@CODEPEKDIGITAL

ALL GIT COMMANDS

WWW.CODEPEK.COM

1. Setup and Configuration

- **git init** — Initialize a new Git repository.
- **git config --global user.name "Your Name"** — Set your Git username.
- **git config --global user.email "your.email@example.com"** — Set your Git email.
- **git config --global core.editor "editor"** — Set the default editor for Git.
- **git config --global --list** — Show all global Git configurations.

2. Basic Snapshotting

- **git add <file>** — Stage a specific file.
- **git add .** — Stage all files in the current directory.
- **git commit -m "commit message"** — Commit the staged changes with a message.
- **git commit -am "commit message"** — Stage and commit all modified and deleted files in one step.
- **git status** — Check the status of the working directory and staging area.
- **git diff** — Show unstaged changes in the working directory.
- **git diff --staged** — Show staged changes for the next commit.
- **git reset <file>** — Unstage a file from the staging area.

3. Branching and Merging

- **git branch** — List all branches.
- **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 in one command.
- **git merge <branch-name>** — Merge a branch into the current branch.
- **git branch -d <branch-name>** — Delete a local branch.
- **git branch -D <branch-name>** — Force delete a local branch.

4. Remote Repositories

- **git remote add <name> <url>** — Add a remote repository.
- **git remote -v** — List all remote repositories.
- **git fetch <remote>** — Fetch changes from a remote repository.
- **git pull <remote> <branch>** — Fetch and merge changes from a remote branch.
- **git push <remote> <branch>** — Push changes to a remote branch.
- **git push -u <remote> <branch>** — Set upstream and push changes.

5. Inspection and Comparison

- **git log** — Show commit history.
- **git log --oneline** — Show a compact view of the commit history.
- **git show <commit-hash>** — Show details of a specific commit.
- **git reflog** — Show the history of all actions in the repository.
- **git blame <file>** — Show who last modified each line of a file.
- **git diff <branch1> <branch2>** — Show differences between two branches.

6. Stashing and Cleaning

- **git stash** — Stash changes in the working directory.
- **git stash list** — List all stashes.
- **git stash apply** — Apply the most recent stash.
- **git stash apply stash@{n}** — Apply a specific stash.
- **git stash drop** — Delete the most recent stash.
- **git stash clear** — Remove all stashes.
- **git clean -f** — Remove untracked files from the working directory.

7. Rewriting History

- **git commit --amend** — Modify the most recent commit.
- **git rebase <branch>** — Reapply commits on top of another base branch.
- **git rebase -i <commit-hash>** — Interactive rebase to edit or squash commits.
- **git reset --soft <commit-hash>** — Reset to a specific commit, keep changes staged.
- **git reset --mixed <commit-hash>** — Reset to a specific commit, keep changes unstaged.
- **git reset --hard <commit-hash>** — Reset to a specific commit, discard all changes.

8. Tagging

- **git tag <tag-name>** — Create a new tag.
- **git tag -a <tag-name> -m "tag message"** — Create an annotated tag with a message.
- **git tag -d <tag-name>** — Delete a local tag.
- **git push <remote> <tag-name>** — Push a tag to a remote repository.
- **git push --tags** — Push all tags to a remote repository.

9. Undoing Changes

- **git checkout -- <file>** — Discard changes to a specific file.
- **git revert <commit-hash>** — Create a new commit that undoes changes from a specific commit.
- **git reset HEAD~1** — Undo the last commit and unstage changes.
- **git reset --hard HEAD~1** — Undo the last commit and discard changes.

10. Miscellaneous Commands

- **git archive** — Create a zip or tar archive of a specific branch or commit.
- **git bisect** — Use binary search to find the commit that introduced a bug.
- **git cherry-pick <commit-hash>** — Apply a specific commit from another branch.
- **git shortlog** — Summarize commit history by author.
- **git describe** — Describe a commit using the most recent tag reachable from it.

11. Aliases

- **git config --global alias.<alias-name> <git-command>** — Set up an alias for a Git command.

For example, git config --global alias.co checkout creates an alias git co for git checkout.

@CODEPEKDIGITAL

DONT FORGET TO LIKE, SHARE AND SAVE IF YOU LIKE THIS POST

WWW.CODEPEK.COM

CODEPEK