**📘 Git Basics & Usage Guide**

**🧠 What is Git?**

**Git** is a **distributed version control system (DVCS)** used to track changes in source code during software development.

**🔧 Why Use Git?**

* Track history of code changes
* Collaborate with multiple developers
* Revert to previous versions
* Work on multiple features using branches
* Integrate with CI/CD systems like Jenkins

**📦 Git vs GitHub**

| **Git** | **GitHub** |
| --- | --- |
| Local tool | Remote hosting service |
| Tracks changes | Stores Git repositories online |
| Runs on your machine | Enables collaboration & pull requests |

**🔁 Basic Git Workflow**

+------------+ +------------+ +-----------+

| Working | ---> | Staging | ---> | Committed|

| Directory | | Area | | History |

+------------+ +------------+ +-----------+

↑ |

+------------<--------- git pull -------+

(Remote Repository)

**🛠️ Common Git Commands**

**📁 Create or Clone**

git init # Initialize a new repo

git clone <repo-url> # Clone a remote repo

**🔍 Check Status**

git status # Show modified files

**➕ Stage and Commit**

git add . # Stage all files

git commit -m "Your message" # Commit changes

**🔄 Branching**

git branch # List branches

git checkout -b feature-x # Create & switch to a new branch

**🔃 Push & Pull**

git push origin main # Push to remote repo

git pull origin main # Pull latest changes

**🛠️ Merge Changes**

git checkout main

git merge feature-x # Merge branch into main

**🌿 Git Branch Example**

**Scenario: Add a new login feature**

git checkout -b login-feature

# Make code changes

git add .

git commit -m "Add login UI and logic"

git push origin login-feature

Submit a **Pull Request** on GitHub to merge into main.

**🧽 Undo Changes**

| **Task** | **Command** |
| --- | --- |
| Unstage file | git reset <file> |
| Undo commit | git reset --soft HEAD~1 |
| Discard local changes | git checkout -- <file> |

**🤝 Git Collaboration Tips**

* Always **pull latest code** before starting
* Use **clear commit messages**
* Prefer **feature branches** for changes
* **Pull Request (PR)** workflow for reviews
* Use .gitignore to exclude unnecessary files

**📂 .gitignore Example**

# Ignore Python cache

\_\_pycache\_\_/

\*.pyc

# Ignore secrets

.env

# Ignore build files

build/

dist/

**✅ Summary**

| **Concept** | **Command** |
| --- | --- |
| Start repo | git init / git clone |
| Track changes | git add → git commit |
| Branching | git checkout -b <name> |
| Sync with remote | git push / git pull |
| Merge changes | git merge <branch> |

**💬 Bonus: Git in Jenkins**

Use Git in your Jenkinsfile:

stage('Checkout') {

steps {

git 'https://github.com/example/project.git'

}

}