

git学习

- 电脑已经安装好Git
- 打开电脑的Git Bash
- 在终端输入“`ssh-keygen -t ed25519 -C "你的邮箱地址"`”命令，注意邮箱地址外面的双引号是你需要加上的——双引号是英文版的。注意命令要从左往右依次输入，否则会报错。这里的邮箱地址是指你的GitHub账户绑定的邮箱地址。
- 注意标红框的这条数据，它指出了你的公钥所在的文件。

```
MINGW64/c/Users/lenovo/Desktop
lenovo@LAPTOP-8N8TJEAN MINGW64 ~/Desktop
$ ssh-keygen -t ed25519 -C "2633455647@qq.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/lenovo/.ssh/id_ed25519):
Created directory '/c/Users/lenovo/.ssh'.
Enter passphrase for "/c/Users/lenovo/.ssh/id_ed25519" (empty for no passphrase)
:
Enter same passphrase again:
Your identification has been saved in /c/Users/lenovo/.ssh/id_ed25519.
Your public key has been saved in /c/Users/lenovo/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:QTvucpQxzW0w1s/v6PZnhr3eBY7w9exCKdBGpXwRg 2633455647@qq.com
The key's randomart image is:
+--[ED25519 256]--+
|      E*      |
|    + . O    |
|   B . O    |
|  + & .    |
| S B .    |
| o B o o   |
| . o + . = *|
| o o . = @ B|
|      . += B O|
+-----[SHA256]-----+

lenovo@LAPTOP-8N8TJEAN MINGW64 ~/Desktop
$ |
```

- 这里存放公钥的文件是图片红色框里的黄色部分。
- 在Git Bash终端中输入“`cat 存放公钥的文件`”，打开文件复制公钥，如图：

```
lenovo@LAPTOP-8N8TJEAN MINGW64 ~/Desktop
$ cat /c/Users/lenovo/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINeduSzxe5R2bQW5yRL6aaE5sPgKpGzm51b1n246z9+
2633455647@qq.com
```

- 将复制好的公钥添加在GitHub网页上，如图：



- 使用 SSH 的好处主要有两点：
 - 免密码推送，执行 `git push` 时不再需要输入用户名和密码了；
 - 提高数据传输速度。它不是必须的，比如在课程中挑战环境是不可保存的，一次性的，这种环境就没有必要创建 SSH 了，因为相较好处来说，还是太麻烦了。

- 在GitHub自己的仓库页面选择克隆，复制SSH里面的git@github.com:uniuan/study.git
- 在Git Bash终端输入“rm -rf 仓库名”删除仓库
- 在Git Bash终端输入“git clone git@github.com:uniuan/study.git”出现如下界面时输入“yes”。

```
lenovo@LAPTOP-8N8TJEAN MINGW64 ~/Desktop
$ git clone git@github.com:uniuan/study.git
Cloning into 'study'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbqZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), 192.79 KiB | 213.00 KiB/s, done.
```

- 查看仓库状态：ls
- git add 文件名：将文件添加到暂存区。
- git reset -- 文件名：撤销暂存区的修改。
- cd 本地仓库名
- git add * （注：别忘记后面的.，此操作是把Test文件夹下面的文件都添加进来）
- git commit -m "提交信息" （注：“提交信息”里面换成你需要，如“first commit”）
- git push -u origin master （注：此操作目的是把本地仓库push到github上面，此步骤需要你输入帐号和密码）

• 使用Git命令上传文件的步骤¹²³⁴⁵:

- ①进入本地的项目目录，右键“Git Bash here”，调出git命令行界面，然后输入git init。
- ②将目录下的所有文件上传，也可以将“.”换成具体的文件名“git add 文件名”
- ③将项目提交到本地仓库git commit -m 'commit file 文件名'。
- ④在GitHub上创建新的repository。
- ⑤点击“Create repository”跳转到一个连接，获取到项目的GitHub地址。
- ⑥将本地的代码关联到GitHub上 git remote add origin 项目的GitHub地址。
- ⑦上传代码到GitHub之前需要先pull。

