

GitHub

Git 备忘单



Git是一个开源的分布式版本控制系统，可以帮助您在笔记本电脑或台式机上进行GitHub操作。本备忘单总结了常用的Git命令行说明，以供快速参考。

安装

在Windows安装GitHub
<https://windows.github.com>

在Mac安装GitHub
<https://mac.github.com>

在所有平台上安装GitHub
<http://git-scm.com>

针对Linux和POSIX系统的Git发行版可以在Git SCM官方网站上获得。

配置工具

为所有本地存储库配置用户信息

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

设置你想附加到提交过程的用户名

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

设置你想附加到提交过程的邮箱

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

启用命令行输出的帮助颜色

分支

分支是使用Git的一个重要部分。您所做的任何提交都将在当前“checked out”的分支中进行。用 `git status` 查看是哪个分支。

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

创建一个分支

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

切换到指定分支并更新工作目录

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

合并指定分支的提交历史到当前分支。这通常是在pull请求中完成的，但这是一个重要的Git操作。

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

删除指定分支

创建仓库

当开始使用一个新的存储库时，你只需要做一次;要么在本地，然后推到GitHub，要么克隆一个现有的存储库。

```
$ git init
```

转换一个已存在的目录为git创库。

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

克隆(下载)已经存在于GitHub上的存储库，包括所有的文件、分支和提交

.gitignore 文件

有时，排除使用Git跟踪的文件可能是个好主意。这通常在一个名为.gitignore的特殊文件中完成。你可以在github.com/github/gitignore上找到.gitignore文件的有用模板。

同步变化

同步本地仓库和GitHub.com上的远程仓库。

```
$ git fetch
```

从远程跟踪分支下载所有历史记录

```
$ git merge
```

将远程跟踪分支合并到当前的本地分支

```
$ git push
```

上传本地分支的所有提交到GitHub

```
$ git pull
```

用GitHub上对应的远程分支的所有新的提交更新你当前本地的工作分支。

git pull 是 git fetch和git merge的合并

进行变

浏览并检查项目文件的变化

<code>\$ git log</code>	列出当前分支的版本历史
<code>\$ git log --follow [file]</code>	列出文件的版本历史包括重命名
<code>\$ git diff [first-branch]...[second-branch]</code>	显示两个分支之间的内容差异
<code>\$ git show [commit]</code>	输出指定提交的元数据和内容变更
<code>\$ git add [file]</code>	快照这个文件为版本控制做准备
<code>\$ git commit -m "[descriptive message]"</code>	在版本历史中记录文件快照

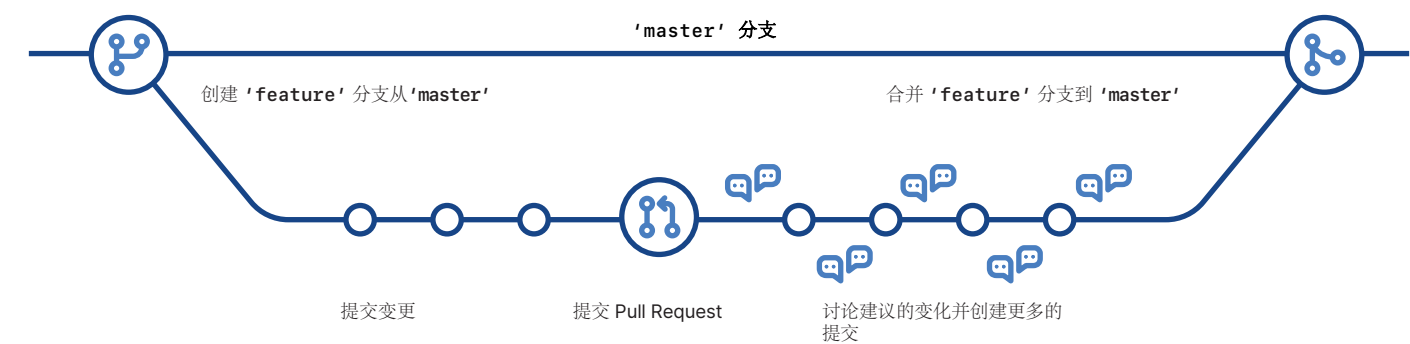
恢复提交

删除错误和技巧性地替换历史提交

<code>\$ git reset [commit]</code>	取消[commit]后的所有提交，保留本地的改变
<code>\$ git reset --hard [commit]</code>	放弃所有的历史和变更退回到指定提交

注意! 变更历史会有副作用。如果你需要改变已经存在于GitHub（远程的）上的提交，小心处理。如果你需要帮助，访问 [github.community](https://github.com/community) 或者联系支持人员。

GitHub 流程



术语

- git:** 一个开源的分布式版本控制系统
- GitHub:** 一个对Git仓库托管和合作的平台
- commit:** 一个Git对象，一个对你整个仓库压缩到一个SHA的快照
- branch:** 指向提交的轻量性可移动指针
- clone:** 一个仓库的本地版本包括所有提交和分支
- remote:** 一个GitHub上的共同仓库，所有团队成员都可以通过它交换他们的变化
- fork:** 一个GitHub上由不同用户所拥有的仓库副本
- pull request:** 一个比较和讨论在一个分支中引入差异的地方，包括审查、评论、集成测试等等
- HEAD:** 当时用git checkout时，表示当前工作目录的HEAD指针可以移动到不同的分支、标签或提交

GitHub Training

想了解更多关于使用GitHub和Git的信息吗?给培训团队发电子邮件或访问我们的网站了解学习活动时间表和私人课程的可用性。

- ✉ services@github.com
- 📄 services.github.com