

Git入门

■ 适用人群

- 听说过 Git 并且
 - 想要入门的同学
 - 想了解 GitHub 团队协作的同学
- 没听说过 Git 但
 - 有版本控制需求的同学
 - 想要在闲暇时间学习一门必备技能的同学

■ 前期准备

- 安装 Git 到本地: <https://gitforwindows.org/>
- 注册 GitHub 账号: <https://github.com/>

■ 您将获得

- 了解使用 Git 管理本地代码的基本方法
- 了解使用 Git 进行团队协作、提交 pr 的基本方法
- 了解使用 Git 的一些经验和新手容易遇到的坑点

目录

- Git简介
- 学习使用Git管理本地代码
- 学习使用GitHub进行团队协作
- 向开源项目提交PR
- Git的其他玩法

Git 简介

- Git是目前最为流行的版本控制系统。
- 为什么需要版本控制系统?
 - 想象如下情景：我们在编写一个程序，突然想到了一个可能更好的实现方案，需要在原来的基础上进行修改。
 - 备份源文件？如果是涉及多个文件改动会很麻烦。
 - 如何记录有哪些改动？

Git 简介

■ Git 用途

- 本地代码管理
- 团队协作
- 参与开源项目开发
- 打造个人博客、电子书等

学习使用Git管理本地代码

- 初始化本地库: `git init`
- 添加到暂存区: `git add <filename>`
- 提交到本地库: `git commit -m "xxxx"`
- 查看分支: `git branch`
- 新建分支: `git branch xxxx`
- 切换分支: `git checkout xxxx`
- 合并分支到当前分支: `git merge xxxx`

学习使用GitHub进行团队协作

- 什么是GitHub
 - 远程仓库 && 开源社区
- 创建远程库（勾选Public和README.md）
- 建立GitHub与本地的关联
 - 创建密钥文件
 - `$ ssh-keygen -t rsa -C youremail@example.com`
 - 获取公钥文件
 - `$ cd ~/.ssh`
 - `$ ls`
 - `$ cat id_rsa.pub`
 - 将ssh-rsa后、用户名@主机名前的字符串复制，粘贴到Github-Settings-SSH and GPG keys-New SSH key-Key 中，Title随意取，点击Add SSH key，关联完成。

学习使用GitHub进行团队协作

- 在关联本地PC与GitHub的前提下，任何新创建的本地库都可以与远程库关联。
 - `$ git remote add origin xxx`
 - origin通常指自己的GitHub仓库
- 将本地文件推送到远程库
 - `git push [-u] origin <本地分支名>: <远程分支名>`
 - 如果省略远程分支名则表示将本地分支推送到与之存在“追踪关系”的远程分支（通常二者同名）。

```

1 //合并前
2 - o - o - o - H - A - B - C (master)
3
4           \
5             P - Q - R (origin/master)
6
7 //使用merge合并
8 - o - o - o - H - A - B - C - X (master)
9
10          \           /
11           P - Q - R --- (origin/master)
12
13 //使用rebase合并
14 - o - o - o - H - P - Q - R - A' - B' - C' (master)
15
16
17           |
18           (origin/master)

```

■ 团队合作

- 添加贡献者
- 更新本地仓库 `$ git pull origin master`
 - `git pull <分支名>` 相当于 `git fetch+git merge <分支名>`
 - `git pull --rebase <分支名>` 相当于 `git fetch+git rebase <分支名>`
- 解决冲突
 - 一旦发现冲突，Git会停止merge并会让你去解决冲突；在解决完冲突后，用git-add命令去更新本地暂存区，然后无需git-commit，只需要使用`$ git merge --continue`

向开源仓库/master分支提交pr

- 从原项目fork到自己的GitHub仓库。
- 从自己的GitHub仓库clone到本地。git clone 自己仓库下该项目的http链接。
- 建立自己仓库与原仓库（上游仓库）的连接。
 - `git remote add upstream 原http链接`。
 - `upstream`指原仓库
- 创建dev分支作为你的开发分支，修改代码。
- 推送到自己的GitHub仓库： `git push origin 刚创建的分支名`。
- 提出pr。
 - 找到New pull request，然后compare处选择刚提交的分支。点击Create pull request，就可以等待别人review你的代码后merge入上游仓库了。

Git的其他玩法

- 解锁Git技能后，您可以轻松上手下列玩法：
 - *GitHub* 个人博客
 - Hexo博客框架：<https://hexo.io/zh-cn/>
 - *GitHub* 电子书
 - <https://www.gitbook.com/>
- 如何同时使用多个Git代码托管平台
 - *Github*、*Gitlab*、*Gitee* 等