# 配置

【显示配置信息】

git config --list

【配置用户信息】

git config --global user.name "kira"

git config --global user.email k527971724@gmail.com

【查看用户信息】

git config --global user.name

git config --global user.email

【修改用户信息】

git config --global --replace-all user.name "输入你的用户名"

git config --global --replace-all user.email "输入你的邮箱"

# 创建仓库

## 【创建本地仓库】repository

cd进入目录

git init

## 【.git文件夹】

仓库创建完成后，会在目录下生成一个隐藏的.git文件夹。

这个文件夹是用来记录本地所有的Git操作的，可以通过”ls -al”命令来查看。如果想要删除本地仓库，删除这个文件夹就行了。

## 【.gitignore文件】

【用法】

指定哪些文件不用加入版本控制

实际项目中，很多文件都是不需要版本管理的，比如Python的.pyc文件和一些包含密码的配置文件等等。

指定不加入版本控制的文件：

1）过滤整个文件夹 mtk/

2）过滤所有.zip文件 \*.zip

3）过滤某个具体文件 mtk/do.c

指定加入版本控制的文件：（多了感叹号）

1）!\*.zip

2）!mtk/one.txt

使用场景：

/mtk/

!/mtk/one.txt

只对!/mtk/one.txt使用版本控制，对/mtk/下的其他文件忽略

【创建仓库时就要添加.gitignore】

如果不慎在创建.gitignore文件之前就push了项目，那么即使你再加入.gitignore文件也不会起作用理。因此一定要养成在项目开始就创建.gitignore文件的习惯。

【添加.gitignore】windows系统

1. 在项目根目录下面创建gitignore.txt文件

2. 把你需要排除的文件名保存到gitignore.txt文件

3. 在项目根目录下面按住Shift键并邮件然后选择“在此处打开命令窗口”

4. 执行命令 ren gitignore.txt .gitignore

# 版本控制

## 【工作流】

本地仓库由 git 维护的三棵"树"组成。

【工作目录】持有实际文件

【暂存区（Index）】它像个缓存区域，临时保存你的改动（执行add后）

【HEAD】它指向你最后一次提交的结果（执行commit后）

【github】远程仓库（执行push后）

## 【提交代码】本地的版本控制，还没有提交到远程仓库

git add 文件名 提交文件

git add 目录名 提交目录

git add . 提交当前目录所有内容

git commit -m"your message"

【查看自上次提交后经过修改的文件的状态（commit后）】

git status

红色文件名 还没add

绿色文件名 已经add（还没commit）

【查看文件的具体改动情况（文件是未add的）】

git diff 路径/文件名

【撤销还没add的修改】

git checkout 路径/文件名

【撤销add】

git reset HEAD 全部取消

git reset HEAD 文件名 取消一个文件

【查看提交记录】

git log

查看某条记录的具体修改内容

git log 1fa380b502a00b82bfc8d84c5ab5e15b8fbf7dac -1 -p

【分支】

建立分支

git branch version1.0

切换分支

git checkout version1.0

合并分支

git checkout master

git merge version1.0

删除分支

git branch -D version1.0

【关于签出】

假如commit 1a 1b 1c 1d 之后

cheakout 1a

那么1b 1c 1d都会永久消失

# 远程仓库

【拷贝远程仓库】

git clone https://github.com/k527971724/flasky.git

会把flasky整个文件夹拷贝到当前目录下

流程：clone add commit push pull

【关联远程仓库】remote 遥远的

git remote add origin https://github.com/k527971724/flasky.git

流程：init remote

【同步到远程仓库】

git push origin master

（git push -u origin master如果新建的远程仓库是空的，要加上-u这个参数）

origin部分代表远程版本库的Git地址

master部分代表同步到哪一个分支上。

【同步到本地】

如果当前分支与远程分支存在追踪关系，git pull就可以省略远程分支名

git pull origin

# 换行符

# 【autoCRLF】

#提交时转换为LF，检出时转换为CRLF

git config --global core.autocrlf true

#提交时转换为LF，检出时不转换

git config --global core.autocrlf input

#提交检出均不转换

git config --global core.autocrlf false

# 【safeCRLF】

#拒绝提交包含混合换行符的文件

git config --global core.safecrlf true

#允许提交包含混合换行符的文件

git config --global core.safecrlf false

#提交包含混合换行符的文件时给出警告

git config --global core.safecrlf warn