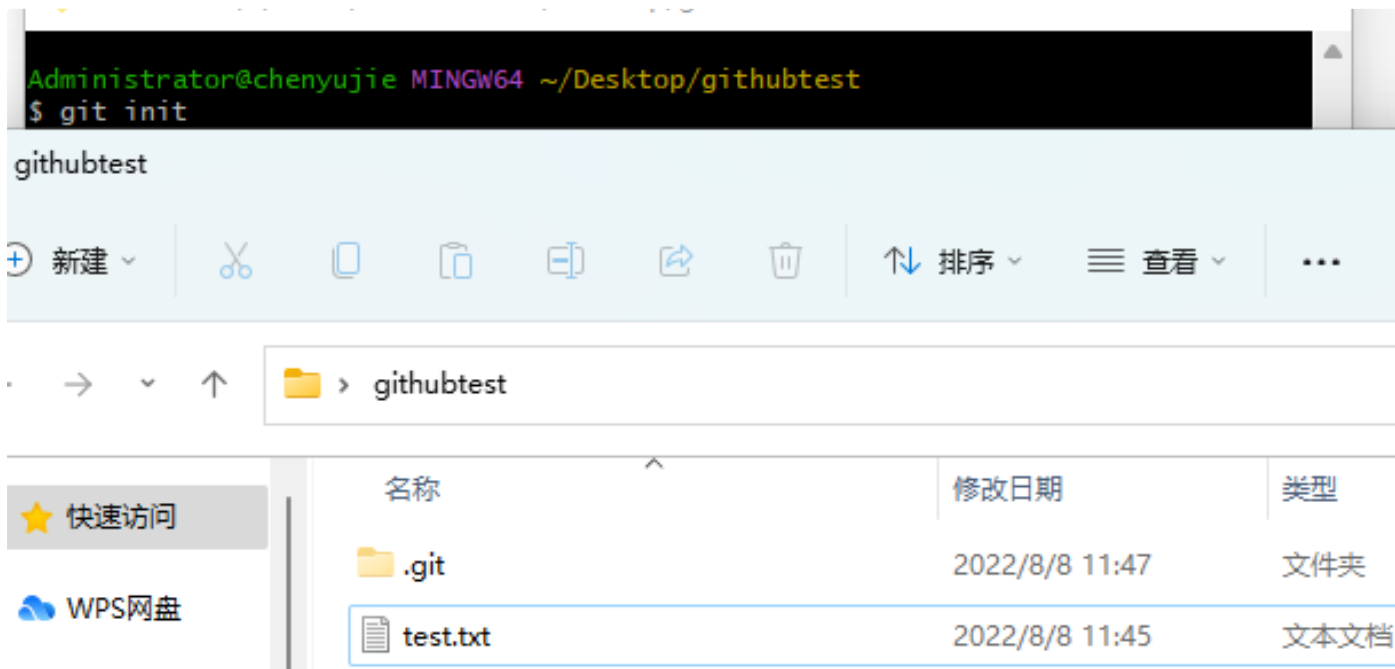


# github 学习心得

在电脑上安装 git 时候 右击进行启动

在一个全新的文件夹里 需要初始化仓库 git init 然后电脑上会生成一个隐藏的文件夹.git 代表配置等文件



然后在 git 命令里面 连接自己的 github 仓库 指定用户名和名字

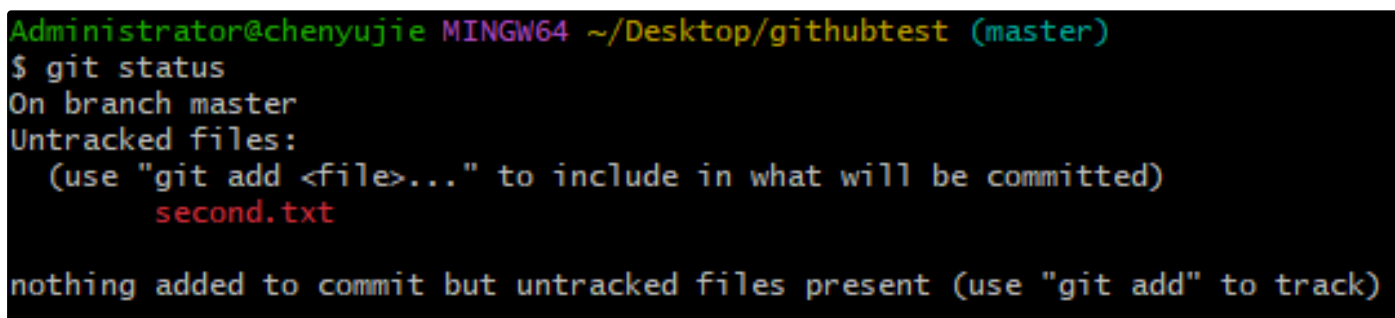
```
$ git config --global user.email "200921743@qq.com"
```

```
$ git config --global user.name "LoveClears"
```

