# GIT CHEAT SHEET

#### CREATE

## Clone 远程仓库

\$ git clone ssh://user@domain.com/xx.git

初始化本地 git 仓库(新建仓库)

\$ git init

## LOCAL CHANGES

#### 查看当前版本状态(是否修改)

\$ git status

显示所有未添加至 index 的变更

\$ git diff

比较与上一个版本的差异

\$ git diff HEAD^ / HEAD -- ./lib

增加更改过的文件至 index

\$ git add . / add \* ...

### 提交

\$ git commit -m 'xxx'

合并上一次提交(用于反复修改)

\$ git commit -amend -m 'xxx'

将 add 和 commit 合为一步

\$ git commit -am 'xxx'

# COMMIT HISTORY

# 显示日志

\$ git log

显示某个提交的详细内容

\$ git show <commit>

在每一行显示 commit 号,提交者,最早提交日期

\$ git blame <file>

## **BRANCHES & TAGS**

#### 显示本地分支

\$ git branch

### 切换分支

\$ git checkout <HEAD>

#### 新建分支

\$ git branch <new-branch>

## 创建新分支跟踪远程分支

\$ git branch --track <new> <remote>

## 删除本地分支

\$ git branch -d <branch>

#### 给当前分支打标签

\$ git tag <tag-name>

## UPDATE & PUBLISH

# 列出远程分支详细信息

\$ git remote -v

显示某个分支信息

\$ git remote show <remote>

添加一个新的远程仓库

\$ git remote add <remote> <url>

获取远程分支 但不更新本地分支 另需 merge

\$ git fetch <remote>

获取远程分支,并更新本地分支

\$ git pull <remote> <branch>

推送本地更新到远程分支

\$ git push <remote> <branch>

删除一个远程分支

\$ git push <remote> :<branch>

推送本地标签

\$ git push --tags

## MERGE & REBASE

合并分支到当前分支,存在两个

\$ git merge <branch>

合并分支到当前分支,存在一个

\$ git rebase <branch>

回到执行 rebase 之前

\$ git rebase --abort

解决矛盾后继续执行 rebase

\$ git rebase --continue

使用 mergetool 解决冲突

\$ git mergetool

使用冲突文件解决冲突

\$ git add <resolve-file>

\$ git rm <resolved-file>

### **UNDO**

将当前版本重置为 HEAD (用于 merge 失败)

\$ git reset --hard HEAD

将当前版本重置至某一个提交状态(慎用!)

\$ git reset --hard <commit>

将当前版本重置至某一个提交状态,代码不变

\$ git reset <commit>

重置至某一状态,保留版本库中不同的文件

\$ git reset --merge <commit>

重置至某一状态,重置变化的文件,代码改变

\$ git reset --keep <commit>

丢弃本地更改信息并将其存入特定文件

\$ git checkout HEAD <file>

撤销提交

\$ git revert <commit>