

你不知道的git

王彬彬

版本控制器(VCS)

- ◇ localized version control system
- ◇ centralized version control system
- ◇ distributed version control system

Working tree
(contains the
files you
work with and
do changes)

Staging area

Git repository

Step 1

Step 2

Add selected
changes to the index

Commit staged changes to
repository

命令

- ◇ `git add filename / .`
 - ◇ `git commit -m "注释"`
 - ◇ `git commit --amend -m 'xxx'` 合并上一次提交
 - ◇ `git status` 常看当前状态
 - ◇ `git log` 显示提交日志
 - ◇ `git diff` 比较修改
 - ◇ `git show` 查看信息
-
1. 修改，但没有用 `git add` 将修改添加到暂存区 `git checkout -- filename`
 2. 修改，已经使用 `git add` 将修改添加到暂存区 `git reset HEAD < filename >`
 3. 撤销 commit, `git reset HEAD^` `git reset 哈希值` `git reset --hard / --soft`
`git revert HEAD` 撤销远程

分支

- ◇ `git branch <branchName>` 创建分支
- ◇ `git checkout <branchName>` 切换分支
- ◇ `git branch -d <branchname>` 删除分支
- ◇ `git merge <branch>` 合并分支（把branch分支合并到当前分支）
- ◇ `git rebase <branch>` 合并分支（把当前分支合并到branch分支）

Rebase 实际上就是取出一系列的提交记录，“复制”它们，然后在另外一个地方逐个的放下去。

提交树上移动

- ◇ HEAD 总是指向当前分支上最近一次提交记录。大多数修改提交树的 Git 命令都是从改变 HEAD 的指向开始的
- ◇ `git checkout C1` 从分支上分离 HEAD 指向提交记录
- ◇ 使用 `^` 向上移动1个提交记录
- ◇ 使用 `~<num>` 向上移动多个提交记录 如: `~3`
- ◇ `Git checkout master^` 指向master的父节点 `master^^` 是 master 的第二个父节点
- ◇ `git branch -f master HEAD~3` 将master分支强制指向HEAD的第3级父提交
- ◇ (经验移动 HEAD 在移动分支)

整理提交记录

git cherry-pick C2 C4 C7

git rebase -i HEAD~4 整理记录

git tag v1 C1

远程

git pull; git fetch 和 git merge 的缩写

git pull -- rebase; git fetch 和 git rebase 的缩写

Git pull --rebase; git push 快速更新master 分支并推送到远程

git pull origin branchName 把branch分支推到 远程仓库

- ◇ git checkout -b foo o/master
- ◇ foo分支 远程分支 o/master
- ◇ git branch -u o/master foo
- ◇ foo 跟踪 o/master

显示所有分支

- ◇ `git log --oneline --abbrev-commit --all --graph --decorate --color`
- ◇ `gitk --simplify-by-decoration -all`
- ◇ `git config --global alias.st status` 定义缩写

暂存

`git stash` 暂存当前修改，将所有至为HEAD状态

`git stash list` 查看所有暂存

`git stash show -p stash@{0}` 参考第一次暂存

`git stash apply stash@{0}`