git（版本控制软件）

生成公私钥：ssh-keygen

代码仓库

（生僻词句）：

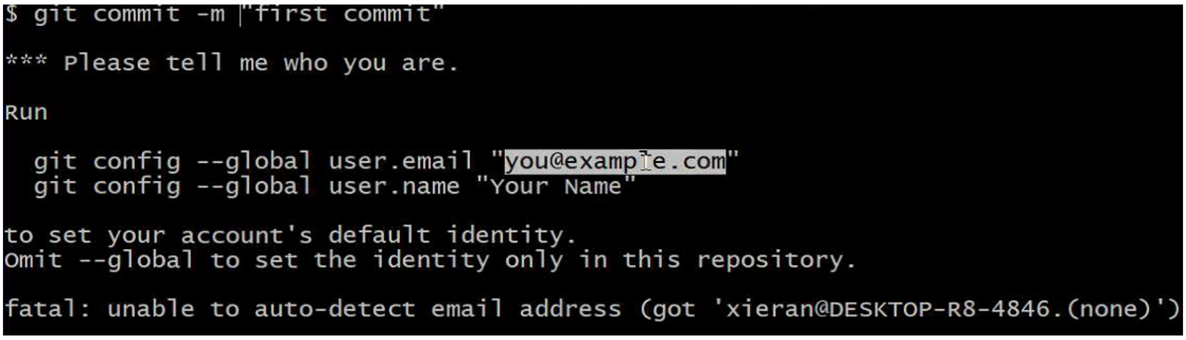
- Untracked files 未被跟踪的文件
- LF 换行
- CRLF 回车换行

初始化： git init （initial）在某个文件夹初始化一个仓库（一份代码只执行一次）

第一次使用时候：需要全局配置登录信息：用户名，用户邮箱

git config --global user.name "xam"

git config --global user.email "1457007317@qq.com"

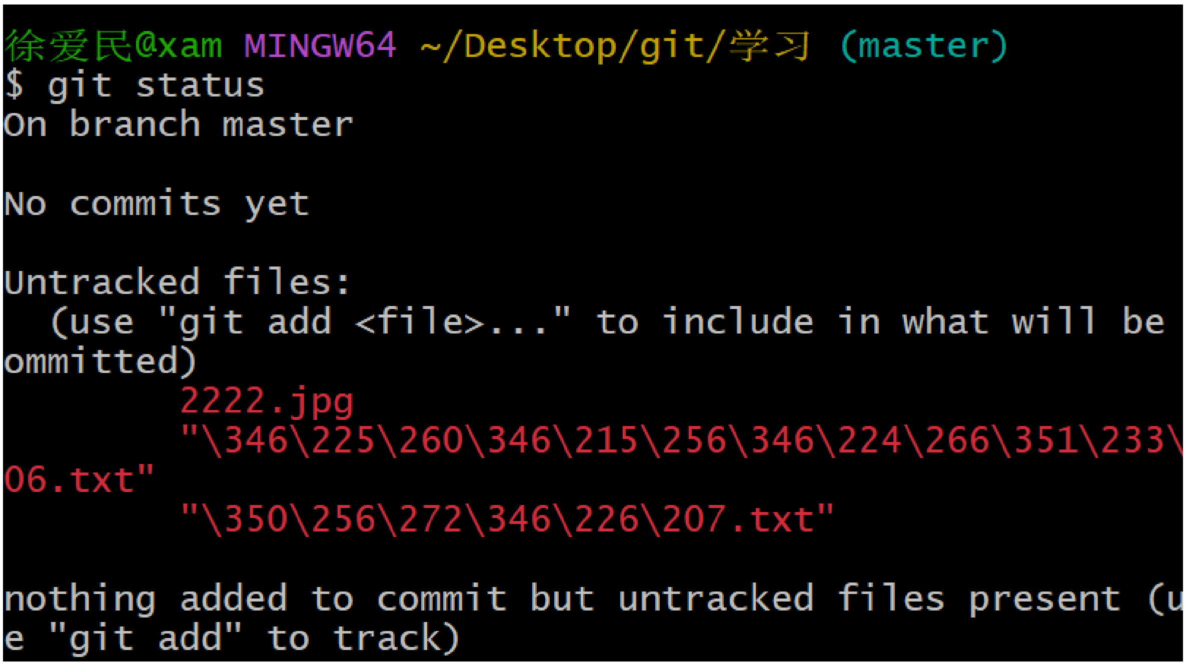


```
$ git commit -m "first commit"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'xieran@DESKTOP-R8-4846.(none)')
```

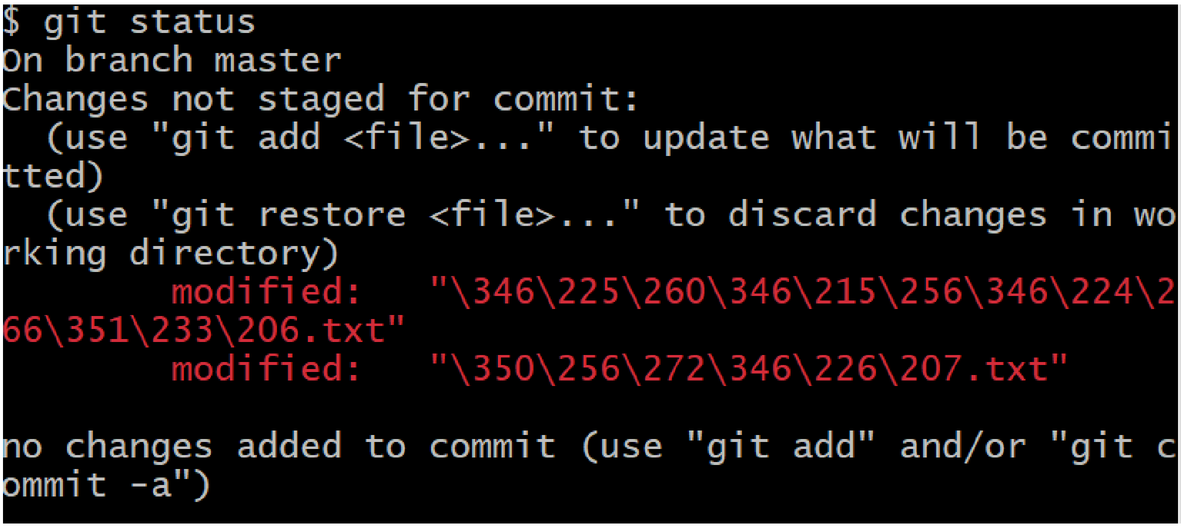
查看当前仓库状态(暂存区 与工作目录)： git status

```
**git status**
```

