一、 目标：

1. Learn the knowlege of git. (*)
2. Use github. (*)

二、 学习材料：

廖雪峰 git 教程。

网址：https://www.liaoxuefeng.com/wiki/896043488029600

三、 学习笔记：

1. Git 是分布式版本控制系统（软件），SVN 集中式版本管理软件。
2. 安装 git. 配置 git 环境 $ git config –global(local) user.name "Your Name" $ git config –global(local) user.email "email@example.com".
3. 创建版本库，repository，git 无法跟踪二进制文件的变化，只知道有改变，要真正使用版本控制系统，就要以纯文本方式编写文件。git init (.git 文件夹，不要动，有这个文件才是 git repository)，git add <file>，git commit –m <message>
4. git status: 查看 git 状态，git diff: 查看修改的内容(默认工作区与暂存区)。
5. 一个 commit 理解为一个快照，git log: 查看提交的信息，commit id，HEAD 指针指向当前分支的一个 commit，HEAD^, HEAD~2，git reset --hard <commit id>/<HEAD~number>：回到某个 commit 版本，git reflog: 查看命令历史，以便确定要回到未来的哪个版本。
6. 工作区：working directory，就是你在电脑里能看到的目录。暂存区：stage（index）add 后的文件在暂存区中，版本库。
7. git 是管理修改的。
8. git checkout -- file: 让这个文件回到最近一次 git commit 或 git add 时的状态, git reset HEAD <file> 可以把暂存区的修改撤销掉（unstage），重新放回工作区。
9. 删除文件：要从版本库中删除该文件，那就用命令 git rm <file>删掉，并且 git commit。git checkout -- file,还原被删文件，从版本库或暂存库中还原。
10. Git 支持多种协议，默认的 git://使用 ssh，但也可以使用 https 等其他协议，但通过 ssh 支持的原生 git 协议速度最快。
11. 要关联一个远程库，使用命令 git remote add origin git@server-

name:path/repo-name.git；或 https://…… 关联后，使用命令 git push -u origin master 第一次推送 master 分支的所有内容；此后，每次本地提交后，只要有必要，就可以使用命令 git push origin master 推送最新修改。origin 表示远程库。

12. git clone 命令
13. 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>，如果要丢弃一个没有被合并过的分支，可以通过 git branch -D <name>强行删除。
14. 当 Git 无法自动合并分支时，就必须首先解决冲突。解决冲突后，再提交，合并完成。解决冲突就是把 Git 合并失败的文件手动编辑为我们希望的内容，再提交。用 git log --graph 命令可以看到分支合并图。
15. 开在实际开发中，我们应该按照几个基本原则进行分支管理：首先，master 分支应该是非常稳定的，也就是仅用来发布新版本，平时不能在上面干活；干活都在 dev 分支上，也就是说，dev 分支是不稳定的，到某个时候，比如 1.0 版本发布时，再把 dev 分支合并到 master 上。
16. 合并分支时，加上--no-ff 参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而 fast forward 合并就看不出来曾经做过合并。
17. 当手头工作没有完成时，先把工作现场 git stash 一下，然后去修复 bug，修复后，再 git stash pop，回到工作现场，git stash list。
18. 在 master 分支上修复的 bug，想要合并到当前 dev 分支，可以用 git cherry-pick <commit>命令，把 bug 提交的修改“复制”到当前分支，避免重复劳动。
19. 查看远程库信息，使用 git remote -v；从本地推送分支，使用 git push origin branch-name，如果推送失败，先用 git pull 抓取远程的新提交；在本地创建和远程分支对应的分支，使用 git checkout -b branch-name origin/branch-name，本地和远程分支的名称最好一致；建立本地分支和远程分支的关联，使用 git branch --set-upstream branch-name origin/branch-name；从远程抓取分支，使用 git pull，如果有冲突，要先处理冲突。
20. git rebase 操作可以把本地未 push 的分叉提交历史整理成直线。
21. tag 就是一个让人容易记住的有意义的名字，它跟某个 commit 绑在一起。

22. 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；用命令 git show <tagname>可以看到说明文字。
23. 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。
24. .gitignore 文件。
25. git push origin ： refs/for/<branchname>
26. fork, pull request.