

Repository (仓库)

用 Repository 存放项目代码，每个项目对应一个仓库。

Star (收藏)

收藏项目，方便查找

Fork (复制/克隆项目)

将别人的仓库复制到自己然后放在自己的

Pull Request (发起请求)

如果别人在你的基础上做了些改进，后来觉得改得不错，想让更多人受益，于是就想起改进（合并）到原有项目里，他发起一个 Pull Request，则原有项目创建人（也就是你）可以收到这个请求，选择接受后可以添加他的改进。

Watch (关注)

如果 Watch 了某个项目，那以后只要这个项目有任何更新，你都能收到通知。

Issue (解决方案)

别人的项目，如果你发现了一些问题 bug，可以提个 issue；然后别人可以修改、修复，ok 了后可以一个个的 close 掉。

一个 git 仓库 对应 一个开源项目。

仓库管理 (~~Create new file~~)

Create new file

git 仓库

↑ git commit -m "描述"

-暂存区

git add + 文件名

↑

working Directory

git status

查看当前状态

Edit file

直接点击文件名 (列表中). 可编辑文件

git

git 是一个免费开源的 版本控制系统 VCS

↓

是一种记录一个或若干个文件内容
变化, 以便将来使用 特定版本
修改情况的系统

记录文件历史变化

随时可恢复到任何一个历史状态

多人协作开发更高效

错误恢复

git 初始化 及 仓库的创建和操作

1. 设置用户名

git config --global user.name '████████' (用户名)

2. 设置用户邮箱

git config --global user.████email '████' (邮箱)

3. 查看设置： git config --list

初始化一个新的仓库

1. 创建文件夹

2. 在文件夹内初始化git (创建git仓库)

命令： git init + <文件夹名>

会在文件夹内生成一个 .git 的隐藏文件夹。

向仓库中添加文件

1. 创建文件

2. 添加到暂存区

git add + 文件名(带扩展)

3. 将暂存区文件提交到仓库

git commit -m + 文件名(带扩展)

"提交描述"

备注：若想删去暂存区的文件（将暂存区文件恢复到工作区）

可以用： git rm --cached + 文件名(带扩展)

若想要删去暂存区文件同时又要删去本地文件

git rm + 文件名(带扩展)

修改仓库文件：

① vim + 文件名(带扩展名)：修改文件
↓ wq：修改后保存退出
⇒ 这一步可以在电脑上自由修改

git add + 文件名(带扩展名)：重新~~加~~修改后文件
↓
累加

git commit -m + “描述”：提交修改后的文件到仓库

删除仓库文件

1. 删除文件

- rm -rf + 文件名(带扩展名) → 也可用鼠标删除。

2. ~~删除从 git 中删除文件~~

git rm + 文件名(带扩展名)

3. 提交操作

git commit -m + “描述”

远程仓库

Git 克隆操作

目的：将远程仓库（GitHub 上项目）复制到本地

操作流程：

① git clone + 项目网址 : 自动将 GitHub 上项目克隆到本地

将本地仓库同步到 Git 远程仓库（如 GitHub 上）中

操作：① 先将远程仓库克隆本地

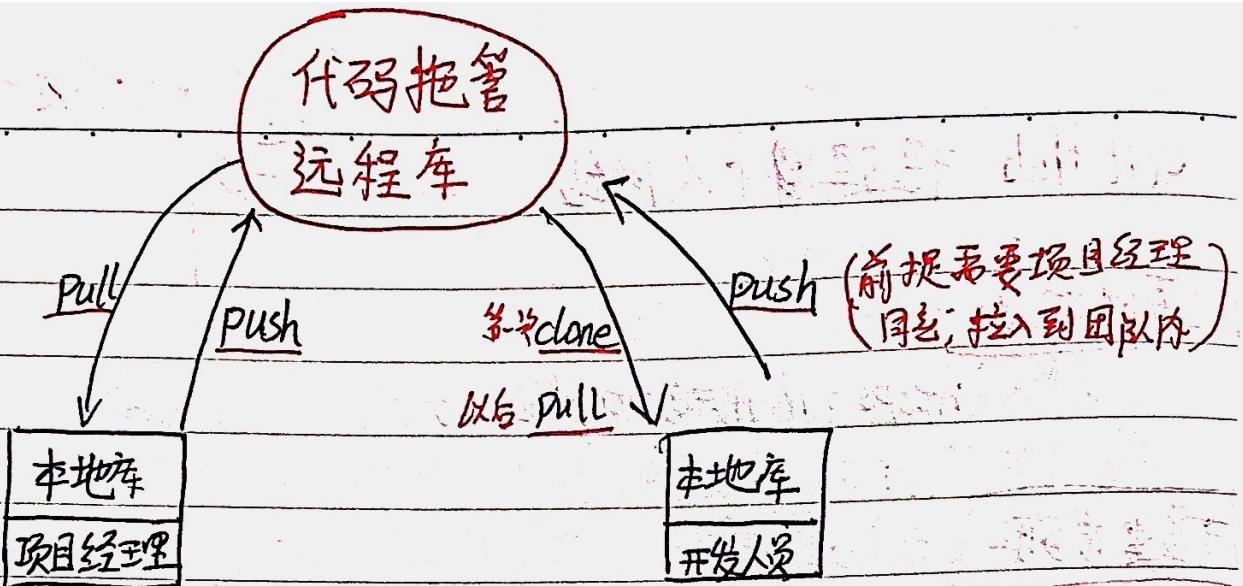
② 在本地修改项目仓库。

③ 将本地同步

↓ git push + 需等待一段时间

代码托管

远程库

**项目经理 流程**

1. 创建一个远程库 (例如在github上创建)
2. 创建本地git仓库 (可进行提交等操作)
3. git remote add demo + 远程仓库地址(网址)
在本地库创建远程库别名

