1. What is Git?

- Git is a **version control system (VCS)** that helps developers track changes in code, collaborate with others, and manage different versions of a project.
- Git हे एक **Version Control System (VCS)** आहे जे डेव्हलपरना कोडमधील बदल ट्रॅक करण्यास, इतरांसोबत काम करण्यास आणि प्रोजेक्टच्या विविध आवृत्त्या (versions) व्यवस्थापित करण्यास मदत करते.

---

2. Why do we need Git? / Git का वापरतो?

- To track changes in files
- To work in teams
- To revert back when something breaks
- To try new features safely
- To backup and share code

- फाइलमधील बदल ट्रॅक करण्यासाठी
- टीमसोबत काम करण्यासाठी
- काहीतरी चुकीचे झाले तर जुनी आवृत्ती परत मिळवण्यासाठी
- नवीन फीचर्स चाचपण्यासाठी
- कोड शेअर आणि बॅकअप ठेवण्यासाठी

---

3. Git Workflow ?

```
Working Directory → Staging Area → Local Repository → Remote Repository
```

(edit) → (git add) → (git commit) → (git push)

---

4. What is Branch?

A **branch** is like a separate line of development — it lets you work on a feature without disturbing the main project.

ब्रँच म्हणजे स्वतंत्र लाईन ऑफ डेव्हलपमेंट — यात तुम्ही नवीन बदल करू शकता, पण मुख्य कोडवर परिणाम होत नाही.

5. Difference between Git & GitHub?

	Git	GitHUB
Definition	A Version Control System that tracks code changes on your local computer.	A cloud-based platform that hosts Git repositories online.
Type	Software / Tool	Online Hosting Service
Installation	Needs to be installed locally.	Accessible via web browser.
Functionality	Tracks and manages local code versions.	Stores and shares Git projects online.
Internet Requirement	Works without Internet.	Requires Internet.
Example Commands	git init, git add, git commit	git push, git pull, git clone

---

6. What is Merge Conflict?

A **merge conflict** happens in **Git** when two or more people change the **same part of a file** in different branches, and Git cannot automatically decide which change to keep.

Merge conflict तेव्हा होते जेव्हा दोन किंवा अधिक लोक एका फाइलचा तोच भाग वेगवेगळ्या branches मध्ये बदलतात, आणि Git ठरवू शकत नाही की कोणता बदल ठेवायचा.

---

7. What is Git Stash?

`git stash` is a Git command that **temporarily saves your uncommitted changes** so that you can work on something else without committing your current work.
Later, you can **apply those changes back** to your branch.

`git stash` हा Git command आहे जो तुमचे **अद्याप commit न केलेले बदल तात्पुरते जतन** करतो, जेणेकरून तुम्ही दुसऱ्या कामावर जाऊ शकता.
नंतर तुम्ही हे बदल पुन्हा branch मध्ये लागू करू शकता.

---

8. What is HEAD in Git?

In Git, **HEAD** is a pointer that **represents the current branch and the latest commit** you are working on.
It tells Git **"this is the snapshot of the project I am currently on."**

Git मध्ये **HEAD** हा एक pointer आहे जो **सध्याच्या branch आणि शेवटच्या commit** ला दर्शवतो.
HEAD Git ला सांगतो की **"मी सध्या या project च्या या version वर काम करत आहे."**

---

9. Remote Repository ?

Remote repository is an online version of your Git project (like GitHub). It helps share code with others.

रिमोट रेपॉजिटरी म्हणजे तुमच्या प्रोजेक्टची ऑनलाइन आवृत्ती (जसे GitHub). यातून इतरांशी कोड शेअर करता येतो

10. Difference between Git Merge, Git Rebase and Git Revert

Feature	Git Merge	Git Rebase	Git Revert
Purpose	Combine branches	Reapply commits to create linear history	Undo a specific commit
History	Keeps full history, may create merge commits	Rewrites history, linearizes commits	Keeps history intact, adds a new "revert" commit
Commit Created?	Yes, merge commit	Usually no new merge commit	Yes, a new revert commit
Safe for Shared Branch?	Yes	Can be risky (rewriting history)	Yes
Command Example	git merge feature	git rebase main	git revert <commit-hash>
Use Case	Combine feature branch into main	Clean up history before merging	Undo a commit safely without deleting history

11. Difference between (CVCS) & (DVCS)?

Feature / Aspect	Centralized Version Control System (CVCS)	Distributed Version Control System (DVCS)
Definition	A single central server stores all versions of code; users get the latest version.	Each user has a complete copy of the repository including history.
Examples	SVN (Subversion), CVS, Perforce	Git, Mercurial, Bitbucket
Repository Type	Only one central repository.	Each user has a local repository + central repository (optional).
Internet Dependency	Requires internet connection to commit and update code.	Works offline; commits are stored locally and pushed later.
Speed	Slower — since most operations need server access.	Faster — most operations happen locally.
Single Point of Failure	If central server crashes, data may be lost or inaccessible.	No single point of failure — every user has full backup.

12. Difference between **Git Pull and Git Fetch?**

Feature	Git Fetch	Git Pull
Definition	Downloads changes from remote but **does not merge**	Downloads and **merges changes** into local branch
Effect on Local Branch	No changes to working files	Updates working files immediately
Safety	Safe, can review changes first	Can create conflicts if local changes exist
Workflow	git fetch → git merge manually	Single command: git pull (fetch + merge)
Use Case	Check for remote updates without affecting local branch	Update local branch with remote changes quickly
Conflict Possibility	None	Possible, may require manual resolution
Speed	Usually faster, no merging	Slower if merging required

---

13. How to Avoid Merge Conflicts ?

Always pull the latest changes before starting your work.

Commit small and frequent changes.

Communicate with your team.

Use separate branches for new features.

14. Common Git Commands

Command	
`git init`	Create new Git repo
`git clone <url>`	Copy remote repo
`git status`	Check current changes
`git add <file>`	Add to staging area
`git commit -m "msg"`	Save version
`git log`	Show commit history
`git branch`	List branches
`git checkout <branch>`	Switch branch
`git merge <branch>`	Merge branch
`git push`	Upload to remote
`git pull`	Download updates

Comment [A1]: