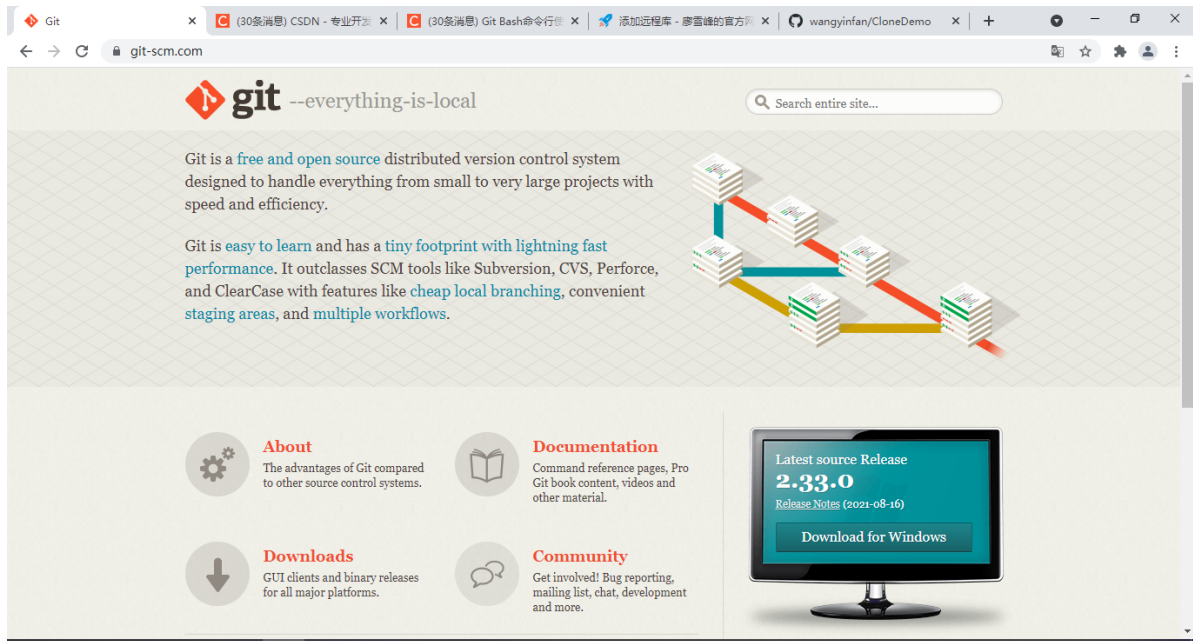
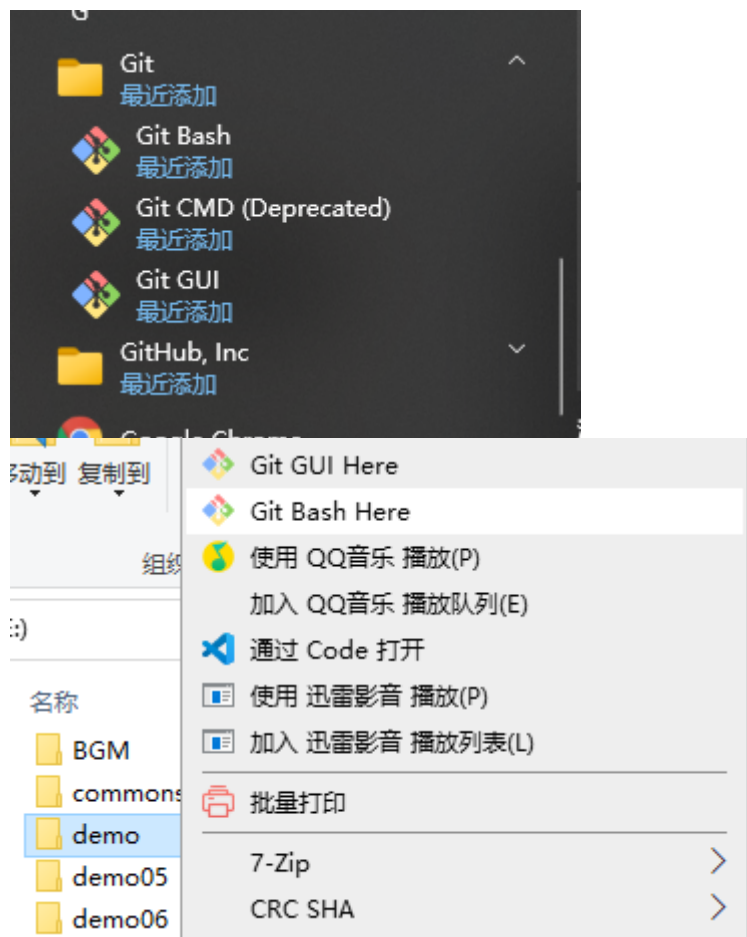


