

Git 学习手册:

1. **git init** 初始化, 在目标文件夹使用此命令, 文件夹内会新增一个.git隐藏文件

```
python@ubuntu:~/git_learn$ git init
初始化空的 git 仓库于 /home/python/git_learn/.git/
```

2. **git add** 后跟着需要上传的文件上传文件

git commit -m '注释' 确认上传文件

```
python@ubuntu:~/git_learn$ git add cat.txt
python@ubuntu:~/git_learn$ git commit -m '版本2'
[master a5f71c3] 版本2
1 file changed, 1 insertion(+)
```

3. **git log** 查询所有已上传的版本记录

```
python@ubuntu:~/git_learn$ git log
commit a5f71c3a41c376bb7f163911753fee9149c7ac19
Author: xwp <xwp_fullstack@163.com>
Date: Mon Oct 15 21:34:05 2018 +0800

    版本2

commit 0522d5575b051135ae13e3a9875b87f330fb159f
Author: xwp <xwp_fullstack@163.com>
Date: Mon Oct 15 21:29:59 2018 +0800

    版本一
```

```
python@ubuntu:~/git_learn$ git log --pretty=oneline
1b1462d9cf3577ef19542d19109b470a137e56bd 分支
53bc36d3d8c8009033c99f4bcc915436c6fd9b43 删除cat1.txt
75d6c32e95ad416ace5b53d38c7eb7e5c0c70533 版本4
5dfa13d493e718adcd43bee201852857498149a7 版本3
a5f71c3a41c376bb7f163911753fee9149c7ac19 版本2
0522d5575b051135ae13e3a9875b87f330fb159f 版本一
```

4. **git reset --hard HEAD^** 版本回退到上一个版本 **git reset --hard HEAD^^** 版本回退到上上一个版本

git reset --hard HEAD~1 版本回退到上一个版本 **git reset --hard HEAD~2** 版本回退到上上一个版本

```
python@ubuntu:~/git_learn$ git reset --hard HEAD^
HEAD 现在位于 0522d55 版本一
```

```
python@ubuntu:~/git_learn$ git reset --hard HEAD~1
HEAD 现在位于 0522d55 版本一
```

git reset --hard 后跟着版本的文件编码前几位可以回退到那个版本

```
python@ubuntu:~/git_learn$ git reset --hard 0522d557
HEAD 现在位于 0522d55 版本一
```

5. git reflog 查询之前的操作记录

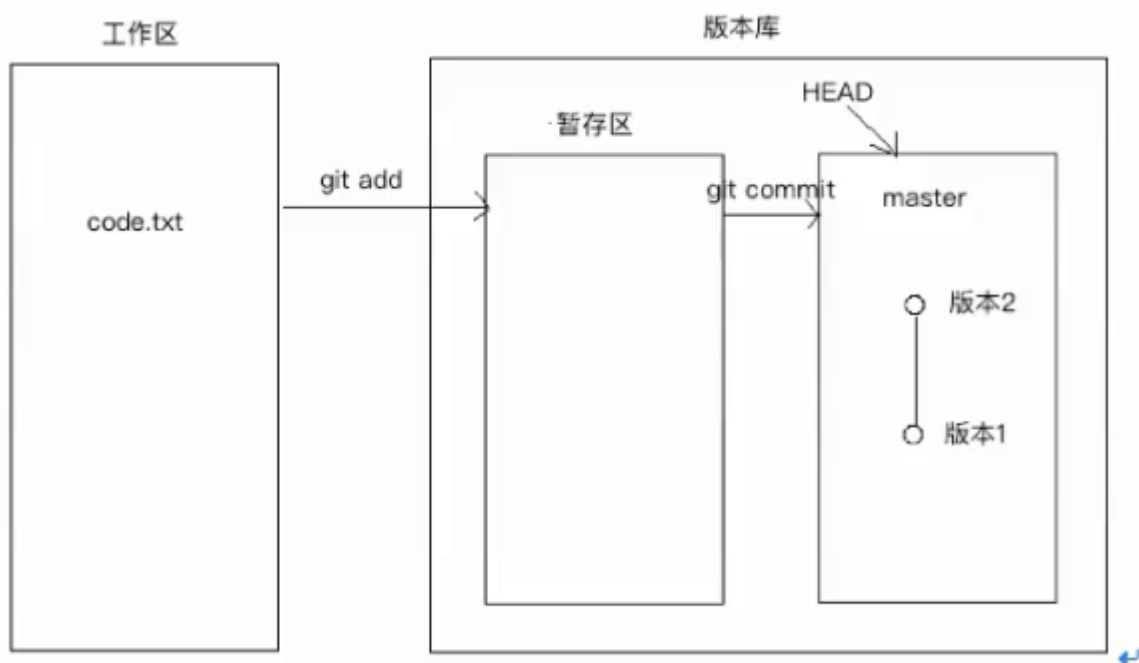
注意：每个记录前有文件的编码，可以使用git reset --hard 命令回退到想退回的版本文件

```
python@ubuntu:~/git_learn$ git reflog
0522d55 HEAD@{0}: reset: moving to HEAD~1
a5f71c3 HEAD@{1}: reset: moving to a5f71c3a4
0522d55 HEAD@{2}: reset: moving to HEAD^
a5f71c3 HEAD@{3}: commit: 版本2
0522d55 HEAD@{4}: commit (initial): 版本一
```

