Git Complete Guide for Beginners

1. What is Git?

Git is a distributed version control system used to track changes in source code during software development. It helps multiple developers work on the same project without interfering with each other.

2. Why Use Git?

- Tracks your code changes
- Allows collaboration
- Keeps a history of every version
- Allows branching and merging for feature isolation

3. Git Setup

Before using Git, you need to configure your identity.

Command:

```
git config --global user.name "Your Name"
```

```
git config --global user.email "you@example.com"
```

4. git init

Initializes a new Git repository in your project folder.

Command:

```
git init
```

5. Adding Files and Staging

To start tracking files:

```
git add filename.txt
```

To add all files:

```
git add -A
```

6. What is Staging and Commit?

Staging is selecting what will be committed. Commit is saving the snapshot of staged changes.

7. git commit

Saves the staged changes with a message.

```
git commit -m "Initial commit"
```

8. git status

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

9. git log

Shows the history of commits.

```
git log
```

```
git log -p
```

10. git diff

Shows the differences between files.

```
git diff
```

11. git checkout

Switches branches or restores files.

```
git checkout branch-name
```

12. Removing Files

Remove from working directory and staging:

```
git rm filename.txt
```

Remove from staging only:

```
git rm --cached filename.txt
```

13. .gitignore

A file listing patterns for files to ignore from tracking.

14. Branching and Merging

To create a branch:

```
git branch feature
```

Switch to branch:

```
git checkout feature
```

Merge into main:

```
git checkout main
```

```
git merge feature
```

15. Remote Repositories

Connect to GitHub or remote repo.

```
git remote add origin https://github.com/user/repo.git
```

16. git clone

To copy a remote repo:

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

17. git push

Sends your local commits to remote repository.

```
git push origin main
```

18. git pull and git fetch

Updates local repo with remote changes.

```
git pull
```

```
git fetch
```

19. SSH vs HTTPS

SSH is secure and password-less once setup. HTTPS needs GitHub login.

20. Other Important Commands

Undo a commit:

```
git revert commit_id
```

View all history including deleted commits:

```
git reflog
```

21. Mini Project: Portfolio Website with Git

Let's walk through a small real-world example to practice all the Git commands you've learned.

Project Goal

We will create a simple project called 'portfolio-site' and use Git to track, branch, merge, and push the project to GitHub.

Step-by-Step Commands

1. Create a Project Folder

```
mkdir portfolio-site
cd portfolio-site
```

2. Initialize Git

```
git init
```

3. Configure Git (if not done)

```
git config --global user.name "Your Name"
git config --global user.email "you@example.com"
```

4. Create First File

```
echo "<h1>Welcome to My Portfolio</h1>" > index.html
```

5. Stage and Commit

```
git add index.html
git commit -m "Initial commit: add homepage"
```

6. Create a GitHub Repo

```
Go to GitHub and create a new repository named 'portfolio-site'.
```

7. Connect Remote and Push

```
git remote add origin https://github.com/username/portfolio-site.git
git branch -M main
git push -u origin main
```

8. Create a New Branch

```
git checkout -b add-about-page
```

9. Add About Page

```
echo "<h2>About Me</h2>" > about.html
git add about.html
git commit -m "Add About page"
```

10. Merge Branch to Main

```
git checkout main
git merge add-about-page
```

11. Push Merged Code

```
git push
```

12. Simulate Conflict

```
Edit index.html locally and on GitHub differently, then run:
git pull
```

13. Resolve Conflict

```
Manually edit conflict markers, then:
git add index.html
git commit -m "Resolved merge conflict"
```

14. Create .gitignore File

```
echo "*.log" > .gitignore
git add .gitignore
git commit -m "Add .gitignore to ignore log files"
```

15. Remove File from Git

```
git rm oldfile.txt
git commit -m "Removed oldfile.txt"
```

16. Remove File Only from Staging

```
git rm --cached notes.txt
```

17. View History

```
git log
git log -p
```

18. View Changes

```
git diff
```

19. Stash Changes Temporarily

```
git stash
git stash apply
```

20. Clone Project

```
git clone https://github.com/username/portfolio-site.git
```

This mini project covers all key Git operations: initialization, staging, committing, branching, merging, resolving conflicts, pushing to remote, ignoring files, and cloning.