下载

下载链接: <https://git-scm.com/>



安装



左图：成功安装后同时添加的三个软件，我们只使用Git Bash执行命令；

右图：Git bash的界面，一般右键需要提交的文件夹，选择“git bash here”即可开始输入命令

配置

配置用户名

```
git config --global user.name 'name'
```

```
WYF@DESKTOP-V95D05B MINGW64 /e/test
$ git config --global user.name 'wangyinfan'
```

配置邮箱

```
git config --global user.email 'email'
```

```
WYF@DESKTOP-V95D05B MINGW64 /e/test
$ git config --global user.email '1959198478@qq.com'
```

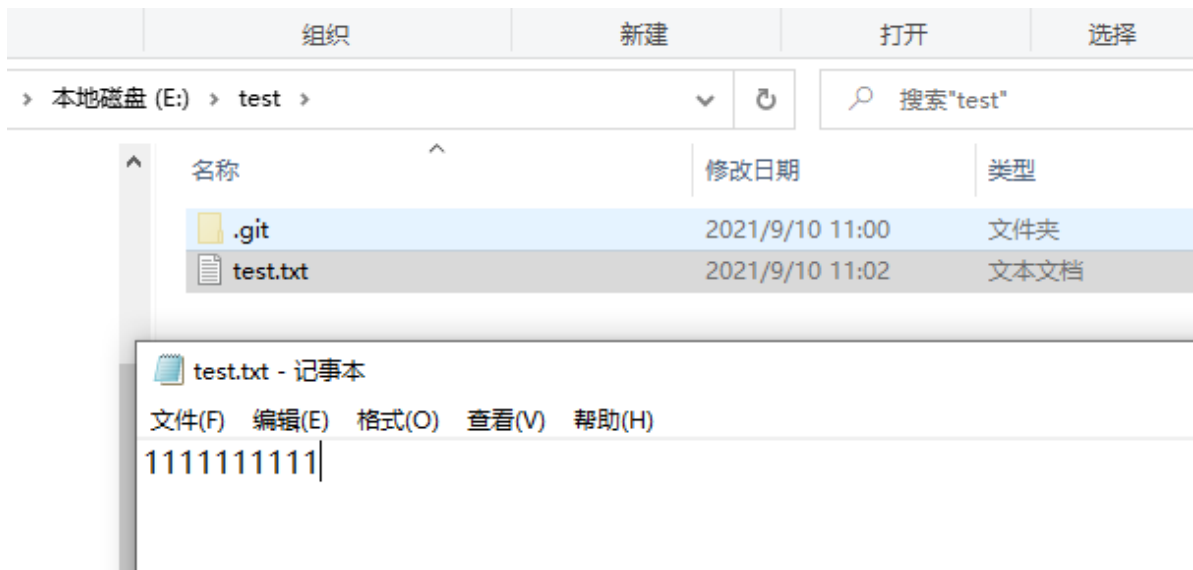
初始化本地版本库

```
git init
```

```
WYF@DESKTOP-V95D05B MINGW64 /e/test
$ git init
Initialized empty Git repository in E:/test/.git/
```

创建文件并提交本地库

第一步：在文件夹中创建文件，添加内容



第二步：分别提交到暂存区，本地库

```
git add file //将文件提交到暂存区
```

```
git commit -m "这是注释" // 将文件提到本地库
```

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git add test.txt

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git commit -m "第一行"
[master (root-commit) 768c20a] 第一行
1 file changed, 1 insertion(+)
create mode 100644 test.txt
```

修改文件并提交本地库

```
git add file
```

```
git commit -m
```

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git diff test.txt
diff --git a/test.txt b/test.txt
index aa75558..58fa72a 100644
--- a/test.txt
+++ b/test.txt
@@ -1,2 @@
-1111111111
\ No newline at end of file
+1111111111
+2222222222
\ No newline at end of file

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git add test.txt

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git commit -m "添加第二行"
[master 521675e] 添加第二行
1 file changed, 2 insertions(+), 1 deletion(-)
```

版本回退或前进

查看带有版本号的显示日志 (从最近开始) `git reflog`

回退上一版本 `git reset --hard HEAD^`

回退 / 前进到某一版本: `git reset --hard 版本号`

(1) 查看带有版本号的显示日志

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git reflog
521675e (HEAD -> master) HEAD@{0}: commit: 添加第二行
768c20a HEAD@{1}: commit (initial): 第一行
```

每一个操作对应的版本号

(2) 回退上一版本

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git reset --hard HEAD^
HEAD is now at 768c20a 第一行
```

