

学习日志

姓名：辛昊洋

学号：1813090

日期：6.15

学习内容安排：

<https://www.liaoxuefeng.com/wiki/896043488029600/897889638509>

536

上午：

Git 简介、安装 Git、创建版本库、版本回退、工作区与暂存区、管理修改、撤销修改、删除文件

下午：

远程仓库、分支管理、标签管理、使用 GitHub、忽略特殊文件、配置别名、使用 SourceTree

学习反馈：

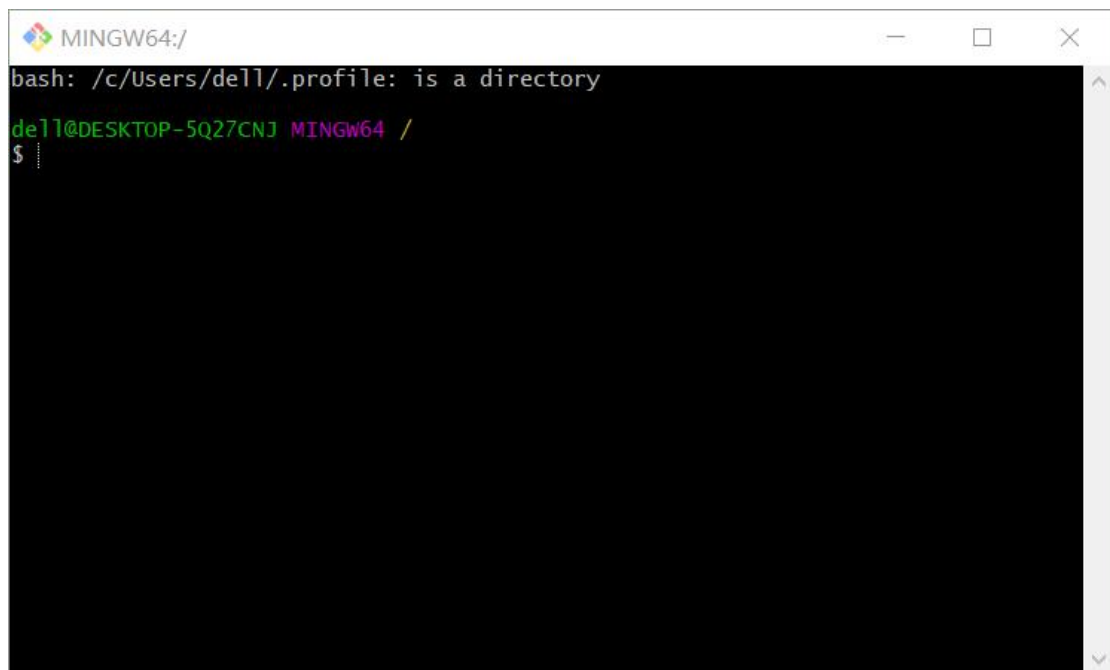
已掌握知识：

一、 Git 简介 Git 是分布式版本控制系统。（不必联网，分制管理强大）

二、 安装 Git：

windows 上安装 git，官网直接下载，默认选项安装。

安装完成后，开始菜单：找到 Git->Git Bash，出现如下窗口，则安装成功。



```
MINGW64:/
bash: /c/Users/dell/.profile: is a directory
dell@DESKTOP-5Q27CNJ MINGW64 /
$
```

设置：

```
$ git config --global user.name "DrPepper8888"
```

```
$ git config --global user.email"1379449332@qq.com"
```

三、 创建版本库

一个目录，这个目录里面的所有文件都可以被 Git 管理起来，每个文件的修改、删除，Git 都能跟踪，以便任何时刻都可以追踪历史，或者在将来某个时刻可以“还原”。

命令： `git init` --变成仓库

添加文件需要以下两步：

前面讲了我们把文件往Git版本库里添加的时候，是分两步执行的：

第一步是用 `git add` 把文件添加进去，实际上就是把文件修改添加到暂存区；

第二步是用 `git commit` 提交更改，实际上就是把暂存区的所有内容提交到当前分支。

- `git add <file>`每次添加一个文件
- 或者： `git add .` 将这个文件全部的东西都添加到仓库
- `git commit -m<message>`文件提交到仓库

```
$ git commit -m "wrote a readme file"
[master 97edddc] wrote a readme file
3 files changed, 6 insertions(+)
create mode 100644 course/~$mplate.docx
create mode 100644 readme1
create mode 100644 readme1.md
```

