**Git基本操作（二）**

**（4）把暂存区的内容提交到本地版本库**

当暂存区域已经准备妥当可以提交时，在此之前，请一定要确认还有什么修改过的，或新建的文件还没有git add到暂存区，否则提交的时候不会记录这些还没暂存起来的变化。

所以，每次准备提交前，先用git status查看下，需要提交的文件是不是都已暂存起来了，然后再运行提交命令git commit -m '备注信息'。

**1）将文件添加到缓存区并提交到本地版本库**

将readme.txt文件加入到暂存区，

并执行git commit -m '本次提交的说明'，将readme.txt文件提交到本地版本库。

|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git add readme.txt warning: LF will be replaced by CRLF in readme.txt. The file will have its original line endings in your working directory  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git commit readme.txt -m 'My first commit.new file readme.txt' warning: LF will be replaced by CRLF in readme.txt. The file will have its original line endings in your working directory [master (root-commit) e84b93b] My first commit.new file readme.txt  1 file changed, 1 insertion(+)  create mode 100644 readme.txt |

重点说明最下面三行内容：

1. master：表示master（主干）分支。
2. root-commit：root表示根，意思是该版本库的第一次提交。
3. e84b93b：提交操作的版本号概要。
4. My first commit.new file readme.txt：本次提交的说明信息。
5. 1 file changed：一个文件被修改。
6. 1 insertions(+)：增加了1行内容，+号表示增加，-号表示减少。
7. create mode 100644 readme.txt：readme.txt文件创建模式为100644， 100代表regular file（普通文件），644代表文件权限。

|  |
| --- |
| 提示：提交时记录的是放在暂存区域的快照。  即：每一次执行提交操作时，都是对项目作一次快照，以后可以回到这个状态，或者与这个状态进行比较。 |

**2）将文件提交到本地版本库后工作区、暂存区状态**

执行git status命令查看工作区、暂存区状态。

|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git status On branch master  # 暂存区中没有可提交的内容 # 工作目录和本地版本库是一样的，没有修改、新建等操作。 nothing to commit, working tree clean |

**3）修改文件内容后查看工作区、暂存区状态**

修改readme.txt文件内容后，再执行git status命令查看工作区、暂存区状态。

|  |
| --- |
| Bash # 修改文件内容 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ echo "git world" >> readme.txt  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ cat readme.txt hello git world git world  # 查看工作区、暂存区状态。 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git status On branch master Changes not staged for commit:  (use "git add <file>..." to update what will be committed)  (use "git restore <file>..." to discard changes in working directory)  modified: readme.txt # 文件名为红色  no changes added to commit (use "git add" and/or "git commit -a") |

说明：

1. Changes not staged for commit:modified: readme.txt readme.txt文件被修改，但是没有添加到暂存区
2. use "git add <file>..." to update what will be committed 表示对readme.txt文件，你可以使用git add <file>命令， 将文件更新添加到暂存区。 和第（2）步的 use "git add <file>..." to include in what will be committed 是有一点差别的。
3. use "git restore <file>..." to discard changes in working directory 表示可以通过git restore <file>...命令，放弃工作目录中文件的更改。 就是还原文件。Git早期版本该命令为git checkout --<file>...。
4. no changes added to commit (use "git add" and/or "git commit -a") 表示你没有添加要提交的更改， and/or表示你可使用“git add”把变更文件添加到暂存区， 在通过"git commit -a"提交到本地版本库， 你也可以通过"git commit -a"直接把变更的文件提交到本地版本库。 注意使用git commit -a 命令用加文件名。

**4）把修改后的文件提交到暂存区**

|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git add readme.txt warning: LF will be replaced by CRLF in readme.txt. The file will have its original line endings in your working directory  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git status On branch master Changes to be committed:  (use "git restore --staged <file>..." to unstage)  modified: readme.txt # 文件名为绿色 |

说明：

1. Changes to be committed: modified: readme.txt readme.txt的修改已被Git追踪到
2. use "git restore --staged <file>..." to unstage 你可以用git restore --staged <file>...命令， 将文件从暂存区撤出，但不会撤销文件的更改。

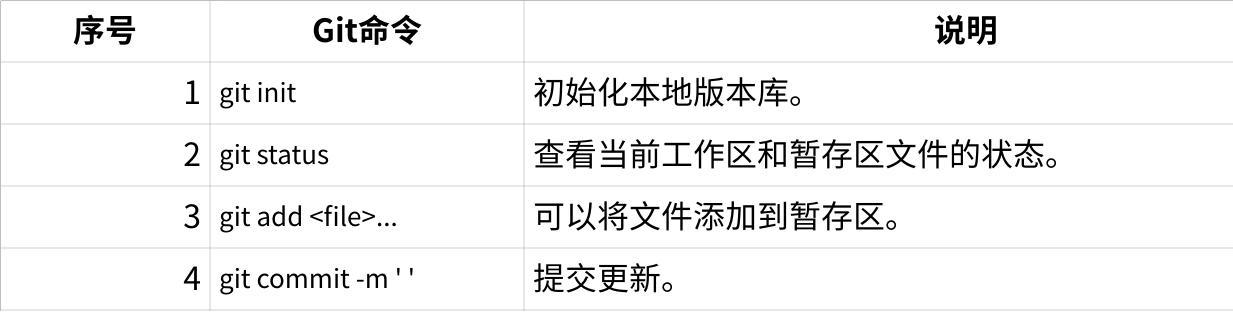
**5）将修改过后的文件提交到本地版本库**

|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git commit readme.txt -m 'modified readme.txt' warning: LF will be replaced by CRLF in readme.txt. The file will have its original line endings in your working directory [master e704334] modified readme.txt  1 file changed, 1 insertion(+) |

可以看到，master后没有 (root-commit)了，因为根提交只有一次。

|  |
| --- |
| 新创建的文件先添加到暂存区，然后添加到本地版本库。  而已经提交到本地版本库的文件，修改了，可以按照上面操作，也可以直接提交到本地版本库。 |

**2、总结本文用到的Git命令**



**点击图片可查看完整电子表格**

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/77a05a813a40 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。