告诉Git自己是谁

2021年1月20日

22:59

git config --global user.name "ronchy"

git config --global user.email "1781106906@qq.com"

注意！有空格！

注意git config命令的--global参数，用了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置，当然也可以对某个仓库指定不同的用户名和Email地址。

创建版本库（Create a repository）

2021年1月20日

23:01

**第一步：创建一个空的目录**

$ mkdir learngit 创建目录

$ cd learngit 进入learngit目录

$ pwd 显示当前目录

**第二步：通过git init命令把这个目录变成Git可以管理的仓库（初始化）**

$ git init

Initialized empty Git repository **in** /Users/michael/learngit/.git/

瞬间Git就把仓库建好了，而且告诉你是一个空的仓库（empty Git repository），可以发现当前目录下多了一个.git的目录，这个目录是Git来跟踪管理版本库的，没事千万不要手动修改这个目录里面的文件，不然改乱了，就把Git仓库给破坏了。

把文件提交到仓库

2021年1月21日

11:10

learngit目录下，创建文件。(放在目录下的原因：让git能够找到此文件，随后通过git add命令添加到仓库中）

把一个文件放入learngit仓库中：

第一步： $ git add readme.txt （告诉Git，把文件添加到仓库中）

第二步： $ git commit -m"wrote a readme file" （本次提交的说明）

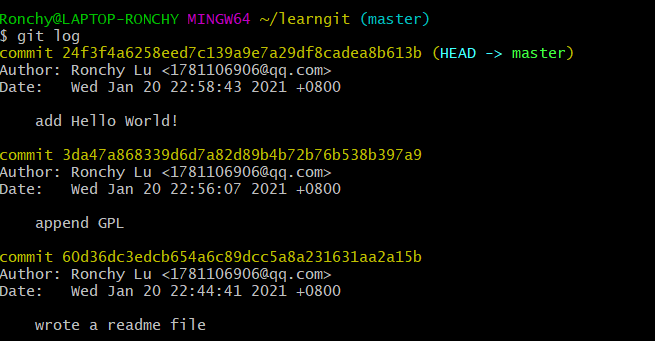
查看日志

2021年1月21日

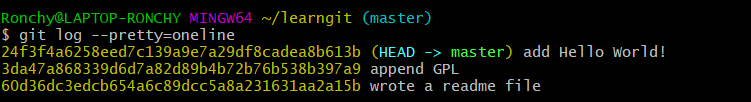
11:03

**查看管理日志：**

$ git log



或 $ git long --pretty=oneline



回退版本

2021年1月21日

11:26

|  |  |
| --- | --- |
| HEAD | 当前版本 |
| HEAD^ | 上个版本 |
| HEAD^^ | 上上个版本 |
| HEAD^^^ | 上上上个版本 |
| HEAD~100 | 前100个版本 |

版本回退指令：

$ git reset --hard HEAD^ (回退到上一个版本)

$cat readme.txt （查看当前内容)

如果后悔了，想恢复到之前的版本：

$ git reset --hard 版本号

e.g. $ git reset --hard 24f3f (不需要全写，前5位即可)