## git 基础操作学习

在 git 仓库文件夹内 新建了文件以后 都会提示错误

untracked files 未被追踪的文件



需要将新建的文件添加到暂存区

```
git add second.txt
```

```
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        second.txt

nothing added to commit but untracked files present (use "git add" to
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git add second.txt
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   second.txt
```

git commit 将暂存文件进入到提交区以后 才会真正意义上被 git 管理

```
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git commit -m "second"
[master e2169b1] second
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 second.txt
```

撤销工作区的修改

git checkout --second.txt

对于已经在暂存区的文件进行回退

## 对文件进行版本回退学习

在仓库中新建了一个 test 的文件,并添加了内容后,对文件进行 commit-m 将文件提交到暂存区

```
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git commit -m "第二次修改"
[master (root-commit) 65d45ed] 第二次修改
1 file changed, 1 insertion(+)
create mode 100644 test.txt

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git log
commit 65d45ed6ca3c3ebba1800bab1d957f8fe8b031d6 (HEAD -> master)
Author: LoveClears <200921743@qq.com>
Date:   Mon Aug 8 14:27:42 2022 +0800

    第二次修改
```

通过 log 发现 文件已成功提交 我将进行第二次提交后发现 无法提交

后面通过查阅资料发现,修改过的文本,需用用 ADD 重新添加一次

```
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git commit -m "第三次修改"
[master f8225f7] 第三次修改
1 file changed, 5 insertions(+), 1 deletion(-)

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ ls
test.txt

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git log
commit f8225f7d8f1f55c75bed82fda69926ae3d684404 (HEAD -> master)
Author: LoveClears <200921743@qq.com>
Date:   Mon Aug 8 14:32:08 2022 +0800

    第三次修改

commit 65d45ed6ca3c3ebba1800bab1d957f8fe8b031d6 (fenzhi)
Author: LoveClears <200921743@qq.com>
Date:   Mon Aug 8 14:27:42 2022 +0800

    第二次修改
```

日志里面已有查询记录 在 git reset 回退汇总选择对应的版本 然后查看文档发现回退成功

每次 commit 的时候 会生成一个 commitid

git reset --hard "提交对应 ID" 就可以进行一个版本回退的话可以随时回退或者进行版本切换 ID 可以在 git log 日志中查看

```
commit 02fa0423e26dda4db95116e11b392064ccb7e732 (HEAD -> master)
Author: LoveClears <200921743@qq.com>
Date: Tue Aug 9 11:15:55 2022 +0800

    second file there

commit e2169b1dcfc496b305401ea5fe4e6a85833e9eb3
Author: LoveClears <200921743@qq.com>
Date: Tue Aug 9 10:48:47 2022 +0800

    second

commit 65d45ed6ca3c3ebba1800bab1d957f8fe8b031d6 (fenzhi)
Author: LoveClears <200921743@qq.com>
Date: Mon Aug 8 14:27:42 2022 +0800

    第二次修改

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git reset --hard e2169b1dcfc496b305401ea5fe4e6a85833e9eb3
HEAD is now at e2169b1 second

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
```

如果因为意外关闭 git 窗口,没有最新版本的 commitid 的时候可以通过 git reflog 进行所有的 commitid 查询

```
$ git reflog
02fa042 (HEAD -> master) HEAD@{0}: reset: moving to 02fa0423e26dda4db95116e11b392064ccb7e732
e2169b1 HEAD@{1}: reset: moving to e2169b1dcfc496b305401ea5fe4e6a85833e9eb3
e2169b1 HEAD@{2}: reset: moving to e2169b1dcfc496b305401ea5fe4e6a85833e9eb3
02fa042 (HEAD -> master) HEAD@{3}: commit: second file there
e2169b1 HEAD@{4}: reset: moving to HEAD
e2169b1 HEAD@{5}: reset: moving to HEAD
e2169b1 HEAD@{6}: commit: second
65d45ed (fenzhi) HEAD@{7}: reset: moving to 65d45ed
f8225f7 HEAD@{8}: commit: 第三次修改
65d45ed (fenzhi) HEAD@{9}: checkout: moving from fenzhi to master
65d45ed (fenzhi) HEAD@{10}: checkout: moving from master to fenzhi
65d45ed (fenzhi) HEAD@{11}: commit (initial): 第二次修改

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$
```