6. 工作区和暂存区

工作区：电脑中的目录文件夹

版本库：工作区内有一个隐藏目录.git，这个不是工作区，而是git的版本库



git status 查看当前工作树的状态（新增一个cat1.txt文件，修改cat.txt文件）

```
python@ubuntu:~/git_learn$ git status
位于分支 master
尚未暂存以备提交的变更：
  （使用 "git add <文件>..." 更新要提交的内容）
  （使用 "git checkout -- <文件>..." 丢弃工作区的改动）

    修改：        cat.txt

未跟踪的文件：
  （使用 "git add <文件>..." 以包含要提交的内容）

    cat1.txt

修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）
```

git add 可以一次上传多个文件，也可以上传一个文件夹里的所有文件

此时文件放在暂存区中

```
python@ubuntu:~/git_learn$ git add cat1.txt cat.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
要提交的变更：
  （使用 "git reset HEAD <文件>..." 以取消暂存）

    修改：      cat.txt
   新文件：    cat1.txt
```

git commit 将文件上传到版本库中

此时工作区内无文件需要提交

```
python@ubuntu:~/git_learn$ git commit -m '版本3'
[master 5dfa13d] 版本3
 2 files changed, 2 insertions(+)
 create mode 100644 cat1.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
无文件要提交，干净的工作区
```

7. git 管理文件的修改，它只会提交暂存区的修改来创建版本

比如一个文件修改后git add到暂存区，然后对文件再次修改，但是没有git add到暂存区，此时git commit只会将之前暂存区的文件上传，而再次修改的文件没有上传，而是留在工作区

```
python@ubuntu:~/git_learn$ git status
位于分支 master
要提交的变更：
  （使用 "git reset HEAD <文件>..." 以取消暂存）

    修改：      cat.txt

尚未暂存以备提交的变更：
  （使用 "git add <文件>..." 更新要提交的内容）
  （使用 "git checkout -- <文件>..." 丢弃工作区的改动）

    修改：      cat.txt

python@ubuntu:~/git_learn$ git commit -m '版本4'
[master 75d6c32] 版本4
 1 file changed, 1 insertion(+)
python@ubuntu:~/git_learn$ git status
位于分支 master
尚未暂存以备提交的变更：
  （使用 "git add <文件>..." 更新要提交的内容）
  （使用 "git checkout -- <文件>..." 丢弃工作区的改动）

    修改：      cat.txt

修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）
```

8. 撤销修改

a. 工作区里的文件撤销修改: **git checkout --** 要撤销的文件

```
python@ubuntu:~/git_learn$ git status
位于分支 master
尚未暂存以备提交的变更:
  (使用 "git add <文件>..." 更新要提交的内容)
  (使用 "git checkout -- <文件>..." 丢弃工作区的改动)

    修改:      cat.txt

修改尚未加入提交 (使用 "git add" 和/或 "git commit -a")
python@ubuntu:~/git_learn$ git checkout -- cat.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
无文件要提交, 干净的工作区
```

b. 暂存区里的文件撤销修改: 首先 **git reset HEAD** 要撤销的文件 到工作区, 然后 **git checkout --** 要撤销的文件 将文件撤销

```
python@ubuntu:~/git_learn$ git add cat.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
要提交的变更:
  (使用 "git reset HEAD <文件>..." 以取消暂存)

    修改:      cat.txt

python@ubuntu:~/git_learn$ git reset HEAD cat.txt
重置后取消暂存的变更:
M       cat.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
尚未暂存以备提交的变更:
  (使用 "git add <文件>..." 更新要提交的内容)
  (使用 "git checkout -- <文件>..." 丢弃工作区的改动)

    修改:      cat.txt

修改尚未加入提交 (使用 "git add" 和/或 "git commit -a")
python@ubuntu:~/git_learn$ git checkout -- cat.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
无文件要提交, 干净的工作区
```

