

我们已经成功地添加并提交了一个 `readme.txt` 文件，现在，是时候继续工作了，于是，我们继续修改 `readme.txt` 文件，改成如下内容：

```
Git is a distributed version control system.  
Git is free software.
```

现在，运行 `git status` 命令看看结果：

```
$ git status  
  
On branch master  
  
Changes not staged for commit:  
  
    (use "git add <file>..." to update what will be committed)  
  
    (use "git checkout -- <file>..." to discard changes in working directory)  
  
        modified:   readme.txt  
  
no changes added to commit (use "git add" and/or "git commit -a")
```

`git status` 命令可以让我们时刻掌握仓库当前的状态，上面的命令输出告诉我们，`readme.txt` 被修改过了，但还没有准备提交的修改。

虽然 Git 告诉我们 `readme.txt` 被修改了，但如果能看看具体修改了什么内容，自然是很好的。比如你休假两周从国外回来，第一天上班时，已经记不清上次怎么修改的 `readme.txt`，所以，需要用 `git diff` 这个命令看看：

```
$ git diff readme.txt  
  
diff --git a/readme.txt b/readme.txt  
  
index 46d49bf..9247db6 100644  
  
--- a/readme.txt  
+++ b/readme.txt  
  
@@ -1,2 +1,2 @@  
  
-Git is a version control system.
```

```
+Git is a distributed version control system.
```

```
Git is free software.
```

`git diff` 顾名思义就是查看 difference，显示的格式正是 Unix 通用的 diff 格式，可以从上面的命令输出看到，我们在第一行添加了一个 `distributed` 单词。

知道了对 `readme.txt` 作了什么修改后，再把它提交到仓库就放心多了，提交修改和提交新文件是一样的两步，第一步是 `git add`：

```
$ git add readme.txt
```

同样没有任何输出。在执行第二步 `git commit` 之前，我们再运行 `git status` 看看当前仓库的状态：

```
$ git status

On branch master

Changes to be committed:

  (use "git reset HEAD <file>..." to unstage)

        modified:   readme.txt
```

`git status` 告诉我们，将要被提交的修改包括 `readme.txt`，下一步，就可以放心地提交了：

```
$ git commit -m "add distributed"

[master e475afc] add distributed

1 file changed, 1 insertion(+), 1 deletion(-)
```

提交后，我们再用 `git status` 命令看看仓库的当前状态：

```
$ git status

On branch master

nothing to commit, working tree clean
```

Git 告诉我们当前没有需要提交的修改，而且，工作目录是干净（working tree clean）的。

小结

- 要随时掌握工作区的状态，使用 `git status` 命令。
- 如果 `git status` 告诉你有文件被修改过，用 `git diff` 可以查看修改内容。