## 分支管理学习

分支就是在 git 下进行多人合作项目的一个方式

在不同的分支上每次的操作都是新的

创建一个 panda 的分支 并切换到分支 panda

```
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git checkout -b panda
Switched to a new branch 'panda'

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (panda)
$ git branch
  fenzhi
  master
* panda
```

git branch 可以查看当前分支情况,并且可以看到目前所处的分支



The image shows a terminal window and a Notepad window. The terminal window shows the user switching to the 'panda' branch and listing files. The Notepad window shows the content of 'test.txt' in the 'panda' branch.

```
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$ ls
second.txt  test.txt
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$ git checkout master
Already on 'master'
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$ git checkout panda
Switched to branch 'panda'
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$
```

The Notepad window shows the content of 'test.txt' in the 'panda' branch:

```
first test.txt
萨达多撒的
今天天气针不戳 针不戳啊
```

在 panda 分支下的 test.txt 内容



The image shows a terminal window and a Notepad window. The terminal window shows the user switching to the 'master' branch and listing files. The Notepad window shows the content of 'test.txt' in the 'master' branch.

```
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$ ls
second.txt  test.txt
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$ git checkout master
Already on 'master'
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$ git checkout panda
Switched to branch 'panda'
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$ git checkout master
Switched to branch 'master'
Administrator@chenyujie MINGW64 ~/Desktop/githubtest
$
```

The Notepad window shows the content of 'test.txt' in the 'master' branch:

```
first test.txt
```

这是在 master 主分支下面的 test.txt 文件

说明在不同的分支上进行版本管理,只有在自己的分支上才会生效

## 分支合并和冲突解决

默认 master 是主分支 一般用于版本 fault

每个人在各自的分支开发完成之后,需要将内容合并到主分支上去

先新增一个开发人员 panda

git checkout -b panda

进行你修改并且 add commit

此时 panda 分支中多了个 commitid,现在需要把 panda 内容进行发布 就要进行内容合并到 master 主分支上

快速合并 在 master 上进行

git merge panda \

```
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git merge panda \
>
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
Automatic merge failed; fix conflicts and then commit the result.

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master|MERGING)
$ git merge panda
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master|MERGING)
```

这边提示代码有错误: 无法合并, 因为您有未合并的文件。

然后通过 git status 发现有没提交的文件 先进行 add 然后 commit 放到提交区

之后查看日志 发现已经合并成功

```
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master|MERGING)
$ git commit -m "合并"
[master e5a5a7d] 合并

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git merge panda \
>
Already up to date.

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git merge panda \
> git status
merge: git - not something we can merge

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git log
commit e5a5a7d268769fdbfaec39481adbbfdca9ffd8f (HEAD -> master)
Merge: 18ac0c2 7b6b313
Author: LoveClears <200921743@qq.com>
Date: Tue Aug 9 16:45:39 2022 +0800

合并
```



```
test.txt - 记事本
文件 编辑 查看

first test.txt

<<<<<< HEAD
小陈写了个文件
=====
萨达多撒的

今天天气针不戳 针不戳啊

小陈也写了一个文件
>>>>>> panda
```

模拟多方合并

需求 先创建 2 个分支

```
git checkout -b panda
```

```
git checkout -b pig
```

分别用 panda 和 pig 分支 输入个内容

然后将他们俩内容进行合并到 master

```
test.txt - 记事本
文件 编辑 查看
MINGW64:/c/Users/Administrator/Desktop/githubtest
git: 'log--oneline' is not a git command. See 'git --help'.

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git log --oneline
3c6c8fd (HEAD -> master) panda合并
57f8892 (panda) panda 11:04
e5a5a7d 合并
7b6b313 小陈第二次提交
18ac0c2 小陈写了个东西
f45c9da 小陈写了一个东西
68eaa61 分支测试
02fa042 second file there
e2169b1 second
65d45ed 第二次修改

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git merge pig
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
Automatic merge failed; fix conflicts and then commit the result.

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master|MERGING)
$ git log --pretty=oneline
3c6c8fdc6d78db52acb9a309101c98f54174b215 (HEAD -> master) panda合并
panda11:04 2022/8/10
<<<<<< HEAD
>>>>>> panda
=====

pig 11:17 2022/8/10
>>>>>> pig
```

遇到合并冲突的问题 需要手动解决,并且进行 add.commit 的操作

```

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master|MERGING)
$ git add test.txt

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master|MERGING)
$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:   test.txt

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master|MERGING)
$ git commit -m "panda合并"
[master 3c6c8fd] panda合并

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git status
On branch master
nothing to commit, working tree clean

```

## 操作远程和本地仓库

了解 push, pull 和 clone

让本地仓库和远程仓库有关联

本地仓库中 git remote add origin git@github.com:LoveClears/share.git 进行远程链接

```

MINGW64/c/Users/Administrator/Desktop/githubtest

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git remote add origin git@github.com:LoveClears/share.git
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$

```

git remote 发现返回 origin (origin 代表远程仓库和本地仓库地址进行一个关联)

```

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git remote
origin

```

git push -u origin master 代表将本地仓库的内容推送到远程仓库

```

Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git push -u origin master
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.



Please make sure you have the correct access rights
and the repository exists.

