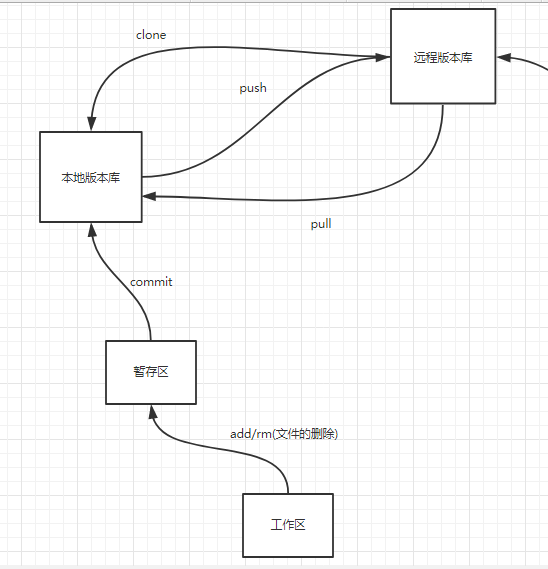
# 版本控制

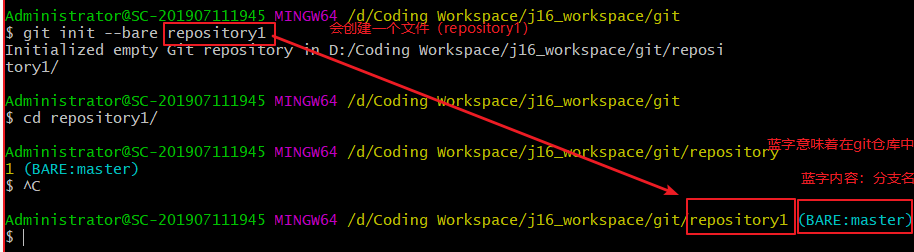
# Git



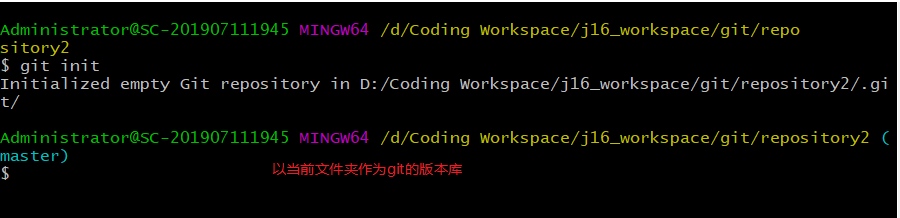
# Git命令

## Git init(初始化仓库)

### git init –bare 仓库名



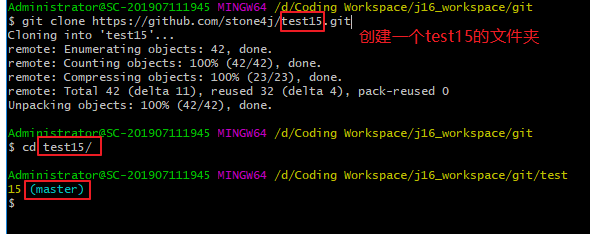
### git init



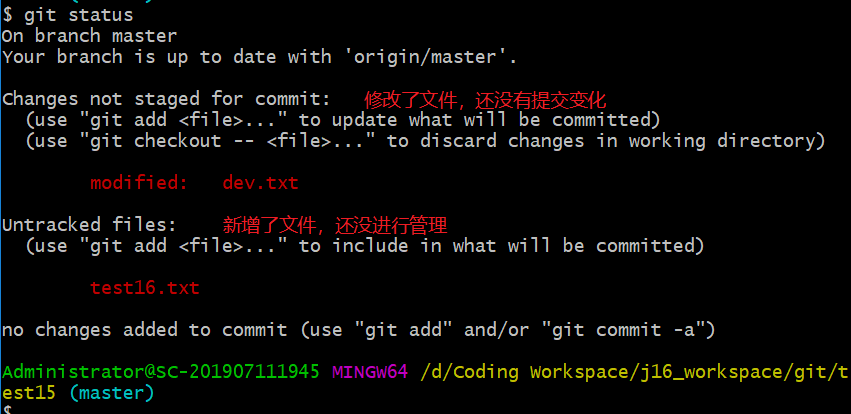
## git clone 远程仓库地址

从远程仓库拉取一个仓库下来作为本地仓库

是最常用获取远程仓库的方式



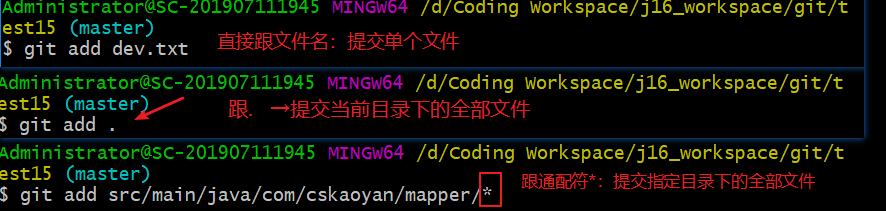
## git status



## git add 文件名

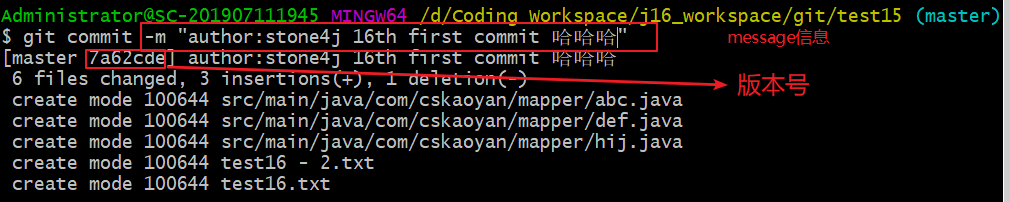
git add . 提交当前文件夹下的全部文件

git add src/\*



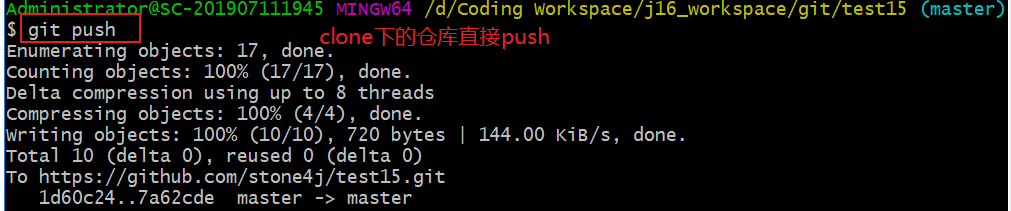
可以通过git status查看代码提交到暂存区的状态

## git commit –m “消息”



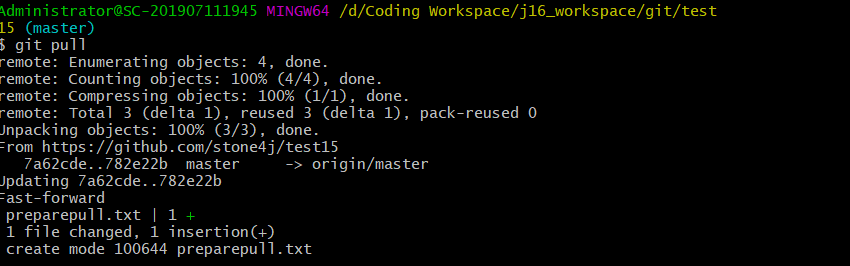
## git push

将本地仓库的修改提交到远程仓库



## git pull

从远程仓库拉取最新的更新



## git Config

需要配置邮箱和用户名

在第一次commit之前

1. 通过命令修改 git config –global user.name(email) 值
