<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>

（廖雪峰Git教程）



Git是什么？

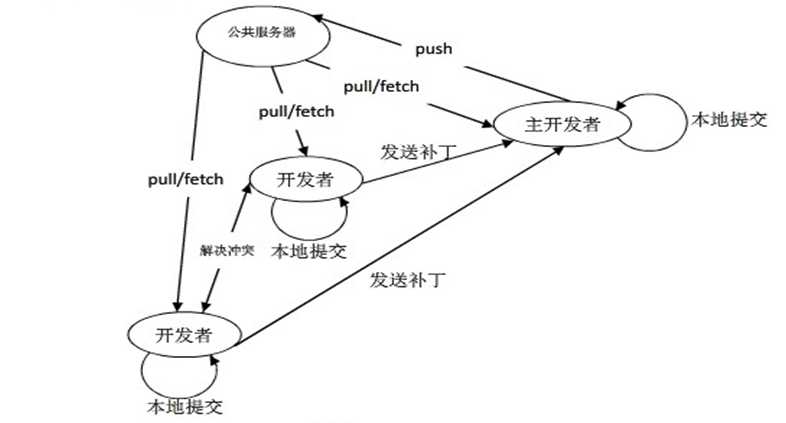
⭐️世界上最先进的分布式版本控制系统

拓展：**版本控制工具（CVS、SVN、GIT）简介**

<https://blog.csdn.net/wanglin_lin/article/details/48845343>

集成式（CVS、SVN）：所有的版本信息都放在服务器上。只要一commit，更改就已经提交到服务器。（无网络时不可以提交）

分布式（Git）：（下图）在本地做修改后，先commit到本地的版本库，然后push到（公共）服务器。（无网络时也可以提交到本地镜像，待有网时再push到服务器）



安装Git：

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/00137396287703354d8c6c01c904c7d9ff056ae23da865a000>

【拓展】

1. 安装了homebrew✔️

homebrew的使用：<https://zhuanlan.zhihu.com/p/46239814>

2. Xcode集成了Git。Xcode是Apple官方IDE，功能非常强大，是开发Mac和iOS App的必选装备，而且是免费的！

————————————————————————————

**创建版本库**

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013743256916071d599b3aed534aaab22a0db6c4e07fd0000>

Sum：

使用git init命令，初始化一个Git仓库。

添加文件到Git仓库，分两步：

1. 使用命令git add <file>，注意，可反复多次使用，添加多个文件；
2. 使用命令git commit -m <message>，完成。

版本库：又名仓库，英文名**repository**，你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，创建空目录：

先手动创建一个合适位置的文件夹（桌面上新建文件夹“GitHub”）

$ cd Desktop/GitHub

$ mkdir learngit

$ cd learngit

$ pwd

/Users/mac/Desktop/GitHub/learngit

Step 2，用git init命令将这个目录变成一个Git可以管理的repo

$ git init

Initialized empty Git repository **in** /Users/mac/Desktop/GitHub/learngit/.git/

注：所有的版本控制系统包括Git，只能跟踪文本文件的改动，比如TXT文件，网页，所有的程序代码等。图片、视频这些二进制文件只能用Git管理文件，无法跟踪具体改动。

**把文件添加到版本库**

**先编写一个文本文件**

具体操作。在learngit目录下新建一个纯文本文件（readme.txt），内容如下：

Git **is** a version control system.

Git **is** free software.

注：一定要放到Git仓库（learngit）目录下，子目录也行

* **Step 1，添加（add）文件到repo**

$ git add readme.txt

# Unix的哲学是“没有消息就是好消息” （hhhh

* **Step 2，把文件提交（commit）到repo**

$ git commit -m “wrote a readme file”

返回：

[master (root-commit) 2d1af90] wrote a readme file

1 file changed, 2 insertions(+)

create mode 100644 readme.txt

注：-m后面输入的是本次提交的说明，以后能从历史记录里方便地找到改动记录。

补充1：GitHub官方指南里对commit的定义：

On GitHub, saved changes are called commits. Each commit has an associated commit message, which is a description explaining why a particular change was made. Commit messages capture the history of your changes, so other contributors can understand what you’ve done and why.

补充2：为什么Git添加文件需要add，commit一共两步呢？因为commit可以一次提交很多文件，所以可以多次add不同的文件，比如：

$ git add file1.txt

$ git add file2.txt file3.txt

$ git commit -m "add 3 files."

