag] //默认添加到最新的

* + - 1. 添加到以前的版本

git tag [tag] <ID>

* + 1. 查看标签

git tag

* + 1. 删除标签

git tag -d [tag]

* + 1. 推送标签到远程服务器

git push origin [tag]

* 1. 分支操作
     1. 创建一个分支

git branch [branch]

* + 1. 查看分支

git branch
git branch -av //查看所有分支的情况

* + 1. 切换分支

git checkout [branch]

* + 1. 删除分支

git branch -d [branch]
不能删除当前分支
不能删除当前代码有修改的分支

* + 1. 强制删除分支

git branch -D [branch]

* + 1. 创建并切换分支

git checkout -b [branch]

* + 1. 分支合并
       1. 把Dev分支的代码合并到Master中

git merge dev

* + 1. 删除远程分支

git push origin --delete [branch]

* 1. 其他操作
     1. 使用命令字符 q来宗旨git bash 中的git log 操作。
     2. 使用命令字符：wq来终止commit的操作。
     3. 使用clear 清除git bash 的命令
     4. 使用git diff找到不同
     5. 回退到某一个文件之后，又需要返回到最新的某个版本
        1. 通过git reset --hard <ID>回到指定的版本
        2. 通过git reflog查询每一次命令的记录，获取<ID>
        3. 通过git reset --hard<ID>回到最新的某个版本

1. 常见问题及解决
   1. git@github.com: Permission denied (publickey) fatal: Could not read from remote repository

使用SSH地址下载GitHub项目报错,原因是没有配置SSH keys

* + 1. git config --global --list //验证邮箱，没有配置就要配置邮箱
    2. ssh-keygen -t rsa -C "email" //生成密钥
    3. 到git 仓库添加密钥
  1. 合并分支时的冲突

<<<<<<< HEAD
 ***//当前分支的代码这是Master分支的代码***
=======
**//d*ev分支的代码这是dev1分支代码****>*>>>>>> dev

* + 1. git merge -abort //保留当前的冲突代码，忽略其他分支的内容
    2. 删除<<<<< >>>>>>后，add/commit 代码
  1. 最好不要用git pull(直接从远程仓库拉取代码到工作区），可以使用git fetch（从远程仓库拉到本地仓库）+git merge(合并本地仓库和工作区的代码）