

Git

2025年12月26日 星期五 12:23

git 是一种协议，一个工具，规定了怎么把东西上传，下拉，然后日志的记录和撤回之前的操作的一个协议，具体的代码放到哪里其实是自己定义的服务器，比如 github 就是代码托管，你免费给代码放上去，放到他的服务器上仅此而已，git 原生的甚至没有可视化页面，和 mysql 一样都没有，可视化页面可以自己下载对应的软件方便去管理，可能这也是一种分层封装的思想，专注于自己的业务，其他的有不同的需求由不同的用户去对应创建工具。

svn 是集中式的版本控制，由一个服务器控制所有的数据信息，git 相比而言是分布式的，所有的主机都拥有全部的信息，可以不害怕服务器的信息的丢失，但是安全性上没有 svn 的安全性高！

Git bash : Linux 风格的命令行

Git cmd: Windows 风格的命令行

Git gui: 图形界面的 git

绿色文件，蓝色目录，白色文件

git 原理：

工作区：存放本地内容

暂存区：修改的还未提交（本质就是一个文件而已）

仓库区：本地提交的修改内容

远程仓库：服务器上的内容

配置 ssh 公钥

由 git 产生公钥，然后到代码托管平台去设置公钥，本地还有一个私钥

这样子 托管平台才能识别到这个 git 是谁，两者相互连接

git 的命令：

Git config --global (用户)

Git config --system (系统设置)

Git init 初始化

Git commit - 将暂存区内容添加到仓库中。

Git add . - 添加文件到暂存区。

(每一次提交都要带上对应的信息说明这次的提交，方便后续的整理和了解)

git clone	拷贝一份远程仓库，也就是下载一个项目。
git add	添加文件到暂存区
git status	查看仓库当前的状态，显示有变更的文件。
git diff	比较文件的不同，即暂存区和工作区的差异。
git difftool	使用外部差异工具查看和比较文件的更改。
git range-diff	比较两个提交范围之间的差异。
git reset	回退版本。
git rm	将文件从暂存区和工作区中删除。
git mv	移动或重命名工作区文件。
git notes	添加注释。
git checkout	分支切换。
git switch (Git 2.23 版本引入)	更清晰地切换分支。
git restore (Git 2.23 版本引入)	恢复或撤销文件的更改。
git show	显示 Git 对象的详细信息。
git remote	远程仓库操作
git fetch	从远程获取代码库
git pull	下载远程代码并合并
git push	上传远程代码并合并
git submodule	管理包含其他 Git 仓库的项目

git branch --merged
git branch --no-merged
git branch -m master master_copy
git checkout -b master_copy
git checkout -b master master_copy
git checkout features/performance
git checkout --track hotfixes/BJVEP933
git checkout v2.0

显示所有已合并到当前分支的分支
显示所有未合并到当前分支的分支
本地分支改名
从当前分支创建新分支master_copy并检出
上面的完整版
检出已存在的features/performance分支
检出远程分支hotfixes/BJVEP933并创建本地跟踪分支
检出版本v2.0
从远程分支develop创建新本地分支devel并检出

```
git checkout -- README          # 检出head版本的README文件 (可用于修改错误回退)
git merge origin/master        # 合并远程master分支至当前分支
git cherry-pick ff44785404a8e  # 合并提交ff44785404a8e的修改
git push origin master         # 将当前分支push到远程master分支
git push origin :hotfixes/BJVEP933 # 删除远程仓库的hotfixes/BJVEP933分支
```