Git 101

课前阅读材料

- Beginner's Guide to using Git
- git - 简明指南

课堂材料

使用Git管理本地的一个目录

- Git仓库：
  - 工作区
  - 版本库
  - 暂存区
- Git命令：
  `init`, `status`, `add`, `commit`, `log`, `diff`
- Shell命令：
  `cd`, `ls`, `mkdir`, `touch`, `pwd`

（练习一）

在以下练习中请尽可能使用你学到的Shell命令

1. 在本地创建一个名为 `YOUR_NAME-learning-git` 的目录，并用Git管理目录；
2. 在目录中创建一个 `exercise-1.txt` 的文件，在其中添加任意文字，
   并将添加的文字提交到版本库；
3. 分别创建 `exercise-2.txt`, `exercise-3.txt` 文件，在其中添加任意文字，
   并将添加的文字提交到版本库；
4. 对 `exercise-3.txt` 做任意修改
   使用 `git diff` 命令查看变化，
   然后将变化提交到版本库，请注意使用有意义的提交信息；
5. 反复重复练习上一步，直到你非常熟悉上一步的操作，
   你可以尝试在一个提交中修改多个文件。

管理修改

- Git：
  - HEAD
  - master
  - Commit hash
- Git命令：
  `checkout`, `reset`, `reset --hard`

（练习二）

1. 任意修改`exercise-2.txt`文件，
   然后使用Git命令撤销修改，
   注意在过程中使用恰当的Git命令查看状态和比较变化；
2. 任意修改`exercise-2.txt`文件，添加到Staging区，
   然后使用Git命令撤销修改， 注意在过程中使用恰当的Git命令查看状态和比较变化；
3. 任意修改`exercise-2.txt`文件，提交修改，
   然后使用Git命令撤销修改， 注意在过程中使用恰当的Git命令查看状态和比较变化；
4. 重复练习以上三个题目，直到你非常熟悉这些操作；
5. 使用恰当的Git命令，查看各个提交的历史状态，在各个历史提交中反复切换；

远端仓库

- Git：
  - origin
  - conflict
- Git命令：
  `clone`, `remote`, `push`, `pull --rebase`, `rebase`

（练习三）

1. 在本地工作区任意创建三个文件，分别写入任意内容，提交，并推送到GitHub；
2. 如果推送被远端拒绝，请阅读Git的输出信息，然后做适当处理，再推送到远端；
3. 拉取GitHub上的提交到本地；如果此时你能看到别人提交的文件，
   请在别人提交的文件上做任意修改，并将修改提交后推送到远端；
   如果此时你看不到任何别人的提交，请稍等一会，再次拉取GitHub上的提交到本地；
4. 如果推送被远端拒绝，请阅读Git的输出信息，然后做适当处理，再推送到远端；
5. 如果在这个过程中遇到冲突，请解决冲突；
6. 重复以上练习，直到你非常熟悉这些操作。

Workflow

1. git pull --rebase
2. (Make some changes)
3. git add && git commit
4. git pull --rebase
5. git push

这只是众多Git工作流实践中的一种，
我们在GTB期间统一使用这种流程来工作，
每个项目会选择自己的Git工作流实践方式，
包括但不限于：

- Trunk-based
- GitLab Workflow
- GitHub Workflow
- Gitflow

它们的基本思想是一致的：

1. 确保追溯一切变化；
2. 便于回退历史版本；
3. 保护公共分支不被污染；
4. 方便引入变化；

GitHub

- GitHub：
  - creating repository
  - repository access rights
- SSH：
  - Secure Shell
  - SSH-Key pair
- Shell命令：
  `less`，`cat`，`ls -al`

（练习四）

1. 在你的GitHub账号下创建一个空仓库，**设置为私有**；
2. 在本地生成ssk-key，并将公钥配置到GitHub上；
3. 将你创建的仓库克隆到本地；
4. 在本地，向仓库任意提交一些修改，并推送到远端；
5. 重复上一步，使你的仓库中至少有三个提交；

补充材料

> 本节课只包含了 Git 最基本的概念和操作，在实际项目中，你常常会遇到更进阶的 Git 技能，如 Branch, Rebase, Cherry-pick, Stash 等等。熟练地使用这些技能（Git 命令），会让你的工作效率提高很多，至少会让你在日常与人 Pair 工作的过程中显得非常资深。

- explainshell.com
- Markdown
- dangitgit.com
- learngitbranching
- explain-git
- SSH原理与运用
- 公开密钥加密