目录

[1初始化用户名和邮箱，（为了告诉Git是谁在操作Git仓库） 1](#_Toc88729072)

[2创建目录 1](#_Toc88729073)

[3进入目录 1](#_Toc88729074)

[4初始化Git仓库（执行了这段代码，Git才知道该目录为需要它来管理的目录（Git仓库）） 1](#_Toc88729075)

[5常用的命令操作 2](#_Toc88729076)

[5.1 git status :查看当前状态 2](#_Toc88729077)

[5.2 git add 文件名 :添加到缓存区 2](#_Toc88729078)

[5.3 git commit -m "注释内容" :提交到版本库 2](#_Toc88729079)

[5.4 git log :查看版本，确定需要回到的时刻点 2](#_Toc88729080)

[5.5 git reset --hard 版本号 :回退操作 2](#_Toc88729081)

[5.6 git reflog :查看历史操作 3](#_Toc88729082)

[5.7 git clone 线上仓库地址 :克隆线上仓库到本地 3](#_Toc88729083)

[5.8 git push ：向线上仓库提交内容 3](#_Toc88729084)

[5.9 git pull ：拉取线上仓库内容 4](#_Toc88729085)

[6创建一个公私钥文件 ssh-keygen -t rsa -C "邮箱" 然后一路回车 4](#_Toc88729086)

[7分支相关指令 4](#_Toc88729087)

[7.1查看分支 git branch 4](#_Toc88729088)

[7.2创建分支 git branch 分支名 4](#_Toc88729089)

[7.3切换分支 git chechout 分支名 4](#_Toc88729090)

[7.4删除分支 git branch -d 分支名 4](#_Toc88729091)

[7.5合并分支 git merge 被合并的分支名 4](#_Toc88729092)

[8忽略文件 4](#_Toc88729093)

# 1初始化用户名和邮箱，（为了告诉Git是谁在操作Git仓库）

$ git config --global user.name "用户名"

$ git config --global user.email "邮箱"

！用户名和邮箱建议与github账户用户名和邮箱一致

//可以用$ git config --global user.name 和 $ git config --global user.email查看用户名及邮箱

# 2创建目录

$ mkdir 目录名

# 3进入目录

$ cd 目录名

# 4初始化Git仓库（执行了这段代码，Git才知道该目录为需要它来管理的目录（Git仓库））

$ git init

//执行完后可以发现目录中多了一个.git目录(该目录为隐藏目录，需要在查看中打开查看隐藏项目菜单)

！不要随意更改该目录下内容

# 5常用的命令操作

## 5.1 git status :查看当前状态

## 5.2 git add 文件名 :添加到缓存区

//git add 可以添加一个文件，也可以同时添加多个文件

git add 文件名

git add 文件名1 文件名2 文件名3 ...

git add. //添加当前目录到缓存区

## 5.3 git commit -m "注释内容" :提交到版本库

## 5.4 git log :查看版本，确定需要回到的时刻点

git log :详细显示

多屏显示的控制方式：

空格：向下翻页

b: 向上翻页

q: 退出

git log --pretty=oneline :一行显示，版本号详细

git log --oneline :一行显示，版本号为7位数

## 5.5 git reset --hard 版本号 :去到某一个版本

//回退之后，如果需要再回到最新版本，需要查看历史操作

git reset –hard HEAD^ 回退一个版本

git reset –hard HEAD^^ 回退三个版本……

git reset –hard HEAD~n 回退n个版本

## reset命令三个参数

--soft:仅仅在本地库移动HEAD指针

--mixed：在本地库移动HEAD指针并重置暂存区

--hard：在本地库移动HEAD指针并重置暂存区和工作区

## 5.6 git reflog :查看历史操作

git reflog :显示一行，会显示移动到某个版本需要几步

## 5.7 git clone 线上仓库地址 :克隆线上仓库到本地

//如果报错：unable to access 'https://github.com/YDFlame13/test.git/': OpenSSL SSL\_read: Connection was reset, errno 10054

//尝试打开cmd，刷新DNS解析缓存 -->输入 ipconfig /flushdns

## 5.8 git push ：向线上仓库提交内容

//如果报错，首次使用容易报错

//unable to access 'https://github.com/YDFlame13/test.git/': The requested URL returned error: 403

//需要修改".git/config"文件内容

//将

//[remote"origin"]

// url=https://github.com/用户名/仓库名.git

//修改为：

//[remote"origin"]

// url=https://用户名:密码@github.com/用户名/仓库名.git

//继续报错：Authentication failed for 'https://github.com/YDFlame13/test.git/'

//最新版本github将密码改成了 令牌token需要自己去gitub上获取令牌token具体操作参见：https://blog.csdn.net/weixin\_41010198/article/details/119698015

## 5.9 git pull ：拉取线上仓库内容

！上班先git pull 下班一定要git push

## 5.10 git diff比较文件差异

Git diff文件名: 将工作区的文件和暂存区进行比较

Git diff本地库中历史版本 文件名: 将工作区与中的文件和本地库中某个历史版本相比较

不带文件名: 比较目录下全部文件

# 6创建一个公私钥文件 ssh-keygen -t rsa -C "邮箱" 然后一路回车

# 7分支相关指令

## 7.1查看分支 git branch/git branch -v

## 7.2创建分支 git branch 分支名

## 7.3切换分支 git chechout 分支名

## 7.4删除分支 git branch -d 分支名

## 7.5合并分支 git merge 被合并的分支名

# 8忽略文件

用于项目中万年不变的场景

需要在需要忽略的目录下创建文件.gitignore 该文件用于声明忽略文件规则

常用规则：

/mtk/ 过滤掉整个文件目录

\*.zip 过滤所有的.zip文件

/mtk/do.c 过滤某个具体文件

!index.php不过滤某个具体文件