**Git Commands Guide**

**COMMITS & STAGING**

* **git add <file name> 📂  
  Move the specified file to the staging area.**
* **git add . 📂  
  Move all files to the staging area.**
* **git commit -m "MSG" 💾  
  Save changes with a commit message.**
* **git init 🆕  
  Initialize a new Git repository.**

**LOG & STATUS**

* **git status 📊  
  Show the current status of the files in the working directory.**
* **git log 📜  
  Show the commit log of the current repository or branch.**

**Removing Commits**

* **git reset (--hard || --soft) HEAD~N 🔄  
  Remove the last N commits.**
  + **--hard: Discard changes in the working directory.**
  + **--soft: Keep changes in the working directory.**
* **git reset <commit ID> ⏪  
  Revert to a specific commit if you deleted a commit by mistake.**
* **git reflog 🕒  
  Show the history of all actions, including commits that were reset.**
* **git reset --hard <commit ID> 🔄  
  Reset to a specific commit using its ID.**

**Checkout**

* **git checkout <commit ID> ⏩  
  Move to the state of the project at the specified commit.**
* **git checkout <branch name> 🌿  
  Switch to the specified branch.**

**Branching**

**Types of Branches**

* **Local branch: 🏠 Exists in your local repository.**
* **Remote branch: 🌐 Exists in the remote repository.**
* **Remote tracking branch: 🔄 A copy of the remote branch used for merging.**
* **Local tracking branch: 📡 Local branches that track remote branches because of the -u setup stream.**

**Branch Commands**

* **git branch 🌿  
  Show all local branches.**
* **git branch <branch name> 🌿  
  Create a new branch.**
* **git checkout <branch name> 🌿  
  Switch to the specified branch.**
* **git checkout -b <branch name> 🌿  
  Create and switch to a new branch in one step.**
* **git switch <branch name> 🔄  
  Switch to another branch.**
* **git switch -c <branch name> 🔄  
  Create and switch to a new branch in one step.**

**Show Branches**

* **git branch 🌿  
  Show local branches only.**
* **git branch -a 🌿  
  Show all branches (local and remote).**
* **git branch -r 🌿  
  Show remote branches only.**

**Publish Branch**

* **git push origin <branch name> 🌐  
  Push a branch to the remote repository.**

**Fetch Updates**

* **git fetch 🔄  
  Fetch updates from the remote repository.**
* **git fetch origin 🔄  
  Fetch updates from the remote repository named origin.**

**Delete Branch**

* **git branch -d <branch name> 🗑️  
  Try to delete a specified branch (dry run).**
* **git branch -D <branch name> 🗑️  
  Force delete a specified branch.**
* **git push origin --delete <branch name> 🗑️  
  Delete a branch from the remote repository.**

**Restore Deleted Branch**

* **git checkout <commit ID> ⏩  
  Move to the last commit of the deleted branch.**
* **git reflog 🕒  
  Get the commit ID of the deleted branch.**
* **git switch -c <branch name> 🔄  
  Create a new branch from the commit ID.**

**Rename Branch**

* **git branch -M <new branch name> ✏️  
  Rename a branch.**

**Remote Repository**

* **git remote add origin <repo link> 🌐  
  Link the local repository to a remote repository.**
* **git push -u origin main 🌐  
  Push changes to the remote repository and set up tracking.**

**Rebase**

* **git switch <branch name> 🔄  
  Switch to the branch you want to rebase.**
* **git rebase <source branch> 🔄  
  Rebase the current branch onto the source branch.**

**Cherry-pick**

* **git cherry-pick <commit ID> 🍒  
  Apply changes from a specific commit to the current branch.**

**Miscellaneous**

* **git ls-files 📂  
  Show tracked files that have been moved to the staging area.**
* **git status 📊  
  Show the current state of tracked and untracked files.**

**Corrections**

1. **git reset <commit ID> ⏪**
   * **Correction: This command is used to move the HEAD to a specific commit, not to restore a deleted commit. To restore a deleted commit, use git reflog to find the commit ID and then git checkout <commit ID>.**
2. **git branch -d <branch name> 🗑️**
   * **Correction: This command deletes a branch only if it has been merged. If the branch has unmerged changes, use git branch -D <branch name>.**
3. **git push origin --delete <branch name> 🗑️**
   * **Correction: This command deletes a branch from the remote repository. Ensure you have deleted it locally first using git branch -D <branch name>.**
4. **git rebase <source branch> 🔄**
   * **Correction: Ensure you are on the branch you want to rebase before running this command.**
5. **git cherry-pick <commit ID> 🍒**
   * **Correction: This command applies the changes from a specific commit to the current branch. Ensure you are on the correct branch before running this command.**

**Merging**

**Before you merge, make sure you are on the branch where you want to merge changes. For example, if you want to merge a branch into the main branch, you need to be on the main branch and run the merge command there.**

* **git merge <source branch> 🔄  
  Merge changes from the source branch into the current branch.**

**Removing Untracked Files**

* **git clean -dn 🧹  
  Try to remove untracked files or directories (dry run).**
* **git clean -df 🧹  
  Delete untracked files or directories.**

**Types of Merge**

* **Fast-forward merge ⏩  
  Happens when you merge a branch into the main branch and there have been no changes in the main branch since the branch was created.**

**Discarding File Changes**

* **git restore <file name> 🗑️  
  Discard changes in the specified file.**
* **git restore . 🗑️  
  Discard all changes in all files.**
* **git checkout <file name> ⏪  
  Discard unstaged updates in the file and revert to the last commit state.**
* **git rm <file name> 🗑️  
  Delete the specified file, then use git commit to save changes.**
* **git restore --staged <file name> 🗑️  
  Unstage the file from the staging area.**
* **git reset <file name> 🗑️  
  Remove the file from the staging area.**

**Resetting Commits**

* **git reset HEAD~N 🔄  
  Remove N number of commits from the top.**
* **git reset --soft HEAD~N 🔄  
  Remove commits from the working directory only, keeping changes in the staging area.**
* **git reset --hard HEAD~N 🔄  
  Remove commits from both the working directory and staging area, discarding modified changes.**

**Stashing**

* **git stash 📦  
  Temporarily save changes.**
* **git stash apply 📦  
  Restore saved changes.**
* **git stash list 📋  
  Show a list of saved changes.**
* **git stash apply <stash ID> 📦  
  Restore a specific saved change using its ID.**
* **git stash drop <stash ID> 🗑️  
  Delete a specific saved change from the list.**
* **git stash clear 🗑️  
  Delete all saved changes.**
* **git stash push -m "message" 📦  
  Save changes with a message.**
* **git stash pop 📦  
  Restore and delete the most recent saved change.**

**Upstream**

* **git push --set-upstream origin main 🌐  
  Set up a channel between the local and remote repository.**
* **git push -u origin main 🌐  
  Set up a channel between the local and remote repository in a shorter form.**

**Pulling Changes**

* **git pull 🔄  
  Combines git fetch and git merge to fetch updates from the remote repository and merge them into the local repository.**

**GitHub**

* **git clone <repo link> 📥  
  Clone a repository from the remote repository.**