#_ Common Git Use Cases: [+50] Challenges

📁 Use Case 1 - Cloning a Repository

Aim: You want to clone a repository to your local machine.
Solution: Use git clone [repo_url] to clone the repository.

🔄 Use Case 2 - Pulling Changes from Remote

Aim: You want to pull changes from the remote repository.
Solution: Use git pull to pull the changes.

🚀 Use Case 3 - Pushing Changes to Remote

Aim: You've made changes and want to push them to the remote repository.
Solution: First, use git add . to stage all changes. Then, use git commit -m "[message]" to commit the changes. Finally, use git push to push the changes.

🔄 Use Case 4 - Syncing a Fork with Original Repo

Aim: You've forked a repo and want to sync it with the original repository.
Solution: Use git remote add upstream [original_repo_url], then git fetch upstream, then git merge upstream/main or git rebase upstream/main, and finally git push.

🌿 Use Case 5 - Creating a New Branch

Aim: You want to create a new branch.
Solution: Use git branch [new_branch_name] to create a new branch.

🔀 Use Case 6 - Switching Branches

Aim: You want to switch to a different branch.
Solution: Use git checkout [branch_name] to switch to the desired branch.

By: Waleed Mousa

📤 Use Case 7 - Staging Changes

Aim: You've made changes **and** want **to** stage them.
Solution: **Use** git add [**file**] **to** stage a specific **file** **or** git add . **to** stage **all** changes.

💾 Use Case 8 - Committing Changes

Aim: You've staged changes and want to **commit** them.
Solution: **Use** git **commit** -m "[commit_message]" **to commit** the changes.

↩️ Use Case 9 - Undoing a Commit

Aim: You've made a **commit and** want **to undo** it.
Solution: **Use** git **reset HEAD**~1 **to undo** the **last commit**, keeping the changes **in** your working directory.

💔 Use Case 10 - Reverting a Commit

Aim: You want to revert changes introduced **by a** specific commit.
Solution: Use git revert [commit_hash] to revert **the** changes introduced **by the** commit.

🔄 Use Case 11 - Updating Your Branch with the Latest Main Branch Changes

Aim: You want to update your branch **with the** latest changes from **the** main branch.
Solution: First, **switch** to **the** main branch **with** git checkout main. Then, pull **the** latest changes **with** git pull. Finally, **switch** back to your branch **with** git checkout [your_branch] **and** merge **the** main branch changes **with** git merge main.

🔄 Use Case 12 - Rebasing Your Branch on Top of Another Branch

Aim: You want **to** rebase your branch **on** top **of** another branch.
Solution: Use git rebase [other_branch] **to** rebase your branch.

👀 Use Case 13 - Viewing the Commit History

Aim: You want **to view** the commit **history**.
Solution: Use git log **to view** the commit **history**.

🔍 Use Case 14 - Searching the Commit History

Aim: You want to find commits that added **or** removed **a** specific piece **of text**.
Solution: Use git log -S"[text]" to find **the** commits.

💥 Use Case 15 - Resolving Merge Conflicts

Aim: You have merge conflicts **and** want to resolve them.
Solution: Use git mergetool to resolve **the** conflicts **with a** graphical interface.

📝 Use Case 16 - Amending the Last Commit

Aim: You want to **change** the **last** commit.
Solution: **Use** git **commit** --amend to amend the last commit.

🔖 Use Case 17 - Creating a Tag

Aim: You want to create **a** tag **for a** specific commit.
Solution: Use git tag [tag_name] [commit_hash] to create **a** tag.

🚀 Use Case 18 - Pushing a Tag to the Remote

Aim: You've created a tag **and** want **to** push it **to** the remote repository.
Solution: **Use** git push origin [tag_name] **to** push the tag.

📚 Use Case 19 - Stashing Changes

Aim: You have changes that you're not ready to **commit** yet **and** want **to save** them **for** later.
Solution: **Use** git stash **to** stash the changes **and** git stash pop **to apply** the stashed changes.

🍒 Use Case 20 - Cherry-Picking a Commit

Aim: You want to apply the changes from a specific commit without merging the entire branch.
Solution: Use git cherry-pick [commit_hash] to apply the changes.

🔎 Use Case 21 - Searching the Git History

Aim: You want to search the Git history for a specific term.
Solution: Use git grep [term] $(git rev-list --all) to search the entire Git history.

📝 Use Case 22 - Editing an Older or Multiple Commits

Aim: You want to edit an older commit or multiple commits.
Solution: Use git rebase -i HEAD~[number_of_commits] to start an interactive rebase.

💾 Use Case 23 - Saving Uncommitted Changes without Stashing

Aim: You have uncommitted changes that you want to save but you don't want to use stash.
Solution: Use git diff > [patch_name].patch to save the changes and git apply [patch_name].patch to apply the saved changes.

💼 Use Case 24 - Ignoring Files

Aim: You want to ignore specific files or directories.
Solution: Add the files or directories to a .gitignore file in your repository root.

🗃️ Use Case 25 - Removing a File from the Repository

Aim: You want to remove a file from the repository.
Solution: Use git rm [file] to remove the file and then commit the change.

By: Waleed Mousa

📤 Use Case 26 - Removing a File from Git Without Deleting It

Aim: You want to remove **a** file from Git but **not** delete **it** from your local file **system**.
Solution: Use git rm --cached [file] to remove the file from Git.

🔄 Use Case 27 - Changing the Branch Base

Aim: You've branched off from one branch **and** want **to** change the base **to** another branch.
Solution: **Use** git rebase --onto [new_base] [old_base] to change the branch base.