• 使用Git命令将本地文件传输到GitHub

- 1.克隆仓库：git clone GitHub仓库的SSH
- 2.进入仓库主目录：cd 仓库名
- 3.创建一个文件并查看仓库状态：echo 'hello world' > one.txt
- ls
- 4.git status
- 5.把新增文件添加到暂存区：git add one.txt
- git status
- 如果对多个文件或目录进行了增删改，可以使用git add .命令全部添加到暂存区。
- 如果要撤销暂存区的修改怎么办？执行git reset -- [文件名] 或者 git rm --cached [文件名] 命令即可

- `git diff`查看工作区被跟踪文件的修改详情
- 修改 README.md 文件: `echo '你要输入的内容' >> README.md`
- 将工作区的两处修改（新增文件 one.txt, 修改文件 README.md）全部添加到暂存区:`git add .`
- `git log`查看版本区的提交历史记录

```
shiyanolou:shiyanolou/ (master*) $ git config --global user.email "1195581533@qq.com"
shiyanolou:shiyanolou/ (master*) $ git config --global user.name "Manchangdx" [13:55:36]
shiyanolou:shiyanolou/ (master*) $ git config -l [13:55:58]
shiyanolou:shiyanolou/ (master*) $ [13:56:15]
```

- 接下来需要对 Git 进行一些本地配置:
- user.email: 写入你自己注册 GitHub 账号的邮箱
- user.name: 你自己的 GitHub 账号名字
- 这两个命令设置你的身份信息如下图。`git config -l` 可以查看配置信息
- 完成后, 系统自动生成 Git 的配置文件, 就是主目录中的隐藏文件 .gitconfig

```
shiyanolou:shiyanolou/ (master*) $ cat -n ~/.gitconfig
1 [user]
2     email = 1195581533@qq.com
3     name = Manchangdx
shiyanolou:shiyanolou/ (master*) $
```

- 现在执行 `git commit` 命令生成一个新的提交, 一个必须的选项 `-m` 用来提供该提交的备注:

```
shiyanolou:shiyanolou/ (master*) $ git status
位于分支 master
您的分支与上游分支 'origin/master' 一致。

要提交的变更:
(使用 "git reset HEAD <文件>..." 以取消暂存)

    修改:      README.md
    新文件:    one.txt

shiyanolou:shiyanolou/ (master*) $ git commit -m 'commit file one.txt'
[master 06f2d6a] commit file one.txt
2 files changed, 2 insertions(+)
create mode 100644 one.txt
shiyanolou:shiyanolou/ (master*) $ git status
位于分支 master
您的分支领先 'origin/master' 共 1 个提交。
(使用 "git push" 来发布您的本地提交)

无文件要提交, 干净的工作区
shiyanolou:shiyanolou/ (master*) $
```

- 提交后, 暂存区的修改被清空, 执行 `git log` 查看提交记录, 紫色框中的十六进制序列号就是提交版本号, 这是很重要的信息, 每个提交都有自己单独的版本号, 就像公民身份证号一样:

```
commit 5c041ade38538dde0ff3dfb6c5003d32641a1422 (HEAD -> master)
Author: Manchangdx <1195581533@qq.com>
Date:   Sun Jan 27 14:01:53 2019 +0800

    commit file one.txt

commit 2b96af01439c999daee2f6d503e454147e713eb5 (origin/master, origin/HEAD)
Author: Manchangdx <1195581533@qq.com>
Date:   Sun Jan 27 01:26:09 2019 +0800

    Initial commit
(END)
```

