

# Git使用教程

本教程为作者备忘使用所编写，仅涉及初步使用的相关命令

## 基础设置

### Git

使用Git，首先请设置 用户名 与 邮箱

关于用户名：虽然不影响使用，但是在多人合作时，用户名有利于不同的开发者区分提交者的身份

关于邮箱：邮箱将作为 git push 的公钥，请认真填写并确保其与代码托管平台所使用的邮箱一致，同时可以使其他开发者更加快捷方便的联系到你

```
git config --global user.name "用户名"
git config --global user.email "邮箱"
```

### GitHub

我们在使用远程仓库时需要设置一个密钥以便于链接我们的项目仓库并进行推送或修改

实现这一功能通常使用以下两种方法

#### Token法

**Token法每次使用很麻烦，不建议用，请使用SSH**

首先，打开GitHub中个人中心的 set 选项

选择 the developer settings 一项

选择 personal access tokens 后点击 tokens

选择右上角的 Generate new token(classic)

填写 note 名称，同时设置有效时间，勾选 repo 选项

最后点击页面底部绿色的 Generate token 即可完成创建密钥

生成后请复制密钥

致又忘了的自己：如果太抽象了，给你挂个链接

[https://www.bilibili.com/video/BV1pX4y1S7Dq/?spm\\_id\\_from=333.337.search-card.all.click&vd\\_source=3dca3996a1331ebf7584868b945b5642](https://www.bilibili.com/video/BV1pX4y1S7Dq/?spm_id_from=333.337.search-card.all.click&vd_source=3dca3996a1331ebf7584868b945b5642)

进度条是10:00

## SSH协议法

### 第一步：创建密钥对

首先在本地桌面右键点击 `Open git bash here` 打开git

输入以下命令创建 SSH key

```
ssh-keygen
```

此时会显示 `Enter file in which to save the key`, 请保持默认, 回车

之后会要求输入一个密码, 个人建议留空直接回车三次

此时理应成功创建一个的密钥对

### 第二步：配置GitHub公钥

根据以下路径打开文件夹

```
C:\Users\[账户名]\.ssh
```

里面应当有两个文件

```
id_ed25519 与 id_ed25519.pub
```

第一个没有后缀名的为私钥, 请保留好, 请勿泄露

第二个为公钥, 打开文件对里面的内容全部复制

下一步登录GitHub点击右上角头像的中的 `settings`

点开后选择左侧栏中的 `SSH and GPG keys`

点击 `SSH keys` 栏中右侧写着 `New SSH key` 的绿色按钮

`Title` 一栏请随意填写, `key` 一栏请粘贴公钥文件中的内容

随后点击绿色的 `Add SSH key`

验证密码后添加成功

## 第一步：创建项目仓库

首先需要创建本地仓库, 其分为两种方式

第一种: clone 别人的/自己托管的 项目仓库

第二种: 创建本地文件夹并init

### clone项目仓库

```
git clone [项目地址]
```

## 创建本地文件夹

在任意位置创建本地文件夹后，使用右键选项中的 `open git bush here`

在打开的命令窗口中输入

```
git init
```

此时，git会自动创建一个名为 `.git` 的隐藏文件，这就已经创建一个代码仓库了

## 远程仓库

本案例远程操控平台以GitHub为例

首先在GitHub上创建一个新的仓库，复制其https地址，如果使用SSH则需要创建配对密钥

添加远程仓库的指令为

```
git remote add [仓库名]
```

修改仓库名

```
git remote [原仓库名] [修改后的仓库名]
```

## 第二步：提交、推送与删除文件

首先将文件添加至暂缓区

在文件夹内打开git，如果使用vscode，请打开文件后同时打开终端

输入以下命令

添加特定文件

```
git add [文件名全称.后缀名]
```

添加多个文件

```
git add .
```

其中 `.` 意为当前目录整个文件夹

删除文件跟踪

```
git rm [文件名全称.后缀名]
```

此时，文件已经添加进入暂存区

如果想将文件从暂缓区取回，则使用以下命令

```
git reset HEAD [文件名全称.后缀名]
```

在此之后将文件提交至仓库

输入以下命令

提交文件

```
git commit
```

提交文件的同时添加备注

```
git commit -m "备注内容"
```

使用status可以查看目前仓库的状态

```
git status //可以查看目前分支
```

在单独使用 `git commit` 后，终端将自动跳转至 `vim` 编辑器中

按下 `i` 键进入输入模式，此时可以对本次提交书写备注

如果想要保存并推出编辑器

按下 `ESC` 键并输入 `:wq`

自此，我们成功的在git中进行了一次提交

如果使用远程仓库，则需要与远程仓库同步，此操作名为推送，命令如下

```
git push [仓库名] [分支名]
```

## 第三步：查看添加日志

使用 `log` 命令可以查看提交的记录

```
git log
```

查看更详细的信息

```
git log --stat
```

查看某一条具体的修改

```
git diff [commit id]
```

## 第四步：回退到指定节点

可以实现回退功能的命令有两条

当前分支回退到指定位置并放弃修改

```
git reset --hard [commit id]
```

当前分支回退到指定位置并保留修改

```
git reset [commit id]
```

此命令会保留

```
git checkout [commit id]
```

## 第五步：分支的概念与使用

---

查看目前的所有分支

```
git branch
```

通常来说有以下几个分支

`master` 分支：一般用来保存经过测试的稳定代码

`develop` 分支：一般用来保存开发过程中的代码

### 创建

创建分支的指令如下

```
git checkout -b [分支名]
```

### 切换

```
git checkout [分支名]
```

### 合并

```
git cherry-pick [某次提交的commit id]
```

此命令需要先跳转至主分支，使用目标某一支指定版本的commit id后，目标版本将合并至主分支

如果需要下载，请回到第一步，直接clone仓库就行

持续更新中.....