git操作整理

git 配置

git config --global user.name "yuluo"

git config --global user.email "1481556636@qq.com"

git config --global color.ui auto　启用命令行着色输出

git 操作

暂存区

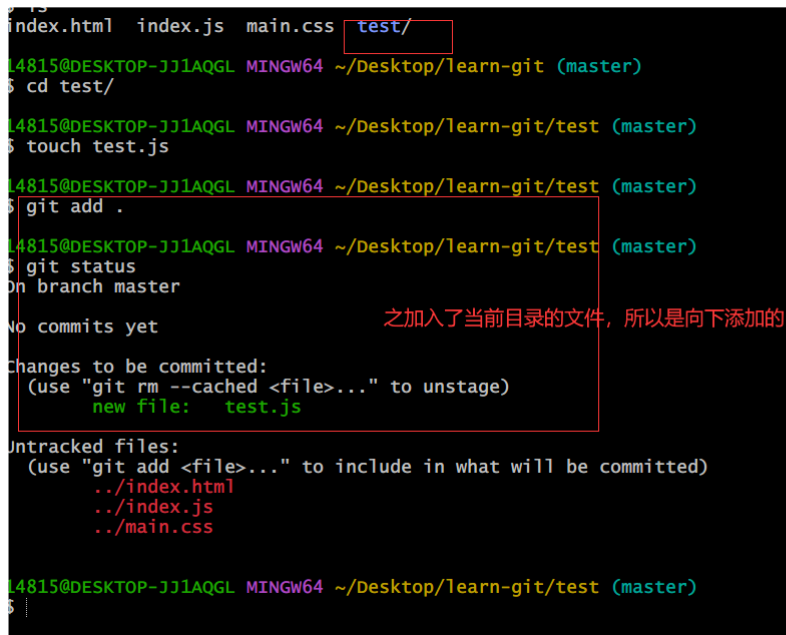git init . 初始化git仓库

git status 看仓库状态

git add index.html　把index.html 文件加入暂存区，**git添加文件默认是从当前目录向下添加的**，

如果在下级目录想加入上级目录的所有内容，使用git add -A

git rm --cached index.html　从暂存区撤回

git add ./* 将当前目录所有文件加入暂存区

git rm -r --cached . 将所有文件从暂存区移除

本地版本库

> 从暂存区获取文件提交到本地版本库

git log 获取此次git commit的hash值,

git show 查看和hash值对应的commit信息

git diff 查看修改后的文件和本地版本库文件的差异

git restore index.js 放弃js的更改

git commit --amend -m "message" 修改已经提交的commit信息

push

git branch -M main 变更主分支为master

```
git remote add origin git@github.com:yuluo-yx/test.git

git branch -M main

git push -u origin main
```
这样做需要配置ssh，我一般使用https协议

branches

git branch -r 查看远程分支，在绑定了远程仓库名之后才可以看到远程分支

git branch -a 查看全部分支，包括本地和远程

git branch feature-a　分支名使用_和-命名，不使用特殊符号，创建一个 feature-a分支

git checkout feature-a 切换到feature-a分支

git checkout - 切换分支

git branch -d "name" 删除分支

git merge feature-a 合并feature-a到主分支

git checkout -b feature-xyz 创建一个feature-xyz分支并切换过去

rebase 解决文件冲突

同时修改了同一个文件，会产生冲突，在拉取时，会自动尝试合并，失败之后文件就会像下面这样，下面的hash为另一个人修改时的commit hash值，head为自己修改的部分，====为分隔符


```
14815@DESKTOP-JJ1AQGL MINGW64 ~/Desktop/learn-git (feature-xyz)
$ git pull https://github.com/yuluo-yx/test.git feature-xyz
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 722 bytes | 80.00 KiB/s, done.
From https://github.com/yuluo-yx/test
 * branch            feature-xyz -> FETCH_HEAD
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

14815@DESKTOP-JJ1AQGL MINGW64 ~/Desktop/learn-git (feature-xyz|MERGING)
$ cat index.html
<!DOCTYPE html>
<html>
        <head>
                <title>Page Title</title>
        </head>
        <body>
<<<<<<< HEAD
                <p>this a git rebase test by local machine!</p>
=======
                <p>this a git rebase test!</p>
>>>>>>> 2ffe34c7ae857c4b72aad41996e6f24a7451561b
        </body>
</html>

14815@DESKTOP-JJ1AQGL MINGW64 ~/Desktop/learn-git (feature-xyz|MERGING)
$
```

