1. 在本地新建文件夹，把远程仓库的克隆到本地:
   1. Git clone URL
2. 本地修改后同步到远程仓库：
   1. Git status (查看本地是否跟踪到)
   2. Git add . (把本地新增的文件添加到缓存)
   3. 再执行 git status 查看是否已存放至暂存区
   4. Git commit –m “changes” (使用 git add 命令将想要快照的内容写入缓存区， 而执行 git commit 将缓存区内容添加到仓库中。)
   5. Git push (把本地代码放到远程仓库)
3. 在本地仓库获取同步最新远程仓库：
   1. Git pull (在本地pull远程仓库最新的代码到本地仓库，相当于git fetch + merge)
   2. Git fetch 在本地仓库获取最新的代码同步（常用）

至此，修改全部结束。

回退命令：

$ git reset --hard HEAD^ 回退到上个版本

$ git reset --hard HEAD~3 回退到前3次提交之前，以此类推，回退到n次提交之前

$ git reset --hard commit\_id 退到/进到 指定commit的sha码

git 基本操作注意点总结:

git clone <source repository> <destination repository>

复制本地仓库的命令方式。

<source repository>：想克隆的本地仓库路径

<destination repository>：想克隆去另一个地方的路径。例如 git clone d:/git e:/git11 是将 d:/git 的仓库（即包含隐藏文件 .git 的目录）克隆到 e:/git11 目录下。

注意：

1、<destination repository> 目录必须没有在文件系统上创建，或创建了但里面为空，不然会克隆不成功。

2、与从远程拉取仓库不同，路径的最后不用写 .git 来表明这是一个仓库。

git status –s

获得简短的状态输出。

git diff

git diff：查看工作区与暂存区的不同。

git diff –cached [<commit>]：查看暂存区与指定提交版本的不同，版本可缺省（为HEAD）。

git diff <commit>：查看工作区与指定提交版本的不同。

git diff <commit>..<commit>：查看2个指定提交版本的不同，其中任一可缺省（为HEAD）。

git diff <commit>...<commit>：查看2个不同分支指定提交版本的不同，其中任一可缺省（为HEAD），该命令相当于git diff $(git-merge-base A B) B。

git commit –am ""

直接提交全部修改，相当于 add 和 commit 一起执行了。

注意：全部文件为 tracked 才行，你新建了文件为 untracked 时，该命令不会执行。

git checkout

与 git reset 不同，reset 是替换整个目录树，多余的文件将被删除。而 checkout 只是替换指定的文件，对多余的文件保留不做任何处理。

git rm

把文件从工作区和暂存区中删除。使用 —cached 只从暂存区中删除。使用 –rf <directory> 可删除指定目录下的所有文件和子目录。

git mv <source> <destination>

在工作区和暂存区中进行移动或重命名。若 <destination> 不为一个目录名，则执行重命名。如果为一个目录名，则执行移动。