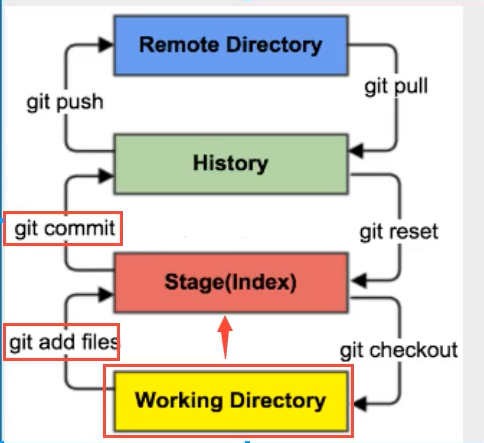
原理: 指针



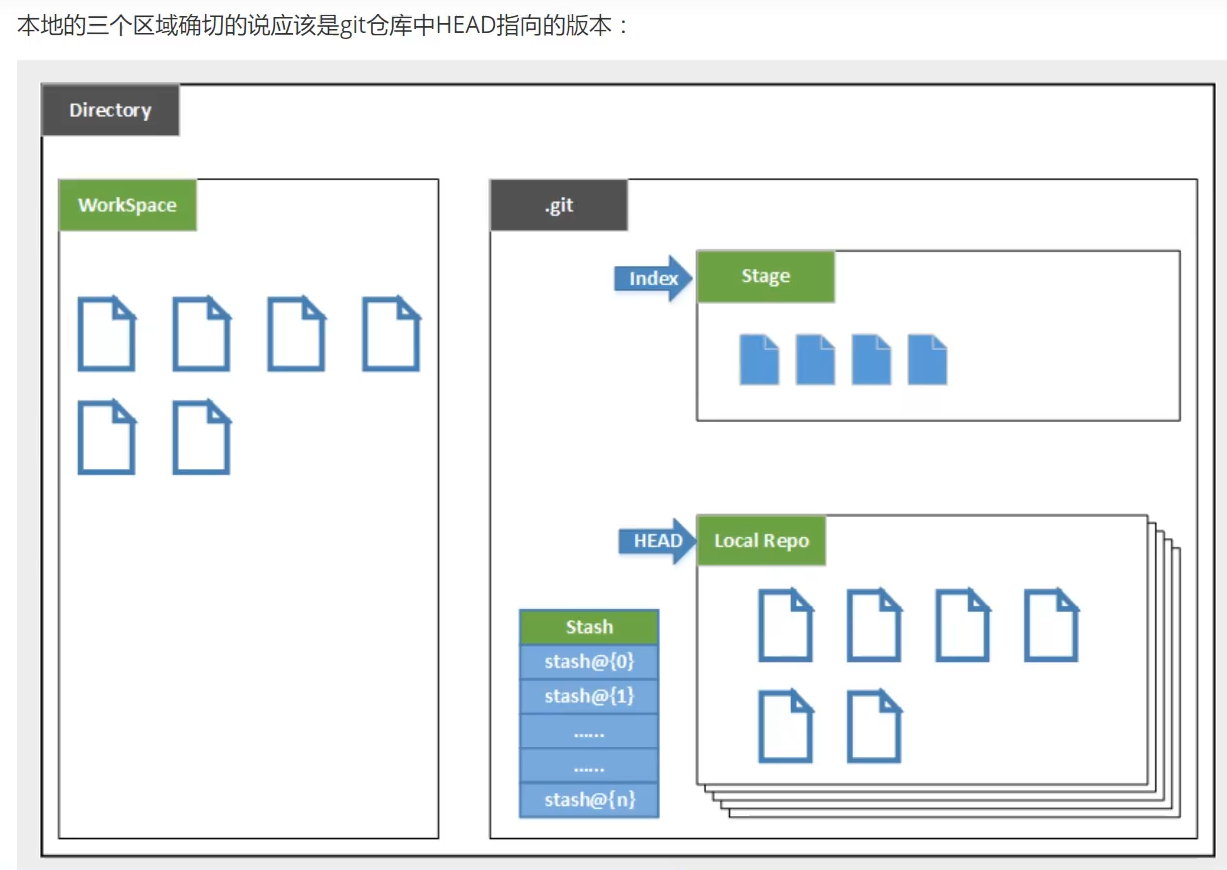
工作区，版本库（stage，master）

2021年1月21日

12:54

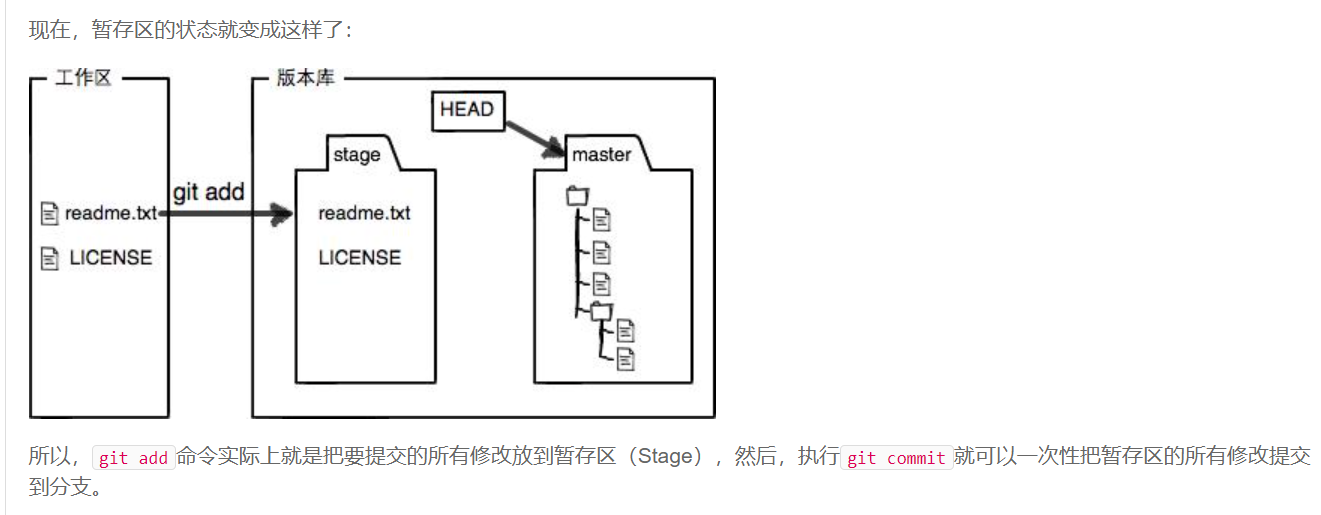


* 工作目录能看到的文件就是工作区。
* stage 暂存区
* master 分支



git add (把工作目录文件放入暂存区）

git commit (把stage中的提交到分支目录里）



git restore readme.txt 恢复工作区的修改

添加到Github远程库

2021年1月24日

12:46

<https://blog.csdn.net/qq_43768946/article/details/90411154>

