

Getting Started with Git and GitHub

Module 2 Cheat Sheet: Git Commands and Managing GitHub Projects

Package/Method	Description	Code Example
git add	Used to move changes from the working directory to the staging area	<code>git add sample.md</code>
git add .	Allows to move the changed files into the staging area on GitHub repositories	<code>git add .</code>
git am	Used to apply patches emailed to the repository	<code>git am < patchfile.patch</code>
git branch	Allows to create an isolated environment within the repository to make changes	<code>git branch <new-branch></code>
git checkout	Allows to see and change existing branches	<code>git checkout <existing-branch></code>
git checkout main	Allows to switch to the main branch	<code>git checkout main</code>
git clone	Allows to create a copy of the remote repository	<code>git clone <repository-url></code>
git commit	Allows you to take staged snapshots if changes and commit them to the project	<code>git commit -m "Your commit message here"</code>
git config --global user.email	Example 1: Sets a global email configuration for Git Example 2: Sets a global username configuration for Git	Example 1: <code>git config --global user.email "your.email@example.com"</code> Example 2: <code>git config --global user.name "Your Name"</code>
git daemon	Used to allow anonymous download from the repository	<code>git daemon --reuseaddr --verbose</code>
git diff	Helps others to review your code to identify and compare the changes	<code>git diff example.txt</code>
git fetch	Used to transfer the changes from the remote repo to your local repo	<code>git fetch <options> <remote name> <branch name></code>
git fetch upstream/master	Used to grab upstream branches	<code>git fetch upstream master:upstream-master</code>
git format-patch	Generates or prepares e-mail submission if you adopt Linux kernel-style public forum workflow	<code>git format-patch -n <number_of_commits></code>
git http-backend	Provides a server-side implementation of Git-over-HTTP, allowing both fetch and push services	<code>git clone --bare /path/to/repos/myrepo.git</code> <code>cd myrepo.git</code> <code>git update-server-info</code>
git init	Used to clone an existing repository	<code>git init <directory></code>
git instaweb	Allows to set up web front-end to Git repositories	<code>git instaweb -p 8080</code>
git log	Enables to browse previous changes to a project	<code>git log -p filename</code>

Package/Method	Description	Code Example
git merge	Used to merge changes in the active branch into another branch	<code>git merge feature_branch</code>
git merge upstream/master	Merges changes from the 'upstream/master' branch to the current branch	<code>git merge upstream/master</code>
git pull	Used to transfer the changes from the remote repo to your local repo, and merge them to a branch	<code>git pull origin main</code>
git pull downstream	Pulls changes from a downstream repository, specifically from the master branch of that repository	<code>git pull downstream main</code>
git pull upstream	Pulls changes from the "upstream" repository into the current branch	<code>git pull upstream main</code>
git push	Used to push all the committed changes into the repository	<code>git push origin your_branch_name</code>
git remote	A command to manage a set of tracked repositories	<code>git remote add upstream https://github.com/original/repo.git</code>
git remote add origin <URL>	Adds a remote repository named "origin" with the specified URL	<code>git remote add origin https://github.com/yourusername/your-repo.git</code>
git remote add upstream	Adds the original repository as a new remote repository labeled upstream	<code>git remote add upstream https://github.com/original/repo.git</code>
git remote rename	The git remote rename command is followed by the name of the remote repository(origin) you want to rename and the new name(upstream) you want to give it	<code>git remote rename origin new-origin</code>
git remote -v	Allows to view the remotes associated with the local repository	<code>git remote -v</code>
git request-pull	Example 1: Creates a summary of changes for your upstream to pull Example 2: Generates a summary of pending changes for an email request	Example 1: <code>git request-pull origin/main your-branch</code> Example 2: <code>git request-pull <base> <head> <repository></code>
git rerere	Reuses recorded resolution of previously resolved merge conflicts	<code>git rerere</code> <code>git rerere diff</code>
git reset	Undoes changes that were made to the files in your working directory	<code>git reset HEAD~1</code>
git revert	Used to undo botched commits	<code>git revert HEAD</code>
git send-email	Example 1: Sends your email submission without corruption by your MUA Example 2: Sends a collection of patches as emails	Example 1: <code>git send-email --to=recipient@example.com path/to/patchfile.patch</code> Example 2: <code>git send-email --to recipient@example.com patches/*.patch</code>
git-shell	Used as a restricted login shell for shared central repository users	<code>sudo usermod -s /usr/bin/git-shell gituser</code>

Package/Method	Description	Code Example
git status	Allows to see the state of your working directory and the staged snapshot of the changes	<code>git status</code>
git version	Displays the current Git version installed on your system	<code>git --version</code>
git web	Provides a web front-end to Git repositories	<code>git instaweb --port=8080</code>



Skills Network