Changes not staged for commit：未提交的更改



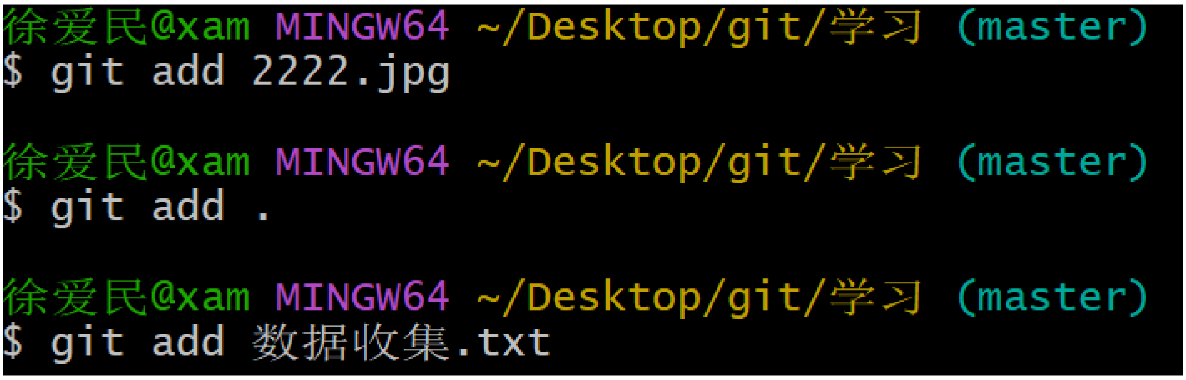
```
徐爱民@xam MINGW64 ~/Desktop/git/学习 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be
ommitted)
        2222.jpg
        "\346\225\260\346\215\256\346\224\266\351\233\
06.txt"
        "\350\256\272\346\226\207.txt"

nothing added to commit but untracked files present (u
e "git add" to track)
```



```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be commi
tted)
  (use "git restore <file>..." to discard changes in wo
rking directory)
        modified:   "\346\225\260\346\215\256\346\224\2
66\351\233\206.txt"
        modified:   "\350\256\272\346\226\207.txt"

no changes added to commit (use "git add" and/or "git c
ommit -a")
```

将文件添加到暂存区：git add

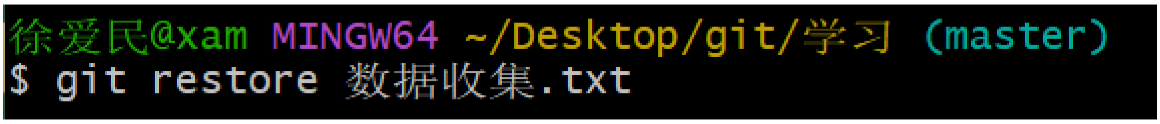
将所有文件添加到暂存区：git add .



```
徐爱民@xam MINGW64 ~/Desktop/git/学习 (master)
$ git add 2222.jpg

徐爱民@xam MINGW64 ~/Desktop/git/学习 (master)
$ git add .

徐爱民@xam MINGW64 ~/Desktop/git/学习 (master)
$ git add 数据收集.txt
```

还原某个文件(将文件从暂存区出来覆盖工作区对应文件)：git restore < file >

： 并不会提示相应合并信息而是直接合并

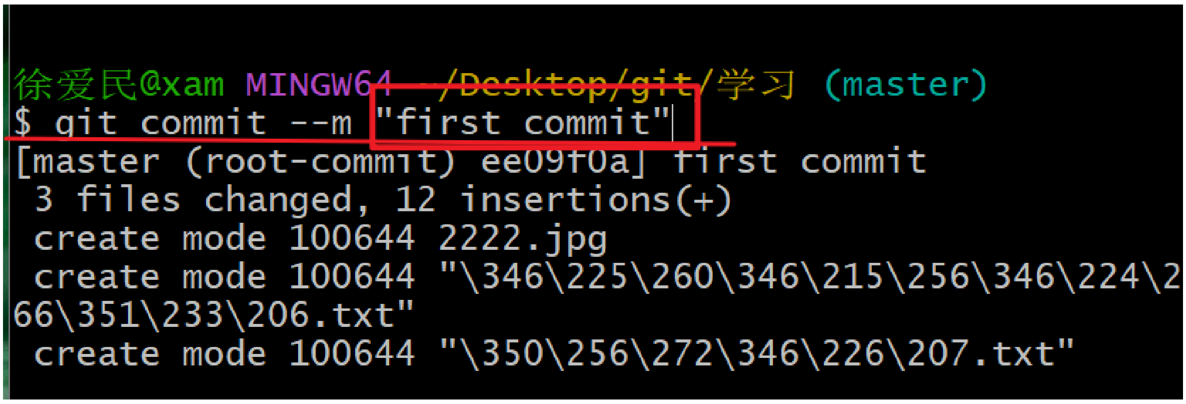


```
徐爱民@xam MINGW64 ~/Desktop/git/学习 (master)
$ git restore 数据收集.txt
```

提交:git commit -m "commit message"

===>"first coomit"   --->提交信息

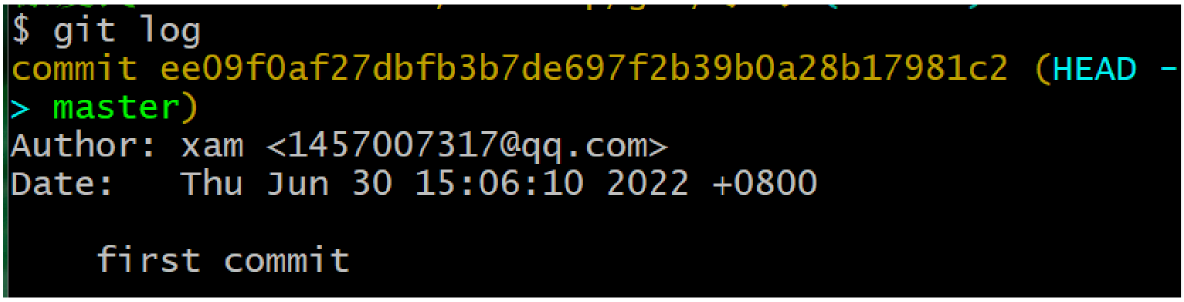
===> --m   ==   --message



```
徐爱民@xam MINGW64 ~/Desktop/git/学习 (master)
$ git commit --m "first commit"
[master (root-commit) ee09f0a] first commit
 3 files changed, 12 insertions(+)
 create mode 100644 2222.jpg
 create mode 100644 "\346\225\260\346\215\256\346\224\2
66\351\233\206.txt"
 create mode 100644 "\350\256\272\346\226\207.txt"
```

查看提交log日志：git log

(查看当前仓库的提交记录，如果多的话可以通过上下键翻页)



```
$ git log
commit ee09f0af27dbfb3b7de697f2b39b0a28b17981c2 (HEAD -
> master)
Author: xam <1457007317@qq.com>
Date:   Thu Jun 30 15:06:10 2022 +0800

    first commit
```

查看工作目录与暂存区前后文件具体差异： git diff

GIt bash here:

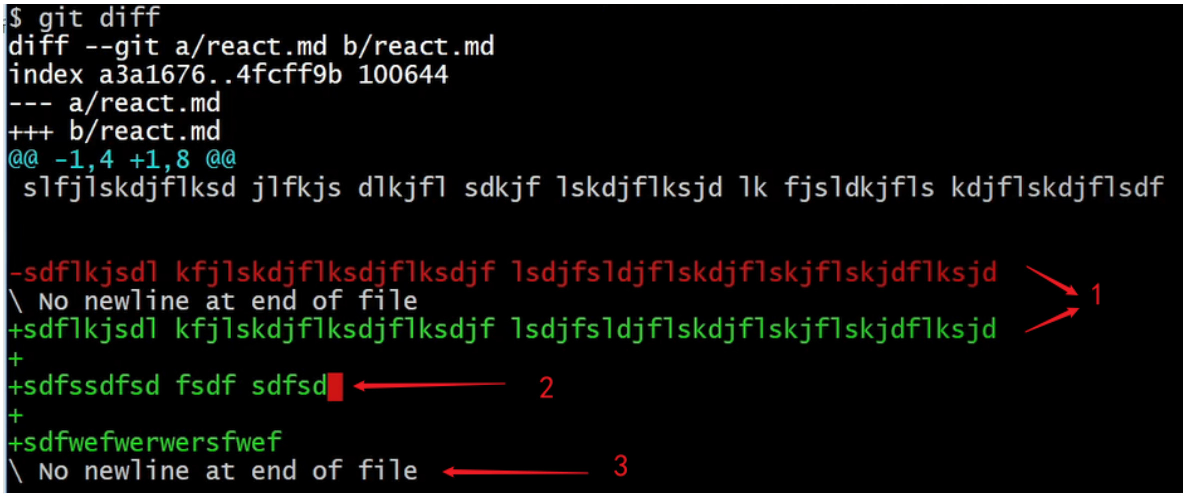
+表示增加

-表示删除

1.此处比较前后有无回车

2.行内末尾多余的空格

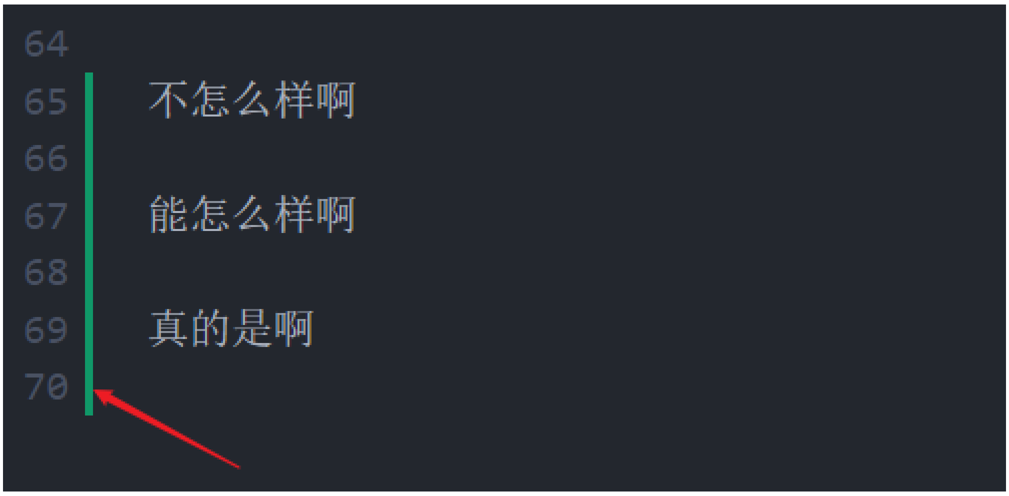
3.提示最后一行未添加换行



```
$ git diff
diff --git a/react.md b/react.md
index a3a1676..4fcff9b 100644
--- a/react.md
+++ b/react.md
@@ -1,4 +1,8 @@
 slfjlskdjflksd jlfkjs dlkjfl sdkjf lskdjflksjd lk fjsldkjfls kdjflskdjflsdf


-sdflkjsdl kfjlskdjflksdjflksdjf lsdjfsldjflskdjflskjflskjdflksjd
\ No newline at end of file
+sdflkjsdl kfjlskdjflksdjflksdjf lsdjfsldjflskdjflskjflskjdflksjd
+
+sdfssdfsd fsdf sdfsd
+
+sdfwefwerwersfwef
\ No newline at end of file
```

