

GIT CHEAT SHEET

DIANJING FAN
YIJIA HE

GIT INTRODUCTION

- What is Git?

- Distributed version control system (DVCS)
- Free and open source
- Tracking and managing changes of codes
- Coordinating work among programmers collaboratively
- Benefits
 - A complete long-term change history of every file
 - Branching and merging
 - Traceability
 - Reduce communication overhead and increase release velocity

- Git installation and GUIs

- Windows: git-scm.com/download/win
- MacOS: git-scm.com/download/mac
- Linux/Unix: git-scm.com/download/linux

After successful installation, we can run all Git commands in command line. Git has built-in GUI (graphical user interface) tools for committing (Git GUI) and browsing (gitk); in addition, we can use third-party GUI clients for different platform-based experience, for example, GitHub Desktop, SourceTree, GitKraken, etc.

- Setup

Command line arguments that use to configure user information used across all repositories:

`git config --global user.name "[firstname last]"`
set a name that is identifiable for credit when review version history

`git config --global user.email "[valid-email]"`
set an email address that will be associated with each history marker

`git config --global color.ui auto`
set automatic command line coloring for Git for easy reviewing

GIT TERMINOLOGY

Term	Meaning
Branch	A version of the repository that diverges from the main working project
Checkout	Switch between branches in a repository
Cherry-Picking	Apply some commit from one branch into another branch
Clone	A copy of a repository or the action of copying a repository
Fetch	Download and copy that branch's files to your workstation
Fork	Create a copy of a repository
Head	The last commit of the repository in which you are working
Index	A staging area between the working directory and repository
Master	The default/primary branch of all repositories
Merge	Take and add the changes from one branch into another branch
Origin	A reference to the remote repository from a project was initially cloned
Pull/Pull Request	A process for a developer to notify team members that they have completed a feature and add the changes to the master branch
Push	Upload local repository content to a remote repository
Rebase	The process of moving or combining a sequence of commits to a new commit

GIT TERMINOLOGY

Term	Meaning
Remote	A copy of the original branch stored on a code hosting service like GitHub etc
Repository	A directory that stores all the files, folders, and content needed for your project
Stash	Switch branch without committing the current branch
Tag	Mark a commit stage as important
Upstream	Where you cloned the repository/push your Git changes from the origin
Downstream	Any project that integrates your work with other works
Hook	A script that runs automatically every time a particular event occurs in a Git repository
Main	The default development branch

COMMAND LINE ARGUMENT

- SETUP & INITIALIZATION

Initializing and cloning repositories

`git init`

initialize an existing directory as a Git repository

`git clone [url]`

retrieve an entire repository from a hosted location via URL

- STAGE & SNAPSHOT

About snapshots and the Git staging area

`git status`

show modified files in working directory, staged for your next commit

`git add [file]`

add a file as it looks now to your next commit (stage)

`git reset [file]`

unstage a file while retaining the changes in working directory

`git diff`

diff of what is changed but not staged

`git diff --staged`

diff of what is staged but not yet committed

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

commit your staged content as a new commit snapshot

- BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

`git branch`

list your branches and it will appear next to the currently active branch

`git branch [branch-name]`

create a new branch at the current commit

`git checkout`

switch to another branch and check it out into your working directory

`git merge [branch]`

merge the specified branch's history into the current one

`git log`

show all commits in the current branch's history

GIT CHEAT SHEET

DIANJING FAN
YIJIA HE

COMMAND LINE ARGUMENT

- INSPECT & COMPARE

Examining logs, diffs and object information

`git log branchB..branchA`

show the commits on branchA that are not on branchB

`git log --follow [file]`

show the commits that changed file, even across renames

`git diff branchB...branchA`

show the diff of what is in branchA that is not in branchB

`git show [SHA]`

show any object in Git in human-readable format

- TRACKING PATH CHANGES

Versioning file removes and path changes

`git rm [file]`

delete the file from project and stage the removal for commit

`git mv [existing-path] [new-path]`

change an existing file path and stage the move

`git log --stat -M`

show all commit logs with indication of any paths that moved

- IGNORING PATTERNS

Preventing unintentional staging or committing of files

`git config --global core.excludesfile [file]`

system wide ignore patern for all local repositories

- SHARE & UPDATE

Retrieving updates from another repository and updating local repos

`git remote add [alias] [url]`

add a git URL as an alias

`git fetch [alias]`

fetch down all the branches from that Git remote

`git merge [alias]/[branch]`

merge a remote branch into your current branch to bring it up to date

`git push [alias] [branch]`

Transmit local branch commits to the remote repository branch

`git pull`

fetch and merge any commits from the tracking remote branch

COMMAND LINE ARGUMENT

- REWRITE HISTORY

Rewriting branches, updating commits and clearing history

`git rebase [branch]`

apply any commits of current branch ahead of specified one

`git reset --hard [commit]`

clear staging area, rewrite working tree from specified commit

- TEMPORARY COMMITS

Temporarily store modified, tracked files in order to change branches

`git stash`

Save modified and staged changes

`git stash list`

list stack-order of stashed file changes

`git stash pop`

write working from top of stash stack

`git stash drop`

discard the changes from top of stash stack

REFERENCE

- <https://education.github.com/git-cheat-sheet-education.pdf>
- <https://git-scm.com/downloads>
- <https://en.wikipedia.org/wiki/GitCharacteristics>
- <https://www.javatpoint.com/git-terminology>
- <https://acloudguru.com/blog/engineering/git-terms-explainedbranch>
- <https://www.atlassian.com/git/glossary/terminology>
- <https://www.atlassian.com/git/tutorials/source-code-management>

ADVANCED GIT

Git SSH Key

- An SSH key is an access credential for the secure shell network protocol.
 - For remote communication between machines on an unsecured open network
 - For remote file transfer, network management, and remote operating system access
- Git is capable of using SSH keys instead of traditional password authentication when pushing or pulling to remote repositories.

- Git Archive

- An archive file combines multiple files into a single file
 - Creat distributable packages of git repositories
 - Target specific refs of a repository and only package the contents of that ref