在版本控制系统中,大约90%的操作都是在本地仓库中进行的:暂存, 提交,查看状态或者历史记录等等。除此之外,如果仅仅只有你一个人 在这个项目里工作, 你永远没有机会需要设置一个远程仓库。 我的GitHub账号为m18945097312@163.com 首先需要注册一个GitHub账号 只有当你需要和你的开发团队共享数据时,设置一个远程仓库才有意 义。你可以把它想象成一个 "文件管理服务器",利用这个服务器可以与 开发团队的其他成员进行数据交换。 A在自己的计算机中创建本地仓库 A在github中创建远程仓库 A将本地仓库推送到远程仓库 多人协作开发的流程 B克隆远程仓库到本地进行开发 B将本地仓库中开发的内容推送到远程仓库 A将远程仓库中的最新内容拉去到本地 创建本地仓库和远程仓库 git push 远程仓库地址 分支名称 git push 远程仓库地址别名 分支名称 git push -u 远程仓库地址别名 分支名称 推送本地仓库到远程仓库 -u 记住推送地址及分支,下次推送只需要输入git push即可 Class:13-git和GitHub day1 day2 GitHub git remote add 远程仓库地址别名 远程仓库地址 克隆远端数据仓库到本地: git clone 仓库地址 拉去操作 拉取远程仓库中最新的版本: git pull 远程仓库地址 分支名称 在多人同时开发一个项目时,如果两个人修改了同一个文件的同一个地 解决冲突 方,就会发生冲突。冲突需要人为解决。 生成秘钥: ssh-keygen 秘钥存储目录: C:\Users\用户\.ssh ssh免密登录 公钥名称: id\_rsa.pub 私钥名称: id\_rsa 将不需要被git管理的文件名字添加到此文件中,在执行git命令的时 候,git就会忽略这些文件。 Git忽略清单 git忽略清单文件名称: .gitignore

git add .

将工作目录中的文件全部添加到暂存区:

版本管理是一种记录文件变化的方式,以便将来查阅特定版本的文件内 什么是版本管理 文档数量多且命名不清晰导致文档版本混乱 人为维护文档版本的问题 每次编辑文档需要复制,不方便 多人同时编辑同一个文档, 容易产生覆盖 Git是一个版本管理控制系统(缩写VCS),它可以在任何时间点,将 文档的状态作为更新记录保存起来,也可以在任何时间点,将更新记录 什么是Git 恢复回来。 git仓库 暂存区 工作目录 用于存放提交记录 临时存放被修改文件 被Git管理的项目目录 Git基本工作流程 配置提交人的姓名 git config --global user.name 姓名 Git的基本配置 配置提交人的邮箱 git config --global user.email 邮箱 查看配置信息 git config --list 1、Git init 初始化Git仓库 2、Git status 查看文件状态 Git提交过程 3、Git add 文件名称 添加要管理的文件 Git的使用 4、Git commit -m 提交说明 向git仓库提交修改 5、Git log 查看提交日志 用暂存区中的文件覆盖工作目录中的文件: git checkout 文件 将文件从暂存区中删除: git rm — cached 文件 Git撤销 将 git 仓库中指定的更新记录恢复出来,并且覆盖暂存区和工作目录: git reset --hard commitID 为了便于理解,大家暂时可以认为分支就是当前工作目录中代码的一份 使用分支,可以让我们从开发主线上分离出来,以免影响开发主线。 本地数据库 主分支 (master): 第一次向 git 仓库中提交更新记录时自动产生的一 2024/07/22 2024/08/02 个分支。 12:06 17:31 11:52 14:13 html css js Git分支 、开发分支 (develop): 作为开发的分支, 基于 master 分支创建。 Git进阶 feature branches 功能分支(feature):作为开发具体功能的分支,基于开发分支创建 git branch 查看分支 git branch 分支名称 创建分支 git checkout 分支名称 切换分支 git merge 来源分支 合并分支 git branch -d 分支名称 删除分支(分支被合并后才允许删 除) (-D 强制删除) 分支命令 在git中,可以暂时提取分支上所有的改动并存储,让开发人员得到一 个干净的工作副本,临时转向其他工作。 使用场景:分支临时切换 暂时保存更改 存储临时改动: git stash 恢复改动: git stash pop