git remote -v 查看远程库信息

4. git push -u demo master 同步到demo.master分支

开发人员

1. 将远程库克隆到本地 [通过 ZIP (这种方式不适合)
通过 "git clone + 网址" 克隆]
2. 修改代码或一些操作
3. 提交到本地库
4. git push 将本地分支仓库推送至远程仓库

注：前提需要项目经理将开发人员拉入团队内

最佳做法是进入远程库的 setting - collaborators

Add collaborator - 获得链接发给要邀请的人

注：但此时项目经理的本地库还没有更新，可以用

git pull 从远程获取最新版本并整合到本地

git 一些其他常用命令

(从暂存区跳回工作区)

① git checkout

: 重新工作区，以暂存区为准，重写工作区

② HEAD: 指当前的版本头



git reset

: 将当前 HEAD 复位到指定状态。

可撤销之前的一些操作

③ git log : 显示提交日志信息，之前提交的信息



拓展

git log --pretty=oneline

简洁显示日志信息，带 Hash 值。

git log --oneline : 带 Hash 值 缩短显示
日志信息

④ 查看(设置恢复)历史记录(之前或之后)

git reset --hard + 版本 Hash 值:

回退到指定版本 (之前 / 之后)



git reflog:

git reflog: 可以查看所有分支的所有操作记录 (包括 Commit 和 reset)
包括已经被删除的 Commit 记录。

而 git log 则不能查看已经删除了的 Commit 记录

文件比较

git diff：用于显示提交和工作树~~仓库~~之间的更改。

比较的是工作目录中当前文件和暂存区~~仓库~~

快照之间的差异，也就是修改之后还没有上~~到~~
暂存起来的变化内容

git diff + 文件名：比较具体文件工作区与暂存区~~仓库~~内容的差异

git diff HEAD + 文件名：比较具体文件在暂存区
与~~仓库~~最新本地库区~~仓库~~的差异

GitHub 搭建个人网站

注意

访问域名：

`https://用户名.github.io`

搭建步骤

(1) 创建个人站点 → 仓库 (注：仓库名必须是：用户名.github.io)

(2) 在仓库下新建 `index.html` 文件

Project Pages 项目站点

说明：

每一个仓库提供一个 web 界面

域名

`https://用户名.github.io/仓库名`

搭建步骤

1) 进入项目页，点击 settings

2) 在 settings 页，点击 `Launch · automatic page · generator`

3) 新建站点 基础信息 设置

自动生成主题

4) 选择主题

5) 生成网页 → publish · page

✓ 新版 GitHub 有改动

IDEA 工具 git

前提：配置 git 环境 将 git 安装路径导入到 IDEA 中

创建仓库：VCS → Import into Version Control → Create Git Repository
为项目(在项目文件夹内)创建一个本地 git 库

修改 添加 提交等操作

可以对整个项目也可以对单个文件

说明：add 后文件名变为红色

commit 后
未添加时文件为红色

& add 后文件为绿色

commit 后文件又变为红色

项目经理

创建 github 远程仓库 (不一定与 IDEA 中项目名称相同)

项目经理

将本地仓库项目 push 到远程库

前提：右击项目 → git → Repository → Remote : 配置要操作
github 仓库网址

右击项目 → git → Repository → Push

开发人员

远程仓库 clone 项目

VCS → checkout · from → git... ① 将远程仓库克隆到本地
 基本操作：从远程仓库拉取本地仓库

开发人员

修改项目

添加 / 提交等操作与项目经理类似

开发人员

将本地仓库同步到远程仓库

前提还要项目经理 邀请到 团队中

项目经理

同步远程仓库到本地

右侧项目 → git → Repository → pull

仓库 ← VCS → 本地仓库

代码冲突

在团队协同工作时，多位团队成员同时修改同一文件造成本地文件与 SVN 系统中的文件版本不一致而导致文件无法提交的情况

①

PS: IDEA 中会弹出^{警告框} Push Rejected.

解决方法

Accept Yours : 以自己的代码为准
点击 Merge | Accept Theirs : 以别人的代码为准
(显示当前的冲突 | Merge | 进一步修改
的具体信息)

关于冲突的建议



1. 修改文件之前先进行一次 update 操作 | Git Pull | 开发人员和项目经理都可用

2. 修改文件后及时 commit, 不要在本地停留过长时间
并 push