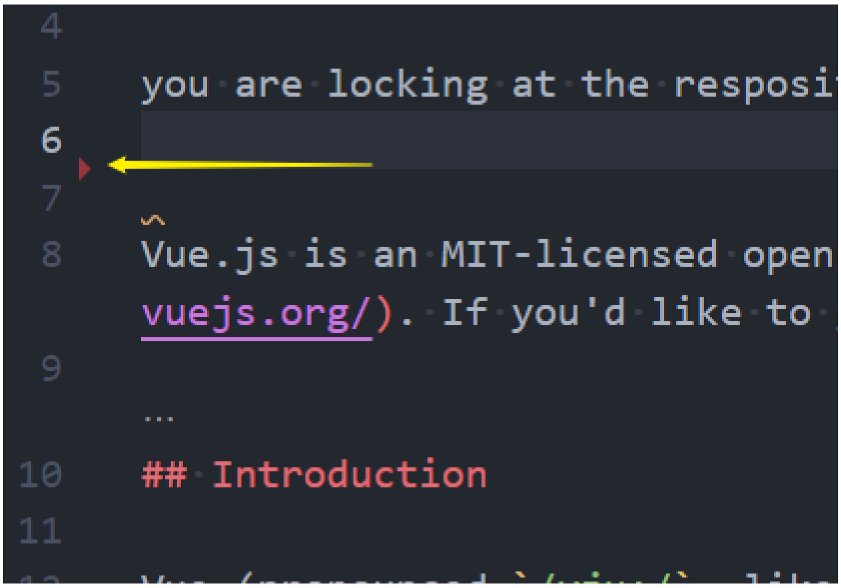
VScode中对比两个文件差异-->右键文件选中-选择以进行比较

绿色：新增



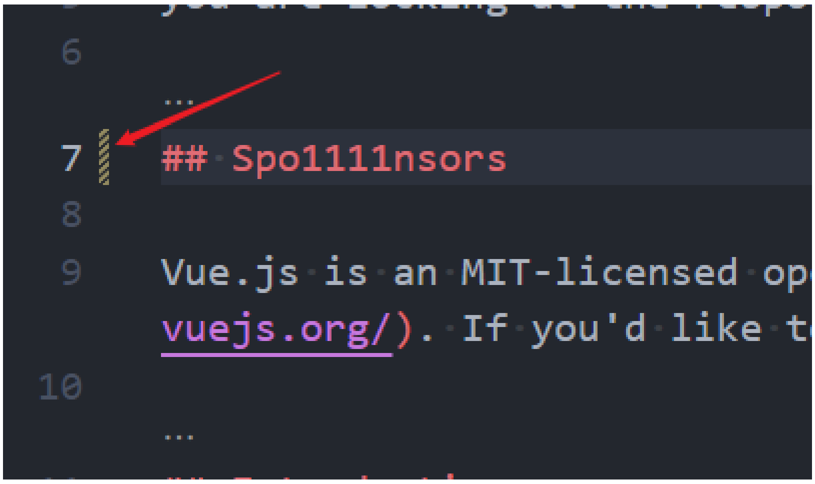
```
不怎么样啊

能怎么样啊

真的是啊
```

红色：删除



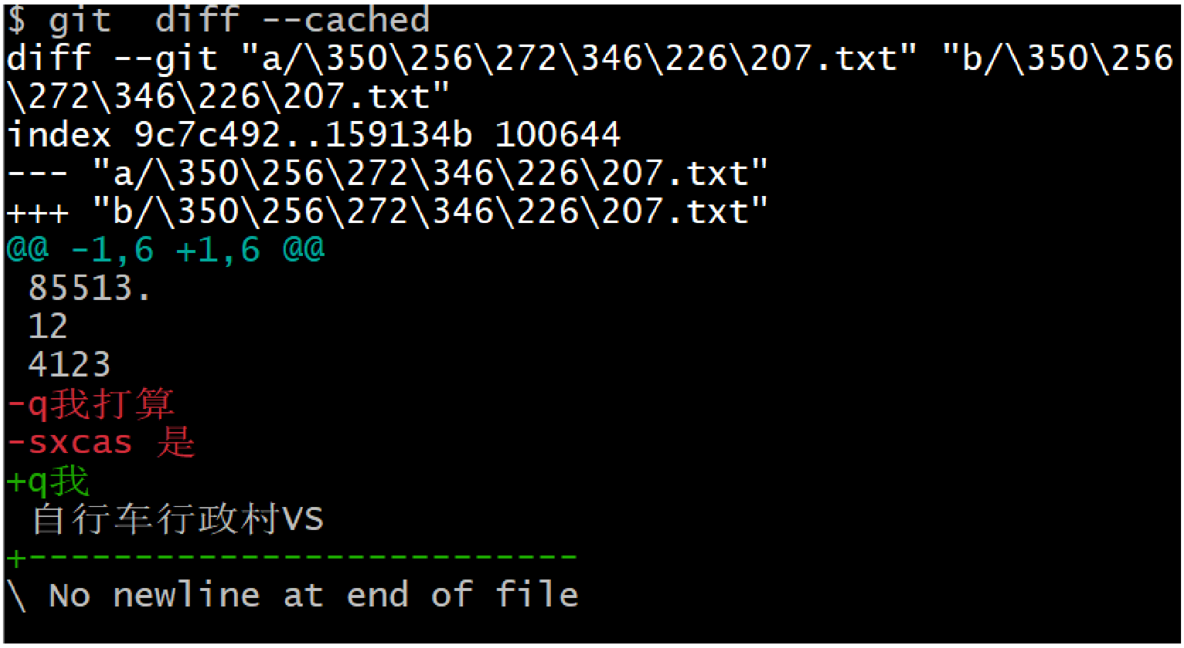
```
you are locking at the resposi

Vue.js is an MIT-licensed open
vuejs.org/). If you'd like to

...
## Introduction
```

黄色：修改

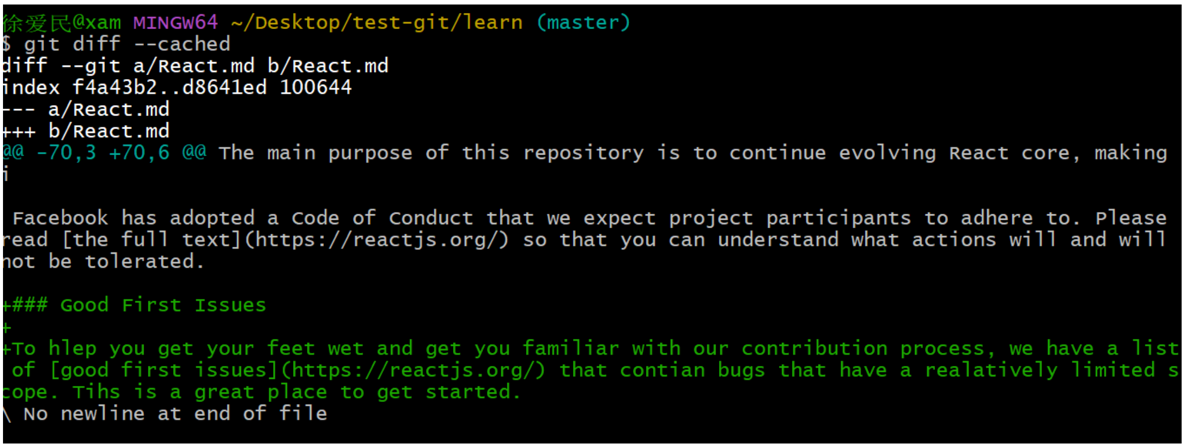


```
...
## Spo1111nsors

Vue.js is an MIT-licensed op
vuejs.org/). If you'd like t

...
```

查看暂存区与最后一提交的具体差异：git diff --cached

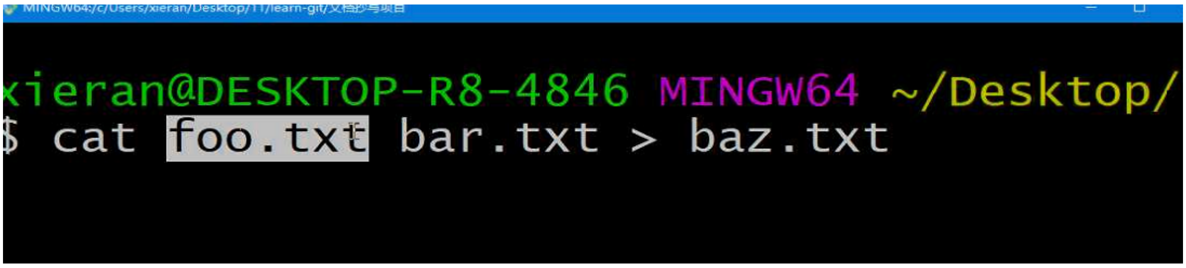


```
$ git  diff --cached
diff --git "a/\350\256\272\346\226\207.txt" "b/\350\256
\272\346\226\207.txt"
index 9c7c492..159134b 100644
--- "a/\350\256\272\346\226\207.txt"
+++ "b/\350\256\272\346\226\207.txt"
@@ -1,6 +1,6 @@
 85513.
 12
 4123
-q我打算
-sxcas 是
+q我
 自行车行政村vs
+--------------------------
\ No newline at end of file
```



```
徐爱民@xam MINGW64 ~/Desktop/test-git/learn (master)
$ git diff --cached
diff --git a/React.md b/React.md
index f4a43b2..d8641ed 100644
--- a/React.md
+++ b/React.md
@@ -70,3 +70,6 @@ The main purpose of this repository is to continue evolving React core, making
i

 Facebook has adopted a Code of Conduct that we expect project participants to adhere to. Please
read [the full text](https://reactjs.org/) so that you can understand what actions will and will
not be tolerated.

+### Good First Issues
+
+To hlep you get your feet wet and get you familiar with our contribution process, we have a list
 of [good first issues](https://reactjs.org/) that contian bugs that have a realatively limited s
cope. Tihs is a great place to get started.
\ No newline at end of file
```

合并两个文件若无换行，拼接的时候会出错

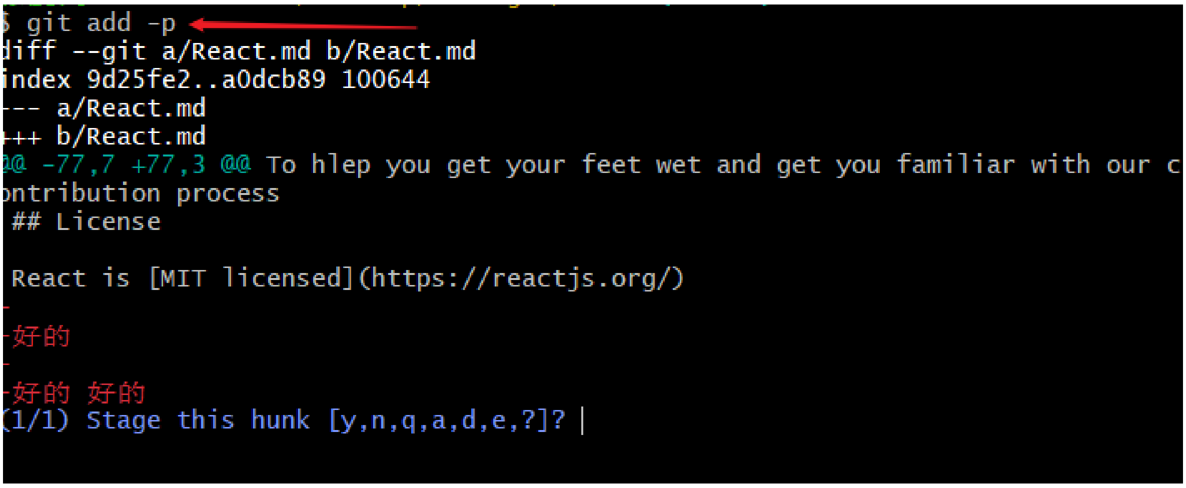


```
xieran@DESKTOP-R8-4846 MINGW64 ~/Desktop/
$ cat foo.txt bar.txt > baz.txt
```

