Taskin Mohammad Mubassir
Tracking number: J2DFECB
Assignment-2

Git commands

1. **git init**

- Initializes a new Git repository in the current directory.
- Sets up the necessary Git metadata in a hidden .git directory, allowing Git to start tracking changes in the project. Use this command when starting a new project or adding version control to an existing project.

2. **git clone [http link of a project]**

- Creates a copy of an existing repository from a remote server.
- Downloads a remote repository and its history to a local machine. The [http link of a project] is the URL of the repository to be cloned. This is commonly used to start working on a project that is hosted on platforms like GitHub or GitLab.

3. **git remote add origin [repo link]**

- Adds a remote repository URL to a local repository.
- Links a local project to a repository hosted on GitHub. The name origin is an alias for the URL. This alias can be used in other commands to refer to the remote repository.

4. **git add [file name]**

- Stages a specific file, preparing it for commit.
- Tells Git to include changes to the specified file in the next commit. Multiple files can also be staged at once by specifying each file name or using a wildcard (e.g., git add . to stage all changes).

5. **git commit -m ["some msg"]**

- Records change to the repository with a descriptive message.
- Saves the staged changes to the repository with a message explaining what was done. The message should be concise but descriptive enough to understand the changes made.

6. **git push origin main**

- Pushes local commits to the remote repository's main branch.
- Uploads local commits to the remote repository specified by origin, updating the main branch. This is how changes are shared with others.

7. **git branch [new branch name]**

- Creates a new branch with the given name.
- Creates a new branch in the repository. Branches allow for developing features, fixing bugs, or experimenting with changes independently from the main codebase.

8. **git branch -M [new name]**

- Renames the current branch to the specified new name.
- Change the name of the current branch to something more appropriate or descriptive. The -M option stands for "move," effectively renaming the branch.

9. **git branch -d [branch name]**

- Deletes the specified branch.
- Removes a branch from the repository. Branches that have been merged into another branch or are no longer needed are typically deleted.

10. **git checkout [branch name]**

- Switches to the specified branch and updates the working directory.
- Changes the working directory to the specified branch, allowing work on that branch's code. Use git checkout -b [new branch name] to create and switch to a new branch in one step.

11. **git merge [branch name]**

- Merges the specified branch into the current branch.
- Integrates changes from one branch into the branch currently being worked on. This command is used to combine the work done on different branches.

12. **git reset [file or multiple file names]**

- Unstages the specified files, keeping the changes in the working directory.

- Removes files from the staging area without discarding the changes. Useful if files were accidentally added to the staging area and need to be unstaged.

13. git reset --hard [# id from git log, which is unique]

- Resets the repository to a specific commit, discarding all changes since then.
- Reverts the repository to a previous state, identified by the commit ID from the git log. This discards all changes made after that commit, including those in the working directory and staging area.

14. git status

- Displays the state of the working directory and the staging area.
- Shows which files are modified, staged, or untracked. This is a quick way to see what changes have been made and what needs to be committed.

15. git pull

- Fetches and merges changes from the remote repository into the current branch.
- Updates the local repository with the latest changes from the remote repository. Combines git fetch (downloading changes) and git merge (applying those changes) in one step.