设置编辑器：

$ git config --global core.editor Code

检查配置信息：

git config --list

使用不同的文本编辑器：

git config --global core.editor editorname

具体检查某一项配置：git config <key>

例如：git config user.name 查看用户名

检查状态：

git status

git status –s 或者git status –short将得到更加紧凑的格式

新添加的未跟踪文件前面有 ?? 标记，新添加到暂存区中的文件前面有 A 标记，修改过的文件前面有 M 标记。你可能注意到了 M 有两个可以出现的位置，出现在右边的 M 表示该文件被修改了但是还没放入暂存区，出现在靠左边的 M 表示该文件被修改了并放入了暂存区。例如，上面的状态报告显示： README 文件在工作区被修改了但是还没有将修改后的文件放入暂存区,lib/simplegit.rb 文件被修改了并将修改后的文件放入了暂存区。而

Rakefile 在工作区被修改并提交到暂存区后又在工作区中被修改了，所以在暂存区和工作区都有该文件被修改了的记录。

$ git status -s

M README

M M Rakefile

A lib/git.rb

M lib/simplegit.rb

?? LICENSE.txt

git diff

git diff 尚未暂存的文件

将通过文件补丁的格式显示具体哪些行发生了改变。

git diff --staged 查看已经暂存，待添加到下次提交的文件

git difftool –tool-help来查看系统支持的其它方式输出

克隆项目

git clone [URL]

将克隆的项目重命名

git clone [URL] MyName

提交更新：

git commit

将提交信息与命令行放在同一行：

git commit -m

跳过暂存区域

git commit –a

从已跟踪的文件清单中移除(并且在本地磁盘中也删除了)

git rm

从已暂存的文件清单中删除(并且在本地磁盘中也删除了)

git rm –f

将文件从git仓库中删除，但是保留在本地磁盘上

git rm –cached

将文件进行更改文件名

git mv file\_source\_name file\_dest\_name

查看提交历史

git log

显示每次提交的内容差异：

git log –p

显示每次提交的简略统计信息：

git log –-stat

不同于默认方式(git log)方式显示

git log –pretty = oneline

或者：short full fuller format

按照时间做出限制：

git log –-since(after)=time

git log –-until(before)=time

按照作者：

git log –-author=“name”

撤销操作

git commit –amend

查看远程仓库

git remote

显示远程仓库使用的git保存简写与URL：

git remote –v

添加远程仓库：

git remote add <shortname><URL>

从远程仓库中获取数据：

git fetch [remote-name]

推送到远程仓库：

git push [remote\_name][branch\_name]

查看远程仓库：

git remote show [remote-name]

远程仓库的移除与重命名：

git remote rename old\_name new\_name