



# 10 Essential Git Commands You Should Know

SWIPE



# 1

## Initial Setup

- Check installed Git version  
`git --version`
- Set your name for all Git repositories
  - `git config --global user.name "Your Name"`
- Set your email for all Git repositories
  - `git config --global user.email "you@example.com"`
- Enable color highlighting in Git output
  - `git config --global color.ui auto`



# 2

## Repository Operations

---

- Initialize a new Git repository  
`git init`
- Clone a remote repository
  - `git clone <repository-url>`



# 3

## Working with Files

- Stage a specific file for commit
  - `git add <file>`
- Stage all changed files in the current directory for commit
  - `git add .`
- Commit the staged changes with a message
  - `git commit -m "Commit message"`
- Commit all tracked changes with a message (skipping `git add`)
  - `git commit -a -m "Commit message"`
- Show the status of the working directory and staging area
  - `git status`
- Show changes between working directory and staging area
  - `git diff`



# 4

## Branching and Merging



- List all branches  
`git branch`
- Create a new branch
  - `git branch <branch-name>`
- Switch to a specific branch.
  - `git checkout <branch-name>`
- Create and switch to a new branch
  - `git checkout -b <branch-name>`
- Merge a branch into the current branch
  - `git merge <branch-name>`
- Delete a branch
  - `git branch -d <branch-name>`



# 5

## Remote Repositories

---

- Add a remote repository
  - `git remote add origin <remote-url>`
- List remote repositories
  - `git remote -v`
- Push changes to a remote branch
  - `git push origin <branch-name>`
- Push to a remote branch and track it
  - `git push -u origin <branch-name>`
- Fetch and merge changes from the remote repository
  - `git pull`
- Fetch changes from the remote (does not merge)
  - `git fetch`



# 6

## Viewing History

---

- Show commit history
  - `git log`
- Show commit history in a concise format
  - `git log --oneline`
- Show details of a specific commit
  - `git show <commit-hash>`



# Undoing Changes

---

- Unstage a file (keep changes in working directory)  
`git reset <file>`
- Reset the working directory to the last commit, discarding all changes
  - `git reset --hard`
- Reset the current branch to the specified commit, keeping changes staged
  - `git reset --soft <commit-hash>`
- Discard changes in a file and revert to the last commit
  - `git checkout -- <file>`





# 8

## Stashing Changes

---

- Save uncommitted changes for later use
  - `git stash`
- Apply stashed changes and remove them from the stash
  - `git stash pop`
- List all stashed changes
  - `git stash list`
- Delete a specific stash entry
  - `git stash drop`



# 9

## Tagging

---

- Create a tag for a specific commit  
`git tag <tag-name>`
- List all tags
  - `git tag`
- Push a tag to the remote repository
  - `git push origin <tag-name>`



# 10

## Git Ignored Files

---

- A file where you specify files/folders to be ignored by Git
  - `.gitignore`



*Thank you!*

David Ezekiel

