



安装

GitHub Desktop

desktop.github.com

Git 全平台版

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

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

启用有帮助的彩色命令行输出

分支

分支是使用 Git 工作的一个重要部分。你做的任何提交都会发生在当前“checked out”到的分支上。使用 `git status` 查看那是哪个分支。

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

创建一个新分支

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

切换到指定分支并更新工作目录(working directory)

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

将指定分支的历史合并到当前分支。这通常在拉取请求(PR)中完成，但也是一个重要的 Git 操作。

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

删除指定分支

创建仓库

当着手于一个新的仓库时，你只需创建一次。要么在本地创建，然后推送到 GitHub；要么通过 clone 一个现有仓库。

```
$ git init
```

在使用过 `git init` 命令后，使用以下命令将本地仓库与一个 GitHub 上的空仓库连接起来：

```
$ git remote add origin [url]
```

将现有目录转换为一个 Git 仓库

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

Clone（下载）一个已存在于 GitHub 上的仓库，包括所有的文件、分支和提交(commits)

.gitignore 文件

有时一些文件最好不要用 Git 跟踪。这通常在名为 `.gitignore` 的特殊文件中完成。你可以在 github.com/github/gitignore 找到有用的 `.gitignore` 文件模板。

同步更改

将你本地仓库与 GitHub.com 上的远端仓库同步

```
$ git fetch
```

下载远端跟踪分支的所有历史

```
$ git merge
```

将远端跟踪分支合并到当前本地分支

```
$ git push
```

将所有本地分支提交上传到 GitHub

```
$ git pull
```

使用来自 GitHub 的对应远端分支的所有新提交更新你当前的本地工作分支。git pull 是 git fetch 和 git merge 的结合

进行更改

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]
```

展示两个分支之间的内容差异

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

输出指定commit的元数据和内容变化

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

将文件进行快照处理用于版本控制

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

将文件快照永久地记录在版本历史中

重做提交

清除错误和构建用于替换的历史

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

撤销所有 [commit] 后的的提交，在本地保存更改

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

放弃所有历史，改回指定提交。

与社区，以及历史与能中不个改后来。想不你而又又以 GitHub (远端) 仓库的定义，有连接环作。如果你需要帮助，可访问 [github.community](https://github.com/community) 或联系支持(support)。

术语表

- **git**: 一个开源的分布式版本控制系统
- **GitHub**: 一个托管和协作管理 Git 仓库的平台
- **commit 提交**: 一个 Git 对象，是你整个仓库的快照的哈希值
- **branch 分支**: 一个轻型可移动的 commit 指针
- **clone**: 一个仓库的本地版本，包含所有提交和分支
- **remote 远端**: 一个 GitHub 上的公共仓库，所有小组成员通过它来交换修改
- **fork**: 一个属于另一用户的 GitHub 上的仓库的副本
- **pull request 拉取请求**: 一处用于比较和讨论分支上引入的差异，且具有评审、评论、集成测试等功能的地方
- **HEAD**: 代表你当前的工作目录。使用 `git checkout` 可移动 HEAD 指针到不同的分支、标记(tags)或提交

Made with ❤️ by 🐱s and friends

  COMMUNITY

© 2022 GitHub, Inc. Jekyll & Primer.