Git Study Notes

Course: 极客时间 苏玲 Git入门

# Config User Information

## Config user.name and user.email

$ git config --global user.name ‘your\_name’

$ git config --global user.email ‘your\_email@domain.com’

## Config Scope

$ git config --local : only applicable for the current repository

$ git config --global : applicable for all repositories for the current user

$ git config --system : applicable for all users available in the system

Default value is ‘local’.

Use ‘--list’ to show config setup.

$ git config --list --local

$ git config --list --global

$ git config --list –system

# Create Git Repository

## Use Git to manage an existing project folder

$ cd [Project Folder Path]

$ git init

## Use Git to create a new project folder

$ cd [some folder path]

$ git init your\_project # will create a folder with same name under the current path

$ cd your\_project

# Add files to repository

## Scenarios – 4 commits to build an html page.

Step 1: Add index.html and git-logo

Step 2: Add style.css

Step 3: Add script.js

Step4: Modify index.html and style.css

Git commit

版本历史

暂存区

Git add files

工作目录

$ cp [from file path] [destination file path]

$ cp ../git-sample/bingindex.html . (copy bingindex.html file from ‘git-sample’ folder to current folder)

$ cp -r ../git-sample/images . (copy ‘images’ folder to current folder)

$ git add [file name] or [folder name] (add file to stage)

$ git add bingindex.html images

$ ls (list information for the file)

$ ls -al

$ mkdir [OPTION] [DIRECTORY] (create the directories if they do not already exist.)

$ mkdir styles (create a folder named ‘styles’)

$ git add -u (directly add the updated files which are already under control of Git)

# Raname a file

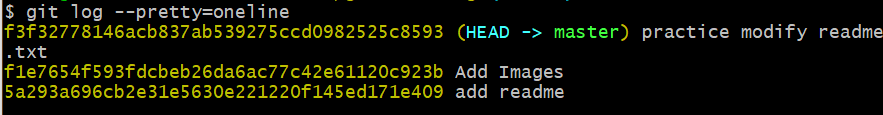
$ git mv [original file name] [new file name]

* Not need to use git add after git mv. You can commit this change directly.

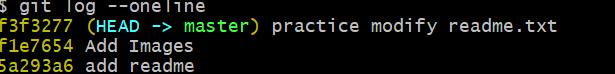
# View Log

$ git log (view all detail logs for the current branch)

$ git --pretty=oneline



$ git log --oneline (view log in one line)



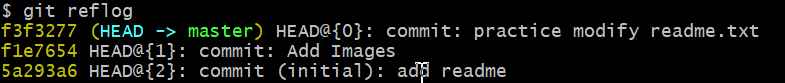
$ git log -n4 (view log for the latest 2 commits)

$ git log -n4 --oneline

$ git log --all (view all detail logs for all the branches)

$ git log --all --graph (view all detail logs in graph view)

