

Git 1-15

1 概述

git的特点

- 1 最优的存储能力
- 2 非凡的性能
- 3 开源的
- 4 很容易做备份
- 5 支持离线操作
- 1 很容易定制工作流程

git的衍生的产品

- github 存储，商业伙伴提供CICD
DevOps 支持全流程生命周期的品台也就诞生了
- gitlab 自带 gitlab CI

2 安装git

<https://git-scm.com/book/zh/v2>

3 git 最小配置

配置user信息

- 配置user.name git config --global user.name 'your_name' global 有什么作用?
- 配置user.email git config --global user.email 'your_email@domain.com'

config的三个作用域

- 缺省等同于local
 - git config --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 对系统所有登录的用户有效
- 显示config的配置，加 --list
 - git config --list --local
 - git config --list --global
 - git config --list --system

4 建git仓库并且配置local用户信息

- git init
- git config --local user.name 'xxd'
- git config --local user.email 'hbxyxuxiaodong@163.com'
- local 优先级高于global

5 工作区和暂存区

git add -u

update 表示把被git跟踪的文件一起提交到暂存区

1 给文件重命名的简便方法

- 原来需要三步
 - mv readme readme.md
 - git add readme.md
 - git rm readme
- 把工作目录和暂存区域操作 清除掉
 - git reset --hard
- git 只需要一步
 - git mv readme readme.md

3 通过git log 查看版本演变历史

- git log 查看版本历史
- git log --oneline 以简洁的方式查看版本提交记录
- git log -n4 查看最近的4个提交版本记录
- git log -n2 --oneline 组合方式
- git log --all 查看所有分支的提交记录
- git log --graph 以图形化界面的方式查看
- git log --all --graph 组合方式
- git log --oneline --all -n4 --graph 组合方式
- git log --decorate 显示指向这次提交的所有引用 如分支，标签等
- git log --decorate --oneline --graph --all
- git branch
 - git branch 查看有多少分支
 - git branch -v 查看分支和对应的版本号以及提交记录
 - git branch -av 查看本地分支以及远程分支 对应的版本号和提交记录
- git checkout -b temp 415c5 切换并从指定版本记录上创建新的分支
- git commit -am '添加到暂存区并提交' add & commit

3 gitk 通过图形界面工具来查看版本历史

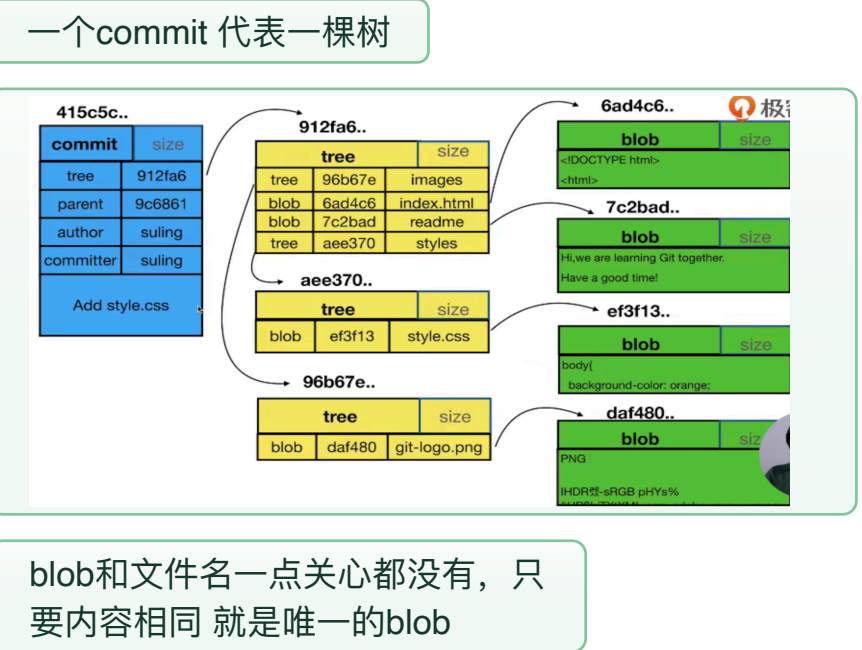
在当前git 仓库 输入gitk 以图形化界面方式查看

new view => All refs

4 探秘.git目录

- cat .git/HEAD
- cat .git/config
- git config --local user.name
- cd .git/refs/heads cat master
- git cat-file -t 59a4b3 tag (查看类型)
- git cat-file -p 59a4b3 object 27d2f (查看内容)
- cat .git/objects 最重要的
- git 核心三个对象 commit tree blob
- git 里面有一个很神奇地方 只要任何文件的文件内容相同 在git的眼里就是唯一的blob

5 commit 、 tree 、 blob



1 数一数tree的个数

- 新建的git仓库，有且仅有一个commit，仅仅包含/doc/readme,请问内含多少个tree，多少个blob?
- find .git/objects -type f 查看.git/objects 下面类型是文件的到底有多少个

2 分离头指针

- git checkout 415c5c 本意就是我们工作在一个没有分支的情况下
- Head 现在没有指向任何分支，而是落在某个commit 上 这个就是分离头指针
- 当我们想做尝试性的变更 可以很好的使用分离头指针
- 分离头指针指的是我们的变更没有基于分支做，所以分支切换的时候 会被git垃圾回收， 如果觉得变更很重要， 切记和分支绑在一起

3 进一步理解head 和branch

- head 不仅仅可以指代新分支的最后一次提交，同时还可以不跟任何分支挂钩 处于分离头指针的情况下是指在了某个具体的commit上 切换分支的时候head 会跟随分支变的
- git diff HEAD HEAD~1/HEAD^1

4 怎么删除不需要的分支

- git branch -d [branch] 会有提示 not fully merged.
- git branch -D [branch] 强制删除某个分支

5 怎么修改最新的commit message

- git commit --amend 进入命令行 修改 wq!