————————————————————————————

**时光机穿梭**

查看repo的当前状态：

$ git status

查看repo做了哪些修改：

$ git diff

————————————

**版本回退**

穿梭前，查看当前版本的commit历史，以便确定要回退到哪个版本：

$ git log

返回上一版本

$ git reset ——hard HEAD^

上上一版本：HEAD^^     以此类推……

上100个版本：HEAD~100

或者（可以重返较新的版本）

$ git reset ——hard commit\_id

查看【命令历史】，得到各版本id（确定要回到未来的哪个版本）：

$ git reflog

查看文件内容：

$ cat readme.txt

————————————

**工作区 v. 暂存区**

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013745374151782eb658c5a5ca454eaa451661275886c6000>

**工作区**：

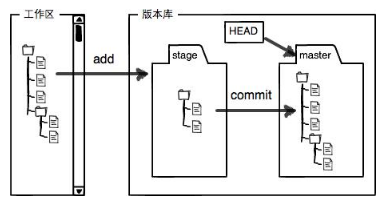
就是你在电脑里能看到的目录，比如learngit文件夹就是一个工作区。

**暂存区**：

**工作区**有一个隐藏目录.git，这个不算工作区，而是Git的**版本库**。

Git的版本库里存了很多东西，其中最重要的就是称为stage（或者叫index）的**暂存区**，

⭐️暂存区是Git非常重要的概念，弄明白了暂存区，就弄明白了Git的很多操作到底干了什么。



————————————

**撤销修改**

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/001374831943254ee90db11b13d4ba9a73b9047f4fb968d000>

情况一：

对工作区的内容（这里是readme.txt）进行了修改，但没有add到暂存区更没有commit到版本库

solution：丢弃对工作区的修改（回到最近一次git commit的状态）

$ git checkout -- file

情况二：

把修改add到暂存区了，但没有commit到版本库

solution（两步）：

step1. 先回退到情况一（文件内容有修改，但未add到暂存区，即 Changes not staged for commit）

git reset HEAD <file>

step2. 重复情况一的操作（checkout）

情况三：

已经把修改commit到版本库，还好还没有推送远程版本库

solution：回退到上一版本

$ git reset -hard HEAD^/commit\_id

注：修改和撤销全程中都可以时不时git status一下~

————————————

**删除文件**

删除文件库中的文件

$ rm test.txt

接着，从版本库中删除该文件

$ git rm test.txt

然后提交

$ git commit -m “remove <filename>"

删错了怎么办？

思路：版本库里还有，git checkout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”。

$ git checkout —— <filename>

————————————

**远程仓库**

注：这部分开始讲GitHub

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/001374385852170d9c7adf13c30429b9660d0eb689dd43a000>

主要内容：

1. 注册GitHub账号
2. 设置（添加）SSH Keys

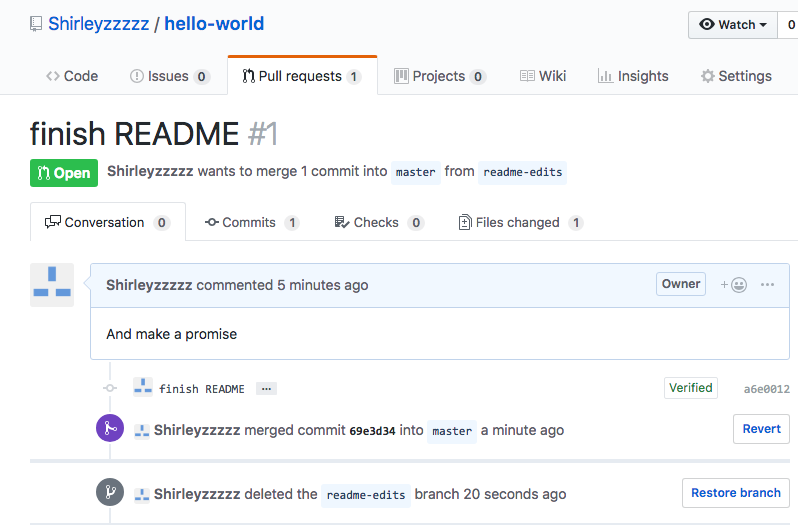
我的GitHub账号：

* [zhongchenli@pku.edu.cn](mailto:zhongchenli@pku.edu.cn)
* 常用的password

GitHub官网使用指南（12.7读完+实操）

<https://guides.github.com/activities/hello-world/>

练习结果页面：<https://github.com/Shirleyzzzzz/hello-world/pull/1>



拓展：

GitHub Explore：参与开源项目

<https://github.com/explore>

————————————

**添加远程库**

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013752340242354807e192f02a44359908df8a5643103a000>