配置：

$ git config --global --list 查看邮箱是否与Github注册时的一致

$ git config --global user.name "xxxx"

$ git config --global user.email "rongqi1949@gmail.com"

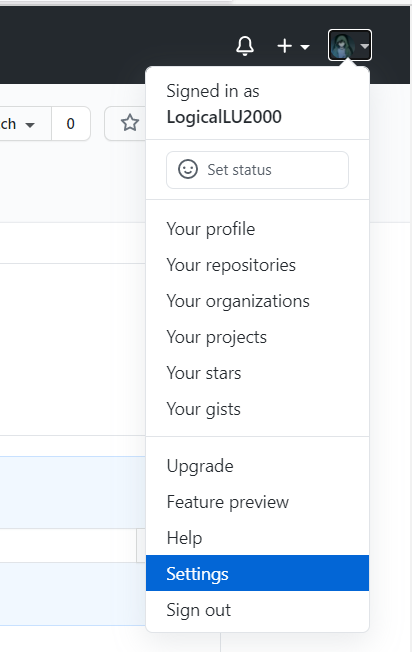
-------------------------------------------------------------------------------------

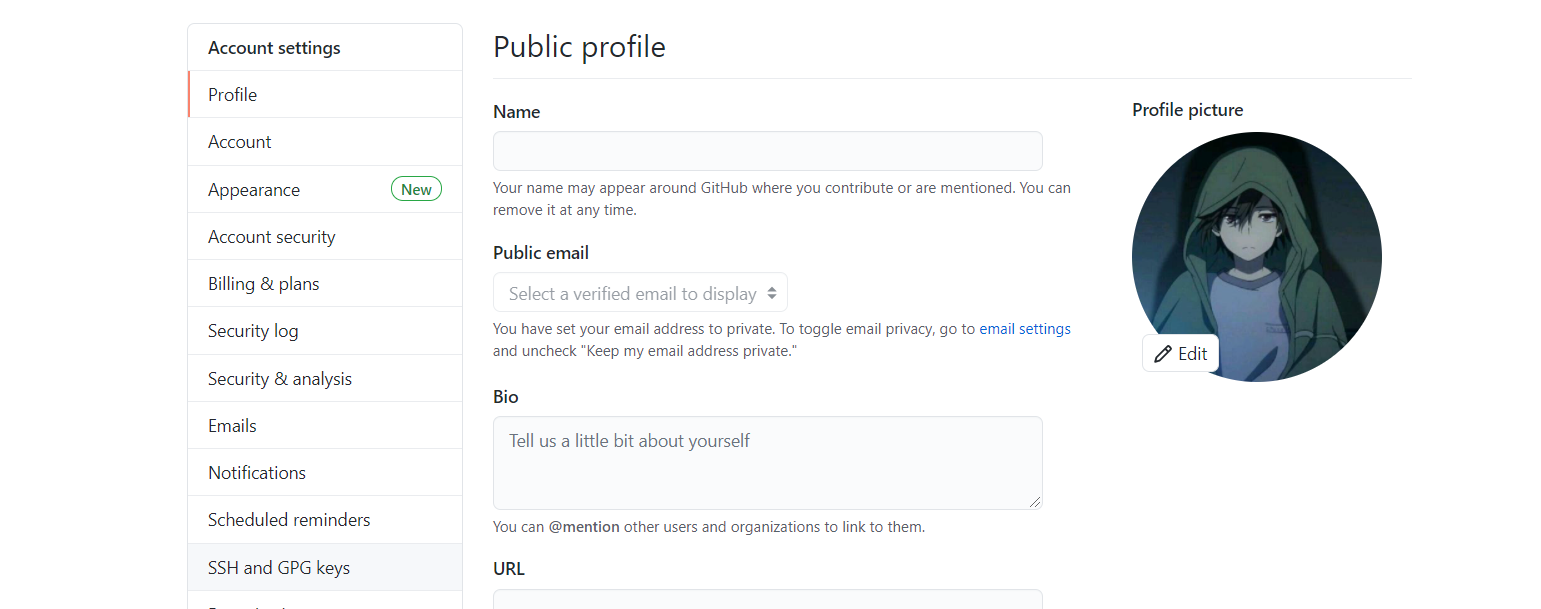
$ ssh-keygen -t rsa -C "rongqi1949@gamil.com" （注意：大写的C !)