git add -p：diff比较相应的修改进行暂存修改（自动识别）---进行原子性提交

（当代码变动时你想创建提交时，这个提交应该尽可能的小量，并且包含一个不可分割的特性（feature）、修复（fix）或优化（improved））

stage暂存



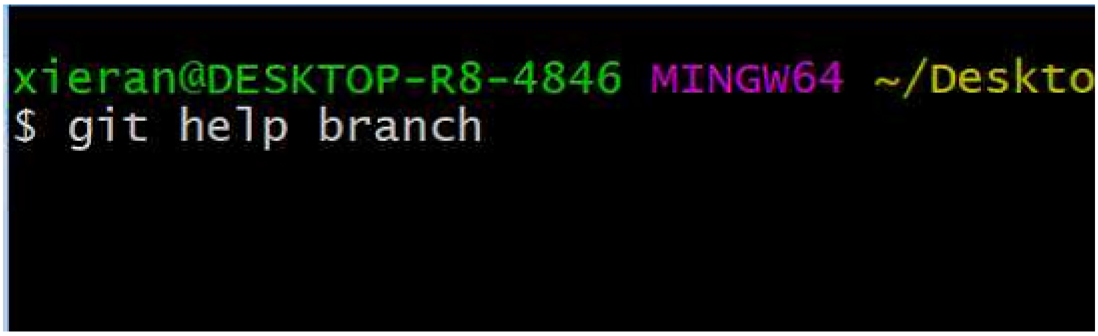
```
$ git add -p
diff --git a/React.md b/React.md
index 9d25fe2..a0dcb89 100644
--- a/React.md
+++ b/React.md
@@ -77,7 +77,3 @@ To hlep you get your feet wet and get you familiar with our c
ontribution process
 ## License

 React is [MIT licensed](https://reactjs.org/)
-
-好的
-
-好的 好的
(1/1) Stage this hunk [y,n,q,a,d,e,?]? |
```

```
Stage this hunk [y,n,a,d,/,j,J,g,e,?]? ?

y - stage this hunk

n - do not stage this hunk

a - stage this and all the remaining hunks in the file

d - do not stage this hunk nor any of the remaining hunks in the file

g - select a hunk to go to

/ - search for a hunk matching the given regex

j - leave this hunk undecided, see next undecided hunk

J - leave this hunk undecided, see next hunk

k - leave this hunk undecided, see previous undecided hunk

K - leave this hunk undecided, see previous hunk

s - split the current hunk into smaller hunks

e - manually edit the current hunk

? - print help
```

```
Stage this hunk [y,n,q,a,d,/,j,J,g,e,?]? n
```

查看命令帮助: git hlep xxx

```
git hlep xxx
```



```
xieran@DESKTOP-R8-4846 MINGW64 ~/Deskto
$ git help branch
```

回退版本/提交：

回退到当前最新提交：git reset --hard HEAD

回退到上次提交：git reset --hard HEAD^

回退到上n次提交：git reset --hard HEAD~n

回退到某次提交：git reset --hard commitid

回到未来版本：git reset --hard commitid

撤销修改：

1.工作区文件撤销没有提交到暂存区/没有git add

--git checkout 文件名

2.将暂存区文件撤销到工作区

--git checkout 文件名

--git reset HEAD 文件不带--hard

3.删除文件从版本库中删除文件

--git rm修改后需要提交

GitHub上传操作(初次)

将仓库名添加进README.md文件中

```
echo "# learn-git" >> README.md
```

初始化仓库

```
git init
```

创建README.md文档（若仓库中有则不需要创建）

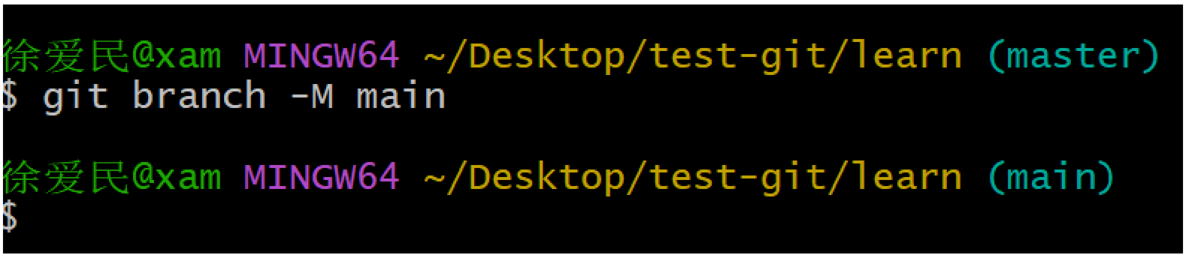
```
git add README.md
```

commit提交

```
git commit -m "first commit"
```

重命名分支(将oldName设置成newName)

```
git branch -M main
```



```
徐爱民@xam MINGW64 ~/Desktop/test-git/learn (master)
$ git branch -M main

徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$
```

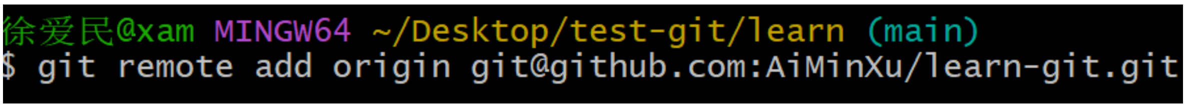
添加一个远程仓库的地址: git remote add origin 地址

--git remote add 地址

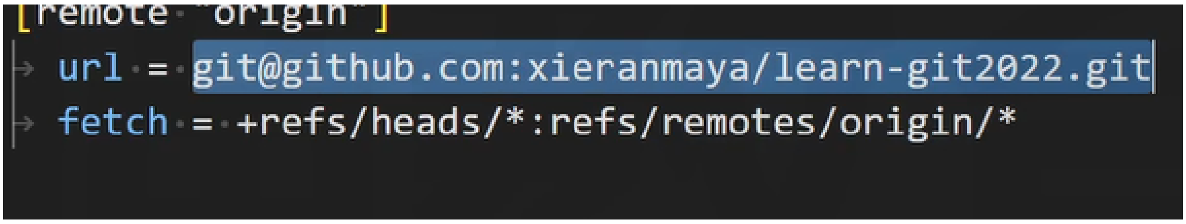
origin（源）--是为后面的地址取的名字

可以在仓库.git文件夹内.config文件中看到相应信息

```
git remote add origin git@github.com:AiMinXu/learn-git.git
```



