\$ cd /Users/Desktop/git演练/公司/weibo # 建立空白代码库(专门用于团队开发) 01. 建立代码仓库 \$ git init --bare # 切换目录 \$ cd /Users/Desktop/git演练/经理 # "克隆"代码库到本地 02. 项目经理准备项目 \$ git clone /Users/Desktop/git演练/公司/weibo/ # 个人信息配置(因为要演示一台机器上的多人协作,日常开发可以忽略) \$ git config user.name manager \$ git config user.email manager@163.com •gitignore可以指定哪些文件不纳入版本库的管理 参考网址: https://github.com/github/gitignore .gitignore # 命令行中进入与。git同级的目录 命令行使用 多人开发 GIT \$ cd /Users/Desktop/git演练/经理/weibo echo -e "内容" # 将 gitignore添加到代码库 \$ git add .gitignore 03. 创建项目 提交同时"push"到远程代码仓库 # "克隆"代码库到本地 \$ git clone /Users/Desktop/git演练/张三/weibo/ 04. 新人加入 01> 代码冲突 05. 解决冲突 02> StoryBoard冲突

\$ git init 1> 创建代码仓库 \$ git config user.name manager \$ git config user.email manager@gmail.com 2> 配置用户名和邮箱 * 以上两个命令会将用户信息保存在当前代码仓库中 \$ git config user.name manager \$ git config user.email manager@gmail.com 3> 一次性配置用户信息 * 以上两个命令会将用户信息保存在用户目录下的 $_{\mathbf{gitconfig}}$ 文件中 \$ touch main.c \$ open main.c 4> 创建代码,开始开发 # 查看当前代码库状态 \$ git status \$ git add main.c 5> 将代码添加到代码库 # 将修改提交到代码库 \$ git commit -m "添加了main.c" \$ touch Person.h Person.m \$ git add . \$ git commit -m "添加了Person类" \$ open Person.h \$ git add . \$ git commit -m "增加Person类属性" 6> 添加多个文件 * 注意 使用git时,每一次修改都需要添加再提交,这一 点是与svn不一样的 工作区 暂存区(staged) git 的重要概念及工作原理 分支(HEAD) \$ git config alias.st status **7>** 别名 \$ git config alias.ci "commit -m" # 查看所有版本库日志 \$ git log # 查看指定文件的版本库日志 \$ git log 文件名 # 查看分支引用记录 8> 日志 \$ git reflog # 配置带颜色的log别名 \$ git config --global alias.lg "log --color --graph --pretty=format:'%Cred %h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit" 提示:在git中,版本号是一个由SHA1生成的哈希值 # 回到当前版本,放弃所有没有提交的修改 \$ git reset --hard HEAD # 回到上一个版本 \$ git reset --hard HEAD^ 版本间穿越 # 回到之前第3个修订版本 \$ git reset --hard HEAD~(3) # 回到指定版本号的版本 \$ git reset --hard e695b67