

GitHub

GIT CHEAT SHEET

V 1.1.0

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.

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]"
```

Set the name you want attached to your commit transactions

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

Set the email you want attached to your commit transactions

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

Enable helpful colorization of command line output

CREATE REPOSITORIES

Start a new repository or obtain one from an existing URL

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

Create a new local repository with the specified name

```
git clone [url]
```

Download a project and its entire version history

MAKE CHANGES

Review edits and craft a commit transaction

```
git status
```

List all new or modified files to be committed

```
git diff
```

Show file differences not yet staged

```
git add [file]
```

Snapshot the file in preparation for versioning

```
git diff --staged
```

Show file differences between staging and the last file version

```
git reset [file]
```

Unstage the file, but preserve its contents

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

Record file snapshots permanently in version history

GROUP CHANGES

Name a series of commits and combine completed efforts

```
git branch
```

List all local branches in the current repository

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

Create a new branch

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

Switch to the specified branch and update working directory

```
git merge [branch]
```

Combine the specified branch's history into the current branch

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

Delete the specified branch



GIT CHEAT SHEET

REFACTOR FILENAMES

Relocate and remove versioned files

```
git rm [file]
```

Delete the file from the working directory and stage the deletion

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

Remove from version control but preserve the file locally

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

Change the filename and prepare it for commit

SUPPRESS TRACKING

Exclude temporary files and paths

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

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

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

List all ignored files in this project

SAVE FRAGMENTS

Shelve and restore incomplete changes

```
git stash
```

Temporarily store all modified tracked files

```
git stash pop
```

Restore the most recently stashed files

```
git stash list
```

List all stashed changesets

```
git stash drop
```

Discard the most recently stashed changeset

REVIEW HISTORY

Browse and inspect the evolution of project files

```
git log
```

List version history for the current branch

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

List version history for a file, including renames

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

Show content differences between two branches

```
git show [commit]
```

Output metadata and content changes of the specified commit

REDO COMMITS

Erase mistakes and craft replacement history

```
git reset [commit]
```

Undo all commits after [commit], preserving changes locally

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

Discard all history and changes back to the specified commit

SYNCHRONIZE CHANGES

Register a repository bookmark and exchange version history

```
git fetch [bookmark]
```

Download all history from the repository bookmark

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

Combine bookmark's branch into current local branch

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

Upload all local branch commits to GitHub

```
git pull
```

Synchronize bookmark history and incorporate current branch

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