ChatGPT 3.5

You

listout all usefull git commands

ChatGPT

Certainly! Here's a list of some useful Git commands along with a brief description of each:

1. **Initializing a Repository:**
   - `git init`: Initializes a new Git repository.
2. **Cloning a Repository:**
   - `git clone <repository_url>`: Creates a copy of a remote repository on your local machine.
3. **Basic Snapshotting:**
   - `git add <file>`: Adds changes in a file to the staging area.
   - `git add .` or `git add --all`: Adds all changes to the staging area.
   - `git commit -m "Commit message"`: Commits changes from the staging area to the repository.
4. **Branching:**
   - `git branch`: Lists all local branches.
   - `git branch <branch_name>`: Creates a new branch.
   - `git checkout <branch_name>`: Switches to a different branch.
   - `git merge <branch_name>`: Merges changes from one branch into another.
5. **Remotes:**
   - `git remote`: Lists all remote repositories.
   - `git remote add <remote_name> <repository_url>`: Adds a remote repository.
   - `git push <remote_name> <branch_name>`: Pushes changes to a remote repository.
   - `git pull <remote_name> <branch_name>`: Fetches changes from a remote repository and merges them into the local branch.
6. **Viewing Changes:**
   - `git status`: Shows the status of changes as untracked, modified, or staged.
   - `git log`: Displays the commit history.
   - `git diff`: Shows changes between commits, the working directory, or branches.
7. **Undoing Changes:**
   - `git reset <file>`: Unstages changes.
   - `git reset --hard <commit_hash>`: Resets the repository to a specific commit, discarding changes.
   - `git revert <commit_hash>`: Creates a new commit that undoes changes made in a previous commit.
8. **Stashing Changes:**
   - `git stash`: Temporarily saves changes that are not ready to be committed.
   - `git stash apply`: Applies the latest stash to the working directory.
9. **Tagging:**
   - `git tag <tag_name>`: Creates a lightweight tag at the current commit.
   - `git tag -a <tag_name> -m "Tag message"`: Creates an annotated tag with a message.
10. **Configuration:**
    - `git config --global user.name "Your Name"`: Sets your username.
    - `git config --global user.email "your.email@example.com"`: Sets your email address.
    - `git config --list`: Lists all Git configuration settings.

These are just some of the fundamental Git commands. Git is a powerful version control system with many features, so feel free to explore its documentation for more advanced commands and options.