## Git的安装

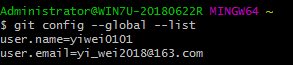
## Git的最小设置

配置user.name 和 user.email

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

$git config --global user.email [‘your\_email@domain.com’](mailto:‘your_email@domain.com’);

**查看当前配置:**



**Config 的三个作用域**

缺省等同于local

$git config --local （local只对某个仓库有效）

$git config --global （global对当前用户所有仓库有效）

$git config --system （system对系统所有登陆的用户有效）

显示config的配置,加 --list

$git config --list --local

$git config --list --global

$git config --list --system

## 建Git仓库

**两种场景:**

1. 把已有的项目代码纳入Git管理

$cd 项目代码所在的文件夹

$git init

1. 新建的项目直接用Git管理

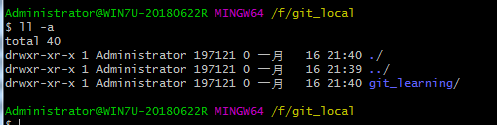
$cd 某个文件夹

$git init your\_project #会在当前路径下创建和项目名称同名的文件夹

$cd your\_project

进入cd /f/git\_loca

$git init git\_learning

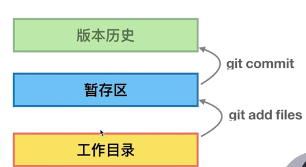


## 通过几次commit来认识工作区和暂存区

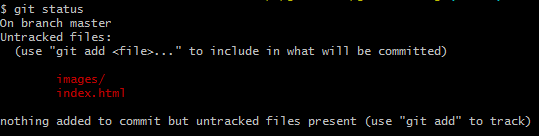
**往仓库里添加文件:**

4次提交,一个像模像样的静态页面生成

1. 加入index.html 和git-logo
2. 加入style.css
3. 加入script.js
4. 修改index.html 和 style.css



查看暂存区

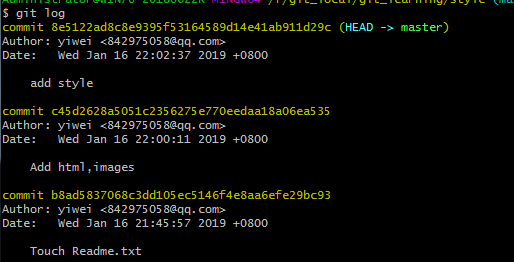


$git add index.html images/ #加入到暂存区

$git commit -m’add index images’ #提交暂存区的内容到版本历史

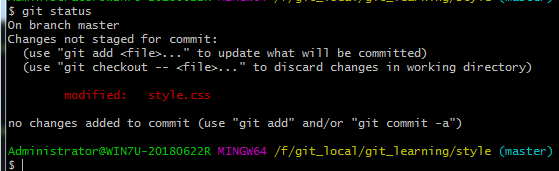
查看当前log

$git log

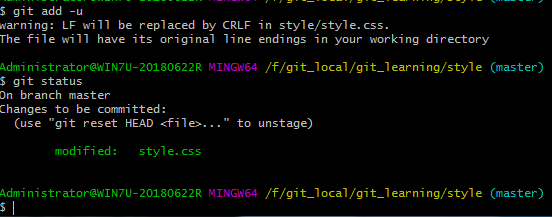


当我修改了文件内容：

$git status #修改信息将被显示

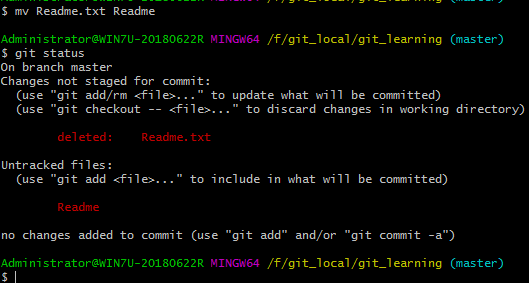


$git add -u #把当前所有修改的文件都加入到暂存区

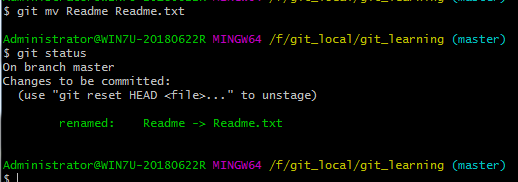


## 给文件重命名简便方法

当我修改一个文件名的时候,git将会做两步操作

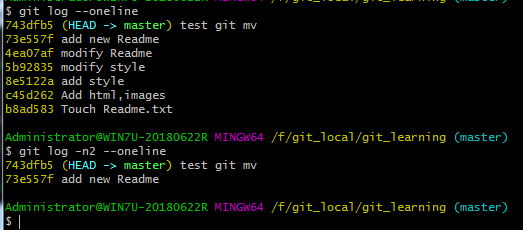


我们可以直接用git mv Readme Readme.txt #git 将会直接在暂存区中帮我们修改



## 通过git log查看版本演变历史

$git log --oneline



创建新的分支:

