

GitHub

GIT备忘录

Git是一个开源的分布式版本控制系统，它可以方便地在您的笔记本电脑或桌面。本备忘单总结了常用的Git命令行指令，以供快速参考。

安装GIT

GitHub提供了桌面客户端，其中包括用于最常见存储库操作的图形用户界面，以及用于高级场景的自动更新命令行的Git命令行版本。

GitHub for Windows

<https://windows.github.com>

GitHub for Mac

<https://mac.github.com>

Linux和POSIX系统的Git发行版可以在Official Git SCM网站上找到。

Git for All Platforms

<http://git-scm.com>

配置工具

为所有本地仓库配置用户信息

```
$ git config --global user.name "[name]"
```

设置提交者的昵称显示在commit信息中

```
$ git config --global user.email "[email address]"
```

设置提交者的email 显示在commit信息中

```
$ git config --global color.ui auto
```

启用命令行输出为高亮彩色显示

创建 Git仓库

启动新存仓库或从现有URL获取仓库

```
$ git init [project-name]
```

使用指定的名称创建新的本地仓库

```
$ git clone [url]
```

下载项目及其整个版本历史记录 克隆远程仓库使用

改变状态

检查编辑并处理commit事务

```
$ git status
```

列出所有要提交的新文件或修改过的文件

```
$ git diff
```

显示所有尚未暂存的文件差异

```
$ git add [file]
```

快照文件，为版本控制做准备 将文件添加到暂存区

```
$ git diff --staged | git diff --cached
```

展示暂存区和最新提交历史版本内容不同

```
$ git reset [file]
```

将文件（file）从暂存区恢复到工作区中。

```
$ git commit -m "[descriptive message]"
```

在版本历史记录中永久记录文件快照，提交到本地仓库中。

分支改变

将一系列提交变成一个组合（分支）

```
$ git branch
```

列出本地仓库中所有的本地分支

```
$ git branch [branch-name]
```

创建一个新分支

```
$ git checkout [branch-name]
```

切换到指定的分支并更新工作目录

```
$ git merge [branch]
```

将指定分支的历史记录合并到当前分支

```
$ git branch -d [branch-name]
```

删除指定的分支

GIT CHEATSHEET

重构文件名

重新定位和删除版本控制文件

```
$ git rm [file]
```

从工作目录中删除该文件并且本地不保留（前提是该文件已commit）

```
$ git rm --cached [file]
```

从版本控制中删除该文件，但在本地保留该文件

```
$ git mv [file-original] [file-renamed]
```

更改文件名称 需要再次执行commit

抑制跟踪

排除临时文件和路径

```
*.log  
build/  
temp-*
```

名为.gitignore的文本文件禁止对匹配指定模式的文件和路径进行放弃版本控制

```
$ git ls-files --other --ignored --exclude-standard
```

列出此项目中所有被忽略的文件

保存工作片段

搁置并恢复不完整的更改

```
$ git stash
```

临时存储所有修改后的跟踪文件（添加到暂存区的文件）

```
$ git stash pop
```

还原最近隐藏的文件

```
$ git stash list
```

列出所有隐藏的变更集

```
$ git stash drop
```

丢弃最近隐藏的变更集

回顾历史

浏览和检查项目文件的历史快照记录

```
$ git log
```

列出当前分支的版本历史记录

```
$ git log --follow [file]
```

列出文件的版本历史记录，包括重命名

```
$ git diff [first-branch]...[second-branch]
```

显示两个分支之间的内容差异

```
$ git show [commit]
```

输出指定提交的元数据和内容更改

重做提交

清除错误提交并创建替换历史记录

```
$ git reset [commit]
```

Undoes all commits after [commit], preserving changes locally in[提交]之后撤消所有提交，在本地保存更改

```
$ git reset --hard [commit]
```

丢弃所有历史记录并将更改返回到指定的提交

同步更改(远程仓库)

注册存储仓库（远程仓库）书签并交换版本历史记录

```
$ git fetch [bookmark]
```

从存储库（远程仓库）下载所有历史记录

```
$ git merge [bookmark]/[branch]
```

将远程仓库的分支组合到当前本地分支中

```
$ git push [alias] [branch]
```

上传所有本地分支提交到GitHub

```
$ git pull
```

从存储库（远程仓库）下载所有历史记录并合并更改

GitHub Training

Learn more about using GitHub and Git. Email the Training Team or visit

✉ training@github.com
🌐 training.github.com