此时，git status显示为：同时修改了文件

步骤:

- 后动修改合并后的文件，之后重新commit push
- git add .
- git status
- git commit -m "resolver conflict"
- git push

变基 详细 - rebase

git pull -r/--rebase https://github.com/yuluo-yx/test.git
main

需要手动解决冲突

代码:

下面是初次提交时的更改，上面是更改之后的内容。

我们可以借助vscode或者idea等工具，或者手动去选择接受它

```
14815@DESKTOP-JJ1AQGL MINGW64 ~/Desktop/learn-git (feature-xyz|REBASE 2/4)
$ cat index.html
<<<<<<< HEAD
<!DOCTYPE html>
<html>
        <head>
                <title>Page Title</title>
        </head>
        <body>
                <p>first pragraph</p>
        </body>
</html>
=======
<!DOCTYPE html>
<html>
        <head>
                <title>Page Title</title>
        </head>
        <body>

        </body>
</html>
>>>>>>> 4d026ce (add some html)

14815@DESKTOP-JJ1AQGL MINGW64 ~/Desktop/learn-git (feature-xyz|REBASE 2/4)
$
```

解决冲突之后:

继续运行:

git add *

git rebase --continue

执行之后,仍然不能成功,因为还有一个冲突没有解决!

```
14815@DESKTOP-JJ1AQGL MINGW64 ~/Desktop/learn-git (feature-xyz|REBASE 2/4)
$ git rebase --continue
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
error: could not apply 97895d7... added p1 another tag
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 97895d7... added p1 another tag

14815@DESKTOP-JJ1AQGL MINGW64 ~/Desktop/learn-git (feature-xyz|REBASE 3/4)
$
```

同样的方式解决之后,

```
14815@DESKTOP-JJ1AQGL MINGW64 ~/Desktop/learn-git (feature-xyz|REBASE 3/4)
$ cat index.html
<<<<<<< HEAD
<!DOCTYPE html>
<html>
        <head>
                <title>Page Title</title>
        </head>
        <body>
                <p>first pragraph</p>
        </body>
</html>
=======
<!DOCTYPE html>
<html>
        <head>
                <title>Page Title</title>
        </head>
        <body>
                <p>this a git rebase test by local machine!</p>
        </body>
</html>
>>>>>>> 97895d7 (added p1 another tag)
```

继续执行：`git rebase --continue`

解决完所有冲突之后，git push


```
14815@DESKTOP-JJ1AQGL MINGW64 ~/Desktop/learn-git (feature-xyz|REBASE 4/4)
$ git rebase --continue
[detached HEAD 868d4b7] Update index.html
 Author: YuLuo <yuluo08290126@gmail.com>
 1 file changed, 7 insertions(+), 4 deletions(-)
Successfully rebased and updated refs/heads/feature-xyz.
```



```
14815@DESKTOP-JJ1AQGL MINGW64 ~/Desktop/learn-git (feature-xyz)
$ git push https://github.com/yuluo-yx/test.git feature-xyz
To https://github.com/yuluo-yx/test.git
 ! [rejected]        feature-xyz -> feature-xyz (fetch first)
error: failed to push some refs to 'https://github.com/yuluo-yx/test.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

这里我们修改了代码，解决了冲突，重新设置了基础。但是push时，却落后了提交，我们自己清楚，但是远程仓库不知道我们改变了基础。

所以选择-f 强制提交