Git Documentation: From Basics to Advanced

1. Introduction to Git

Git is a version control system (VCS) that lets you track changes to your code, collaborate with others, and manage your project history. It's widely used in software development and works locally on your machine. GitHub, GitLab, and Bitbucket are services that host Git repositories online.

2. Basic Git Setup

2.1 Installing Git

To install Git:

- **Windows**: Download from git-scm.com and follow the installation wizard.
- **Linux** (Ubuntu/Debian):

Command sudo apt update , sudo apt install git

2.2 Setting Up Git

After installation, configure your Git settings:

Command

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

git config --global user.email demo.email@example.com

Check Git configuration with:
Command git config –list

3. Working with Git

3.1 Initializing a Git Repository

To start tracking a project with Git, navigate to your project directory and run:

Command git init

This creates a .git folder inside your project directory where Git stores your project's version history.

3.2 Adding Files to the Repository

Command git add .

3.3 Committing Changes

After staging files, you commit them to the repository. This saves a snapshot of your project at a given time.

Command git commit -m "Describe your changes here"

3.4 Checking the Status

To see which files are staged, modified, or untracked:

Command git status

3.5 Viewing the Commit History

Command git log

4. Branching in Git

Branches allow you to work on different versions of your project simultaneously. The default branch is called main (or master in older versions).

4.1 Creating a New Branch

Command git branch <branch-name>

4.2 Switching to a Branch

Command git checkout <branch-name>

Alternatively, you can create and switch to a branch in one command:
Command git checkout -b <branch-name>

4.3 Merging Branches

Once we finish working on a feature, you merge your branch into main.
Command git checkout main
git merge <branch-name>

5. Remote Repositories

5.1 Connecting to a Remote Repository

To connect to a remote repository (like GitHub)
Command
git remote add origin https://github.com/yourusername/your-repo.git

5.2 Pushing Changes to a Remote Repository

To push changes from your local repository to a remote one.
Command git push origin <branch-name>

5.3 Pulling Changes from a Remote Repository

If someone else has pushed changes, you can pull them into your local repository.
Command git pull origin <branch-name>

6. Advanced Git Commands

6.1 Cloning a Repository

If you want to copy a remote repository to your local machine.

Command
git clone https://github.com/username/repo-name.git

6.2 Resetting Commits

To undo commits, use git reset. There are two types.
Soft Reset (keeps changes):

Command git reset --soft HEAD~1

Hard Reset (discards changes):
Command git reset --hard HEAD~1

6.3 Stashing Changes

If you want to save changes temporarily without committing them.
Command git stash

To bring back the stashed changes:
Command git stash pop

7. Rebase Git Commands

7.1 Recommendation

Always use rebase concept in sub-branches and try to avoid in main branch.

7.1 Rebase and branch to new stage

Suppose you need to rebase your branch onto the master branch where you started working after pulling the latest changes. Use the following command.
Command git rebase master
It will make base latest master (head) as its base to prevent unnecessory commit.

7.1 History for all branch work

Reflog (keeps changes)
Command git reflog