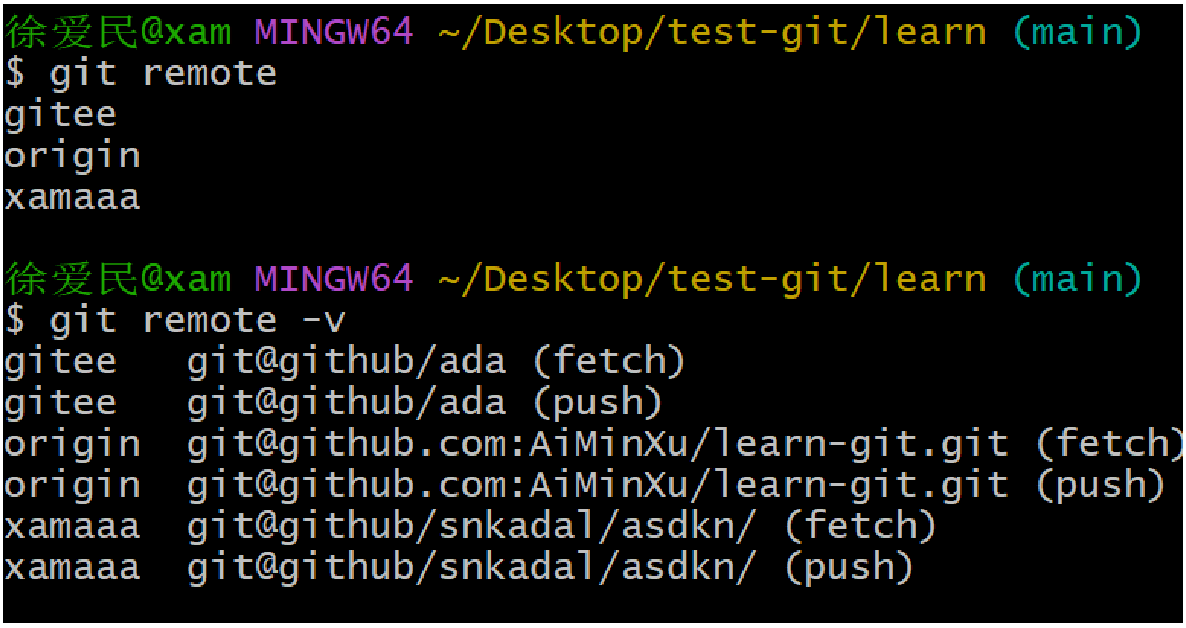
```
徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$ git remote add origin git@github.com:AiMinXu/learn-git.git
```



```
[remote "origin"]
    url = git@github.com:xieranmaya/learn-git2022.git
    fetch = +refs/heads/*:refs/remotes/origin/*
```

查看远程仓库信息:git remote (查看仓库) git remote -v （查看仓库信息）

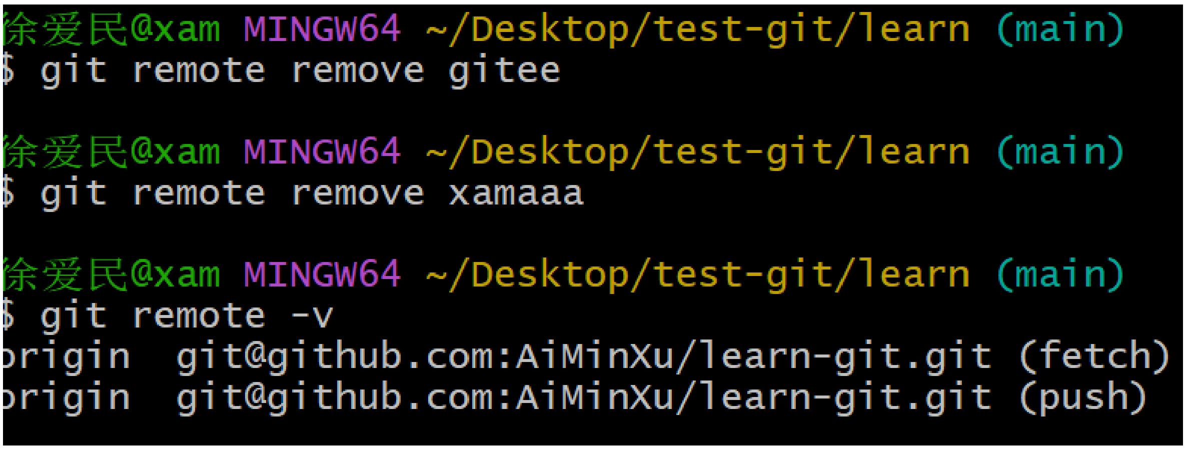
```
git remote  （查看仓库）
git remote -v （查看仓库信息）
```



```
徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$ git remote
gitee
origin
xamaaa

徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$ git remote -v
gitee   git@github/ada (fetch)
gitee   git@github/ada (push)
origin  git@github.com:AiMinXu/learn-git.git (fetch)
origin  git@github.com:AiMinXu/learn-git.git (push)
xamaaa  git@github/snkadal/asdkn/ (fetch)
xamaaa  git@github/snkadal/asdkn/ (push)
```

删除远程仓库: git remote remove 仓库名字

```
git remote remove 仓库名字
```



```
徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$ git remote remove gitee

徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$ git remote remove xamaaa

徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$ git remote -v
origin  git@github.com:AiMinXu/learn-git.git (fetch)
origin  git@github.com:AiMinXu/learn-git.git (push)
```

将分支代码推送到对应仓库：git push （-u origin main)

