Useful Git Commands

Basic Commands

- **`git init`**
  Initialize a new Git repository.

- **`git clone <repository>`**
  Clone an existing repository from a remote location.

- **`git add <file>`**
  Stage changes to a specific file.

- **`git add .`**
  Stage all changes in the current directory.

- **`git commit -m "message"`**
  Commit staged changes with a descriptive message.

- **`git status`**
  Check the status of your working directory and staging area.

- **`git log`**
  View the commit history.

Branching and Merging

- **`git branch`**
  List all branches in the repository.

- **`git branch <branch-name>`**
  Create a new branch.

- **`git checkout <branch-name>`**
  Switch to a different branch.

- **`git checkout -b <branch-name>`**
  Create and switch to a new branch.

- **`git merge <branch-name>`**
  Merge changes from a specified branch into the current branch.

- **`git branch -d <branch-name>`**
  Removing local branch.

Remote Repositories

- **`git remote -v`**
  List remote repositories.

- **`git remote add <name> <url>`**
  Add a new remote repository.

- **`git fetch <remote>`**
  Fetch changes from a remote repository without merging.

- **`git pull <remote> <branch>`**
  Fetch and merge changes from a remote branch.

- **`git push <remote> <branch>`**
  Push your changes to a remote repository.

- **`git push <remote> --delete <branch>`**
  Remove remote branch.

Viewing Changes

- **`git diff`**
  Show changes between the working directory and staging area.

- **`git diff --staged`**
  Show changes between the staging area and the last commit.

- **`git show <commit>`**
  Show changes introduced by a specific commit.

Stashing

- **`git stash`**
  Stash changes in a dirty working directory.

- **`git stash pop`**
  Apply the stashed changes and remove them from the stash.

- **`git stash list`**
  List all stashed changes.

Tagging

- **`git tag`**
  List all tags in the repository.

- **`git tag <tag-name>`**
  Create a new tag.

- **`git push <remote> <tag-name>`**
  Push a tag to a remote repository.

Undoing Changes

- **`git reset <file>`**
  Unstage a file.

- **`git checkout -- <file>`**
  Discard changes in a specific file.

- **`git revert <commit>`**
  Create a new commit that undoes the changes of a specified commit.

- **`git reset --hard <commit>`**
  Reset the working directory to a specific commit, discarding all changes.

Others

- **`git cherry-pick <commit>`**
  Apply the changes introduced by an existing commit.

- **`git blame <file>`**
  Show what revision and author last modified each line of a file.

- **`git clean -fd`**
  Remove untracked files from the working directory.