

Git Commands Guide

1. git clone

Used to create a copy of a remote repository on your local machine.

Syntax:

```
git clone <repository-url>
```

Example:

```
git clone https://github.com/user/repository.git
```

2. git add .

Adds all changes in the working directory (tracked or untracked files) to the staging area.

Syntax:

```
git add .
```

Example:

```
git add .
```

3. git status

Shows the current status of the working directory and staging area.

Syntax:

```
git status
```

Example:

```
git status
```

4. git commit

Records changes from the staging area to the local repository.

Git Commands Guide

Syntax:

```
git commit -m "<commit-message>"
```

Example:

```
git commit -m "Added new feature"
```

5. git push origin

Uploads local commits to the remote repository.

Syntax:

```
git push origin <branch-name>
```

Example:

```
git push origin main
```

6. git fetch

Downloads new data from the remote repository without merging it.

Syntax:

```
git fetch
```

Example:

```
git fetch origin
```

7. git pull

Fetches and merges changes from the remote repository into your current branch.

Syntax:

```
git pull <remote> <branch>
```

Git Commands Guide

Example:

```
git pull origin main
```

8. git branch

Lists, creates, or deletes branches.

Syntax:

```
git branch
```

```
git branch <branch-name>
```

```
git branch -d <branch-name>
```

Example:

```
git branch feature-1
```

9. git checkout <branch-name>

Switches to the specified branch or commit.

Syntax:

```
git checkout <branch-name>
```

Example:

```
git checkout feature-1
```

10. git push origin --delete <branch-name>

Deletes a branch from the remote repository.

Syntax:

```
git push origin --delete <branch-name>
```

Git Commands Guide

Example:

```
git push origin --delete feature-1
```

11. git rebase and git merge

git merge: Combines changes from another branch into the current branch.

git rebase: Moves the base of your branch to the latest commit of another branch.

Syntax:

```
git merge <branch-name>
```

```
git rebase <branch-name>
```

Example:

```
git merge feature-1
```

```
git rebase main
```

12. git log --oneline

Displays the commit history in a compact format.

Syntax:

```
git log --oneline
```

Example:

```
git log --oneline
```

13. git stash pop

Restores the most recent stashed changes and removes it from the stash list.

Syntax:

Git Commands Guide

```
git stash pop
```

Example:

```
git stash pop
```

14. git cherry-pick

Applies a specific commit from one branch into your current branch.

Syntax:

```
git cherry-pick <commit-hash>
```

Example:

```
git cherry-pick alb2c3d
```

15. git reset and git revert

git reset: Moves the branch back to a previous state.

git revert: Creates a new commit that undoes a previous commit.

Syntax:

```
git reset --hard <commit-hash>
```

```
git revert <commit-hash>
```

Example:

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

```
git revert alb2c3d
```

16. Branching Strategies

1. Git Flow: Feature branches created from 'develop' and merged back after completion.

Git Commands Guide

2. GitHub Flow: Main branch is always deployable; feature branches merged via pull requests.
3. Trunk-Based Development: Frequent code integrations into 'main'.
4. Release Branching: Maintains release branches for long-term projects.