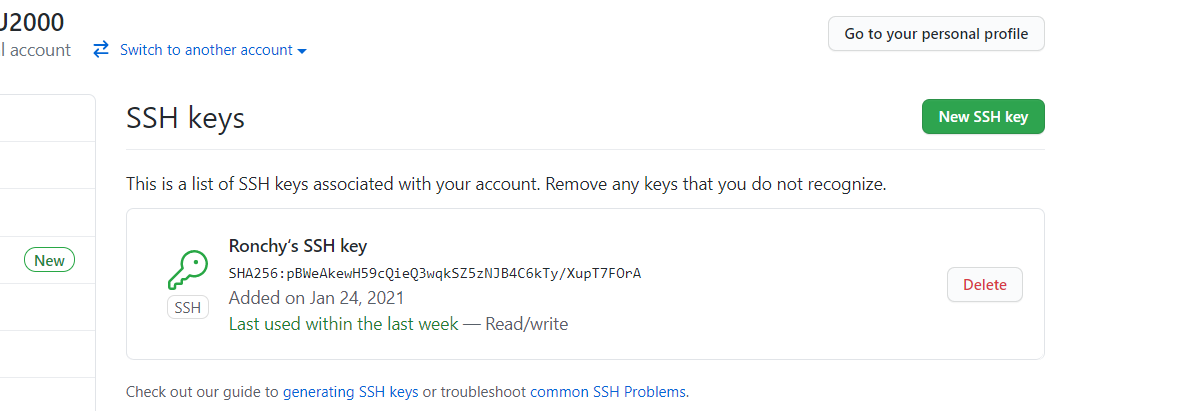
一路回车,直到生成密钥



最后，在git仓库，添加密钥





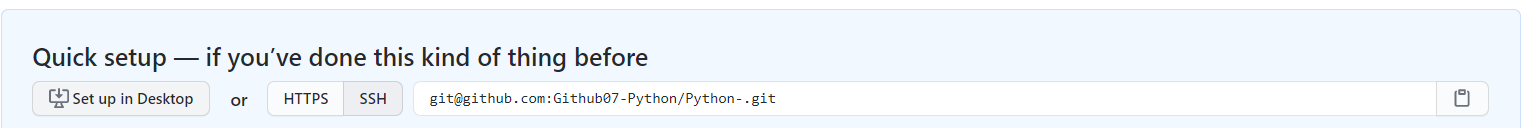


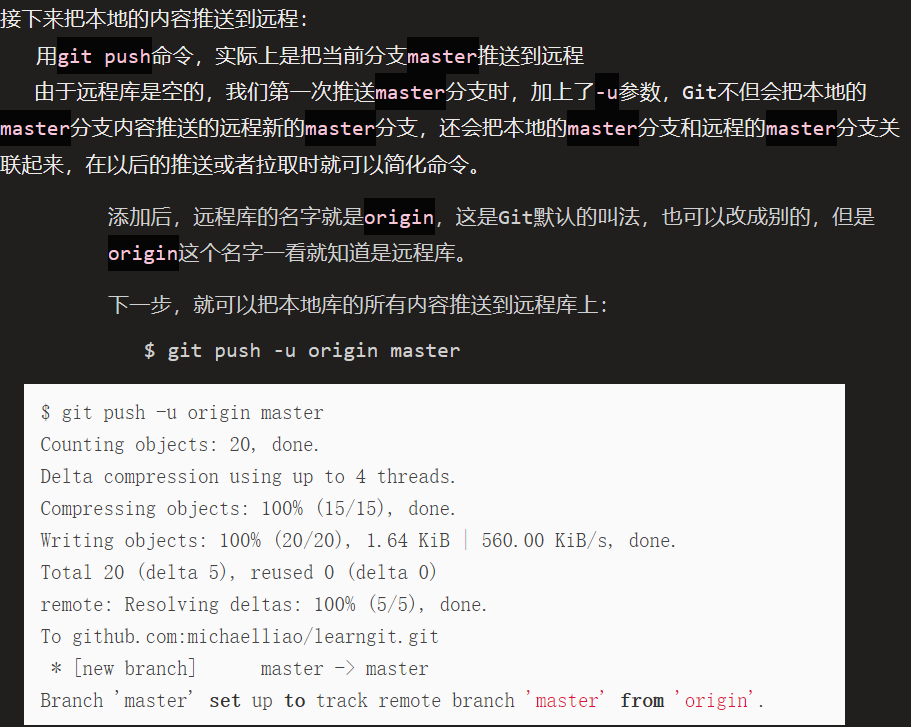
添加即可！至此配置成功！

目前，在GitHub上的这个learnpython仓库还是空的，GitHub告诉我们，可以从这个仓库克隆出新的仓库，也可以把一个已有的本地仓库与之关联，然后，把本地仓库的内容推送到GitHub仓库。

现在，我们根据GitHub的提示，在本地的learnpython仓库下运行命令：

$ git remote add origin git@github.com:Github07-Python/Python-.git





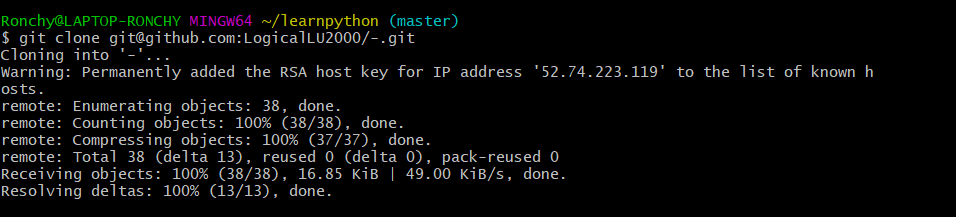
从远程库克隆

2021年1月24日

13:12

命令：

$ git clone git@github.com:LogicalLU2000/-.git



克隆完成！

这是通过ssh克隆，也可以通过http协议克隆。但是ssh最快！

git中文件的4种状态

2021年1月25日

18:30

使用 $ git status [filename] 查看指定文件状态

$ git status 查看所有文件状态

$ git add . 添加所有文件到暂存区

$ git commit -m 提交暂存区的内容到本地仓库 -m 提交信息

|  |  |
| --- | --- |
| Untracked | 未跟踪，此文件在文件夹中，但没有加入到git库，不参与版本控制。通过git add 状态变为 Staged |
| Unmodify | 文件已经入库，未修改。版本库中的文件快照内容与文件夹中完全一致。 两种去处：被修改，变为Modified，使用git rm 移出版本库，则成为Untracked文件 |
| Modified | 文件已修改，仅仅是修改，并没有进行其他操作。 两种去处：通过git add 进入暂存staged状态，使用git checkout 则丢弃修改过，返回到unmodified状态，这个git checkout 即从库中取出文件，覆盖当前修改 |
| Staged | 暂存状态。执行git commit 将修改同步到库中，这时库中的文件＆本地文件又变为一致，文件为Unmodified状态，执行 git reset HEAD filename 取消暂存，文件状态为Modified |

分支

2021年1月26日

16:36

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  | $ git checkout -b [branch name] |
|  |  |  |

|  |  |
| --- | --- |
| $ git merge [dev] | 合并分支到当前分支 |