(3) 回退 / 前进到某一版本

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git reset --hard 521675e
HEAD is now at 521675e 添加第二行

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ cat test.txt
1111111111
2222222222
```

创建分支

创建分支: `git branch <name>`

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git branch dev
```

查看分支: `git branch`

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git branch
dev
* master
```

切换分支: `git checkout <name>` 或者 `git switch <name>`

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git checkout dev
Switched to branch 'dev'
```

创建+切换分支: `git checkout -b <name>` 或者 `git switch -c <name>`

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (dev)
$ git checkout -b fanzhi1
Switched to a new branch 'fanzhi1'
```

合并分支

合并某分支到当前分支: `git merge <name>`

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git merge dev
Updating 521675e..389dbea
Fast-forward
 test.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

删除分支保存分支内容

删除分支: `git branch -d <name>`

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git branch -d dev
Deleted branch dev (was 38fe627).
```

删除分支并保存分支内容 `git merge --no-ff -m "merge with no-ff"`
`<name>`

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git merge --no-ff -m "merge with no-ff" dev
Merge made by the 'recursive' strategy.
 test.txt | 1 +
 1 file changed, 1 insertion(+)

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git branch -d dev
Deleted branch dev (was 38fe627).

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git log --graph --pretty=oneline --abbrev-commit
* 5b399a5 (HEAD -> master) merge with no-ff
|
| * 38fe627 dev添加第三行
|/
* 9031fe1 恢复
* ed4bd45 fanzhi添加内容Merge branch 'fanzhi1'
|
| * d03447d 在fenzhi1添加第三行
| * 389dbea dev添加第三行
|/
* 521675e 添加第二行
* 768c20a 第一行
```

撤销修改

```
git checkout -- file.name
```

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ cat test.txt
1111111111
2222222222
3333333333
4444444444
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git checkout -- test.txt

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ cat test.txt
1111111111
2222222222
3333333333
```

删除文件

```
rm test.txt
```

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ rm test.txt
```

生成并添加ssh公钥

生成 / 添加公钥

```
mkdir key_backup
```

```
ssh-keygen -t rsa -C "1959198478@qq.com"
```

```

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ cd ~/.ssh

WYF@DESKTOP-V95D05B MINGW64 ~/.ssh
$ mkdir key_backup
mkdir: cannot create directory 'key_backup': File exists

WYF@DESKTOP-V95D05B MINGW64 ~/.ssh
$ cp id_rsa* key_backup

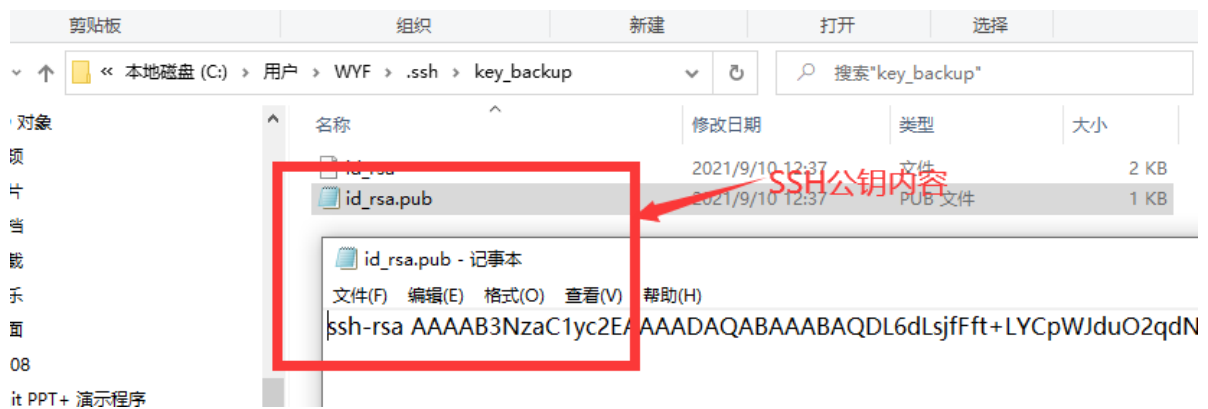
WYF@DESKTOP-V95D05B MINGW64 ~/.ssh
$ rm id_rsa*

WYF@DESKTOP-V95D05B MINGW64 ~/.ssh
$ ssh-keygen -t rsa -C "1959198478@qq.com"
Too many arguments.
usage: ssh-keygen [-q] [-b bits] [-t dsa | ecdsa | ed25519