$ git reflog



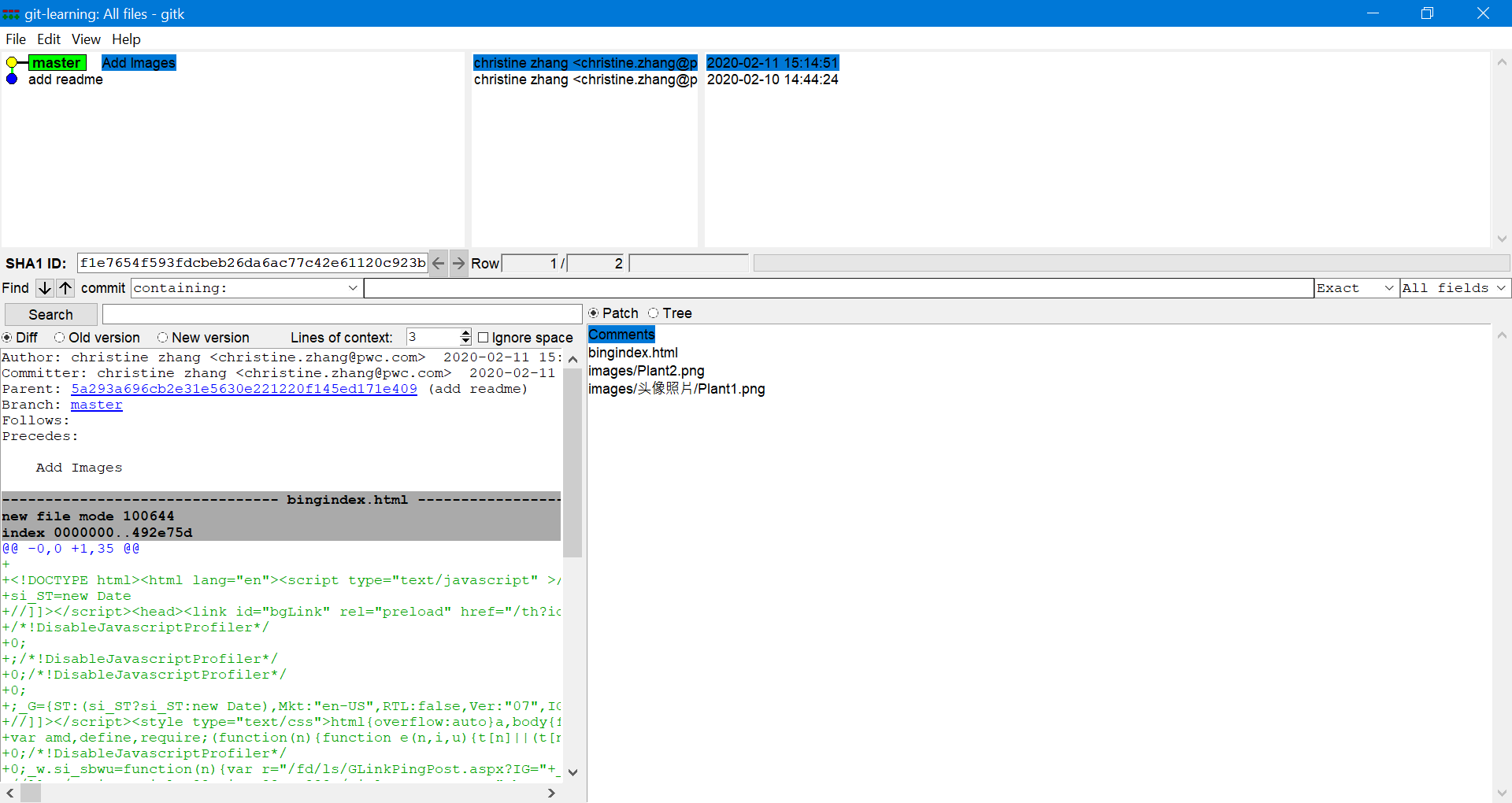
翻页命令：space 向下翻页

b 向上翻页

q 退出

# View log in GUI

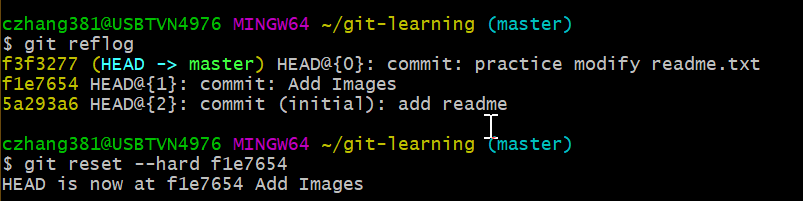
$ gitk (to open GUI tool)