电脑 > 新加卷 (E:) > Practical_training >			搜索"Prac..."
名称	修改日期	类型	
.git	2020/6/15 11:46	文件夹	
1813090_xhy	2020/6/15 11:35	文件夹	
course	2020/6/15 11:56	文件夹	
readme.md	2020/6/15 11:40	MD 文	

四、 版本回退

`$git status`：时刻掌握仓库当前的状态

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   course/~$mplate.docx
    new file:   readme1.md
```

\$git diff 查看上次怎么修改的那个文件

```
$ git diff readme
fatal: ambiguous argument 'readme': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
```

（意思就是没修改过）

\$git log 查看几个版本被提交到 git 仓库了

```
$ git log
commit 97edddcb5b0f06f095899921f128ff6f5bde2c86 (HEAD -> master)
Author: DrPepper8888 <1379449332@qq.com>
Date:   Mon Jun 15 18:02:32 2020 +0800

    wrote a readme file

commit a0f7e135dcea2ff17ce5ee980ad6a7dfad397d92 (origin/master)
Author: DrPepper8888 <1379449332@qq.com>
Date:   Mon Jun 15 11:45:04 2020 +0800

    loading first time

commit a85e987219fa4bd9f262de93eca863e2073a93a4
Author: DrPepper8888 <1379449332@qq.com>
Date:   Mon Jun 15 11:29:57 2020 +0800

    Update readme.md
```

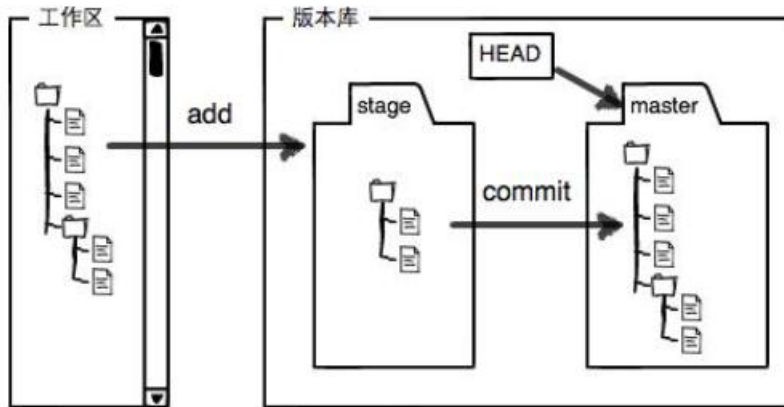
\$git reset - -hard 'commit id'回退到上个版本

\$git reflog 记录每次命令（可以查出来 commit id）

五、 工作区与暂存区

工作区：你电脑里能看到的目录

版本库：



命令 `git add`: 实际上是把修改放到暂存区

命令 `git commit`: 一次性把暂存区所有修改提交到分支

```
$ git commit -m "add a line"
[master a3cd06d] add a line
1 file changed, 2 insertions(+), 1 deletion(-)
```

六、 管理修改

Git 跟踪修改，每次修改，如果不用 `git add` 到暂存区，那就不会加入到 `commit` 中。

七、 撤销修改

```
$ git checkout -- readme.txt
```

让这个文件回到最近一次 `git commit` 或 `git add` 时的状态。

Git 同样告诉我们，用命令 `git reset HEAD <file>` 可以把暂存区的修改撤销掉 (unstage)，重新放回工作区：

八、 删除文件

`$rm file` 工作区删除了

```
dell@DESKTOP-5Q27CNJ MINGW64 /e/Practical_training (master)
$ rm readme1

dell@DESKTOP-5Q27CNJ MINGW64 /e/Practical_training (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    readme1
    deleted:    readme1.md

no changes added to commit (use "git add" and/or "git commit -a")
```

要从版本库删除文件: `$git rm readme1` `$git commit -m "remove readme1"`

```
$ git rm readme1
rm 'readme1'
```

```
$ git commit -m "remove readme1"
[master 494ba27] remove readme1
1 file changed, 4 deletions(-)
delete mode 100644 readme1
```

另一种情况是删错了，因为版本库里还有呢，所以可以很轻松地
把误删的文件恢复到最新版本：