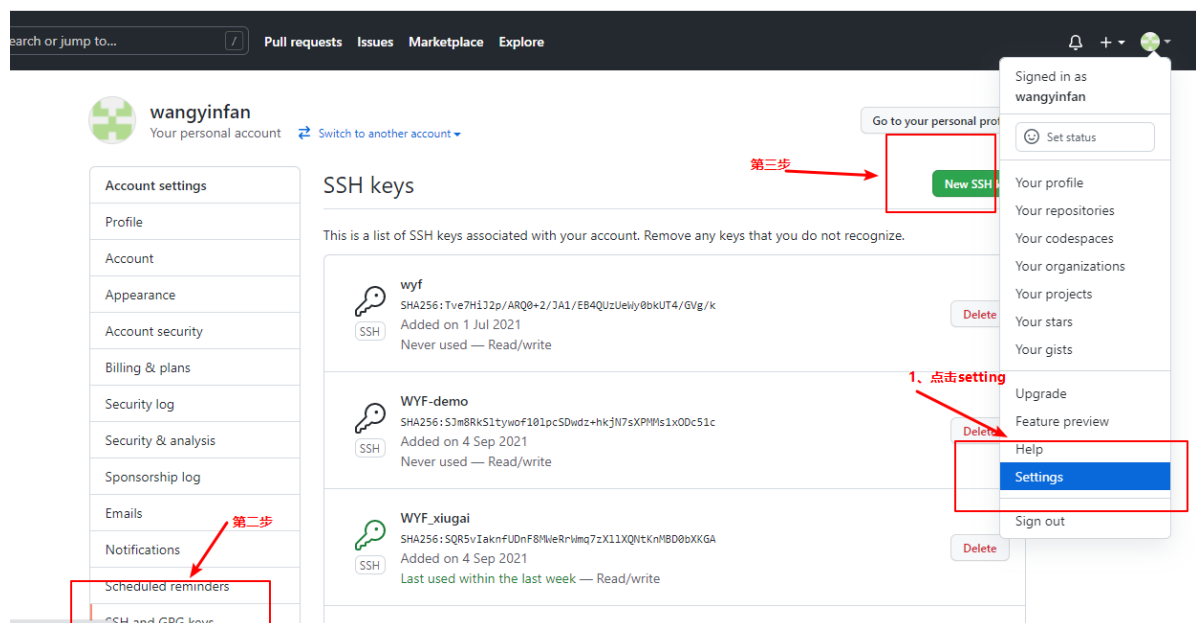
```

如果已有公钥，可删除再添加

这里要注意：.pub文件里面存的就是我们需要的公钥代码



新建公钥



添加公钥

SSH keys / Add new

Title

test02

Key

ssh-rsa

AAAAB3NzaC1yc2EAAAADAQABAAQDAQL6dLsjfFft+LYCpWJduO2qdN176dOEAHcuEBzyCiy/1WTndCL17/JsWTT4
uCVywuSmqo6xbwBynTmBYLwn/xmIUCMHMrizEZKBDDfW7+hYImP/8R5kYBzeclZcqsV/SAH3xSMRmg9r9bs2
CoNvLynGfDPV+BpL0iISHf...ig4DCq8E09BOWxGv/K3xuojamfzDbxu0EPIMRwvuE8UpcliTWuQ55zKFTixcSgt8Cu0Rr
OmPLXLMJv7FLJrcIai+IE/oDrM5OY+tgMaklkQGowvDeEGBiJ+N35izQg7IO5ec8RC7N2Ee3yPdmKSCQcAO00peYKs
gAmSdpP0+7X 1959198478@qq.com

复制过去

id_rsa.pub - 记事本

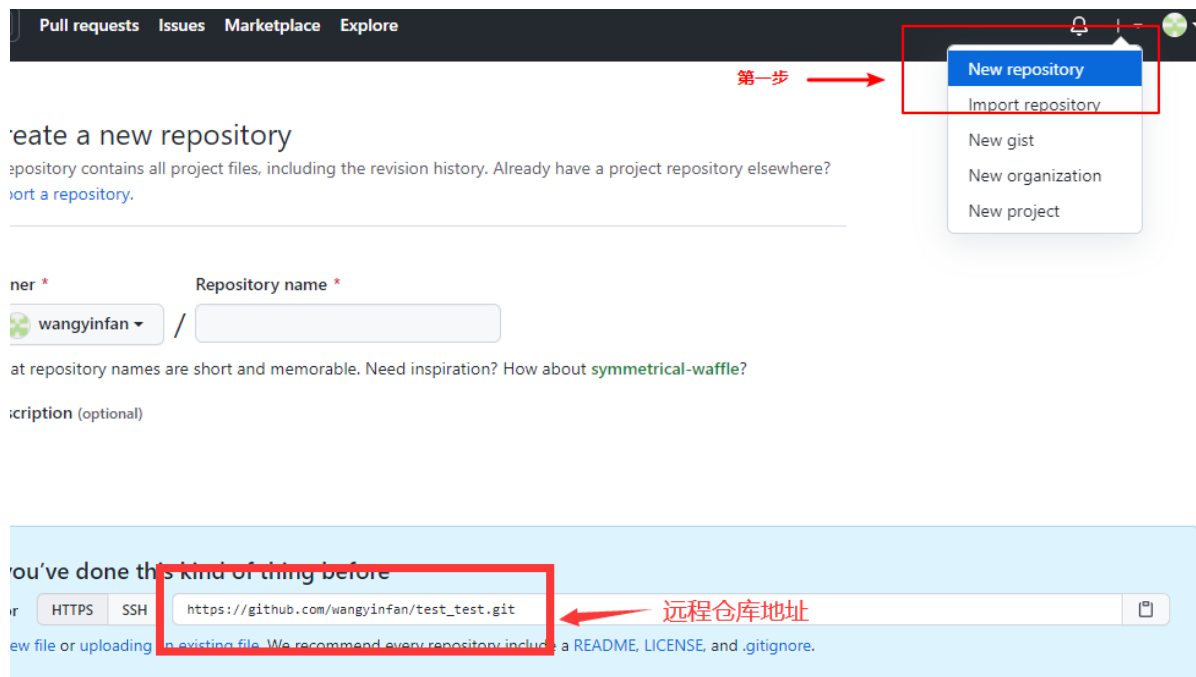
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQL6dLsjfFft+LYCpWJduO2qdN176d

Add SSH key

本地仓库推送GitHub仓库

第一步：新建远程库



Pull requests Issues Marketplace Explore

第一步 → **New repository**

Import repository

New gist

New organization

New project

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * **wangyinfan** / Repository name *

Repository description (optional)

Repository type **New repository**

Repository visibility **Public**

Repository location **writable**

You've done this kind of thing before

Repository address **https://github.com/wangyinfan/test_test.git** ← 远程仓库地址

HTTPS SSH

new file or uploading an existing file. We recommend every repository includes a README, LICENSE, and .gitignore.

repository on the command line

本地仓库关联远程仓库 —— git remote add origin 远程仓库地址

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git remote add origin https://github.com/wangyinfan/test_test.git
```

