

GitHub Assignment

Name : Syedanwar Hanzahusen Mashalkar

=====

GitHub Commands and Their Purposes

Basic Git Commands

Configuration

- `git config --global user.name "Your Name"`
 - **Purpose:** Sets the name you want to attach to your commit transactions.
- `git config --global user.email "your.email@example.com"`
 - **Purpose:** Sets the email you want to attach to your commit transactions.
- `git config --global core.editor "editor"`
 - **Purpose:** Sets the default text editor for Git.

Repository Management

- `git init`
 - **Purpose:** Initializes a new Git repository.
- `git clone <repository-url>`
 - **Purpose:** Clones a repository into a new directory.

Staging and Committing

- `git status`
 - **Purpose:** Shows the working directory status.
- `git add <file>`
 - **Purpose:** Adds a file to the staging area.
- `git add .`
 - **Purpose:** Adds all files to the staging area.
- `git commit -m "Commit message"`
 - **Purpose:** Commits the staged changes with a descriptive message.
- `git commit -a -m "Commit message"`
 - **Purpose:** Stages and commits all changes in tracked files.

Branching

- `git branch`
 - **Purpose:** Lists all local branches in the repository.
- `git branch <branch-name>`
 - **Purpose:** Creates a new branch.
- `git checkout <branch-name>`
 - **Purpose:** Switches to the specified branch.
- `git checkout -b <branch-name>`
 - **Purpose:** Creates and switches to a new branch.
- `git merge <branch-name>`
 - **Purpose:** Merges the specified branch into the current branch.
- `git branch -d <branch-name>`
 - **Purpose:** Deletes the specified branch.

Remote Repositories

- `git remote`
 - **Purpose:** Lists all remote repositories.
- `git remote add <name> <url>`
 - **Purpose:** Adds a new remote repository.
- `git remote -v`
 - **Purpose:** Shows URLs of remote repositories.
- `git fetch <remote>`
 - **Purpose:** Fetches changes from the remote repository.
- `git pull <remote> <branch>`
 - **Purpose:** Fetches and merges changes from the remote branch.
- `git push <remote> <branch>`
 - **Purpose:** Pushes commits to the remote branch.

Undoing Changes

- `git reset <file>`
 - **Purpose:** Unstages a file.
- `git reset --hard`
 - **Purpose:** Resets the staging area and the working directory to the last commit.
- `git reset --soft HEAD~1`
 - **Purpose:** Undoes the last commit, keeping changes in the working directory.
- `git revert <commit>`
 - **Purpose:** Creates a new commit that undoes changes made by a specific commit.

Viewing History

- `git log`
 - **Purpose:** Shows the commit history.
- `git log --oneline`
 - **Purpose:** Shows the commit history in a compact format.
- `git diff`
 - **Purpose:** Shows changes between commits, commit and working directory, etc.
- `git show <commit>`
 - **Purpose:** Shows changes in a specific commit.

Tags

- `git tag`
 - **Purpose:** Lists all tags.
- `git tag <tag-name>`
 - **Purpose:** Creates a new tag.
- `git tag -d <tag-name>`
 - **Purpose:** Deletes a tag.
- `git push <remote> <tag>`
 - **Purpose:** Pushes a tag to the remote repository.

Advanced Git Commands

Stashing

- `git stash`
 - **Purpose:** Stashes changes in the working directory.
- `git stash apply`
 - **Purpose:** Applies stashed changes.
- `git stash drop`
 - **Purpose:** Deletes a specific stash.
- `git stash list`
 - **Purpose:** Lists all stashes.

Rebasing

- `git rebase <branch>`
 - **Purpose:** Reapplies commits on top of another base tip.
- `git rebase --abort`
 - **Purpose:** Aborts the rebase process.
- `git rebase --continue`
 - **Purpose:** Continues the rebase process after resolving conflicts.

Cherry-Picking

- `git cherry-pick <commit>`
 - **Purpose:** Applies the changes introduced by a specific commit.

Submodules

- `git submodule add <repository-url>`
 - **Purpose:** Adds a new submodule.
- `git submodule update`
 - **Purpose:** Updates all submodules.