Git练习题

说明：

访问练习git库的方法：

服务器： 192.168.1.81

Git repository: git@192.168.1.81:chenx/git-exercise.git

由于Git客户端工具有很多种，图形化的：sourcetree和 tortoisegit，命令行的：portableGit。可根据自己的需要，选择合适的工具。建议前期为了更好的理解Git的工作原理，建议用命令行的方式。熟悉了原理后可以用图形化的客户端。

练习1：

在你的本机安装git客户端工具：每个平台可以根据自己的喜好任选一个或者多个。

Windows环境：TortoiseGit(图形工具)，SourceTree（图形工具），PortableGit（命令行工具）

MacOS环境：SourceTree（图形工具）Xcode common tools（命令行工具）

注意：SourceTree需要依赖于命令行工具，所以在安装sourcetree时，需要先安装命令行工具。

练习2：

创建你的SSHKEY，并将SSH的公钥导入到GitLab中你的账户下的SSHkey中，设置Git客户端参数。

1、创建你的SSHKey，可以通过sourcetree：工具->创建或导入秘钥; TortoiseGit: start->TortoiseGit->PuTTYgen; 命令行工具：ssh-keygen -t rsa -C “your email address”

2、用你的账户登录GitLab，导入第一步保存的SSH公钥。

3、配置Git客户端工具参数：用户名、邮件地址 、SSH私钥路径。SourceTree：工具->选项->一般tab页内；TortoiseGit:开始菜单->TortoiseGit->Setting->git栏中。

注意：选择全局选项

命令行：

$ git config --global user.name "FirstName Lastname"

$ git config  --global user.email youremailaddress@company.com

 其他选项可以根据自己的喜好设置。不做强制要求，一般采用系统默认参数。

练习3：

克隆远程版本库，查看相关git库信息。

1、 创建一个exercises的目录：

 $ mkdir ~/exercises

2、 进入~/exercises目录

3、克隆版本，可以采用图形界面工具也可以用命令行。

SourceTree：克隆/新建->克隆仓库页，输入相关信息，克隆远程库。

TortoiseGit: 在新建的目录中右键操作，选择Git克隆。

命令行：$ git clone git@192.168.1.81:chenx/git-exercise.git

3、 检查你的版本库信息

$ cat ~/.gitconfig

练习4： 练习常用的git命令/操作：add，commit，status，rm

1. 在Git库中创建一个文件
2. 查看git库的状态变化，SourceTree/TortoiseGit会标识该文件没有被跟踪（请说明如何标识的）。命令行可利用status命令查看。
3. 添加我文件到暂存库，并查看库的变化（请指出变化）
4. 提交文件到本地版本库，查看库的变化（请指出变化）
5. 删除该文件，并提交到版本库，指出版本库的变化。

 练习5

在这个练习中，我们主要关注如何恢复(reset)你的工作目录(working directory）和索引(index)。这在出问题后我们想重新开始的时候是很有用的。我们将实践增加和删除文件。

注意图形界面中实现此练习，相对简单且明显，所以在此不做多讲述，可自行完成。



练习5a

创建一个新文件并暂存到index中。同时从index中取消暂存。

1、 进入代码目录，例如~/exercises/

2、 用编辑器创建一个新文件，保存成newfile.txt

3、 用git status命令查看状态。你的文件现在在上图中的哪个状态下？

4、现在运行git add newfile.txt

5、再运行git status命令查看状态。你的工作目录下发生了什么变化？这次你的文件在上图中的哪个状态下？

6、现在，我们尝试取消暂存状态。运行命令： $ git reset HEAD newfile.txt

这次你的文件在上图中的哪个状态下？

提示：如果运行git reset后面不指定具体的文件，那么所有文件都会被取消暂存。

练习5b

如何恢复所有内容

1、进入~/exercisers/目录。

2、你想取消所有的操作回到上次的提交（HEAD），这可以用命令git reset --hard来实现。

3、现在恢复其他的修改到HEAD。 $ git reset --hard HEAD

4、运行git status检查一下，是所有文件吗？

练习6:

分支的创建,切换,合并.

1. 进入目录~/exercises/目录
2. 运行命令git branch devleop创建develop分支
3. 运行命令git checkeout develop切换到develop分支
4. 修改某个文件，并提交到本地版本库。
5. 切换回master分支。
6. 运行命令git merge develop合并develop分支，
7. 查看develop分支内容是否更新到了master分支上

 练习7：

在这个练习中，我们会查看git库的历史信息，比较工作目录，index和分支。

步骤1：用git log检查提交日志。

1、 进入目录~/exercises/

2、 运行命令git log查看git库的历史信息 $ git log

3、 运行命令git branch查看所有的本地分支，分支名前面有\*的表示当前分支。在这个列表中你可能会看到其他的一些分支，如果有,你可以看看分支的历史： $ git log <分支名称>

4、 查看指定文件或目录的历史也是可以的，可以用下面的命令： $ git log -- file

步骤2：使用git diff来显示提交和提交之间、提交和工作目录之间的修改。

1. 进入~/exercises/目录
2. 修改任意一个文件

3、 用命令git diff显示你的工作目录和index中的区别：

$ git diff

4、 把该文件加入到index中

5、 可以用命令git diff --cached commit来显示暂存在index中的文件和一个指定的

提交的区别（如果不指定提交（commit），那么默认是和HEAD比较） $ git diff --cached

6、 也可以显示分支之间的区别

$ git diff branch1..branch2

练习8：

分支合并中冲突的解决。

1. 进入~/exercises/目录
2. 切换到develop分支
3. 修改任意一个文件中某一行，并提交
4. 切换回master分支，在步骤3修改过的文件中，同一行的位置做响应的修改，并提交。
5. 运行命令 git merge develop合并develop分支到master分支。
6. Git将报合并冲突
7. 处理冲突，运行命令 git add <冲突文件名> 确认冲突解决完毕。
8. 运行命令 git commit完成合并操作。

练习9：

与远程版本库相关操作练习。

练习9a：

推送修改到远程版本库。

1. 运行命令git push origin master将本地master分支推送到远程版本库
2. 如发生冲突，请运行git pull抓取远程版本库的最新提交，完成冲突处理。
3. 继续运行命令 git push origin master推送修改到远程版本库。

练习9b：

在本地创建和远程分支对应的分支

1. 运行git checkout -b branch-name origin/branchname，本地和远程分支的名称最好一致；
2. 建立本地分支和远程分支的关联，运行命令：

git branch --set-upstream-to= origin/branch-name branch-name；