

Q&A

git commit遇到问题

大部分同学都是第一次使用git，所以第一次使用命令`git commit`时会遇到如下问题：

```
D:\Mycode>git commit -m"first commit"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.
```

原因是没有设置用户信息，Git在每次提交时除了记录批注（`-m "xxx"`）外还会记录提交者，便于版本回退。出现该情况只需要根据提示依次敲入`git config --global user.email xxx`和`git config --global user.name xxx`即可，用户名和邮箱为注册github中户时使用的邮箱

Git有什么用？示范教程中的操作好麻烦啊

- git的作用

git最大的用处的就是版本管理，就像一个时光机，让你随时可以回退到过去任何一次提交时的代码状态，当然也可以穿越回来。譬如今天我一股脑写了一堆代码，但发现没什么用，思路完全错误，但我又不想全部重写，只希望回退到昨天的进度继续写，如果你昨天提交了所有的代码修改，上述愿想便可轻易实现

- git可以提交哪些文件？

除了示范教程中的md文件外，几乎对任何文件的任何修改够可以通过git提交，已完成版本控制。譬如大家项目中最关键的源代码——`.cpp`和`.h`文件等，在本地仓库（示范中的`mycode`文件夹）创建好后，大家可以任意向该文件夹中添加任何与项目相关的文件，并参照示范教程提交到仓库中。

- 如何快速的完成一次git提交

由于示范教程中涉及仓库建立、远程仓库链接、分支改名等复杂操作，整个流程较为繁琐。但其实完成某个修改的提交很快捷，只需要记住“`add-commit-push`”一条龙

快速完成一次提交只需执行以下三条指令：

`git add .`: 将当前工作区（简单理解为当前目录）中的所有修改（包含添加文件、删除文件、编辑文件）添加到暂存区。

`git commit -m "注释"`: 将暂存区的内容提交到本地仓库，并备注上注释。至此，你的所有修改历史已经在本地仓库被小心翼翼的记录下来，足以完成版本回退这样的时空穿梭任务。不过，有人会担心自己的PC遭遇不测导致本地文件损坏，历史记录丢失等等，所以远程仓库横空出世。

`git push`: 将本地仓库同步到远程仓库。至此，所有的修改历史和文件都保存在了云端，不用担心没法儿时空穿梭了。

注：注意到示范教程中同步远程仓库的指令是`git push -u origin main`, 实际上`-u origin main`是可以省略的。第一次执行完`git push -u origin main`后，我们执行过一次`git push --set-upstream origin main`，之后的所有同步就可以省略`-u`参数了，直接敲`git push`回车，搞定！

- 何时进行一次git提交

我们建议大家在敲代码时，实现了某个功能或者某个/些函数，比如排序等并验证了它的正常运作后便可以来一次git提交，老套路：`git add .`、`git commit -m"完成了xxx功能"`、`git push`。搞定！

`git add`, `git commit`等命令还有很多其他用法，有余力的可以自行学习教程<https://www.liaoxuefeng.com/wiki/896043488029600>。当然，你只要掌握了上面三条套路命令，完成本项目就不成问题。