

# GitHub

# GIT CHEAT SHEET

V1.1.1

Git 是开源分布式版本控制系统，可促进笔记本电脑或台式机上的 GitHub 活动。此备忘单总结了常用的 Git 命令行说明以供快速参考。

## 安装 GIT

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

### GitHub for Windows

<https://windows.github.com>

### GitHub for Mac

<https://mac.github.com>

Git distributions for Linux and POSIX systems are available on the official Git SCM web site.

### Git for All Platforms

<https://git-scm.com>

## 配置工具

在所有的本地版本库设置用户信息

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

设置你要附加到提交事务的名称

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

设置你要附加到提交事务的邮箱地址

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

启用实用的彩色命令行输出

### 事务Transaction:

指访问并可能更新数据库中各种数据项的一个程序执行单元

## 创建版本库

新建一个新的版本库或获取一个现有URL的版本库

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

创建一个特定名称的本地版本库

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

下载一个项目以及它全部的版本

## 修改

查看编辑并创造提交事务

```
$ git status
```

列出所有将要提交的新建的或修改的文件

```
$ git diff
```

显示暂存的文件的修改差异

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

对文件进行快照以准备版本控制

```
$ git diff --staged
```

显示暂存和上一个文件版本之间的文件差异

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

取消暂存文件，但保留其内容

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

在版本历史记录中永久记录文件快照

快照：对指定数据集的一个完全可用拷贝，该拷贝包含源数据在拷贝时间点的静态影像

## 修改组

命名一系列提交并合并完成的工作

```
$ git branch
```

列出当前版本库中的所有本地分支

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

创建新分支

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

切换到特定分支并更新工作区

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

合并指定分支的历史记录到当前分支

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

删除指定分支

# GIT CHEAT SHEET

## 重构文件名

重新定位和删除版本化文件

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

从工作区删除文件并暂存删除文件

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

从版本控制中删除文件但在本地保留文件

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

更改文件名并准备提交

## 禁用追踪

排除临时文件和路径

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

名为 .git ignore 的文本文件抑制了与指定模式匹配的文件和路径的意外版本控制

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

列出此项目中所有被忽略的文件

## 碎片保存

搁置和恢复未完成的更改

```
$ git stash
```

临时存储所有修改过的跟踪文件

```
$ git stash pop
```

恢复最近保存的文件

```
$ git stash list
```

列出所有隐藏的变更集

```
$ git stash drop
```

丢弃最近存储的变更集

## 回顾历史记录

浏览和检查项目文件的演变

```
$ git log
```

列出当前分支的版本记录

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

列出一个文件的版本历史，包括重命名

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

显示两个分支的内容差异

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

输出指定提交的元数据和内容更改

## 重新提交

删除错误和工艺更换历史

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

在 [commit] 之后撤消所有提交，在本地保留更改

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

丢弃所有历史和更改回指定的提交

## 同步变更

注册存储库书签并交换版本历史记录

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

从版本库书签中下载所有历史记录

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

结合书签分支到当前本地分支

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

上传所有本地分支提交到GitHub

```
$ git pull
```

下载书签历史并合并更改

## GitHub Training

Learn more about using GitHub and Git. Email the Training Team or visit our web site for learning event schedules and private class availability.

✉ [training@github.com](mailto:training@github.com)  
🌐 [training.github.com](https://training.github.com)