

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	1. 1
		1. git add sample.md
git add .	Allows to move the changed files into the staging area on GitHub repositories	1. 1
		1. git add .
git am	Used to apply patches emailed to the repository	1. 1
		1. git am < patchfile.patch
git branch	Allows to create an isolated environment within the repository to make changes	1. 1
		1. git branch <new-branch>
git checkout	Allows to see and change existing branches	1. 1
		1. git checkout <existing-branch>
git checkout main	Allows to switch to the main branch	1. 1
		1. git checkout main
git clone	Allows to create a copy of the remote repository	1. 1
		1. git clone <repository-url>
git commit	Allows you to take staged snapshots if changes and commit them to the project	1. 1
		1. git commit -m "Your commit message here"
git config --global user.email	Example 1: Sets a global email configuration for Git	Example 1:
		1. 1
git config --global user.name	Example 2: Sets a global username configuration for Git	1. git config --global user.email "your.email@example.com"
		Example 2:
git daemon	Used to allow anonymous download from the repository	1. 1
		1. git config --global user.name "Your Name"
git diff	Helps others to review your code to identify and compare the changes	1. 1
		1. git diff example.txt
git fetch	Used to transfer the changes from the remote repo to your local repo	1. 1
		1. git fetch <options> <remote name> <branch name>
git fetch upstream/master	Used to grab upstream branches	1. 1
		1. git fetch upstream master:upstream-master
git format-patch	Generates or prepares e-mail submission if you adopt Linux kernel-style public forum workflow	1. 1
		1. git format-patch -n <number_of_commits>
git http-backend	Provides a server-side implementation of Git-over-HTTP, allowing both fetch and push services	1. 1
		2. 2
		3. 3
		1. git clone --bare /path/to/repos/myrepo.git
		2. cd myrepo.git
		3. git update-server-info

Package/Method	Description	Code Example
git init	Used to clone an existing repository	<div>Copied!</div> <div>1. 1</div>
		<div>1. git init <directory></div>
git instaweb	Allows to set up web front-end to Git repositories	<div>Copied!</div> <div>1. 1</div>
		<div>1. git instaweb -p 8080</div>
git log	Enables to browse previous changes to a project	<div>Copied!</div> <div>1. 1</div>
		<div>1. git log -p filename</div>
git merge	Used to merge changes in the active branch into another branch	<div>Copied!</div> <div>1. 1</div>
		<div>1. git merge feature_branch</div>
git merge upstream/master	Merges changes from the 'upstream/master' branch to the current branch	<div>Copied!</div> <div>1. 1</div>
		<div>1. git merge upstream/master</div>
git pull	Used to transfer the changes from the remote repo to your local repo, and merge them to a branch	<div>Copied!</div> <div>1. 1</div>
		<div>1. git pull origin main</div>
git pull downstream	Pulls changes from a downstream repository, specifically from the master branch of that repository	<div>Copied!</div> <div>1. 1</div>
		<div>1. git pull downstream main</div>
git pull upstream	Pulls changes from the "upstream" repository into the current branch	<div>Copied!</div> <div>1. 1</div>
		<div>1. git pull upstream main</div>
git push	Used to push all the committed changes into the repository	<div>Copied!</div> <div>1. 1</div>
		<div>1. git push origin your_branch_name</div>
git remote	A command to manage a set of tracked repositories	<div>Copied!</div> <div>1. 1</div>
		<div>1. git remote add upstream https://github.com/original/repo.git</div>
git remote add origin <URL>	Adds a remote repository named "origin" with the specified URL	<div>Copied!</div> <div>1. 1</div>
		<div>1. git remote add origin https://github.com/yourusername/your-repo.git</div>
git remote add upstream	Adds the original repository as a new remote repository labeled upstream	<div>Copied!</div> <div>1. 1</div>
		<div>1. git remote add upstream https://github.com/original/repo.git</div>
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	<div>Copied!</div> <div>1. 1</div>
		<div>1. git remote rename origin new-origin</div>
git remote -v	Allows to view the remotes associated with the local repository	<div>Copied!</div> <div>1. 1</div>
		<div>1. git remote -v</div>
git request-pull	Example 1: Creates a summary of changes for your upstream to pull	<div>Example 1:</div> <div>1. 1</div>
		<div>1. git request-pull origin/main your-branch</div>
	Example 2: Generates a summary of pending changes for an email request	<div>Copied!</div> <div>1. 1</div>
		<div>1. git request-pull <base> <head> <repository></div>
git rerere	Reuses recorded resolution of previously resolved merge conflicts	<div>Copied!</div> <div>1. 1</div>
		<div>2. git rerere diff</div>

Package/Method	Description	Code Example
git reset	Undoes changes that were made to the files in your working directory	1. 1
		1. git reset HEAD~1
git revert	Used to undo botched commits	1. 1
		1. git revert HEAD
git send-email	Example 1: Sends your email submission without corruption by your MUA Example 2: Sends a collection of patches as emails	1. 1
		1. git send-email --to=recipient@example.com
		2. path/to/patchfile.patch
		Example 1:
git shell	Used as a restricted login shell for shared central repository users	1. 1
		1. git send-email --to recipient@example.com
		2. patches/*.patch
		Example 2:
git status	Allows to see the state of your working directory and the staged snapshot of the changes	1. 1
		1. git status
git version	Displays the current Git version installed on your system	1. 1
		1. git --version
git web	Provides a web front-end to Git repositories	1. 1
		1. git instaweb --port=8080

