# GIT AND GITHUB

Git is a version control system – it helps developers track changes in their code, collaborate with others,
and go back to earlier versions if needed.

| Term | Simple Meaning |
|---|---|
| Repository | A folder that Git is tracking. |
| Commit | A save point in Git. It records your changes. |
| Branch | A separate line of work. You can make changes without affecting the main code. |
| Merge | Combining changes from one branch into another. |
| Push | Send changes to GitHub (remote repository). |
| Pull | Get latest changes from GitHub. |
| Clone | Copy a remote GitHub repo to your local machine. |
| Remote | A link to a repo hosted on GitHub or other server. |

## STAGES OF GIT:

Git have total 3 stages.

### 1. WORKING DIRECTORY:

Its a folder where we write the code.

here we can't track the files.

---------------------------------------------------

2. STAGING AREA:

here we can track the files.
its next version of your file.

-------------------------------------------------

3. REPOSITORY:

it's a folder where we can store tracked file.

----------------------------------------------------

========================================================================
Git workflow: Create Repo → Create Branch → Work → Commit → Merge → Push

========================================================================

Step 1: Starts a Git repo in your project folder

** git init ** - To create a local repo

=====================================================================

Step 2:  Create a File  ** echo "Hello World" > index.html **

=====================================================================

Step 3:   Track the File

 ** git add index.html  **  - To Tells Git to start tracking the file

=====================================================================

Step 4:  Commit (Save Your Changes)

** git commit -m "Initial commit" **

======================================================================================

Step 5: Create a Branch for feature/login in code

** git checkout -b feature/login **  - You are now working on a separate branch called feature/login
** git add . **
** git commit -m "feature login added" **

======================================================================================

Step 6: Switch Back to Main Branch

** git checkout main **

======================================================================================

Step 7: Merge Changes from Feature Branch

** git merge feature/login **

======================================================================================

Step 8: Push to GitHub (If Connected)

** git push origin main **

| Branch Type | Use Case |
| ------------------ | ---------------------------------- |
| `main` or `master` | The main stable code |
| `feature/*` | New features like `feature/login` |
| `release/*` | Code ready to release |
| `hotfix/*` | Urgent fixes to production |
| `develop` | Working development branch |

Git workflow: Create Repo → Create Branch → Work → Commit → Merge → Push

## Git Cheat Sheet (Most Useful Commands)

### ◇ Basic Setup

```
git init                     # Initialize a new Git repo

git config --global user.name "Your Name"

git config --global user.email "you@example.com"
```

### ◇ Staging & Committing

```
git status                  # Show current changes

git add file.txt            # Stage a file

git add .                   # Stage all files

git commit -m "Message"     # Commit staged files
```

### ◇ Branching

```
git branch                  # List branches

git branch branch-name      # Create a new branch

git checkout branch-name    # Switch to a branch

git checkout -b new-branch  # Create + switch

============================================================
 ◇ Merging
============================================================


git merge branch-name       # Merge into current branch


==============================================================
 ◇ Remote (GitHub)
==============================================================


git remote add origin URL   # Link local repo to GitHub

git push origin branch      # Push to GitHub

git pull origin branch      # Pull latest from GitHub


==============================================================
 ◇ Undoing Changes
==============================================================


git reset --soft HEAD~1     # Undo last commit, keep files

git reset --hard HEAD~1     # Undo commit and file changes

git checkout -- file.txt    # Undo changes in file



==============================================================
 ◇ Stash
==============================================================


git stash                   # Save uncommitted changes
git stash pop               # Reapply stashed changes
```

====================================================================

💡 GitHub: What’s the Role?

====================================================================

GitHub is an online platform to store your Git repositories so that:

-> You can share your code.

-> You can collaborate with others.

-> You can track changes.

========================================================================