🔄 Use Case 28 - Merging Development Branch to Main Branch

Aim: You've finished development in a branch and want to **merge** it **to** the **main** branch.
Solution: **Switch to** the **main** branch **using** git checkout **main**, **then use** git **merge** [development_branch] **to merge** the changes.

🔄 Use Case 29 - Squashing Commits Using Rebase

Aim: You have several commits **and** want to squash them **into** one.
Solution: Use git rebase -i HEAD~[number_of_commits] to start **an** interactive rebase **and** squash **the** commits.

🗑️ Use Case 30 - Deleting Untracked Files

Aim: You have untracked files **in** your Git repository that you want to delete.
Solution: Use git clean -f to remove untracked files.

📦 Use Case 31 - Checking out a Remote Branch

Aim: You want to checkout **a** branch from **a** remote repository.
Solution: Use git fetch, **then** git checkout [branch_name] to checkout **the** remote branch.

👀 Use Case 32 - Seeing Changes on a File

Aim: You want to see the changes made on a specific file.
Solution: Use git diff [file] to see the changes.

📈 Use Case 33 - Seeing Who Changed a File

Aim: You want to see who made changes to a specific file.
Solution: Use git blame [file] to see who changed the file.

🔗 Use Case 34 - Seeing Changes Between Two Commits

Aim: You want to see the changes between two commits.
Solution: Use git diff [first_commit]..[second_commit] to see the changes.

🔄 Use Case 35 - Reverting to a Previous Commit

Aim: You want to revert to a previous commit.
Solution: Use git checkout [commit_hash] to revert to the previous commit.

🔀 Use Case 36 - Resetting to a Previous Commit and Discarding All Changes

Aim: You want to reset to a previous commit and discard all changes.
Solution: Use git reset --hard [commit_hash] to reset to the commit and discard all changes.

⬆️ Use Case 37 - Pushing a Branch to the Remote

Aim: You've made changes in a branch and want to push it to the remote.
Solution: Use git push -u origin [branch_name] to push the branch to the remote.

🌿 Use Case 38 - Creating and Switching to a New Branch

Aim: You want to create a new branch and switch to it.
Solution: Use git checkout -b [new_branch_name] to create and switch to a new branch.

🔄 Use Case 39 - Fetching the Latest Commits

Aim: You want to fetch the latest commits without merging them.
Solution: Use git fetch to fetch the latest commits.

🔄 Use Case 40 - Reverting Uncommitted Changes to a File

Aim: You've made changes to a file and want to revert them.
Solution: Use git checkout -- [file] to revert the changes.

🔄 Use Case 41 - Updating the Local Repository with Changes from the Remote

Aim: You want to update your local repository with the latest changes from the remote.
Solution: Use git pull to pull the latest changes from the remote.

📝 Use Case 42 - Changing the Last Commit Message

Aim: You've made a mistake in the last commit message and want to change it.
Solution: Use git commit --amend -m "New commit message" to change the last commit message.

💼 Use Case 43 - Checking the Status of the Repository

Aim: You want to check the status of the repository.
Solution: Use git status to see the status of the repository.

💼 Use Case 44 - Staging and Committing Changes in One Command

Aim: You've made changes to tracked files and want to stage and commit them in one command.
Solution: Use git commit -am "Commit message" to stage and commit the changes.

🔄 Use Case 45 - Reapplying Commits on Top of Another Branch

Aim: You've made commits in one branch and want to reapply them on top of another branch.
Solution: Use git rebase [other_branch] to reapply the commits.

🚀 Use Case 46 - Pushing All Local Branches to Remote

Aim: You want to push all your local branches to the remote repository.
Solution: Use git push --all origin to push all branches to the remote repository.

🔀 Use Case 47 - Merging Changes from Another Branch

Aim: You have changes in another branch that you want to merge into your current branch.
Solution: Use git merge [other_branch] to merge the changes.

👀 Use Case 48 - Viewing Changes Between Two Branches

Aim: You want to view the differences between two branches.
Solution: Use git diff [branch1]..[branch2] to view the differences.

🔀 Use Case 49 - Swapping to Previous Branch

Aim: You want to switch back to the branch you were on before the current one.
Solution: Use git checkout - to switch to the previous branch.

📝 Use Case 50 - Amending Author of the Last Commit

Aim: You've committed **with** the wrong author information **and** want **to** correct it.
Solution: **Use** git commit --amend --author="Author Name <email@address.com>" to amend the author of the last commit.

🚀 Use Case 51 - Force Pushing to Remote Branch

Aim: You've made changes **to** your local branch **and** want **to force** push **to** the remote branch.
Solution: **Use** git push origin [branch_name] --force to force push the changes.

🚫 Use Case 52 - Removing All Local Branches Except Current

Aim: You want to delete all local branches except **the** current one.
Solution: Use git branch | grep -v "*" | xargs git branch -D to remove all local branches except **the** current one.

🌿 Use Case 53 - Creating a New Branch from a Specific Commit

Aim: You want to create **a** new branch starting from **a** specific commit.
Solution: Use git checkout -b [new_branch] [commit_hash] to create **the** new branch.

🔄 Use Case 54 - Changing a Commit Message in History

Aim: You have an older **commit with** a wrong message that you want **to** correct.
Solution: **Use** git rebase -i [commit_hash]^ **to start** an interactive rebase, **then replace** "pick" **with** "reword" **for** the **commit** you want **to** change. **Save and exit**, **then update** the **commit** message.