**Git简介**

Git: 分布式版本控制系统，可自动记录每次文件的改动，还可以协作编辑

GitHub网站：为开源项目免费提供Git存储

远程代码仓库

# 本地代码仓库

集中式版本控制系统：版本库是集中存放在中央服务器，需取出放回

分布式版本控制系统：每个人的电脑上都是一个完整的版本库，互相推送实现协作。通常也有一台充当“中央服务器”的电脑

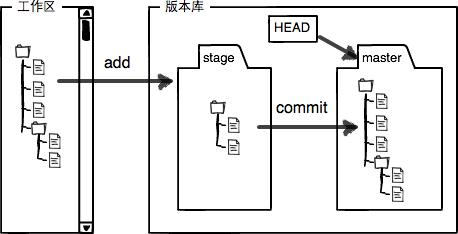
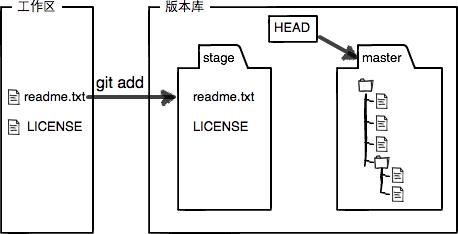
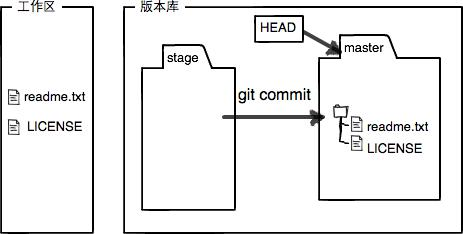
分支：同时处理不同版本的存储库的方法

Git无法知道二进制文件（图片、视频、word等）的具体改动

工作区 电脑里能看到的目录

版本库 工作区有一个隐藏目录.git

stage（或者叫index） 暂存区 (需要提交的文件修改通通放到暂存区，然后，一次性提交暂存区的所有修改)

**命令：**

Is / dir 查看当前目录下文件

pwd 查看当前路径

**# 将文件夹设置为本地仓库**

mkdir ./Desktop/git\_work 创建文件

cd ./Desktop/git\_work 进入文件

git init 设置为仓库

**# 向git中添加文件两步：**

git add readme.txt 把文件修改添加到暂存区

git commit 将文件提交至仓库，快照版本 把暂存区的所有内容提交到当前分支

-m “....”本次提交的说明

git status 时刻掌握仓库当前的状态

git diff head --.txt 知道了对readme.txt作了什么修改

vi readme.txt 进入vi编辑模式，输入“:wq”保存并退出（长按Esc键，同时按两下大写的Z键，即可退出）

git log：显示从最近到最远的提交日志

--pretty=oneline参数简洁版

HEAD 表示当前版本，上一个版本就是HEAD^，上上一个版本就是HEAD^^，当然往上100个版本写100个^比较容易数不过来，所以写成HEAD~100

git reset –hard commit\_id 当前版本回退到某一个版本

cat 查看文件内容

git reflog 记录你的每一次命令

撤销修改：（只能撤回到最新版本）

撤销工作区：git restore .txt

旧版本：git checkout --.txt

撤销暂存区：git restore –staged .txt

旧版本：git reset HEAD .txt

删除文件：

Rm .txt 从工作区删除文件

Git rm .txt 从版本库中删除该文件

并用 git commit -m””提交

添加本地文件夹： git add <name>

删除本地文件夹：git rm -rf <name>

# 远程代码仓库：

GitHub网站 提供Git仓库托管服务，只要注册一个GitHub账号，就可以免费获得Git远程仓库

本地Git仓库和GitHub仓库之间的传输通过SSH加密

生成公匙、私匙

**命令：**

关联本地与远程：git remote add origin git@github.com:dashang233/Gao-shangxin.git

本地仓库分支内容推送至远程仓库 git push -u origin master

-u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令

本地做了提交，通过命令将本地分支最新修改推送至Github: git push origin master

查看远程库信息 git remote -v

解除本地与远程库绑定 git remote rm origin

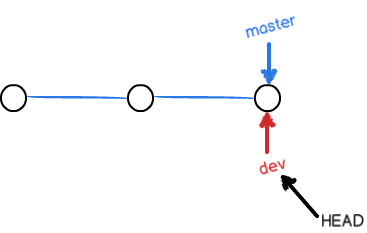
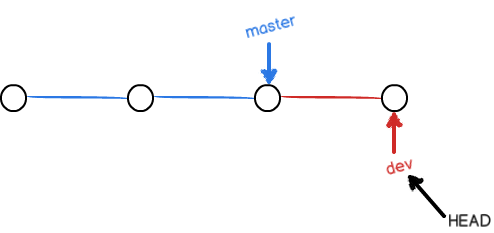
物理删除远程库：登录Github点击删除

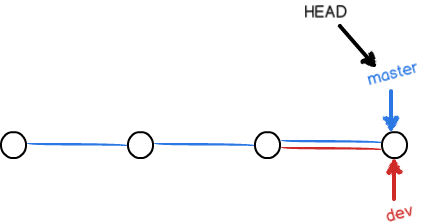
克隆远程库 git clone <http://github.com/dashang233/Gao-shangxin.git>

直接包含三个操作：初始化仓库，连接分支，传送文件

# 分支管理

分支：一条时间线，HEAD指向当前分支，分支指针指向最新提交



创建分支 git branch <name>

切换分支：git switch -c <name> 或合并两步： git switch -c <name>

git checkout <name> 或合并两步： git checkout -b <name>

合并分支： git merge <name1>

解决冲突：手动将合并失败的文件编辑为需要的内容

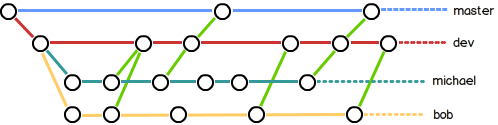
禁用fast forward（防止丢失分支信息）： git merge --no-ff -m”merge with no-ff” dev 产生新的commit故加上-m描述

查看分支合并情况：git log --graph --pretty=oneline --abbrev-commit

删除分支：git branch -d <name>

git branch -D <name> 丢弃一个没有被合并过的分支

分支策略：



**Bug分支**：

Stash 把当前工作现场“储藏”起来，等以后恢复现场后继续工作

Git stash list 查看stash内容

Git stash apply 恢复隐藏的工作环境

Git stash pop 恢复隐藏的工作环境，同时删除stash内容

Git cherry-pick <commit\_id> 复制一个特定的提交到当前分支

**多人协作**

Git remote 查看远程库的信息

git remote -v 显示更详细的信息

推送远程分支 git push origin master

从远程仓库抓取最新提交 git pull

创建本地分支和远程分支的链接git branch --set-upstream-to <branch-name> origin/<branch-name>

git rebase 本地未push的分叉提交历史整理成直线，看上去更直观。缺点是本地的分叉提交已经被修改过了 git rebase master

抓取远程库分支：git fetch orogin master

切换到远程库分支： git checkout origin/master

合并远程分支： git merge origin/master

抓取并合并：git pull origin master

问题：

diff不显示 表示分支两天横杠前后空格作用

先stash再进行debug