

Git Commands with Syntax

1.git init

Syntax: `git init`

Initialize a new Git repository.

2.git clone [url]

Syntax: `git clone <repository_url>`

Clone an existing repository from a remote URL.

3.git status

Syntax: `git status`

Show the status of changes as untracked, modified, or staged.

4.git add [file]

Syntax: `git add <file_name>`

Add a file to the staging area.

5.git add .

Syntax: `git add .`

Add all files to the staging area.

6.git commit -m "message"

Syntax: `git commit -m "commit message"`

Commit changes with a message.

7.git commit -am "message"

Syntax: `git commit -am "commit message"`

Add and commit changes in one command.

8.git log

Syntax: `git log`

Show commit history.

9.git log --oneline

Syntax: `git log --oneline`

Show commit history in a compact format.

10.git log --graph

Syntax: `git log --graph`

Show a graphical representation of branches.

11.git diff

Syntax: `git diff`

Show changes between commits and working directory.

12.git diff --staged

Syntax: `git diff --staged`

Show changes between staging and last commit.

13.git branch

Syntax: `git branch`

List all branches.

14.git branch [branch-name]

Syntax: `git branch <branch_name>`

Create a new branch.

15.git checkout [branch-name]

Syntax: `git checkout <branch_name>`

Switch to a branch.

16.git checkout -b [branch-name]

Syntax: `git checkout -b <branch_name>`

Create and switch to a new branch.

17.git merge [branch-name]

Syntax: `git merge <branch_name>`

Merge a branch into the current branch.

18.git rebase [branch-name]

Syntax: `git rebase <branch_name>`

Reapply commits on top of another base branch.

19.git stash

Syntax: `git stash`

Temporarily save uncommitted changes.

20.git stash pop

Syntax: `git stash pop`

Apply stashed changes and remove them from stash.

21.git stash list

Syntax: `git stash list`

Show list of stashed changes.

22.git stash apply

Syntax: `git stash apply`

Apply stashed changes without removing them.

23.git reset [file]

Syntax: `git reset <file_name>`

Unstage a file.

24.git reset --hard

Syntax: `git reset --hard`

Reset working directory and staging area.

25.git reset --soft HEAD~1

Syntax: `git reset --soft HEAD~1`

Undo last commit, keeping changes staged.

26.git reset --hard HEAD~1

Syntax: `git reset --hard HEAD~1`

Undo last commit and discard changes.

27.git rm [file]

Syntax: `git rm <file_name>`

Remove a file from working directory and staging area.

28.git mv [old-file] [new-file]

Syntax: `git mv <old_file> <new_file>`

Rename or move a file.

29.git remote -v

Syntax: `git remote -v`

Show remote repositories.

30.git remote add [name] [url]

Syntax: `git remote add <name> <url>`

Add a remote repository.

31.git fetch [remote]

Syntax: `git fetch <remote>`

Fetch changes from a remote repository.

32.git pull [remote] [branch]

Syntax: `git pull <remote> <branch>`

Fetch and merge changes from remote.

33.git push [remote] [branch]

Syntax: `git push <remote> <branch>`

Push changes to a remote repository.

34.git push -u origin [branch]

Syntax: `git push -u origin <branch>`

Push and set upstream tracking.

35.git push --force

Syntax: `git push --force`

Force push changes.

36.git tag [tag-name]

Syntax: `git tag <tag_name>`

Create a tag.

37.git tag

Syntax: git tag

List all tags.

38.git show [tag-name]

Syntax: git show <tag_name>

Show tag details.

39.git cherry-pick [commit-hash]

Syntax: git cherry-pick <commit_hash>

Apply a specific commit.

40.git revert [commit-hash]

Syntax: git revert <commit_hash>

Revert a specific commit.

41.git blame [file]

Syntax: git blame <file_name>

Show commit history for each line in a file.

42.git grep "text"

Syntax: git grep "search_text"

Search for a specific text in repository.

43.git config --list

Syntax: git config --list

Show all configurations.

44.git reflog

Syntax: git reflog

Show reference logs.