GitHub

GIT CHEAT SHEET

V 1.1.1

Git是一个开源的分布式版本控制系统，它可以为您的笔记本或桌面的GitHub活动提供便利。本备忘单总结了常用的Git命令行指令，以供快速参考。

Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. This cheat sheet summarizes commonly used Git command line instructions for quick reference.

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版本。

INSTALL GIT

GitHub provides desktop clients that include a graphical user interface for the most common repository actions and an automatically updating command line edition of Git for advanced scenarios.

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
http://git-scm.com

CONFIGURE TOOLING

Configure user information for all local repositories

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

Sets the name you want attached to your commit transactions

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

Sets the email you want attached to your commit transactions

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

启用命令行输出的彩色化

Enables helpful colorization of command line output

CREATE REPOSITORIES

Start a new repository or obtain one from an existing URL

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

Creates a new local repository with the specified name

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

Downloads a project and its entire version history

MAKE CHANGES

Review edits and craft a commit transaction

```
$ git status
```

Lists all new or modified files to be committed

```
$ git diff
```

显示尚未暂存的文件差异

Shows file differences not yet staged

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

为版本化而准备的文件快照

Snapshots the file in preparation for versioning

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

显示暂存和最后一个文件版本之间的文件差异

Shows file differences between staging and the last file version

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

取消文件的阶段，但保留其内容

Unstages the file, but preserve its contents

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

Records file snapshots permanently in version history

GROUP CHANGES

Name a series of commits and combine completed efforts

列出一系列提交并合并完成的工作

```
$ git branch
```

Lists all local branches in the current repository

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

Creates a new branch

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

Switches to the specified branch and updates the working directory

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

Combines the specified branch's history into the current branch

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

Deletes the specified branch

GIT CHEAT SHEET

重构

REFACTOR FILENAMES

重新定位和删除版本化的文件

Relocate and remove versioned files

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

从工作目录中删除文件并执行删除

Deletes the file from the working directory and stages the deletion

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

从版本控制中删除文件，但在本地保留文件

Removes the file from version control but preserves the file locally

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

Changes the file name and prepares it for commit

抑制跟踪

SUPPRESS TRACKING

排除临时文件和路径

Exclude temporary files and paths

```
*.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意外版本控制

A text file named `.gitignore` suppresses accidental versioning of files and paths matching the specified patterns

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

列出此项目中所有被忽略的文件

Lists all ignored files in this project

保存碎片

SAVE FRAGMENTS

搁置并恢复不完整的更改

Shelve and restore incomplete changes

```
$ git stash
```

临时存储所有修改后的跟踪文件

Temporarily stores all modified tracked files

```
$ git stash pop
```

还原最近隐藏的文件

Restores the most recently stashed files

```
$ git stash list
```

列出所有隐藏的变更集

Lists all stashed changesets

```
$ git stash drop
```

丢弃最近隐藏的更改集

Discards the most recently stashed changeset

REVIEW HISTORY

浏览和检查项目文件的演变

Browse and inspect the evolution of project files

```
$ git log
```

Lists version history for the current branch

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

Lists version history for a file, including renames

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

Shows content differences between two branches

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

输出指定提交的元数据和内容更改

Outputs metadata and content changes of the specified commit

REDO COMMITS

删除错误并创建替换历史

Erase mistakes and craft replacement history

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

取消[提交]后的所有提交，在本地保存更改

Undoes all commits after `[commit]`, preserving changes locally

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

Discards all history and changes back to the specified commit

同步更改

SYNCHRONIZE CHANGES

注册存储库书签并交换版本历史记录

Register a repository bookmark and exchange version history

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

从存储库书签下载所有历史记录

Downloads all history from the repository bookmark

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

将bookmark的分支合并到当前本地分支中

Combines bookmark's branch into current local branch

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

Uploads all local branch commits to GitHub

```
$ git pull
```

下载书签历史记录并合并更改

Downloads bookmark history and incorporates changes

bookmark和alias区别？

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

training.github.com