$git checkout -b temp 8e5122a



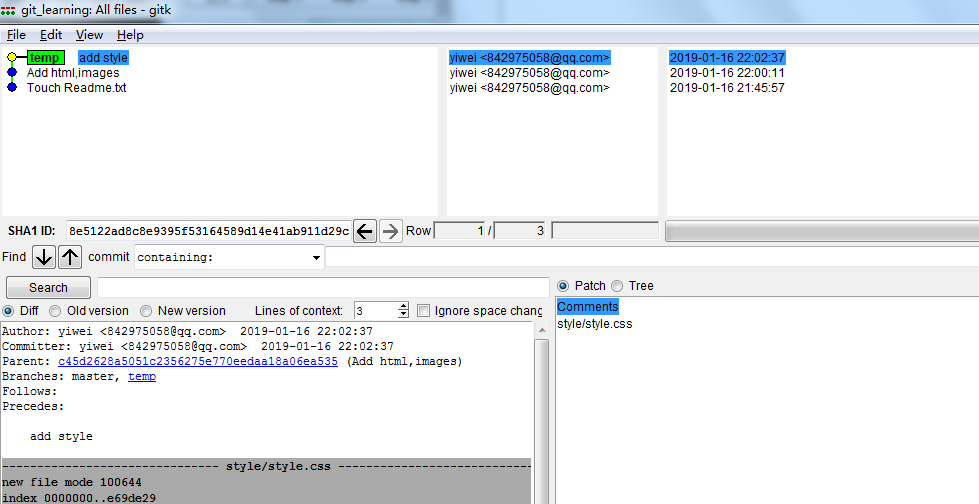
$git log #只能查看当前的分支

$git log -all 查看所有情况

$git log --oneline --all --graph

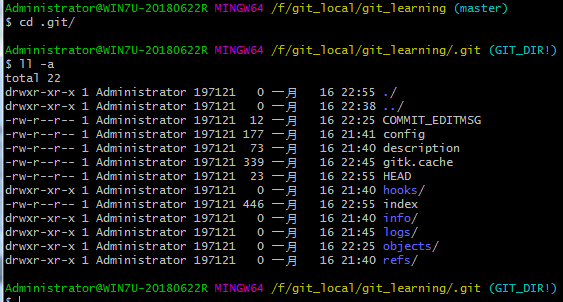
## Gitk:通过图形界面工具来查看版本历史

$gitk



## 探秘.git目录

**.git相关目录:**



**HEAD目录内容:**



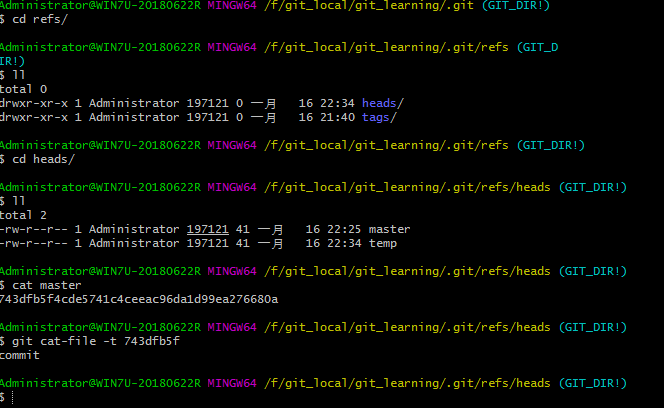
**Config 文件内容** #可以修改这个文件中的内容将影响相关内容



**Refs文件夹里面的内容:**

**heads #代表分支**

**tags #代表开发里程碑**



$git cat-file -t xxxx #查看类型

$git cat-file -p xxxx#查看内容

**Objects**/ #里面存放的是我们要用的内容

