# 2019-02-28-Status+diff

## 一、状态status+查修改diff

* 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

### 1.1、git diff:

git diff #是工作区(work dict)和暂存区(stage)的比较

git diff --cached #是暂存区(stage)和分支(master)的比较

git diff HEAD #查看工作区和版本库里面最新版本的区别

修改后查看状态： modified: git.txt

Dit diff: 工作区与缓存区比较，有区别（修改后未add）

$ git status

On branch master

Changes not staged for commit:

(use "git add ..." to update what will be committed)

(use "git checkout -- ..." to discard changes in working directory)

modified: git.txt

Administrator@Nancy MINGW64 /e/Nancyxiao/study/git (master)

$ git diff

diff --git a/git/git.txt b/git/git.txt

index bb7fdf0..7d9fb6e 100644

--- a/git/git.txt

+++ b/git/git.txt

@@ -1,2 +1,4 @@

Add new one

add two

+

+git diff

\ No newline at end of file

将修改添加到缓存区，查看状态: modified: git.txt

Git diff: 工作区和缓存区，此时无区别(修改后，已add, 工作区=缓存区)

Administrator@Nancy MINGW64 /e/Nancyxiao/study/git (master)

$ git add git.txt

Administrator@Nancy MINGW64 /e/Nancyxiao/study/git (master)

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD ..." to unstage)

modified: git.txt

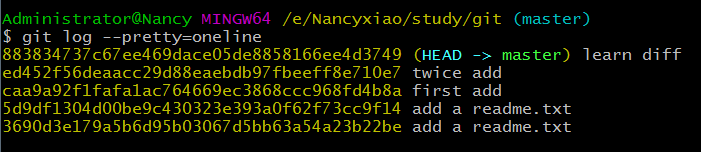
|  |
| --- |
| Administrator@Nancy MINGW64 /e/Nancyxiao/study/git (master)  $ git diff  查看缓存区与版本库的区别：（已add未commit -m）  Administrator@Nancy MINGW64 /e/Nancyxiao/study/git (master)  $ git diff --cached  diff --git a/git/git.txt b/git/git.txt  index bb7fdf0..7d9fb6e 100644  --- a/git/git.txt  +++ b/git/git.txt  @@ -1,2 +1,4 @@  Add new one  add two  +  +git diff  \ No newline at end of file  查看工作区与版本库的区别：（已add未commit -m）  Administrator@Nancy MINGW64 /e/Nancyxiao/study/git (master)  $ git diff HEAD  diff --git a/git/git.txt b/git/git.txt  index bb7fdf0..7d9fb6e 100644  --- a/git/git.txt  +++ b/git/git.txt  @@ -1,2 +1,4 @@  Add new one  add two  +  +git diff  \ No newline at end of file |
| 总结：  d:\Documents\Tencent Files\768445005\FileRecv\MobileFile\Image\SB%`_98H1{O)%VOW`){D8%W.png |
|  |

## 版本回退reset –hard

git log命令显示从最近到最远的提交日志

如果嫌输出信息太多，看得眼花缭乱的，可以试试加上--pretty=oneline参数

$ git log --pretty=oneline



在Git中，用HEAD表示当前版本,上一个版本就是HEAD^，上上一个版本就是HEAD^^，当然往上100个版本写100个^比较容易数不过来，所以写成HEAD~100。

HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令git reset --hard commit\_id。

穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。

要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。