

# 玩转Git

## 一、Git基础

- 01 | Git综述
- 02 | 安装Git
- 03 | 使用Git之前需要做的最小配置
- 04 | 创建第一个仓库并配置local用户信息
- 05 | 通过几次commit来认识工作区和暂存区
- 06 | 给文件重命名的简便方法
- 07 | 通过git log查看版本演变历史
- 08 | gitk: 通过图形界面工具来查看版本历史
- 09 | 探秘.git目录
- 10 | commit、tree和blob三个对象之间的关系
- 11 | 小练习: 数一数tree的个数
- 12 | 分离头指针情况下的注意事项
- 13 | 进一步理解HEAD和branch

## 二、单独使用Git

- 14 | 怎么删除不需要的分支?
- 15 | 怎么修改最新commit的message?
- 16 | 怎么修改老旧commit的message?
- 17 | 怎样把连续的多个commit整理成1个?
- 18 | 怎样把间隔的几个commit整理成1个?
- 19 | 怎么比较暂存区和HEAD所含文件的差异?
- 20 | 怎么比较工作区和暂存区所含文件的差异?
- 21 | 如何让暂存区恢复成和HEAD的一样?
- 22 | 如何让工作区的文件恢复为和暂存区一样?
- 23 | 怎样取消暂存区部分文件的更改?
- 24 | 消除最近的几次提交
- 25 | 看看不同提交的指定文件的差异
- 26 | 正确删除文件的方法
- 27 | 开发中临时加塞了紧急任务怎么处理?
- 28 | 如何指定不需要Git管理的文件?
- 29 | 如何将Git仓库备份到本地?

## 三、Git与GitHub同步

- 30 | 注册一个GitHub账号
- 31 | 配置公私钥
- 32 | 在GitHub上创建个人仓库
- 33 | 把本地仓库同步到GitHub

## 四、Git多人单分支集成协作

- 34 | 不同人修改了不同文件如何处理?
- 35 | 不同人修改了同文件的不同区域如何处理?
- 36 | 不同人修改了同文件的同一区域如何处理?
- 37 | 同时变更了文件名和文件内容如何处理?
- 38 | 把同一文件改成了不同的文件名如何处理?

## 五、Git集成使用禁忌

- 39 | 禁止向集成分支执行push -f操作
- 40 | 禁止向集成分支执行变更历史的操作

## 六、认识GitHub

- 41 | GitHub为什么会火?
- 42 | GitHub都有哪些核心功能?
- 43 | 怎么快速淘到感兴趣的开源项目?
- 44 | 怎样在GitHub上搭建个人博客
- 45 | 开源项目怎么保证代码质量?
- 46 | 为何需要组织类型的仓库?

## 七、使用GitHub进行团队协作

- 47 | 创建团队的项目
- 48 | 怎样选择适合自己团队的工作流?
- 49 | 如何挑选合适的分支集成策略?
- 50 | 启用issue跟踪需求和任务
- 51 | 如何用project管理issue?
- 52 | 项目内部怎么实施code review?
- 53 | 团队协作时如何做多分支的集成?
- 54 | 怎样保证集成的质量?
- 55 | 怎样把产品包发布到GitHub上?
- 56 | 怎么给项目增加详细的指导文档?

## 八、GitLab实践

- 57 | 国内互联网企业为什么喜欢GitLab?
- 58 | GitLab有哪些核心的功能?
- 59 | GitLab上怎么做项目管理?
- 60 | GitLab上怎么做code review?
- 61 | GitLab上怎么保证集成的质量?
- 62 | 怎么把应用部署到AWS上?