

远程库 Github:

仓库(Repositories)、收藏(Star)、克隆项目(Fork)、关注(Watch)、事物卡片(Issue)

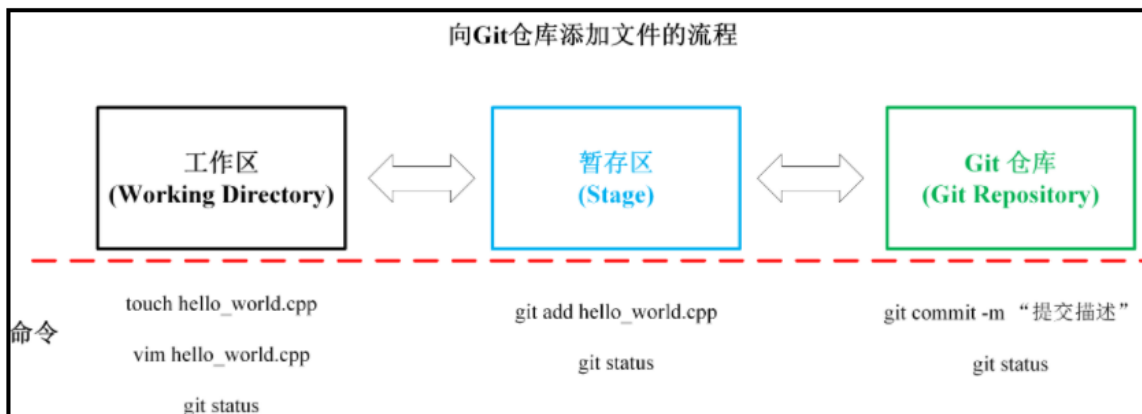
本地库 Git

账户初始化:

配置账户名: `git config --global user. Name "GitHub 用户名"`

配置账户邮箱: `git config --global user. email "GitHub 邮箱"`

初始化仓库: `git init`



查看当前工作状态: `git status`

查看历史提交记录: `git log`

或者 `git log --pretty=oneling` 查看简约版历史记录
`git reflog` 方便回退, 带 HEAD 号

版本的前进和后退: `git reset --hard` 版本号

`git reset --hard^` 表示后退一个版本

`git reset --hard^^` 表示后退两个版本

soft: 移动 HEAD 指针

mixed: 移动 HEAD 指针, 重置暂存区

hard: 移动 HEAD 指针, 重置暂存区, 重置工作区

版本中的文件内容比较:

工作区和暂存区比较: `git diff` 文件名

工作区和 Git 仓库比较: `git diff HEAD` 文件名

工作区和 Git 仓库的不同版本文件比较: `git diff` 版本号 文件名

删除文件:

`rm` 文件名 表示删除的是工作区的文件。

`git rm` 表示删除的是工作区和暂存区的文件。

`git rm -f` 文件名 表示当工作目录和暂存区的同一个文件存在不同内容时, 强制删除工作区和暂存区的文件。

`git rm --cached` 文件名 表示只删除暂存区的文件并且保留工作目录的文件。

分支

创建分支: `git branch 分支名`
查看分支: `git branch -v`
图形化显示分支: `git log --decorate --all --oneline --graph`
切换分支: `git checkout 分支名`
合并分支: `git merge 分支名` 需在 master 下操作
删除分支: `git branch -d 分支名`

本地库(push)到远程库(Github)

起别名: `git remote add origin 远程库地址`
删除别名: `git remote remove origin`
本地库推送到远程库: `git push origin 分支名(master)`
若报错, 是因为远程库中 README.md 本地库没有, 需要 pull 操作
`git pull origin master --allow-unrelated-histories`

远程库克隆(clone)到本地库: `git clone 远程仓库地址`

远程库拉取(pull)到本地库: `git pull origin master`

若远程库和本地库有不同的文件, 可通过强制拉取(pull):
`git pull origin master --allow-unrelated-histories`

也可通过(1) `git fetch origin master` 将远程库文件抓取到本地库缓存区,
(2) `git checkout origin/master` 切换到本地库缓存分支上, `cat 文件名` 查看文件内容
(3) `git merge origin/master` 将缓存区分支文件和本地库文件进行合并

`pull == fetch + merge`