

git学习笔记

葛鑫 2018-1-9, 根据廖雪峰git教程整理

git 使用

- 创建一个文件夹, 使用 `git init` 初始化为仓库, 用 `ls -ah` 可查看目录下有了 `.git` 隐藏文件夹。
- 添加文件 `readme.txt`, 使用 `git add readme.txt` 把文件添加到仓库, 使用 `git commit -m "comment"` 提交到仓库
- 修改文件 `readme.txt` 后。使用 `git status` 查看仓库当前状态。告诉我们 `readme.txt` 被修改过, 但还没有提交。
- 使用 `git diff` 查看修改了什么
- 提交修改和步骤2一样, 先 `git add readme.txt`。此时使用 `git status` 将要被提交的修改内容。下一步 `git commit -m "comment"` 提交仓库, 再 `git status` 可以看到 `nothing to commit`
- `git commit` 不添加 `-m` 后会进入到 `vim`` 模式
- 保存一个快照, 这个快照在git中被称为 `commit`, 一旦把文件改乱了, 可以从最近一个 `commit` 恢复。于是可以使用 `git log` 查看历史记录。或 `git log --pretty=oneline` 查看。

```
1 54abd58e04f314f2eb96b1edd3d1a020c6baf8fc haha
2 c27393d5e8cafcd7513c26277574dfd5ba68ca8 wrote a readme file
```

前面是哈希计算的 `id`, 后面是每次 `commit` 时候的 `comment`

- git中 `HEAD` 表示当前版本, 也就是 `54ab...`, 上一个版本是 `HEAD^`, 上上个就是 `HEAD^^`, 上100个版本是 `HEAD~100`, 现在要回退到 `c273..` 版本, 使用命令 `git reset --haed HEAD^`。
- 此时用 `git log` 查看历史信息, 之前最新的版本已经被抹去了, 但还想要再回去的话, 在命令行窗口找到 `haha` 的 `commit id`, 也就是 `54abd...`, 于是就可以回到未来的指定版本 `git reset --hard 54abd`, 只需要前几位就可以了。历史提交的 `commit` 是一个链表, `HEAD` 是头指针。如果找不到命令行历史提交记录了, 可以用 `git reflog` 命令, 它记录了每一次命令, 则可以找出来以前的id。

git 原理

- 工作区(Working Directory): 即工作目录, 文件夹。

- 版本库(Repository): 工作区有一个隐藏文件夹 `.git`, 就是版本库, 其中最重要是称为 `stage` 的暂存区
 - 第一步 `git add` 时, 实际上把文件添加到暂存区
 - 第二部 `git commit`, 把暂存区所有内容提交到当前分支。

当我们创建git版本库时, git自动为我们创建了唯一一个 `master` 分支。需要提交的文件修改通通放到暂存区, 然后, 一次性提交暂存区的所有修改。

`git diff` 比较的是工作区文件与暂存区文件的区别 (上次git add 的内容) `git diff --cached` 比较的是暂存区的文件与仓库分支里 (上次git commit 后的内容) 的区别

`git diff HEAD -- filename` 是只比较工作区和版本库 (最后一次commit) 的区别

- 如果 第一次修改 -> `git add` -> 第二次修改 -> `git commit`, 第二次修改不会被添加到当前分支, 因为第二次修改没有被 `git add` 到暂存区。
- 假如在 `readme.txt` 进行了修改, 在 `add` 和 `commit` 前想要撤销, 使用 `git checkout -- readme.txt`
 - 命令 `git checkout -- readme.txt` 意思就是, 把 `readme.txt` 文件在工作区的修改全部撤销, 这里有两种情况:
 - 一种是 `readme.txt` 自修改后还没有被放到暂存区, 现在, 撤销修改就回到和版本库一模一样的状态;
 - 一种是 `readme.txt` 已经添加到暂存区后, 又作了修改, 现在, 撤销修改就回到添加到暂存区后的状态

总之, 就是让这个文件回到最近一次 `git commit` 或 `git add` 时的状态。
- 对 `readme.txt` 修改, 且已经 `git add`, 使用 `git reset HEAD readme.txt` 把暂存区修改撤销掉。接着 `git checkout -- readme.txt` 撤销工作区修改。
- 若已经 `git commit` 了, 则回退版本。
- 删除一个文件
 - 首先 `rm test.txt`, 再 `git rm test.txt`, 再 `git commit`
 - 如果删错了, 从版本库恢复 `git checkout -- test.txt`

远程仓库

- 创建ssh key: `ssh-keygen -t rsa -C "iamgexin@qq.com"`, 一路确认, 最后会在用户目录找到 `.ssh` 里找到 `id_rsa` 和 `id_rsa.pub`。
 - `id_rsa` 是私钥, `id_rsa.pub` 是公钥。前者自己保管, 后者可以告诉任何人。
- 登陆GitHub, settings, ssh and GPG keys, 添加公钥。
- 创建一个repo, 根据提示
 - `git remote add origin https://github.com/dovetion/fwd1_dev.git`

- `git push -u origin master` //第一次push加 `-u`
- `origin` 是默认名字
- 把本地内容推送到远程上，用 `git push` 命令，实际上是把当前分支 `master` 推送到远程
- 以后每次本地提交后，用 `git push origin master` 推送最新修改。
- 克隆就不在此记录了

分支管理
