# Git分布式版本控制系统

Git初始化：git init

查看当前状态：git status

添加到暂存区：git add 文件名 //可以添加一个或多个

语法：Git add 文件名

Git add 文件名1 文件名2 文件名3

Git add . //添加全部文件

提交至版本库：git commit -m “注释内容”

版本回退：

1. 查看版本,确定需要回到的时刻点

语法：git log

git log --pretty=oneline

1. 回退操作

语法：git reset --hard 提交编号

回到过去后，要想再回到之前的最新版本的时候，则需要使用指令去查看历史操作得到最新的commit id

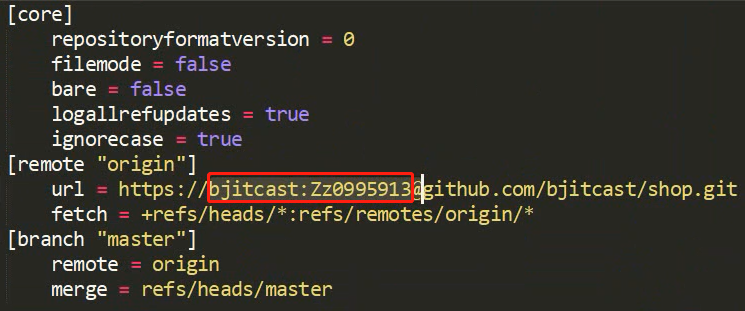
查看历史操作：git reflog

克隆仓库：git clone

提交到线上仓库的指令：git push

首次提交可能出现403错误 解决方法:

打开.git文件内的config文件把账号和密码添加到该地方



拉取线上仓库：git pull

分支管理

Git对文件大小写不区分，需要添加此命令：git config core.ignorecase false

相关指令：

查看分支：git branch

创建分支：git branch 分支名 git checkout -b 分支名 创建并切换分支

切换分支：git checkout 分支名

删除分支：git branch -d 分支名

合并分支：git merge 被合并的分支名

忽略文件

忽略文件需要新建一个名为.gitignore的文件，该文件用于声明忽略文件或不忽略文件的规则，规则当前目录及其子目录生效

注意：改文件因为没有文件名，没办法直接在Windows目录下直接创建，可以通过命令行Git Bash 来 touch创建

/mtk/ 过滤整个文件名

\*.zip 过滤所有的.zip文件

/mtk/do.c 过滤某个具体文件

!index.php 不过滤具体某个文件 (感叹号)

在文件中以#开头的都为注释

企业分支管理

进入master分支

创建分支：git branch one

切换分支：git checkout one

------------开始各自项目代码编写---------

------------代码编写完毕---------

在自己分支进行上传代码：

git add .

git commit -m “one分支代码”

git push –set-upstream origin one //设置本地分支追踪远程分支并提交代码到one分支

git checkout master

git merge origin/one //将one分支代码合并到maste主分支上

git push

