Basic Git Documentation (Interview Preparation)

Git is a distributed version control system used to track changes in source code during software development. It allows multiple developers to collaborate, manage versions, and maintain the history of their code efficiently.

1. What is Git?

Git is a distributed version control system that helps developers track changes and collaborate on code. Every developer has a complete copy of the repository locally.

2. Why use Git?

• Tracks changes in code
• Allows multiple developers to work simultaneously
• Provides backup with local repositories
• Enables branching and merging for parallel development

3. Difference between Git and GitHub

Git is a version control tool. GitHub is a cloud-based hosting service for Git repositories.

4. Basic Git Workflow

1. Create or clone a repository
2. Make changes in working directory
3. Stage the changes
4. Commit the changes
5. Push to remote repository

5. Important Git Areas

• Working Directory – your local files
• Staging Area – holds files before committing
• Repository – stores all commits and history

6. Commonly Used Git Commands

git init – Initialize a new Git repository

git clone [url] – Copy an existing repository

git status – Check current state of working directory

git add [file] – Stage changes

git commit -m "message" – Save staged changes

git push – Upload local commits to remote repository

git pull – Fetch and merge changes from remote

git log – View commit history

git diff – Show file differences

7. Git Branching

Branching allows you to work on different features independently. Use **git branch** to list branches, **git checkout -b [name]** to create and switch to a new branch.

8. Merging in Git

Combines changes from one branch into another. Example: **git merge feature-branch** merges feature-branch into main.

9. Git Stash

Temporarily stores uncommitted changes. Use **git stash** to save and **git stash pop** to reapply them.

10. Git Head

HEAD is a pointer that represents your current branch or commit. Usually points to the latest commit.

11. Merge Conflict

Occurs when Git cannot automatically merge changes. You must manually resolve differences and then commit again.

12. Git Configuration Commands

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

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

13. Viewing History

git log – Shows commit history

git show [commit] – Shows details of a specific commit

14. Undoing Changes

git reset – Unstage or move HEAD

git revert – Create a new commit that undoes previous commit

15. Git Help Command

git help [command] – Displays help for any Git command

Summary:
Git is essential for developers to manage and collaborate on code efficiently. Remember key commands (init, add, commit, push, pull, branch, merge) and understand Git workflow to answer interview questions confidently.