Git Commands

Hridoy Das

Basic Commands

Initializes a new Git repository in the current directory

```
git init
```

Clones an existing remote repository to your local machine

```
git clone <repository-url>
```

Stages a file, preparing it for commit.

```
git add <file>
```

Commits staged changes with a descriptive message

```
git commit -m "<message>"
```

Displays the commit history for the repository, showing commit hashes, authors, dates, and message

```
git log
```

Hridoy Das

Branching & Merging

Lists all the branches in the repository.

```
git branch
```

Create new branch

```
git branch branch-name
```

Delete branch

```
git branch -d branch-name
```

Switches to a different branch or commit.

```
git checkout branch-name
```

Create and switch to a new branch

```
git checkout -b new-branch
```

Merges changes from the specified branch into the current branch.

```
git merge branch-name
```

Reapplies commits from one branch into another in a linear sequence

```
git rebase branch-name
```

Hridoy Das

Remote Repositories

Lists all remote repositories associated with the local repo

```
git remote -v
```

Adds a new remote repository reference to your local repository

```
git remote add origin https://github.com/user/repo.git
```

Pushes local changes (commits) from your current branch to the remote repository

```
git push origin branch-name
```

Fetches the latest changes from the remote repository and merges them into your local branch

```
git pull origin branch-name
```

Fetches the latest changes from the remote repository without merging them into the current branch

```
git fetch origin
```

Hridoy Das

Staging & Resetting

Removes the specified file from the staging area, but keeps the file changes

```
git reset filename
```

Resets the repository to a specific commit, discarding all changes after that commit

```
git reset --hard commit-hash
```

Creates a new commit that undoes changes introduced by a specific commit without changing the commit history

```
git revert commit-hash
```

Stashing & Cleaning

Temporarily stores changes in a "stash" so you can work on something else without committing the changes

```
git stash
```

Applies the stashed changes back into the working directory and removes the stash from the list

```
git stash pop
```

Removes untracked files from the working directory

```
git clean -f
```

Stashing & Cleaning

Temporarily stores changes in a "stash" so you can work on something else without committing the changes

```
git stash
```

Applies the stashed changes back into the working directory and removes the stash from the list

```
git stash pop
```

Removes untracked files from the working directory

```
git clean -f
```