**You said:**

all git cmd defination in one line from basic to advance that is used in real world from basic to advance - just simple defiantion what they do in one or 2 lines just

**ChatGPT said:**

Here’s a quick reference for common Git commands from basic to advanced:

1. **git init** - Initializes a new Git repository.
2. **git clone [url]** - Clones a remote repository to your local machine.
3. **git status** - Displays the current status of your working directory.
4. **git add [file]** - Stages changes (adds them to the next commit).
5. **git commit -m "[message]"** - Commits staged changes with a message.
6. **git log** - Shows the commit history.
7. **git diff** - Displays the differences between the working directory and the index.
8. **git pull** - Fetches and merges changes from a remote repository.
9. **git push** - Pushes local commits to a remote repository.
10. **git branch** - Lists, creates, or deletes branches.
11. **git checkout [branch]** - Switches to the specified branch.
12. **git merge [branch]** - Merges changes from the specified branch into the current branch.
13. **git remote add [name] [url]** - Adds a new remote repository.
14. **git fetch** - Fetches changes from the remote repository but doesn’t merge them.
15. **git reset [file]** - Unstages a file (but doesn’t delete changes).
16. **git reset --hard [commit]** - Resets the current branch to a specific commit and discards changes.
17. **git rebase [branch]** - Applies commits from the specified branch onto the current branch.
18. **git stash** - Temporarily stores changes that are not yet committed.
19. **git stash pop** - Applies stashed changes back into the working directory.
20. **git cherry-pick [commit]** - Applies a specific commit from another branch.
21. **git tag [tagname]** - Creates a tag for a specific commit.
22. **git rebase -i [commit]** - Interactively rebases a series of commits.
23. **git reflog** - Shows the history of HEAD (helpful to recover lost commits).
24. **git bisect** - Helps to find the commit that introduced a bug by binary search.
25. **git revert [commit]** - Reverts a commit by creating a new commit that undoes its changes.

4o

**You said:**

continue please

**ChatGPT said:**

Sure, here are more advanced Git commands:

1. **git config --global user.name "[name]"** - Sets the global username for commits.
2. **git config --global user.email "[email]"** - Sets the global email for commits.
3. **git config --list** - Lists all the Git configuration settings.
4. **git rm [file]** - Removes a file from the working directory and stages the removal.
5. **git mv [old-filename] [new-filename]** - Renames a file and stages the change.
6. **git show [commit]** - Displays detailed information about a specific commit.
7. **git log --oneline** - Displays a simplified log with each commit on one line.
8. **git log --graph** - Displays a visual representation of the commit history.
9. **git log --author="[name]"** - Filters commits by a specific author.
10. **git log --since="2 weeks ago"** - Shows commits from a specified time range.
11. **git merge --no-ff [branch]** - Merges a branch and forces a merge commit even if the merge is a fast-forward.
12. **git pull --rebase** - Fetches and rebases instead of merging.
13. **git push --force** - Forces the push to overwrite remote changes.
14. **git push --tags** - Pushes all local tags to the remote repository.
15. **git submodule add [url] [path]** - Adds a submodule to the repository.
16. **git submodule update** - Updates the submodules to match the current commit.
17. **git ls-files** - Lists all tracked files in the repository.
18. **git clean -fd** - Removes untracked files and directories from the working directory.
19. **git reflog expire --expire=now --all** - Cleans up reflog entries that are no longer needed.
20. **git reset --soft [commit]** - Resets the HEAD to a commit but keeps changes staged.
21. **git reset --mixed [commit]** - Resets the HEAD to a commit and unstages the changes.
22. **git diff --cached** - Shows the differences between staged changes and the last commit.
23. **git bundle create [filename] [ref]** - Creates a bundle of repository data that can be transferred or backed up.
24. **git archive [commit]** - Creates a zip or tar archive from a specific commit.
25. **git remote show [remote]** - Shows detailed information about a remote repository.
26. **git revert --no-commit [commit]** - Reverts a commit without committing immediately.
27. **git clean -n** - Shows which files would be removed by git clean -fd without actually removing them.

These comma