配置git基本信息

```
git config --global user.name "用户名"
git config --global user.email "邮箱"
git config --list   // 查看
git remote -v  // 查看所有远端仓库
git remote add 名称 地址   // 添加远端仓库
git remote set-url 名称 地址 // 修改远端仓库
git remote remove 名称 // 删除远端仓库
```

生成一个提交

```
git add .   // 将所有文件添加到暂存区    or
git add 文件名    // 将指定文件添加到暂存区
git commit -m '提交说明'  // 将暂存区内容添加到本地仓库中，生成一个提交
```

提交远端

```
git push origin 本地分支名：远端分支名     // 将本地提交推送到远端仓库
提交失败解决方案：
    1.git fetch --all
    2.git rebase origin/远端分支名（解决冲突过程如rebase用法）
    3.git push origin 本地分支名：远端分支名

git push origin 本地分支名:远端分支名 --force   // 强推代码
```

```
git clone 地址  // 克隆远程文件到本地
git clone -b 分支名 地址    // 克隆指定分支文件到本地
git fetch --all    // 获取远程代码库，将远端代码更新到本地git库
gitk --all &    // 进入可视化页面
```

分支管理

```
git checkout 分支名    // 切换分支
git checkout -b 新建分支名    // 基于当前位置新建分支
git checkout -b 新建分支名 远端库|commitId   // 基于远端库新建分支
git branch -D 分支名    // 删除本地分支
```

```
git branch  // 查看当前分支

git push -d origin 远端分支名    // 删除远端分支
```

复制提交

```
git cherry-pick commitId     // 将其他提交合并到当前分支
若有冲突，全局搜索====查找解决冲突
git cherry-pick --continue    // 继续执行合并
git cherry-pick --abort   // 中途推出合并
```

同步最新提交

```
git rebase 本地分支名     // 将当前分支与想要合并分支同步
git rebase origin/远端分支名      // 同步到指定分支上的一个提交
若有冲突，全局搜索====查找解决冲突
git add .  // 重新提交
git rebase --continue   // 继续合并
git rebase --abort   // 中途推出
```

合并提交

```
git merge --squash 分支名/id   // 合并多个提交为一个提交，基于第一个提交的前一个节点，向上合并为
git reset --merge    // 退出merge过程

git rebase -i 分支名|commitId 基于最后一个提交，合并到第一次提交的前一个节点，合并为commit状态
    按 i 键进入编辑，将除第一项外的 pick 修改为 s ，按Esc退出编辑，
    按两次shift + zz，合并完成
```

修改提交

```
git reset --mixed HEAD^   // 回退提交，保留修改记录
git reset --soft HEAD^    // 回退提交，保留修改记录至暂存区（add）
git reset --hard HEAD^    // 回退提交，并删除修改记录
    HEAD^ 表示上一个版本，即上一次的commit，也可以写成HEAD~1
    如果进行两次的commit，想要都撤回，可以使用HEAD~2
    也可以使用commitId

git commit --amend    // 修改commit注释

```

```
git checkout -- 文件名  // 撤销某个文件的修改
```

gitk界面中文乱码

```
git config --global gui.encoding utf-8
```

git拉的代码，禁止lf格式的文件变成crlf

```
git config --global core.autocrlf false
```

让git拥有区分大小写的能力

```
git config core.ignorecase false
```