# Git and GitHub Comprehensive Notes

## ****Introduction to Git and GitHub****

### 1. What is Git?

**Answer:** Git is a distributed version control system that tracks changes in files and facilitates collaboration among developers.

### 2. What are the main features of Git?

**Answer:**

* Distributed architecture
* Branching and merging
* Lightweight tags
* Staging area
* High performance for large projects

### 3. What is the difference between Git and GitHub?

**Answer:**

* **Git:** A version control system used locally to track changes.
* **GitHub:** A web-based hosting service for Git repositories with features like pull requests and CI/CD.

### 4. How do you check the version of Git installed?

**Answer:** Run git --version.

### 5. What is a repository in Git?

**Answer:** A repository is a storage space where Git saves all file changes and history.

## ****Initial Setup****

### 6. How do you initialize a Git repository?

**Answer:** Use git init. This creates a .git folder to track changes.

### 7. How do you configure a Git username and email?

**Answer:**

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

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

### 8. How do you clone a repository?

**Answer:** Run git clone <repository-url>.

### 9. What is the .gitignore file?

**Answer:** It specifies files or directories that Git should ignore.

## ****Staging and Committing****

### 10. What is the staging area in Git?

**Answer:** A space where changes are prepared before committing. Use git add <file> to add files to the staging area.

### 11. How do you commit changes in Git?

**Answer:** Use git commit -m "Commit message".

### 12. How do you check the status of a repository?

**Answer:** Use git status.

### 13. What is git diff used for?

**Answer:** It shows changes between the working directory, staging area, and commits.

### 14. How do you view commit history?

**Answer:** Use git log.

## ****Branching and Merging****

### 15. What is a branch in Git?

**Answer:** A branch represents a separate line of development.

### 16. How do you create a new branch?

**Answer:** Run git branch <branch-name>.

### 17. How do you switch branches?

**Answer:** Use git checkout <branch-name> or git switch <branch-name>.

### 18. What is the difference between git merge and git rebase?

**Answer:**

* **git merge:** Combines changes from two branches, keeping commit history.
* **git rebase:** Reapplies commits on top of another branch, rewriting history.

### 19. What is a merge conflict?

**Answer:** A conflict occurs when changes from two branches overlap. Resolve conflicts manually, then commit the resolution.

### 20. How do you delete a branch?

**Answer:**

* **Local:** git branch -d <branch-name>
* **Remote:** git push origin --delete <branch-name>

## ****Tags and References****

### 21. What is a tag in Git?

**Answer:** A reference used to mark a specific point in the repository’s history, typically for releases.

### 22. How do you create a lightweight tag?

**Answer:** Use git tag <tag-name>.

### 23. What is HEAD in Git?

**Answer:** HEAD points to the current branch and commit.

### 24. What is a detached HEAD?

**Answer:** When HEAD points directly to a commit instead of a branch.

### 25. What is git cherry-pick?

**Answer:** It applies a specific commit from one branch into another.

## ****Undoing Changes****

### 26. How do you undo the last commit?

**Answer:**

* **Soft reset:** git reset --soft HEAD~1 (keeps changes staged)
* **Hard reset:** git reset --hard HEAD~1 (discards changes)

### 27. What is git stash?

**Answer:** Temporarily saves changes not ready for commit. Retrieve with git stash pop.

### 28. How do you revert a commit?

**Answer:** Use git revert <commit-hash>.

### 29. What is git clean?

**Answer:** Removes untracked files from the working directory. Use cautiously with git clean -f.

## ****GitHub Basics****

### 30. What is GitHub?

**Answer:** A platform for hosting Git repositories with collaboration tools like pull requests and issue tracking.

### 31. How do you fork a repository on GitHub?

**Answer:** Forking creates a personal copy of a repository under your GitHub account.

### 32. How do you create a pull request on GitHub?

**Answer:** Navigate to the forked repository and click "Pull Request" to propose changes.

### 33. What is GitHub Actions?

**Answer:** A CI/CD platform integrated with GitHub to automate workflows.

### 34. How do you secure your GitHub repository?

**Answer:**

* Use branch protection rules
* Enable 2FA
* Review and minimize access permissions

## ****Advanced Git****

### Logs and History

#### 35. What is git reflog?

**Answer:** Tracks updates to HEAD, including changes not in the commit history.

#### 36. How do you compare two commits?

**Answer:** Use git diff <commit1> <commit2>.

#### 37. What is git blame?

**Answer:** Shows who made changes to each line in a file and when.

### Submodules and Hooks

#### 38. What is a submodule in Git?

**Answer:** A repository embedded within another repository, used for dependency management.

#### 39. What are Git hooks?

**Answer:** Scripts that run automatically at specific Git events, like pre-commit or post-merge.

## ****Collaboration and Workflow****

### 40. What is the purpose of git fetch?

**Answer:** Downloads changes from the remote repository without merging them.

### 41. How do you resolve a merge conflict?

**Answer:** Edit conflicting files, stage the resolved files, and commit the changes.

### 42. What is the difference between SSH and HTTPS cloning in Git?

**Answer:**

* **SSH:** Secure and requires an SSH key.
* **HTTPS:** Easier setup but requires frequent authentication.

### 1. What is the difference between git fetch and git pull?

**Answer:**

* **git fetch:** Downloads updates; does not merge.
* **git pull:** Downloads and merges updates.

### 2. How do you view a specific commit?

**Answer:** Use git show <commit-hash>.

### 3. What is the purpose of .gitignore?

**Answer:** Specifies files/directories to be ignored by Git.

### Advanced Topics

#### **43. What is git archive?**

**Answer:** Creates a compressed file (like tar or zip) containing the contents of a specific branch or commit.

#### **44. How do you rename a branch?**

**Answer:**

* **Local branch:** git branch -m <old-name> <new-name>
* **Remote branch:**
* git push origin :<old-name>

git push origin <new-name>

#### **45. What is a bare repository?**

**Answer:** A repository without a working directory, primarily used as a central repository for collaboration.

#### **46. How do you squash commits?**

**Answer:** Use interactive rebase: git rebase -i <base-branch> and choose squash for commits to merge.

#### **47. What is git bisect?**

**Answer:** A tool to find the commit that introduced a bug by performing a binary search through commit history.

#### **48. How do you undo git add?**

**Answer:** Run git reset <file> to unstage a file.

#### **49. How do you see untracked files?**

**Answer:** Use git status.

#### **50. How do you add all changes to staging?**

**Answer:** Use git add ..