- 观察上图的提交信息, 提交版本是按时间倒序排列的, 也就是最近的提交排在最上面, 你可能需

要查看时间正序排列的信息，那么可以使用 `git log --reverse` 命令。

- 现在介绍一个超级实用、使用频率极高但几乎所有 Git 教程都不重视的命令 `git branch -avv`，它用来查看全部分支信息：

```
* master          5c041ad [origin/master: 领先 1] commit file one.txt
remotes/origin/HEAD -> origin/master
remotes/origin/master 2b96af0 Initial commit
(END)
```

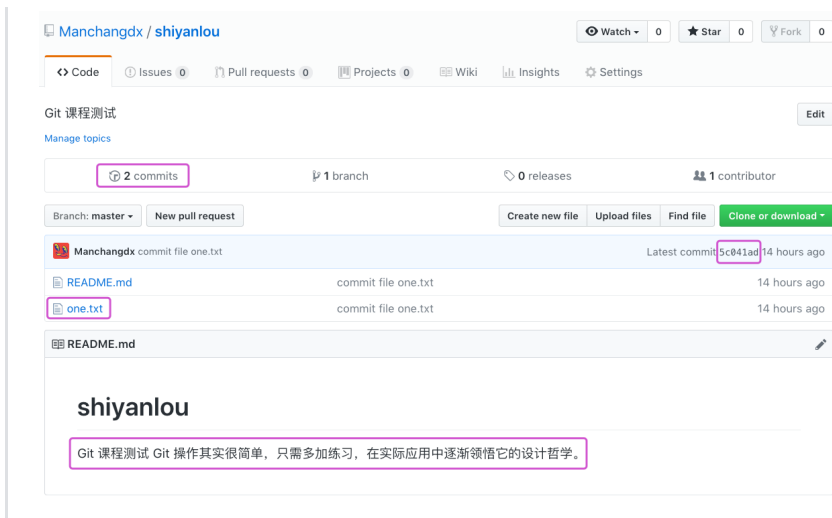
- 上图有三行信息，依次说明：
- 第一行，开头的星号表示当前所在分支，绿色的 `master` 是分支名，之所以是绿色，也是因为它是当前所在分支。后面第二项是版本号，第三项中括号里面蓝色的字，表示此分支跟踪的远程分支的名字，当然啦，这也是克隆远程仓库到本地时的默认设置 -- 创建 `master` 分支并自动跟踪远程同名分支；冒号后面黑色文字表示本地分支领先其跟踪的远程分支一个提交。最后一项是提交时填写的备注信息。
- 第二行，是 Git 指针信息，它指向远程仓库的 `master` 分支，这行信息暂不重要。
- 第三行，远程分支信息，详见第一行的解释。
- 在执行 `commit` 命令时，再介绍一个我并不推荐的选项 `-a`，它的作用是将未添加到暂存区的修改，也就是工作区的修改也一并提交，但会略过未被跟踪的文件，比如新建文件 `one.txt`，此命令的完整格式：`git commit -am xxxxx`。谨慎的做法是按照前文的顺序，修改工作区 - 提交到暂存区 - 随时使用 `git status` 查看仓库状态 - 将暂存区的修改提交到版本区生成一次新的提交。
- 最后一个环节，将本地新增的提交推送到 GitHub 远程仓库中，命令是 `git push`，后面不需要任何选项和参数，此命令会把本地仓库 `master` 分支上的新增提交推送到远程仓库的同名分支上，因为当前所在的分支就是 `master`，而且上文提到，它已经跟踪了远程仓库的同名分支：

```
shiyanolou:shiyanolou/ (master) $ git push
Username for 'https://github.com': Manchangdx
Password for 'https://Manchangdx@github.com':
枚举对象: 6, 完成.
对象计数中: 100% (6/6), 完成.
使用 4 个线程进行压缩
压缩对象中: 100% (3/3), 完成.
写入对象中: 100% (4/4), 431 bytes | 431.00 KiB/s, 完成.
总共 4 (差异 0), 复用 0 (差异 0)
To https://github.com/Manchangdx/shiyanolou.git
 2b96af0..5c041ad master -> master
shiyanolou:shiyanolou/ (master) $
```

- 此命令需要再次输入你的 Github 用户名和密码，密码为隐藏数据，输入时看不到。推送成功后执行 `git branch -avv` 查看分支情况：

```
* master          5c041ad [origin/master] commit file one.txt
remotes/origin/HEAD -> origin/master
remotes/origin/master 5c041ad commit file one.txt
(END)
```

- 如上图所示，本地分支 `master` 与远程分支 `origin/master` 的版本号一致，通常看两个版本号是否一致，只需比对前四位。看一下网页上的情况：



- 完全符合预期。
- 一个小细节，在上图右侧有“14 hours ago”字样，因为这次提交操作是 14 小时前完成的，提交后我睡了一觉，与推送操作的时间无关。
- 以上就是一次完整的修改 - 提交 - 推送操作。一次推送中可以包含多个 git commit 操作，也就是多个提交可以一起推送。
- 如果发现 one.txt 文件内容有误，怎么做？可以修改此文件然后再次添加到暂存区、提交、推送，也可以撤销最近一次提交，修改文件后重新提交推送。现在使用后一种方法来演示撤销提交的操作流程。
- 首先执行 `git reset --soft HEAD^` 撤销最近的一次提交，将修改还原到暂存区。`--soft` 表示软退回，对应的还有 `--hard` 硬退回，后面会讲到，`HEAD^` 表示撤销一次提交，`HEAD^^` 表示撤销两次提交，撤销 `n` 次可以简写为 `HEAD~n`。软退回一个提交后执行 `git branch -avv` 命令查看分支信息：

```
shiyanlou:shiyanlou/ (master) $ git reset --soft HEAD^
shiyanlou:shiyanlou/ (master*) $ git branch -avv
shiyanlou:shiyanlou/ (master*) $ █
```

```
* master          2b96af0 [origin/master: 落后 1] Initial commit
  remotes/origin/HEAD -> origin/master
  remotes/origin/master 5c041ad commit file one.txt
(END)
```

- 可以看到本地仓库的 master 分支的版本号已经发生了变化，变成了前一次提交的版本号，中括号里也有提示信息，本地分支 master 落后其跟踪的远程分支 origin/master 一个提交。
- 执行 `git status` 查看仓库状态，果然上一个提交中的修改全部扔回了暂存区：

```
shiyanlou:shiyanlou/ (master*) $ git status
位于分支 master
您的分支落后 'origin/master' 共 1 个提交，并且可以快进。
（使用 "git pull" 来更新您的本地分支）
```

要提交的变更：
（使用 `"git reset HEAD <文件>..."` 以取消暂存）

```
修改：    README.md
新文件：  one.txt
```

```
shiyanlou:shiyanlou/ (master*) $ █
```

- 再次修改 one.txt 文件，执行 git add . 命令将新的修改添加到暂存区，然后执行 git commit 命令生成新的提交：

```
shiyanolou:shiyanolou/ (master*) $ cat one.txt
hello world
shiyanolou:shiyanolou/ (master*) $ echo 'hello shiyanolou' >> one.txt
shiyanolou:shiyanolou/ (master*) $ cat one.txt
hello world
hello shiyanolou
shiyanolou:shiyanolou/ (master*) $ git add .
shiyanolou:shiyanolou/ (master*) $ git status
位于分支 master
您的分支落后 'origin/master' 共 1 个提交，并且可以快进。
（使用 "git pull" 来更新您的本地分支）

要提交的变更：
（使用 "git reset HEAD <文件>..." 以取消暂存）

    修改：      README.md
    新文件：    one.txt

shiyanolou:shiyanolou/ (master*) $ git commit -m 'commit file one.txt'
[master e290005] commit file one.txt
 2 files changed, 3 insertions(+)
 create mode 100644 one.txt
shiyanolou:shiyanolou/ (master) $ █
```