GIT 用法：

1. 创建本地分支

   git checkout –t origin/远端分支名—— 拷贝一份远端分支代码至本地，本地分支名称和远端名称相同

   git checkout –b 本地分支名 –t origin/远端分支名——拷贝一份远端分支代码至本地，本地分支名称为自定义的名称

2. 本地提交代码

   git add file——添加指定文件至待提交区

   git add . —— 添加所有更改的文件至待提交区

   git commit –m “description” —— 将待提交区的代码提交至当前本地分支，description 为提交的描述信息，谨记用英文

3. 同步远端分支代码

   git fetch –all —— 获取远端所有代码和分支

   git rebase origin/远端分支名—— 同步远端分支名的代码

   执行完该命令后，再执行 git st——查看当前状态，如果出现 conflict 提示，需把冲突解决后再执行 git add file 操作，或者把所有冲突解决后执行 git add . 操作。执行完后再执行 git rebase --continue 操作，继续完成 rebase 操作。

4. 提交代码至远端

   git push origin 本地分支名：远端分支名

   如果本地分支名和远端分支名相同，也可执行如下指令：

   git push origin 本地分支名

5. 切换分支

   git checkout 本地分支名——切换到指定的分支

6. Merge 代码

   作用：git merge A——将 A 分支的代码改动与 B 分支的代码改动进行合并，即在 B 分支上获取 A 分支上的代码改动

   git merge 本地分支名——与本地分支名的分支进行 merge 操作

   git merge origin/远端分支名——与远端分支名的分支进行 merge 操作

   git merge --squash 分支名——将知道分支的代码改动压缩为一个 pick，且去除提交代码的注释信息

   执行完上述命令后，执行 git st 查看当前状态，如果有待提交代码，需执行

   git commit –m “description”指令将代码提交至 merge 后的分支

7. Pick 代码

   作用： git cherry-pick ID——将指定 ID 的代码提交 pick 到当前分支，如有两个分支 A 和 B，现在需要将 A 上的某个代码提交 pick 到分支 B 上，可执行如下操作：

   git checkout A——切换到 A 分支

   git log——查看 A 分支上的代码提交日志

git checkout B——切换到 B 分支
git cherry-pick ID——将 A 分支上指定的 ID 代码提交 Pick 到 B 分支

8. 删除分支
   例如本地有两个分支 A 和 B，需要删除分支 A，执行如下操作：
   git checkout B——切换到 B 分支
   git branch –D A——删除 A 分支

9. 查看分支
   git branch ——查看本地分支
   git branch –a ——查看本地和远端所有分支

10. 查看所有提交状态
    gitk --all