链滴

Git 常用资源

作者：JinFengYi
原文链接：https://ld246.com/article/1545384894397
来源网站：链滴

Git 常用资源

库管理

克隆库

```
git clone https://github.com/php/php-src.git
git clone --depth=1 https://github.com/php/php-src.git # 只抓取最近的一次 commit
```

历史管理

查看历史

```
git log --pretty=oneline filename # 一行显示
git show xxxx # 查看某次修改
```

标签功能

```
git tag # 显示所有标签
git tag -l 'v1.4.2.*' # 显示 1.4.2 开头标签
git tag v1.3 # 简单打标签
git tag -a v1.2 9fceb02 # 后期加注标签
git tag -a v1.4 -m 'my version 1.4' # 增加标签并注释， -a 为 annotated 缩写
git show v1.4 # 看某一标签详情
git push origin v1.5 # 分享某个标签
git push origin --tags # 分享所有标签
```

回滚操作

```
git reset 9fceb02 # 保留修改
git reset 9fceb02 --hard # 删除之后的修改
```

取消文件的修改

```
git checkout -- a.php #  取消单个文件
git checkout -- # 取消所有文件的修改
```

删除文件

```
git rm a.php  # 直接删除文件
git rm --cached a.php # 删除文件暂存状态
```

移动文件

```
git mv a.php ./test/a.php
```

查看文件修改

```
git diff          # 查看未暂存的文件更新
git diff --cached # 查看已暂存文件的更新
```

暂存和恢复当前staging

```
git stash # 暂存当前分支的修改
git stash apply # 恢复最近一次暂存
git stash list # 查看暂存内容
git stash apply stash@{2} # 指定恢复某次暂存内容
git stash drop stash@{0} # 删除某次暂存内容
```

修改 commit 历史纪录

```
git rebase -i 0580eab8
```

分支管理

创建分支

```
git branch develop # 只创建分支
git checkout -b master develop # 创建并切换到 develop 分支
```

合并分支

```
git checkout master # 切换到 master 分支
git merge --no-ff develop # 把 develop 合并到 master 分支，no-ff 选项的作用是保留原分支记录
git rebase develop # rebase 当前分支到 develop
git branch -d develop # 删除 develop 分支
```

克隆远程分支

```
git branch -r # 显示所有分支，包含远程分支
git checkout origin/android
```

修复develop上的合并错误

1. 将merge前的commit创建一个分之，保留merge后代码
2. 将develop reset --force到merge前，然后push --force
3. 在分支中rebase develop
4. 将分支push到服务器上重新merge

强制更新到远程分支最新版本

```
git reset --hard origin/master
git submodule update --remote -f
```

Submodule使用

克隆带submodule的库

```
git clone --recursive https://github.com/chaconinc/MainProject
```

clone主库后再去clone submodule

```
git clone https://github.com/chaconinc/MainProject
git submodule init
git submodule update
```

Git设置

Git的全局设置在~/.gitconfig中，单独设置在project/.git/config下。

忽略设置全局在~/.gitignore_global中，单独设置在project/.gitignore下。

设置 commit 的用户和邮箱

```
git config user.name "xx"
git config user.email "xx@xx.com"
```

或者直接修改config文件

```
[user]
    name = xxx
    email = xxx@xxx.com
```

查看设置项

```
git config --list
```

设置git终端颜色

```
git config --global color.diff auto
git config --global color.status auto
git config --global color.branch auto
```

原文链接：Git 常用资源