

# GitHub Git 备忘单

Git是一个开源的分布式版本控制系统，方便你在笔记本或桌面端进行Github的操作，这个备忘单总结了常用的Git命令行指令，以便快速查询。

## 安装Git

Github提供了包含图形界面的桌面客户端，通过客户端可以完成大部分常用的仓库操作，同时可以自动更新Git的命令行版本，以适应新的场景。

## GitHub for Windows

<http://windows.github.com>

## GitHub for Mac

<http://mac.github.com>

Github的Linux和POSIX版本可以在官方的Git SCM网站上获取。

## Git 全平台版

<http://git-scm.com>

## 配置工具

对所有本地仓库的用户信息进行配置

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

对你的commit操作设置关联的用户名

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

对你的commit操作设置关联的邮箱地址

## 创建仓库

创建一个新的仓库或者从一个现有的链接获取仓库

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

创建一个本地的仓库，并设置名字

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

下载一个项目以及它所有的版本历史

## 更改

检查已有的编辑并执行commit操作

```
$ git status
```

列出所有新建或者更改的文件，这些文件需要被commit

```
$ git diff
```

展示那些没有暂存文件的差异

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

将文件进行快照处理用于版本控制

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

展示暂存文件与最新版本之间的不同

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

将文件移除暂存区，但是保留其内容

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

将文件快照永久地记录在版本历史中

## 批量更改

命名一系列commit以及合并已完成的工作

```
$ git branch
```

列出当前仓库中所有的本地分支

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

建立一个新分支

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

切换到一个特定的分支上并更新工作目录

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

合并特定分支的历史到当前分支

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

删除特定的分支

# 重构文件

重定位并移除版本文件

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

从工作目录中删除文件并暂存此删除

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

从版本控制中移除文件，并在本地保存文件

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

改变文件名并准备commit

# 停止追踪

不包含临时文件和路径

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

文本文件 `.gitignore` 可以防止一些特定的文件进入到版本控制中

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

列出所有项目中忽略的文件

# 保存临时更改

暂存一些未完成的更改

```
$ git stash
```

临时存储所有修改的已跟踪文件

```
$ git stash pop
```

重新存储所有最近被stash的文件

```
$ git stash list
```

---

© 2015 GitHub, Inc. Terms (<https://github.com/site/terms>)

License (<https://training.github.com/kit/license.html>) Security (<https://github.com/security>)

Contact (<https://training.github.com/contact>)

```
(/) $ git stash drop
```

放弃所有最近stash的更改

 [githubtraining \(https://twitter.com/GitHubTraining\)](https://twitter.com/GitHubTraining) [Status \(https://status.github.com\)](https://status.github.com)  
[Blog \(https://github.com/blog\)](https://github.com/blog) [About \(https://github.com/about\)](https://github.com/about)

# 查阅历史

浏览并检查项目文件的发展

```
$ git log
```

列出当前分支的版本历史

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

列出文件的版本历史，包括重命名

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

展示两个不同分支之间的差异

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

输出元数据以及特定commit的内容变化

## 撤销commit

擦除错误并更改历史

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

撤销所有 [commit] 后的的commit，在本地保存更改

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

放弃所有更改并回到某个特定的commit

## 同步更改

注册一个远程的链接，交换仓库的版本历史

```
$ git fetch [remote]
```

下载远程仓库的所有历史

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

合并远程分支到当前本地分支

```
$ git push [remote] [branch]
```

上传所有本地分支commit到Github上

```
$ git pull
```

下载书签历史并合并更改