已经有了本地库，要关联到远程库：

$ git remote add origin http://github.com/Shirleyzzzzz/learngit.git

关联后，（第一次）推送master分支的所有内容

$ git push -u origin master

此后，每次本地提交后，推送最新修改到GitHub：

$ git push origin master

分布式版本系统的意义在于（最大好处之一）：

在本地工作完全不需要考虑远程库的存在，也就是有没有联网都可以正常工作，而SVN在没有联网的时候是拒绝干活的！当有网络的时候，再把本地提交推送一下就完成了同步，真是太方便了！

————————————

**从远程库克隆**

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/001375233990231ac8cf32ef1b24887a5209f83e01cb94b000>

从零开发时，最好的方式是先创建远程库，然后从远程库克隆。

第一步：在GitHub里创建一个new repo（准备好远程库）

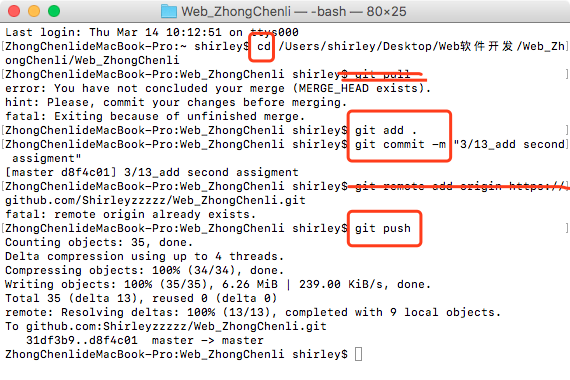
第二步：用命令行克隆一个本地库：

$ git clone  https://github.com/Shirleyzzzzz/gitskills.git（repo的地址，此处以名为gitskills的repo为例）

——————————————

**将代码发布到github**

Web开发课的GitHub链接：  
https://github.com/Shirleyzzzzz/Web\_ZhongChenli.git

3.14上传到远程库操作：  


## **介绍1**

如果是从零开发，那么最好的方式是先创建远程库，然后，从远程库克隆。

首先，登陆GitHub，创建一个新的仓库，名字叫gitskills：



勾选Initialize this repository with a README，这样GitHub会自动创建一个README.md文件。创建完毕后，可以看到README.md文件：



远程库已经准备好了，下一步是用命令git clone克隆一个本地库：

$ git clone git@github.com:michaelliao/gitskills.gitCloning into 'gitskills'...remote: Counting objects: 3, done.remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 3Receiving objects: 100% (3/3), done.

注意把Git库的地址换成自己的，然后进入gitskills目录看看，已经有README.md文件了：

$ cd gitskills

$ ls

README.md

如果有多个人协作开发，那么每个人各自从远程克隆一份就可以了。

GitHub给出的地址不止一个，还可以用https://github.com/michaelliao/gitskills.git这样的地址。实际上，Git支持多种协议，默认的git://使用ssh，但也可以使用https等。

使用https除了速度慢以外，还有个最大的麻烦是每次推送都必须输入口令，但是在某些只开放http端口的公司内部就无法使用ssh协议而只能用https。

### 小结

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，但通过ssh支持的原生git协议速度最快。

参考：廖雪峰-从远程库克隆

https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/001375233990231ac8cf32ef1b24887a5209f83e01cb94b000

## **介绍2**

在github上建立repository以后，要把这个repository下载下来。

首先在本地上建立一个文件夹，进入文件夹以后，git init

再执行 git clone https://github.com/xxxxx/xxx.git

这一步将project下载下来了。

然后进行一些改动后，要把它push上去

执行git add .

git commit -m "change description"

git remote add origin https://github.com/xxxx/xxx.git 要输入帐号及密码。

git push

参考：https://blog.csdn.net/pyufftj/article/details/55802671