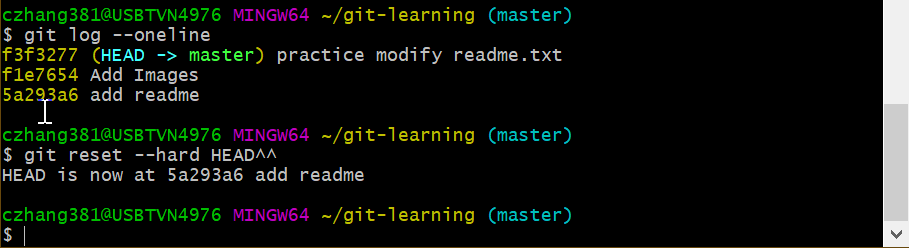
# 版本的前进后退

## 基于索引值

$ git reset --hard [index key]



使用^符号： 只能后退



使用~符号：只能后退

$ git reset --hard HEAD^^ = $ git reset --hard HEAD~2

## Reset命令的三个参数对比

--soft 参数：仅在本地库移动HEAD

--mixed 参数：在本地库、暂存区移动HEAD

--hard 参数：在本地库、暂存区、工作区移动HEAD

# Compare Files

$ git diff [File Name]: compare file in working area with staging area

$ git diff [Version number][File Name]: compare file in working area with local repository of version#

# Branch Management

## Create Branch

$ git branch [Branch Name]

## View Branch

$ git branch -v

## Switch Branch

$ git checkout [Branch Name]

## Merge Branch

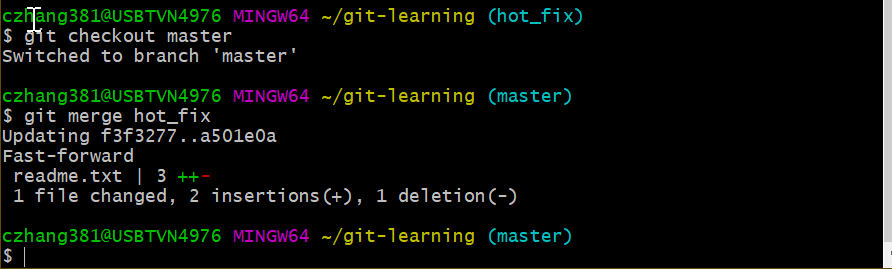
Step1: switch to the destination branch (the branch which is merged to)

$ git checkout [Branch Name1]

Step2: merge the branch (the branch which is merged from)

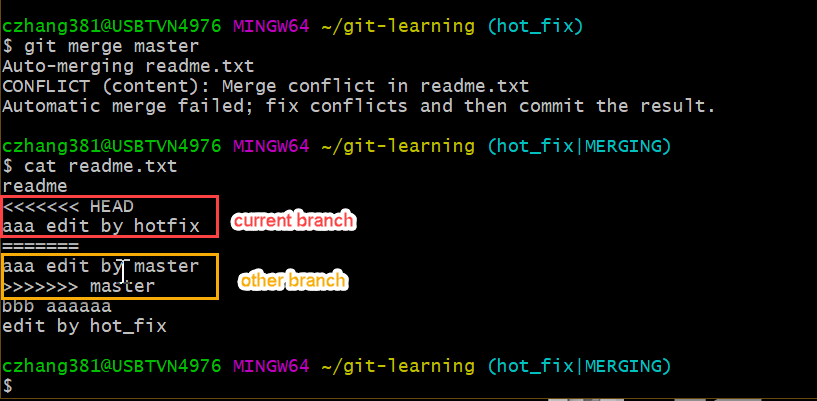
$ git merge [Branch Name2]

* Merge Branch Name2 to Branch Name1



## Resolve Conflict

### How does the conflicted file looks like



### How to resolve conflicts



# Integrate Local Repository with Remote Repository

## Step 1: setup remote repository

$ git remote add original [remote git address]

## Pull from Remote Repository

* Pull = fetch + merge
* Git fetch [Remote Repository Alias] [Remote Repository Branch Name]
* Git merge [Remote Repository Alias/ Remote Repository Branch Name]
* Git pull [Remote Repository Alias] [Remote Repository Branch Name]