本地仓库master主分支推送到远程仓库—— git push -u origin master 远程地址

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git push -u origin master
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (16/16), 1.33 KiB | 680.00 KiB/s, done.
Total 16 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/wangyinfan/test_test.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

本地仓库克隆

git clone 远程仓库地址

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git clone https://github.com/wangyinfan/CloneDemo.git
Cloning into 'CloneDemo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

常见冲突

分支与主分支同时修改文件

1、在分支dev里修改第三行

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git checkout -b dev
Switched to a new branch 'dev'
WYF@DESKTOP-V95D05B MINGW64 /e/test (dev)
$ git add test.txt
WYF@DESKTOP-V95D05B MINGW64 /e/test (dev)
$ git commit -m "在dev修改"
[dev 738c726] 在dev修改
1 file changed, 2 insertions(+), 1 deletion(-)
```

1、在dev分支里修改第三行

2、切换至主分支master，同样修改第三行

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (dev)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ cat test.txt
1111111111
2222222222
3333333333

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git add test.txt

WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git commit -m "master修改"
[master 7dfd415] master修改
1 file changed, 2 insertions(+), 1 deletion(-)
```

2、在主分支master修改第三行

3、合并时提示有冲突

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master)
$ git merge dev
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
Automatic merge failed; fix conflicts and then commit the result.

WYF@DESKTOP-V95D05B MINGW64 /e/test (master|MERGING)
$ cat test.txt
1111111111
2222222222
3333333333
<<<<<<< HEAD
5555555555
=====
4444444444
>>>>>>> dev
```

警告：有合并冲突

提示冲突内容

解决方法：修改冲突部分内容，再次提交即可

```
WYF@DESKTOP-V95D05B MINGW64 /e/test (master|MERGING)
$ cat test.txt
1111111111
2222222222
3333333333
4444444444

WYF@DESKTOP-V95D05B MINGW64 /e/test (master|MERGING)
$ git add test.txt

WYF@DESKTOP-V95D05B MINGW64 /e/test (master|MERGING)
$ git commit -m "解决冲突"
[master 2709cf8] 解决冲突
```

