

Assignment-1 : Basic Git Commands

Here are some basic Git commands along with their definitions:

1. `git init`

- **Definition:** Initializes a new Git repository in the current directory. This command creates a `.git` directory in the current folder, which contains all the necessary repository files.

2. `git clone [url]`

- **Definition:** Creates a copy of an existing Git repository from a specified URL. This command downloads the repository's files and its entire history.

3. `git add [file]`

- **Definition:** Adds a file to the staging area, making it ready to be committed. The file can be specified by name or by using a wildcard pattern.

4. `git commit -m "[message]"`

- **Definition:** Records the changes in the staging area with a descriptive commit message. This command creates a new commit containing the current contents of the index and the given log message describing the changes.

5. `git status`

- **Definition:** Displays the state of the working directory and the staging area. It shows which changes have been staged, which haven't, and which files aren't being tracked by Git.

6. `git log`

- **Definition:** Shows the commit history for the current branch, listing the commits along with their details such as the commit message, author, and date.

7. `git push [remote] [branch]`

- **Definition:** Uploads the local branch commits to the remote repository. This command updates the remote branch with commits made to the local branch.

8. `git pull [remote] [branch]`

- **Definition:** Fetches and integrates changes from the remote repository into the current branch. This command combines `git fetch` and `git merge` to update the local repository.

9. `git branch`

- **Definition:** Lists all the branches in the repository. When used with a branch name, it creates a new branch.

10. `git checkout [branch]`

- **Definition:** Switches to the specified branch and updates the working directory to match. If the branch doesn't exist, it will switch to a new branch when used with the `-b` option.

11. git merge [branch]

- **Definition:** Merges the specified branch into the current branch. This command integrates changes from the specified branch into the current branch.

12. git remote

- **Definition:** Manages the set of tracked repositories. This command lets you view and configure remote repositories, such as those hosted on GitHub.

13. git fetch [remote]

- **Definition:** Downloads objects and refs from another repository. It fetches updates from the remote repository but does not merge them into the current branch.

14. git rebase [branch]

- **Definition:** Reapplies commits on top of another base tip. This command is used to integrate changes from one branch into another by reapplying commits.

15. git diff

- **Definition:** Shows the changes between commits, commit and working tree, etc. It compares the changes in the working directory against the staging area.

16. git reset [file]

- **Definition:** Removes files from the staging area but keeps the changes in the working directory. When used with `--hard`, it resets the index and working directory to match the specified commit.

17. git rm [file]

- **Definition:** Removes a file from the working directory and stages the removal for commit. This command deletes the file from the disk and the repository.

18. git stash

- **Definition:** Temporarily saves changes that are not ready to be committed. This command is useful for saving work in progress without committing it.

19. git tag [name]

- **Definition:** Creates a tag in the repository, often used to mark release points. Tags are used to create a named reference to a specific commit.

20. git config [option] [value]

- **Definition:** Sets configuration options for Git installation, such as username, email, and editor preferences. This command can be used to set both global and repository-specific settings.