$ git config --global user.name "Your Name"

$ git config --global user.email "email@example.com"

命令作用：配置全局用户名和地址

$ git init

Initialized empty Git repository **in** /Users/michael/learngit/.git/

命令作用：初始化git仓库

$ git add readme.txt

命令作用：将文件添加至git仓库（git add把文件添加进去，实际上就是把文件修改添加到暂存区；）

$ git **commit** -m "wrote a readme file"

[master (root-**commit**) cb926e7] wrote a readme file

1 file changed, 2 insertions(+)

**create** mode 100644 readme.txt

命令作用：将文件提交到git仓库（git commit提交更改，实际上就是把暂存区的所有内容提交到当前分支）



git add命令实际上就是把要提交的所有修改放到暂存区（Stage），然后，执行git commit就可以一次性把暂存区的所有修改提交到分支。

$ git status

命令作用：查看仓库当前的状态

$ git diff readme.txt

命令作用：查看文件的修改信息

$ git log

命令作用：查看版本修改记录（可以使用：--pretty=oneline 参数精简输出信息）

$ git re**set** --hard HEAD^

命令作用：git文件版本回退（在Git中，用HEAD表示当前版本，也就是最新的提交3628164...882e1e0（注意我的提交ID和你的肯定不一样），上一个版本就是HEAD^，上上一个版本就是HEAD^^，当然往上100个版本写100个^比较容易数不过来，所以写成HEAD~100）

$ git re**set** --hard “版本号”

命令作用：根据版本号回到git文件的某个版本

$ git reflog

命令作用：记录你的每一次命令

$ git checkout -- readme.txt

命令作用：命令git checkout -- readme.txt意思就是，把readme.txt文件在工作区的修改全部撤销，这里有两种情况：

一种是readme.txt自修改后还没有被放到暂存区，现在，撤销修改就回到和版本库一模一样的状态；

一种是readme.txt已经添加到暂存区后，又作了修改，现在，撤销修改就回到添加到暂存区后的状态。

总之，就是让这个文件回到最近一次git commit或git add时的状态

$ git re**set** HEAD readme.txt

命令作用：git reset命令既可以回退版本，也可以把暂存区的修改回退到工作区。当我们用HEAD时，表示最新的版本

$ git rm test.txt

命令作用：从版本库中删除该文件

$ git checkout -b dev

命令作用：git checkout命令加上-b参数表示创建并切换，相当于以下两条命令：

$ git branch dev

$ git checkout dev

Switched to branch 'dev'

$ git remote add origin git@github.com:michaelliao/learngit.git

命令作用：将本地仓库与远程仓库关联

$ git push origin master

命令作用：把本地master分支的最新修改推送至GitHub，现在，你就拥有了真正的分布式版本库

$ git merge dev

命令作用：合并指定分支到当前分支

Git鼓励大量使用分支：

查看分支：git branch

创建分支：git branch <name>

切换分支：git checkout <name>

创建+切换分支：git checkout -b <name>

合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>