```
$ git checkout -- test.txt
```

其实是用版本库里的版本替换工作区的版本，无论工作区是修改还是删除，都可以“一键还原”。

九、 远程仓库

第 1 步：创建 SSH Key。在用户主目录下，看看有没有 .ssh 目录，如果有，再看看这个目录下有没有 `id_rsa`

和 `id_rsa.pub` 这两个文件，如果已经有了，可直接跳到下一步。如果没有，打开 Shell（Windows 下打开

Git Bash），创建 SSH Key：

```
$ ssh-keygen -t rsa -C "youremail@example.com"
```

你需要把邮件地址换成你自己的邮件地址，然后一路回车，使用默认值即可，由于这个 Key 也不是用于军事目的，所以也无需设置密码。

如果一切顺利的话，可以在用户主目录里找到 `.ssh` 目录，里面有 `id_rsa` 和 `id_rsa.pub` 两个文件，这两个就是 SSH Key 的秘钥对，`id_rsa` 是私钥，不能泄露出去，`id_rsa.pub` 是公钥，可以放心地告诉任何人。

第 2 步：登陆 GitHub，打开“Account settings”，“SSH Keys”页面：

然后，点“Add SSH Key”，填上任意 Title，在 Key 文本框里粘贴 `id_rsa.pub` 文件的内容。

之后：

关联远程库，远程库默认叫做 `origin`

```
git remote add origin git@github.com:DrPepper8888/Practical_training.git
```

本地推送到远程库