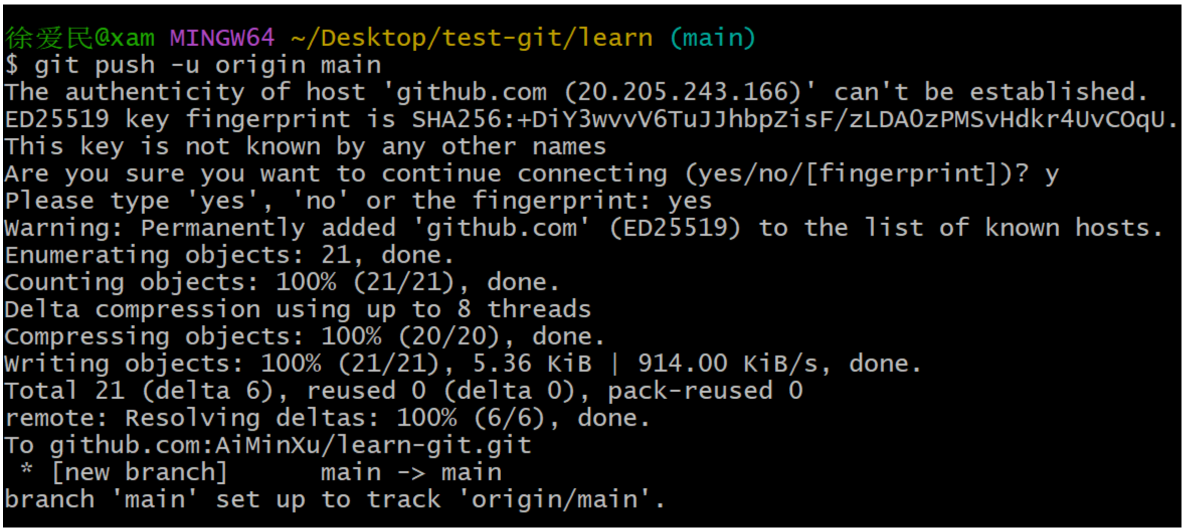
```
git push -u origin main
```

将分支代码推送（上传）到对应仓库，同时将分支与远程仓库绑定，后续只需要git push即可推送

-u(--set-upstream)

```
（git push -u 仓库名字 分支名字）
git push -u origin main
```

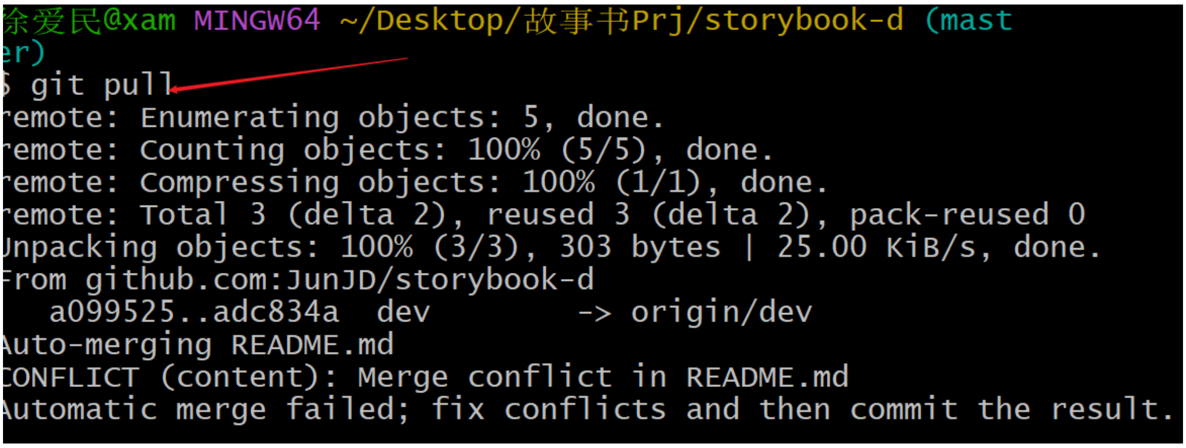
确定公私密钥确定连接，需输入 yes



```
徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$ git push -u origin main
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (21/21), 5.36 KiB | 914.00 KiB/s, done.
Total 21 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To github.com:AiMinXu/learn-git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

将远程仓库最新修改更新到本地：git pull

```
git pull
```



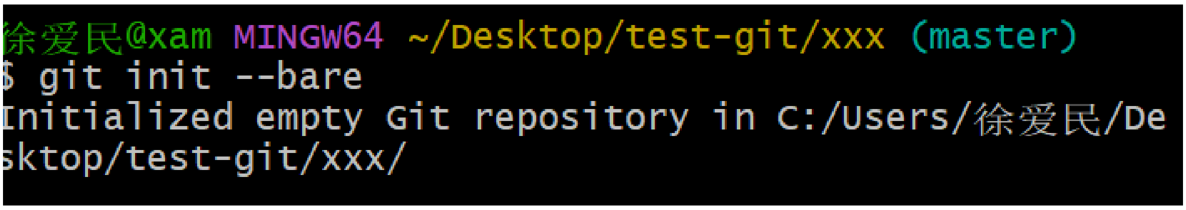
```
徐爱民@xam MINGW64 ~/Desktop/故事书Prj/storybook-d (mast
er)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 2), reused 3 (delta 2), pack-reused 0
Unpacking objects: 100% (3/3), 303 bytes | 25.00 KiB/s, done.
From github.com:JunJD/storybook-d
   a099525..adc834a  dev        -> origin/dev
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

更新远程仓库信息：git fetch

```
git fetch
```

将该文件夹作为仓库，不作为工作目录：git init --bare

```
git init --bare
```

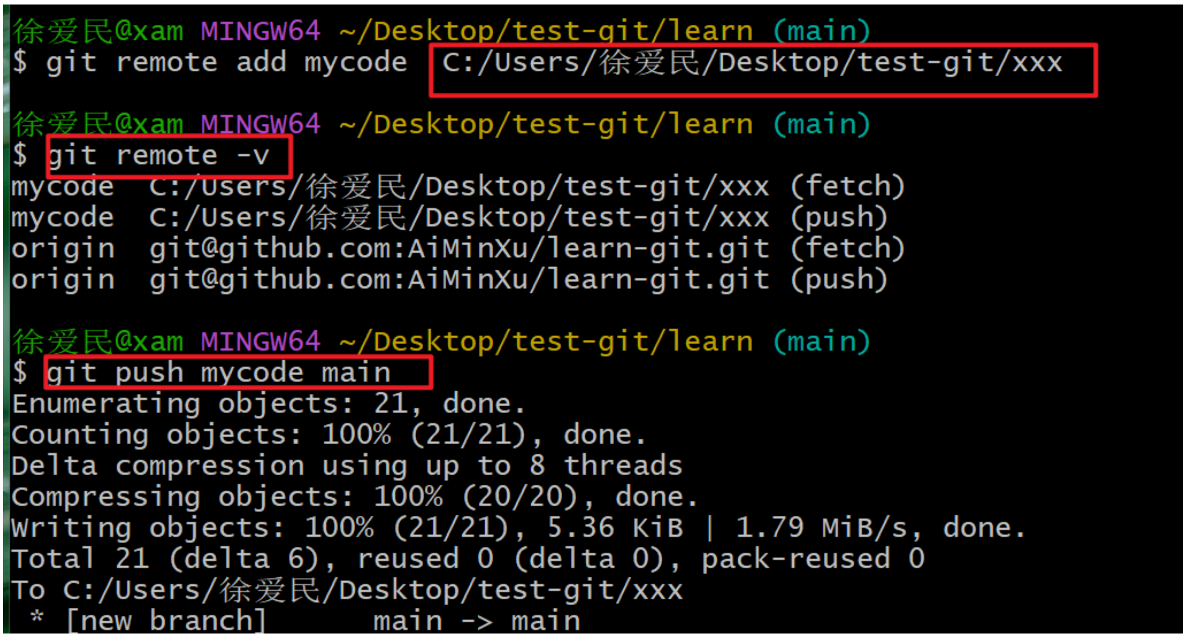


```
徐爱民@xam MINGW64 ~/Desktop/test-git/xxx (master)
$ git init --bare
Initialized empty Git repository in C:/Users/徐爱民/De
sktop/test-git/xxx/
```