```

发现错误 经过查询是公钥出现问题 重新进行生成秘钥 将秘钥复制到 ssh 里面 重新进行文件推送到远程仓库上




This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

 SSH	<b>测试666</b> SHA256:w2G+C41TPYU/rf3PSmjHKBeeghAXC42MtAVOXAiZ4hk Added on 4 Aug 2022 Never used — Read/write	Delete
 SSH	<b>我就要用中文</b> SHA256:Wre8MLLuSAiDKQqruYemhroFr8PLEk4z6G7F8JWIVzU Added on 10 Aug 2022 Never used — Read/write	Delete

push 时候没有报错,

```
Administrator@chenyujie MINGW64 ~/Desktop/githubtest (master)
$ git push -u origin master
Enumerating objects: 36, done.
Counting objects: 100% (36/36), done.
Delta compression using up to 8 threads
Compressing objects: 100% (31/31), done.
Writing objects: 100% (36/36), 3.29 KiB | 1.10 MiB/s, done.
Total 36 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/LoveClears/share/pull/new/master
remote:
To github.com:LoveClears/share.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

 LoveClears txt文件上传	3dce13a 6 days ago	 4 commits
 first.txt	txt文件上传	6 days ago
 linux7月学习实践.docx	Add files via upload	6 days ago
 群晖备份方案调研.docx	群晖备份方案	6 days ago



### Your main branch isn't protected

Protect this branch from force pushing, deletion, or require status checks before merging.

[Protect this branch](#)

LoveClears Merge branch 'main' of github.com:LoveClears/share

a575156 1 minute ago ⌚ 17 commits



first.txt

txt文件上传

6 days ago



linux7月学习实践.docx

Add files via upload

6 days ago



second.txt

second file there

yesterday



test.txt

pig合并

3 hours ago



群晖备份方案调研.docx

群晖备份方案

6 days ago

上传前后对比

## 将远程仓库代码克隆到本地

```
$ git clone git@github.com:LoveClears/share.git
```

> xiaochen > share

名称



.git



first.txt



linux7月学习实践.docx



second.txt



test.txt



群晖备份方案调研.docx



MINGW64:/c/Users/Administrator/Desktop/xiaochen

```
Administrator@chenyujie MINGW64 ~/Desktop/xiaochen
$ git clone git@github.com:LoveClears/share.git
Cloning into 'share'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 49 (delta 11), reused 42 (delta 8), pack-reused 0
Receiving objects: 100% (49/49), 2.70 MiB | 4.63 MiB/s, done.
Resolving deltas: 100% (11/11), done.
```

```
Administrator@chenyujie MINGW64 ~/Desktop/xiaochen
$
```

也能同步推送到 github 内 方便多人协作

但是当远程仓库版本内容和本地内容不一致,需要先 git pull 把远程仓库内容更新到最新版本 然后手动进行解决

1 first test.txt

2

3 &lt;&lt;&lt;&lt;&lt;&lt; HEAD

4 小陈写了个文件

5 =====

6 萨达多撒的

7

8

9

10 今天天气针不戳 针不戳啊

11 \$ git commit -m "push"

12 小陈也写了一个文件

13 &gt;&gt;&gt;&gt;&gt;&gt; panda

14 &lt;&lt;&lt;&lt;&lt;&lt; HEAD

15 =====

17 panda11:04 2022/8/10

18 &lt;&lt;&lt;&lt;&lt;&lt; HEAD

19 &gt;&gt;&gt;&gt;&gt;&gt; panda

20 =====

22 pig 11:17 2022/8/10

23 &gt;&gt;&gt;&gt;&gt;&gt; pig

24

25 xiaochendev

MINGW64:/c/Users/Administrator/Desktop/xiaochen/share

Administrator@chenyujie MINGW64 ~/Desktop/xiaochen/share (main)

\$ git add test.txt

Administrator@chenyujie MINGW64 ~/Desktop/xiaochen/share (main)

\$ git commit -m "push"

[main d9aff5f] push

1 file changed, 2 insertions(+)

Administrator@chenyujie MINGW64 ~/Desktop/xiaochen/share (main)

\$ git push -u origin main

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 8 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 290 bytes | 290.00 KiB/s, done.

Total 3 (delta 2), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (2/2), completed with 2 local objects

To github.com:LoveClears/share.git

a575156..d9aff5f main -&gt; main

branch 'main' set up to track 'origin/main'.

Administrator@chenyujie MINGW64 ~/Desktop/xiaochen/share (main)

\$