### 1,在工作目录中初始化新仓库

$ git init

### 2,检查当前文件状态

$ git status

### 3,跟踪新文件

将文件放入暂存区

$ git add README //跟踪readme文件

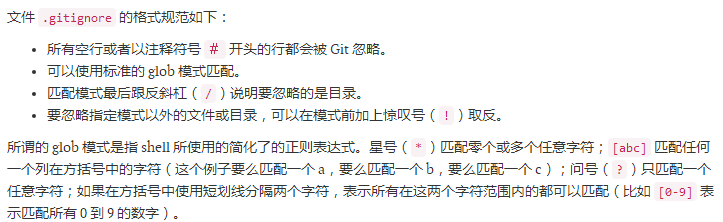
$ git add . //跟踪目录下的所有文件

### 4，忽略某些文件

$ cat .gitignore

\*.[oa] //第一行告诉 Git 忽略所有以 .o 或 .a 结尾的文件。

\*~ //第二行告诉 Git 忽略所有以波浪符（~）结尾的文件，





### 5，查看已暂存和未暂存的更新

git status 的显示比较简单，仅仅是列出了修改过的文件，

可以用 git diff 命令，显示还没有暂存起来的改动

若要看已经暂存起来的文件和上次提交时的快照之间的差异，可以用 git diff --cached 命令。

### 6，跳过使用暂存区域

git commit -a -m 'added new benchmarks'

### 7，移除文件

手动从工作目录中删除文件时，然后再运行 git rm XXXX就可以记录此次移除文件的操作；

如果删除之前修改过并且已经放到暂存区域的话，则必须要用强制删除选项 -f（译注：即 force 的首字母），以防误删除文件后丢失修改的内容

我们想把文件从 Git 仓库中删除，但仍然希望保留在当前工作目录中时：

换句话说，仅是从跟踪清单中删除。比如一些大型日志文件或者一堆 .a 编译文件，不小心纳入仓库后，要移除跟踪但不删除文件，以便稍后在 .gitignore 文件中补上，用 --cached 选项即可：

$ git rm --cached readme.txt //后面可以列出文件或者目录的名字以便稍后在

$ git rm log/\\*.log //删除所有 log/ 目录下扩展名为 .log 的文件。

$ git rm \\*~ //递归删除当前目录及其子目录中所有 ~ 结尾的文件。

### 6，跳过使用暂存区域

### 6，跳过使用暂存区域

### 6，跳过使用暂存区域