TOPIC : GIT COMMANDS

NAME : MULAGADA DILEEP KUMAR

1. Git Configuration

1. git config --global user.name "Your Name" - Sets the user name for commits.
2. git config --global user.email "your.email@example.com" - Sets the user email.
3. git config --global core.editor "vim" - Sets the default editor.
4. git config --list - Displays all Git configurations.
5. git config --global alias.co checkout - Creates an alias for a Git command.

2. Git Setup & Initialization

6. git init - Initializes a new Git repository.
7. git clone <repository_url> - Clones an existing repository.

3. Working with Branches

8. git branch - Lists all branches.
9. git branch <branch_name> - Creates a new branch.
10. git checkout <branch_name> - Switches to another branch.
11. git checkout -b <branch_name> - Creates and switches to a new branch.
12. git merge <branch_name> - Merges another branch into the current one.
13. git branch -d <branch_name> - Deletes a branch.
14. git branch -D <branch_name> - Force deletes a branch.
15. git switch <branch_name> - Switches branches (alternative to checkout).

4. Staging & Committing Changes

16. git status - Shows the status of changes.
17. git add <file> - Stages a file for commit.
18. git add . - Stages all files for commit.
19. git reset <file> - Unstages a file.
20. git commit -m "Commit message" - Commits staged changes.
21. git commit -a -m "Commit message" - Stages and commits all tracked files.

22. git commit --amend -m "Updated commit message" - Modifies the last commit message.

5. Viewing Commit History

23. git log - Displays commit history.
24. git log --oneline - Displays a compact commit history.
25. git log --graph - Shows a graphical representation of commits.
26. git log --author="Author Name" - Shows commits by a specific author.
27. git show <commit_hash> - Displays details of a specific commit.

6. Undoing Changes

28. git checkout -- <file> - Discards changes in a file.
29. git reset --hard - Resets the working directory to the last commit.
30. git reset <commit_hash> - Resets the repository to a specific commit.
31. git revert <commit_hash> - Reverts a specific commit.

7. Synchronizing with Remote Repositories

32. git remote -v - Lists remote repositories.
33. git remote add origin <url> - Adds a new remote repository.
34. git fetch origin - Fetches changes from a remote repository.
35. git pull origin <branch> - Fetches and merges changes from a remote branch.
36. git push origin <branch> - Pushes local changes to a remote branch.
37. git push -u origin <branch> - Pushes a branch and sets upstream tracking.
38. git push --force - Forces a push, overwriting changes.

8. Working with Tags

39. git tag - Lists all tags.
40. git tag <tag_name> - Creates a new tag.
41. git tag -a <tag_name> -m "Tag message" - Creates an annotated tag.
42. git push origin <tag_name> - Pushes a specific tag to a remote repository.
43. git push origin --tags - Pushes all local tags to the remote repository.
44. git tag -d <tag_name> - Deletes a local tag.

45. git push origin --delete <tag_name> - Deletes a remote tag.

9. Stashing Changes

46. git stash - Saves changes temporarily without committing.
47. git stash list - Lists all stashes.
48. git stash pop - Applies the latest stash and removes it from the stash list.
49. git stash apply - Applies a stash but keeps it in the list.
50. git stash drop - Deletes a stash.

10. Git Bisect & Blame

51. git bisect start - Starts a bisect session.
52. git bisect bad - Marks the current commit as bad.
53. git bisect good <commit> - Marks a commit as good.
54. git blame <file> - Shows who last modified each line of a file.

11. Working with Submodules

55. git submodule add <repository_url> - Adds a submodule.
56. git submodule update --init --recursive - Initializes and updates submodules.
57. git submodule foreach git pull origin master - Updates all submodules.

12. Git Clean & Garbage Collection

58. git clean -f - Removes untracked files.
59. git gc - Runs garbage collection to optimize repository.
60. git fsck - Checks for repository corruption.