将该文件夹设置成仓库，并将拉取main分支代码到mycode仓库中

（注意地址中的 \ 需要改成 / 绝对路径）

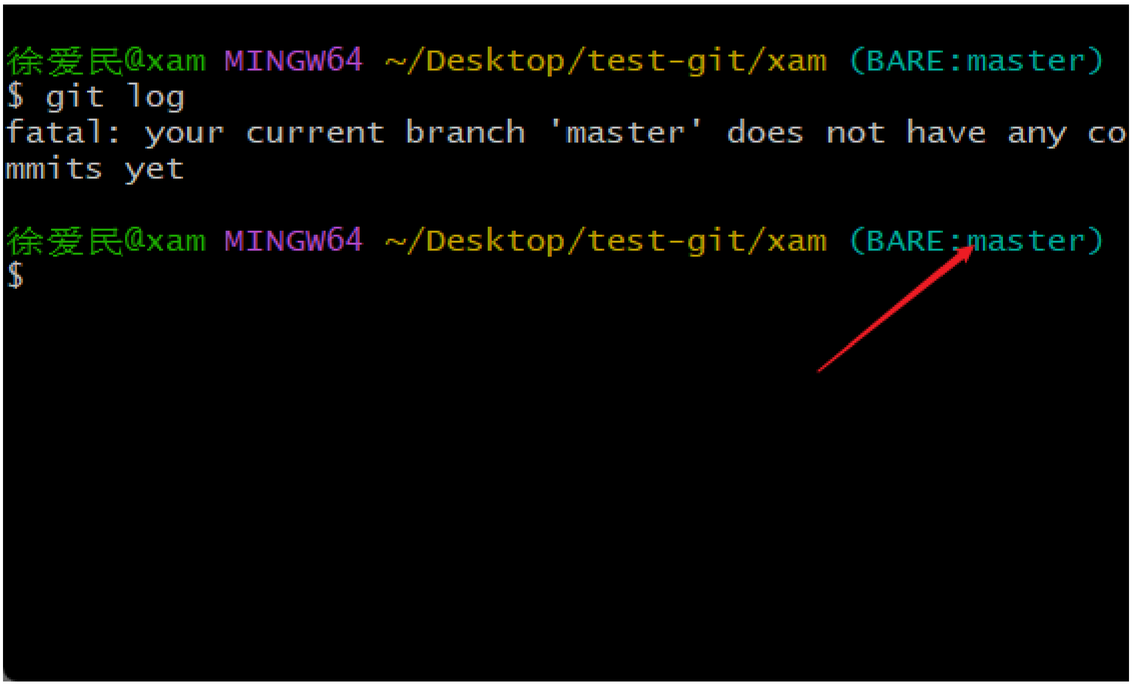
```
git remote add mycode  C:/Users/徐爱民/Desktop/test-git/xxx
```



```
徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$ git remote add mycode  C:/Users/徐爱民/Desktop/test-git/xxx

徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$ git remote -v
mycode  C:/Users/徐爱民/Desktop/test-git/xxx (fetch)
mycode  C:/Users/徐爱民/Desktop/test-git/xxx (push)
origin  git@github.com:AiMinXu/learn-git.git (fetch)
origin  git@github.com:AiMinXu/learn-git.git (push)

徐爱民@xam MINGW64 ~/Desktop/test-git/learn (main)
$ git push mycode main
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (21/21), 5.36 KiB | 1.79 MiB/s, done.
Total 21 (delta 6), reused 0 (delta 0), pack-reused 0
To C:/Users/徐爱民/Desktop/test-git/xxx
 * [new branch]      main -> main
```

文件夹远程仓库状态： git log

```
git log
```



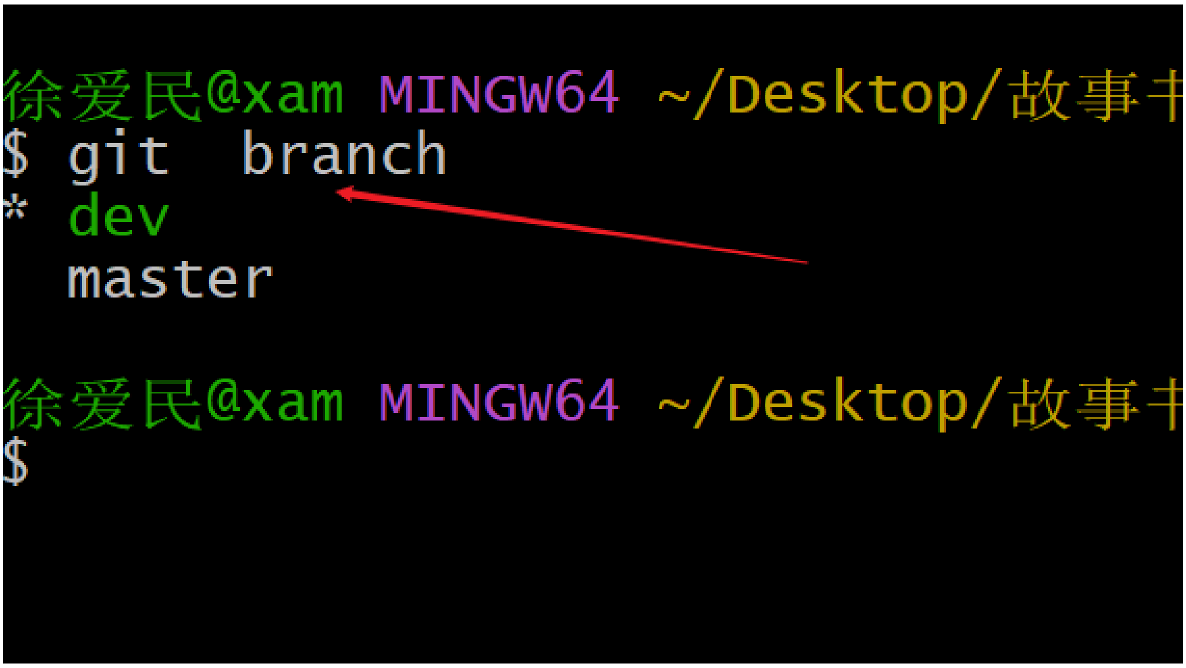
```
徐爱民@xam MINGW64 ~/Desktop/test-git/xam (BARE:master)
$ git log
fatal: your current branch 'master' does not have any co
mmits yet

徐爱民@xam MINGW64 ~/Desktop/test-git/xam (BARE:master)
$
```

git push mycode mian已经执行过现在只需要git push

```
git push
```

查看当前分支：git branch

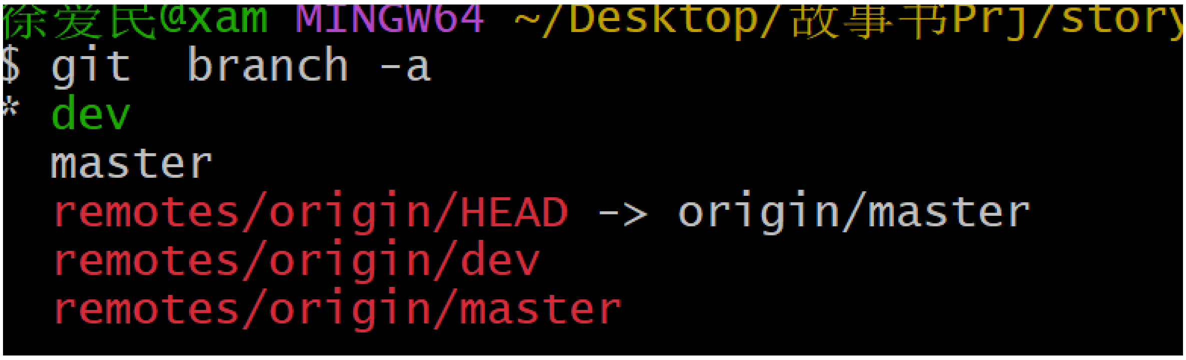
```
git branch
```



```
徐爱民@xam MINGW64 ~/Desktop/故事书
$ git  branch
* dev
  master

徐爱民@xam MINGW64 ~/Desktop/故事书
$
```

查看所有分支信息：git branch -a

```
git branch -a
```



```
徐爱民@xam MINGW64 ~/Desktop/故事书Prj/story
$ git  branch -a
* dev
  master
  remotes/origin/HEAD -> origin/master
  remotes/origin/dev
  remotes/origin/master
```

切换分支：git checkout 分支名

```
git checkout 分支名
```

创建分支：

创建分支：git branch 分支名

注意：在本地仓库操作，创建的都是本地分支

```
git branch 分支名
```

创建并切换分支：git checkout -b 分支名

注意：在本地仓库操作，创建的都是本地分支

```
git checkout -b 分支名
```

合并分支：

git merge 合并某分支到当前分支

```
git merge 合并某分支到当前分支
```

git merge --no-ff 分支名

```
git merge --no-ff 分支名
```

git rebase**若无特殊需要不建议使用**

删除分支：

删除本地分支：

删除未合并分支：git branch -D 分支名

删除已合并分支：git branch -d 分支名

删除远程分支：（此处建议界面操作）

git push origin -d 分支名

git push -d 分支名