c. 版本库里的文件撤销修改: **git reset --hard** 到需要返回的版本

9. 对比文件的不同

a. 对比工作区中的文件和版本库中的文件的不同

git diff HEAD -- 要对比的文件

其中文件前有---的表示为版本库HEAD里的文件, +++的表示工作区里的文件

以下对比表示工作区里的文件比版本库HEAD里的文件多了一行，多了一行'a new line;'前面有加了一个 '+' 号，白色文字表示两个文件都有的内容

```
python@ubuntu:~/git_learn$ git status
位于分支 master
尚未暂存以备提交的变更：
  (使用 "git add <文件>..." 更新要提交的内容)
  (使用 "git checkout -- <文件>..." 丢弃工作区的改动)

    修改：      cat.txt

修改尚未加入提交 (使用 "git add" 和/或 "git commit -a")
python@ubuntu:~/git_learn$ git diff HEAD -- cat.txt
diff --git a/cat.txt b/cat.txt
index 54b93b9..28845cc 100644
--- a/cat.txt
+++ b/cat.txt
@@ -2,3 +2,4 @@ this is the frist line;
    this is the second line;
    this is the third line;
    this is the forth line;
+a new line;
```

b.对比版本库中的不同版本的文件的不同

git diff HEAD HEAD^ -- 要对比的文件(不同版本可以用不同方式表示)

其中文件前有---的表示为当前HEAD文件，+++的表示版本库里的HEAD前一个版本的文件

以下对比表示当前版本里的文件比前一个版本库里的文件多了一行，多了一行'this is the forth line;'前面有加了一个 '-' 号，白色文字表示两个文件都有的内容

```
python@ubuntu:~/git_learn$ git diff HEAD HEAD^ -- cat.txt
diff --git a/cat.txt b/cat.txt
index 54b93b9..c67d07c 100644
--- a/cat.txt
+++ b/cat.txt
@@ -1,4 +1,3 @@
    this is the frist line;
    this is the second line;
    this is the third line;
- this is the forth line;
```

10. 删除文件

使用rm 命令删除文件后，可以使用git checkout -- 文件名恢复文件

```
python@ubuntu:~/git_learn$ ls
cat1.txt  cat.txt
python@ubuntu:~/git_learn$ rm cat1.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
尚未暂存以备提交的变更：
  （使用 "git add/rm <文件>..." 更新要提交的内容）
  （使用 "git checkout -- <文件>..." 丢弃工作区的改动）

    删除：        cat1.txt

修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）
python@ubuntu:~/git_learn$ git checkout -- cat1.txt
python@ubuntu:~/git_learn$ ls
cat1.txt  cat.txt
```

使用rm 命令删除文件后，可以使用git add/rm 文件名将文件放入暂存区，此时可以使用git reset HEAD 文件名取消暂存，文件重回工作区删除状态

```
python@ubuntu:~/git_learn$ rm cat1.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
尚未暂存以备提交的变更：
  （使用 "git add/rm <文件>..." 更新要提交的内容）
  （使用 "git checkout -- <文件>..." 丢弃工作区的改动）

    删除：        cat1.txt

修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）
python@ubuntu:~/git_learn$ git add cat1.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
要提交的变更：
  （使用 "git reset HEAD <文件>..." 以取消暂存）

    删除：        cat1.txt
```

最后git commit -m '注释'将此次删除文件放入版本库中


```
python@ubuntu:~/git_learn$ git commit -m '删除cat1.txt'
[master 53bc36d] 删除cat1.txt
1 file changed, 1 deletion(-)
delete mode 100644 cat1.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
无文件要提交，干净的工作区
python@ubuntu:~/git_learn$ git log
commit 53bc36d3d8c8009033c99f4bcc915436c6fd9b43
Author: xwp <xwp_fullstack@163.com>
Date: Wed Oct 17 22:42:33 2018 +0800

    删除cat1.txt

commit 75d6c32e95ad416ace5b53d38c7eb7e5c0c70533
Author: xwp <xwp_fullstack@163.com>
Date: Tue Oct 16 22:34:56 2018 +0800
```

如果一个文件不在暂存区也没有在版本库，那么文件删除后无法恢复

```
python@ubuntu:~/git_learn$ git status
位于分支 master
无文件要提交，干净的工作区
python@ubuntu:~/git_learn$ touch code.txt
python@ubuntu:~/git_learn$ vi code.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
未跟踪的文件:
  (使用 "git add <文件>..." 以包含要提交的内容)

    code.txt

提交为空，但是存在尚未跟踪的文件（使用 "git add" 建立跟踪）
python@ubuntu:~/git_learn$ rm code.txt
python@ubuntu:~/git_learn$ git status
位于分支 master
无文件要提交，干净的工作区
```

如果一个文件已经被提交到版本库，那么永远不用担心误删，但是要小心，只能恢复到最新版本，你会丢失最近一次提交后你修改的内容

11. 总结：git基本操作

