<https://blog.csdn.net/ZteenMozart/article/details/80459462?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.channel_param&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.channel_param>

<https://blog.csdn.net/youzhouliu/article/details/78952453>

## Windows git安装及使用

安装git

## linux git安装及使用

<https://www.cnblogs.com/liter7/p/6581344.html>

## git 基本使用命令

### 查看版本

git – version

### 查看当前登录用户

Git config --global user.name

### 仓库初始化

Git init

### 创建文件夹

Mkdir test

Cd test

Vi test.txt

### 创建忽略文件

touch .gitignore

vi .gitignore//添加上传是不上传的文件

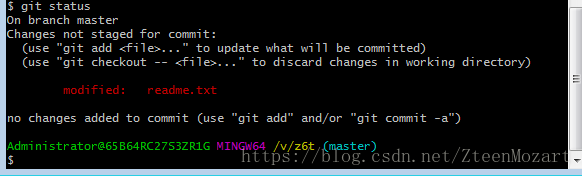
### 添加到缓存区

Git add test.txt

Git add ./test/test.txt//添加指定文件

Git add –A 添加全部

Git status 查看结果



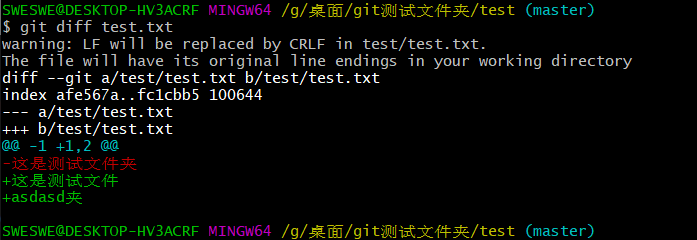
红色表示修改该文件还未上传到暂缓区

绿色表示已经已经上传到暂缓区

### 查看文件修改内容差异

Git diff test.txt

红色为之前内容，绿色为修改后内容



### 删除缓存区中文件

Git rm –-f “test.txt”

不仅将该文件从缓存中删除，还会将物理文件删除

git rm –cached “文件路径” 不删除物理文件，仅将该文件从缓存中删除

### 添加到版本库中

Git commit –m ‘备注信息’

### 比较差异

* 比较的是暂存区和工作区的差异

git diff

* 比较的是暂存区和历史区的差异

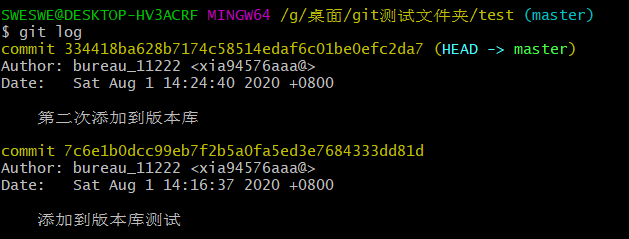
git diff --cached

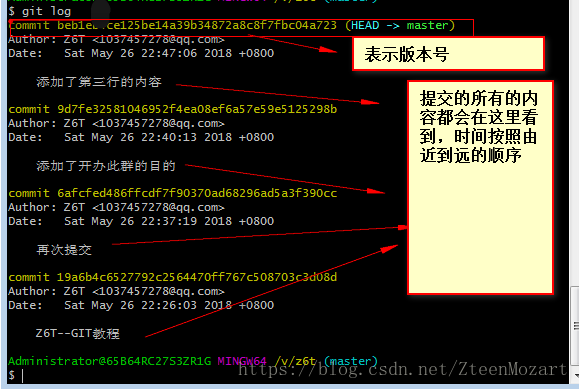
* 比较的是历史区和工作区的差异（修改）

git diff master

## 版本回退

Git log 查看提交到版本库的记录





Git reset –-hard HEAD^ 回退上个版本

Git reset –-hard HEAD^^回退上上个版本

Git reset – -hard HEAD^100 回退上100版本

Git reset – hard HEAD 版本号

Git reflog 回滚到未来

## 工作区与版本库

**工作区：**就是你在电脑上看到的目录，比如目录下testgit里的文件(.git隐藏目录版本库除外)。或者以后需要再新建的目录文件等等都属于工作区范畴。

**版本库(Repository)：**工作区有一个隐藏目录.git,这个不属于工作区，这是版本库。其中版本库里面存了很多东西，其中最重要的就是stage(暂存区)，还有Git为我们自动创建了第一个分支master,以及指向master的一个指针HEAD。

## 放弃工作区的修改

Git checkout -- test.txt 注意文件名前边必须有空格

## 添加远程仓库

Git remote add orgin 仓库地址

## 克隆远程仓库

Git clone 仓库地址

unset GIT\_DIR

cd /home/www/test/demo

git pull

## 创建合并分支

### 创建分支

Git branch test

### 查看当前分支

Git branch

### 切换分支

Git checkout test

### 切换并创建分支

Git checkout –b test

### 提交到分支

git push origin master

### 删除分支

Git branch –d test

### 在分支上提交新的版本

未完待续。。。。。

### 融合分支

Git merge test

## Git合并add commit push命令

git config --global alias.cmp '!f() { git add -A && git commit -m "$@" && git push; }; f'