

## Git configuration and setup

- **git config** --global user.name "Your Name": Set your username globally.
- **git config** --global user.email "youremail@example.com" - Set your email globally.
- **git help**: Display the main help documentation, showing a list of commonly used Git commands.
- **git config --list**: List all settings.

## Initializing a Repository

- **git init**: initializes a new Git repository in the current directory.
- **git init <directory>**: Creates a new Git repository in the specified directory.
- **git clone <url>**: This clones a repository from a remote server to your local machine.

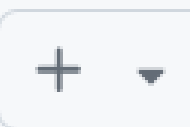
## Basic Git Commands

- **git add <file>**: Adds a specific file to the staging area.
- **git add .**: Adds all modified and new files to the staging area.
- **git status**: Shows the current state of your repository, including tracked and untracked files, modified files, and branch information.
- **git status --ignored**: Displays ignored files in addition to the regular status output.
- **git diff**: Shows the changes between the working directory and the staging area.
- **git diff <commit1\_id> <commit2\_id>**: Displays the differences between two commits.
- **git diff HEAD**: Display the difference between the current directory and the last commit.

- **git commit -m "<message>"**: Creates a new commit with the changes in the staging area and specifies the commit message inline.
- **git restore <file>**: Restores the file in the working directory to its state in the last commit.
- **git reset <commit>**: Moves the branch pointer to a specified commit, resetting the staging area and the working directory to match the specified commit.
- **git reset --hard <commit>**: Moves the branch pointer to a specified commit, discarding all changes in the staging area and the working directory, resetting the repository to the specified commit.

## Branching & Merging

- **git branch**: Lists all branches in the repository.
- **git branch <branch-name>**: Creates a new branch with the specified name.
- **git branch -d <branch-name>**: Deletes the specified branch.
- **git branch -a**: Lists all local and remote branches.
- **git branch -r**: Lists all remote branches.
- **git checkout <branch-name>**: Switches to the specified branch.
- **git checkout -b <new-branch-name>**: Creates a new branch and switches to it.



- **git merge <branch>**: Merges the specified branch into the current branch.
- **git log**: Displays the commit history of the current branch.
- **git log -all**: Displays the commit history of all branches.
- **git stash**: Stashes the changes in the working directory, allowing you to switch to a different branch or commit without committing the changes.
- **git stash list**: Lists all stashes in the repository.
- **git stash pop**: Applies and removes the most recent stash from the stash list.
- **git stash drop**: Removes the most recent stash from the stash list.

## Remote Repositories

- **git fetch**: Retrieves change from a remote repository, including new branches and commit.
- **git fetch <remote>**: Retrieves change from the specified remote repository.
- **git fetch -prune**: Removes any remote-tracking branches that no longer exist on the remote repository.
- **git pull**: Fetches changes from the remote repository and merges them into the current branch.
- **git push**: Pushes local commits to the remote repository.

- **git push <remote>**: Pushes local commits to the specified remote repository.
- **git push <remote> <branch>**: Pushes local commits to the specified branch of the remote repository.
- **git push -all**: Pushes all branches to the remote repository.
- **git remote**: Lists all remote repositories.
- **git remote add <name> <url>**: Adds a new remote repository with the specified name and URL.

## Git Comparison

- **git show**: Shows the details of a specific commit, including its changes.

- **git show <commit>**: Shows the details of the specified commit, including its changes.

## Git Managing History

- **git revert <commit>**: Creates a new commit that undoes the changes introduced by the specified commit.
- **git revert -no-commit <commit>**: Undoes the changes introduced by the specified commit, but does not create a new commit.
- **git rebase <branch>**: Reapplies commits on the current branch onto the tip of the specified branch.