

## GitHub Commands

### 1. Configuration

- a. **Set your username:** `git config --global user.name "Your Name"` Sets the global username for Git commits.
- b. **Set your email:** `git config --global user.email "your.email@example.com"` Sets the global email for Git commits.

### 2. Repository Management

- a. **Initialize a new repository:** `git init` Creates a new Git repository.
- b. **Clone an existing repository:** `git clone <repository_url>` Copies an existing Git repository to your local machine.

### 3. Working with Changes

- a. **Check the status of your working directory:** `git status` Shows the status of changes in the working directory.
- b. **Stage changes:** `git add <file>` Stages a file for the next commit.
- c. **Stage all changes:** `git add .` Stages all changes in the working directory for the next commit.
- d. **Commit staged changes:** `git commit -m "Commit message"` Records staged changes to the repository with a message.

### 4. Branching and Merging

- a. **Create a new branch:** `git branch <branch_name>` Creates a new branch.
- b. **Switch to a different branch:** `git checkout <branch_name>` Switches to the specified branch.
- c. **Create and switch to a new branch:** `git checkout -b <branch_name>` Creates a new branch and switches to it.
- d. **Merge a branch into your current branch:** `git merge <branch_name>` Combines the specified branch into the current branch.

### 5. Remote Repositories

- a. **Add a remote repository:** `git remote add origin <repository_url>` Adds a remote repository.
- b. **Fetch changes from the remote repository:** `git fetch origin` Downloads changes from the remote repository without merging.

- c. **Push changes to the remote repository:** `git push origin <branch_name>` Uploads local branch commits to the remote repository.
- d. **Pull changes from the remote repository:** `git pull origin <branch_name>` Downloads and merges changes from the remote branch.

## 6. Viewing History

- a. **View commit history:** `git log` Shows the commit history.
- b. **View a specific file's commit history:** `git log <file>` Shows the commit history for a specific file.

## 7. Stashing Changes

- a. **Stash changes:** `git stash` Temporarily saves changes not ready to be committed.
- b. **Apply stashed changes:** `git stash apply` Applies stashed changes to the working directory.
- c. **List stashes:** `git stash list` Shows a list of stashed changes.