2. 修改user目录下的.gitconfig文件

## git log

提交日志

包含版本号 提交用户信息 和提交的message

## git push的时候

需要github的登陆信息



## git reset（谨慎操作）

git reset –hard 版本号（相较于下面两个更常用）

git reset –hard HEAD^ 回退一个版本

git reset –hard HEAD~n 回退n个版本

## 删除

文件的删除git rm

修改的撤销：撤销的是工作区还未提交到暂存区的部分git checkout 文件名（谨慎操作）

## 冲突解决

多个用户维护同一部分的代码时发生冲突。

发生冲突时，首先拉取冲突下来git pull

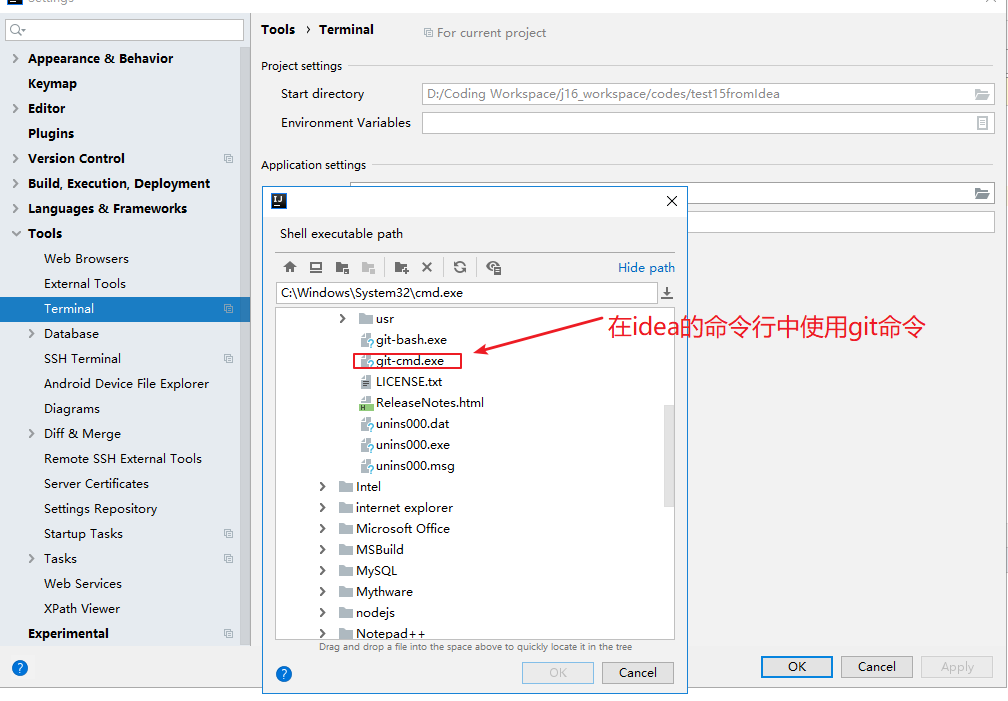
手动解决冲突（修改代码）

重新执行代码提交流程（add → commit → push）

# 文件忽略的配置



# Idea中配置git的terminal



# 分支

## 查看分支

Git branch -a

## 切换分支

Git checkout 分支名

## 创建分支

Git checkout 新的分支名

Git checkout –b 分支名

## 分支的合并

比如说我们现在做的是dev要合并到master上

1. 切换到master分支上
2. 在master上合并dev，git merge dev
3. （解决冲突）提交git push

# 补充

## 快捷键

Shift+insert：粘贴

Ctrl+insert：复制

## Linux命令

Cd 前往目录

touch 创建文件

mkdir 创建文件夹

vim