```
#git push -u origin master
```

十、 分支管理

要克隆一个仓库，首先必须知道仓库的地址，然后使用 `git clone` 命令克隆。Git 支持多种协议，包括 `https`，但 `ssh` 协议速度最快。

Git鼓励大量使用分支：

查看分支：`git branch`

创建分支：`git branch <name>`

切换分支：`git checkout <name>` 或者 `git switch <name>`

创建+切换分支：`git checkout -b <name>` 或者 `git switch -c <name>`

合并某分支到当前分支：`git merge <name>`

删除分支：`git branch -d <name>`

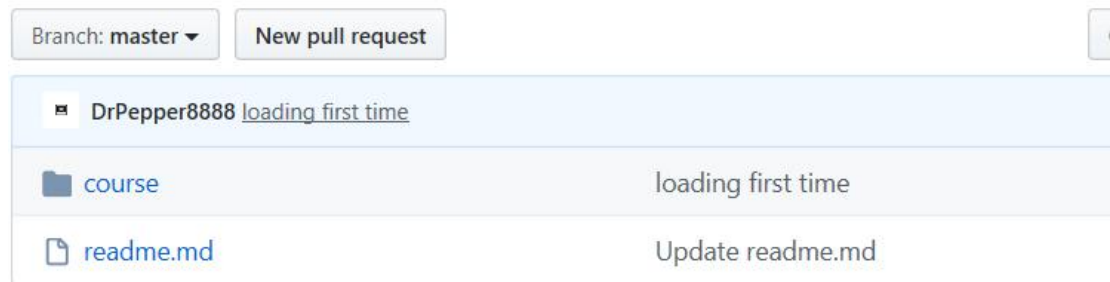
Git 的分支是与众不同的，无论创建、切换和删除分支，Git 在 1 秒钟之内就能完成！无论你的版本库是 1 个文件还是 1 万个文件。

十一、 标签管理

Git 的标签虽然是版本库的快照，但其实它就是指向某个 `commit` 的指针（跟分支很像，但是分支可以移动，标签不能移动），所以，创建和删除标签都是瞬间完成的。

- 命令 `git tag <tagname>` 用于新建一个标签，默认为 `HEAD`，也可以指定一个 `commit id`；
- 命令 `git tag -a <tagname> -m "blablabla..."` 可以指定标签信息；
- 命令 `git tag` 可以查看所有标签。
- 命令 `git push origin <tagname>` 可以推送一个本地标签；
- 命令 `git push origin --tags` 可以推送全部未推送过的本地标签；
- 命令 `git tag -d <tagname>` 可以删除一个本地标签；
- 命令 `git push origin :refs/tags/<tagname>` 可以删除一个远程标签。

十二、 使用 Github



十三、 忽略特殊文件

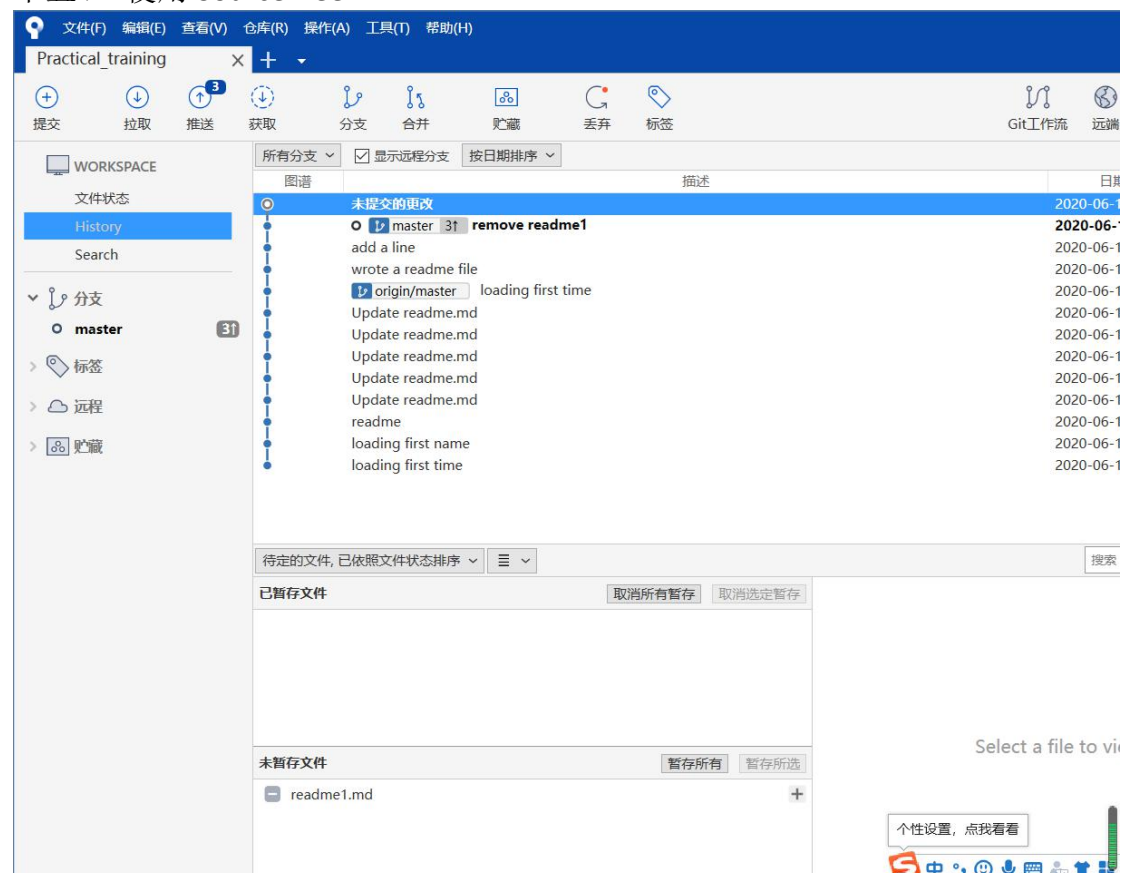
- 忽略某些文件时，需要编写 `.gitignore` ；
- `.gitignore` 文件本身要放到版本库里，并且可以对 `.gitignore` 做版本管理！

忽略文件的规则是：

1. 忽略操作系统自动生成的文件，比如缩略图等；
2. 忽略编译生成的中间文件、可执行文件等，也就是如果一个文件是通过另一个文件自动生成的，那自动生成的文件就没必要放进版本库，比如 Java 编译产生的 `.class` 文件；
3. 忽略你自己的带有敏感信息的配置文件，比如存放口令的配置文件

十四、 配置别名

十五、 使用 SourceTree



未熟悉知识：

分支管理非常精髓，感觉还需要更多的练习才能更深入掌握。
SourceTree 还需要多加应用，以加深理解。

学习心得：

今天关于 Git 的学习非常充实有用，我通过学习和实践建立了小组实训学习的本地 git 版本库，又通过 github 实现了远程仓库，方便小组进行学习和交流。