Git

2019/12/29

学习来源

廖雪峰的Git教程 |https://u.nu/9d-r

一、创建仓库

什么是版本库呢？版本库又名仓库，英文名repository，可以简单理解成一个目录。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”。

Step 1

选择一个合适的地方，创建一个空目录

$ mkdir learngit

$ cd learngit

$ pwd

/Users/michael/learngit

Step 2

通过git init命令把这个目录变成Git可以管理的仓库：

$ git init

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

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。

如果你没有看到.git目录，那是因为这个目录默认是隐藏的，用ls -ah命令就可以看见。

Step 3

测试

编写一个readme.txt文件，内容如下：

> Git is a version control system.
>
> Git is free software.

一定要放到learngit目录下（子目录也行），因为这是一个Git仓库。

1. 用命令git add告诉Git，把文件添加到仓库：

$ git add readme.txt

执行上面的命令，没有任何显示，这就对了，Unix的哲学是“没有消息就是好消息”，说明添加成功。

2. 用命令git commit告诉Git，把文件提交到仓库：

$ git commit -m "wrote a readme file"

[master (root-commit) eaadf4e] wrote a readme file

1 file changed, 2 insertions(+)

create mode 100644 readme.txt

简单解释一下git commit命令，-m后面输入的是本次提交的说明，可以输入任意内容，当然最好是有意义的，这样你就能从历史记录里方便地找到改动记录。git commit命令执行成功后会告诉你，1 file changed：1个文件被改动（我们新添加的readme.txt文件）；2 insertions：插入了两行内容（readme.txt有两行内容）。

嫌麻烦不想输入-m "xxx"，确实可以这么干，但是强烈不建议。因为输入说明对自己对别人阅读都很重要。

二、版本控制

1. 查看状态

- 要随时掌握工作区的状态，使用git status命令。
- 如果git status告诉你有文件被修改过，用git diff可以查看修改内容。

2. 版本回退、恢复

版本控制系统肯定有某个命令可以告诉我们历史记录，在Git中，我们用

$ git log

如果嫌输出信息太多，看得眼花缭乱的，可以试试加上--pretty=oneline参数。

$ git log --pretty=oneline

注：Git必须知道当前版本是哪个版本，在Git中，用HEAD表示当前版本，上一个版本就是HEAD^，上上一个版本就是HEAD^^，当然往上100个版本写100个^比较容易数不过来，所以写成HEAD~100。

3. 现在，我们要把当前版本回退到上一个版本，就可以使用git reset命令：

$ git reset --hard HEAD^

4. 在Git中，当你用$ git reset --hard HEAD^回退到过去版本时，再想恢复到未来版本，就必须找到append GPL的commit id。Git提供了一个命令git reflog用来记录你的每一次命令：

$ git reflog

之后：$ git reset --hard 1094a

- HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令git reset --

hard commit_id。

- 穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。
- 要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。

三、暂存区和工作区

git add 命令实际上就是把要提交的所有修改放到暂存区（Stage），然后，执行 git commit 就可以一次性把暂存区的所有修改提交到分支。

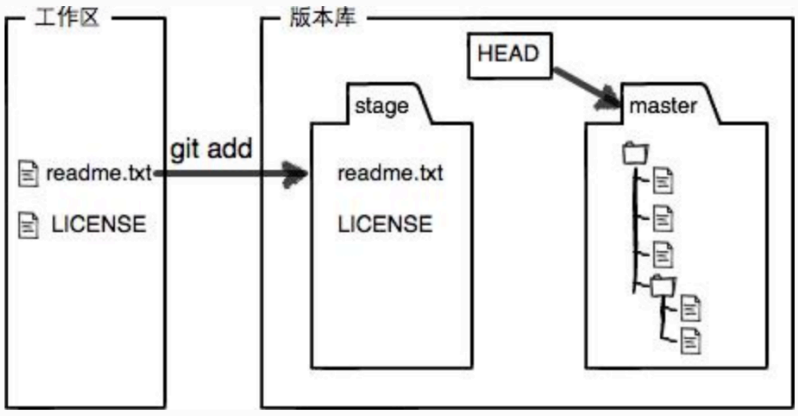
现在，使用两次命令`git add`，把`readme.txt`和`LICENSE`都添加后，用`git status`再查看一下：

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   LICENSE
        modified:   readme.txt
```

现在，暂存区的状态就变成这样了：

现在，暂存区的状态就变成这样了：



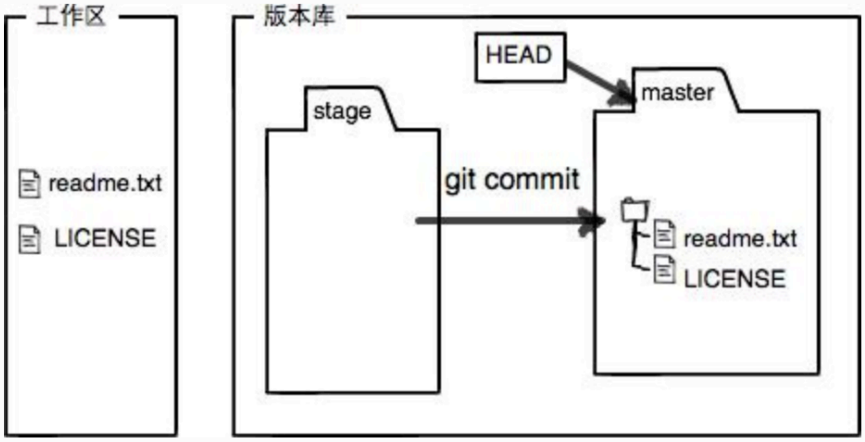
所以，`git add`命令实际上就是把要提交的所有修改放到暂存区（Stage），然后，执行`git commit`就可以一次性把暂存区的所有修改提交到分支。

```
$ git commit -m "understand how stage works"
[master e43a48b] understand how stage works
 2 files changed, 2 insertions(+)
 create mode 100644 LICENSE
```

一旦提交后，如果你又没有对工作区做任何修改，那么工作区就是“干净”的：

```
$ git status
On branch master
nothing to commit, working tree clean
```

现在版本库变成了这样，暂存区就没有任何内容了：



四、撤销修改

- 场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令 git checkout -- file 。
- 场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步。

第一步用命令 git reset HEAD ，就回到了场景1，第二步按场景1操作。

- 场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，参考版本回退一节，不过前提是没有推送到远程库。