FDV- 103: Git & GitHub Onboarding Guide

A guide to help get up and running with Git and GitHub for version control and collaboration.

# Section 1: Install and Set Up Git

## Step 1: Install Git

* Windows: https://git-scm.com/download/win
* macOS: Use Homebrew: brew install git
* Linux (Debian/Ubuntu): sudo apt install git

## Step 2: Configure Git

* git config --global user.name "Your Full Name"
* git config --global user.email "your@email.com"

## Step 3: Set Preferred Editor (Optional)

* git config --global core.editor "nano"

## Step 4: Verify Configuration

* git config --list

# Section 2: Basic Git Commands

1. Initialize repository: git init
2. Check status: git status
3. Track a file: git add <filename>
4. Commit changes: git commit -m "Your message"
5. View history: git log
6. See diffs: git diff
7. Create a branch: git branch <branch-name>
8. Switch branch: git checkout <branch-name>
9. Merge a branch: git merge <branch-name>

# Section 3: Set Up GitHub

## Step 1: Create an Account

* Sign up at https://github.com

## Step 2: Set Up SSH Key (Recommended)

1. ssh-keygen -t ed25519 -C "your@email.com"
2. Add the key to GitHub at: https://github.com/settings/keys
3. Copy public key: cat ~/.ssh/id\_ed25519.pub

## Step 3: Test SSH Connection

* ssh -T git@github.com

# Section 4: Connecting Local Git to GitHub

## Clone a Repository

* git clone git@github.com:your-username/repo-name.git

## Add Remote to Existing Project

* git remote add origin git@github.com:your-username/repo-name.git

## Push Changes to Remote

* git push -u origin main

# Section 5: Working with Branches and Pull Requests

## Create a New Feature Branch

* git checkout -b feature/your-feature

## Push Your Branch

* git push origin feature/your-feature

## Open a Pull Request

* Go to GitHub → Your Repo → Compare & pull request

## Pull Latest Changes from Main

* git checkout main  
  git pull origin main

## Rebase Main into Feature (Advanced)

* git checkout feature/your-feature
* git fetch origin
* git rebase origin/main

# Best Practices

* Commit often with meaningful messages.
* Always pull the latest changes before pushing.
* Use feature branches for new work.
* Keep your local `main` or `master` branch clean.
* Use `.gitignore` to avoid committing sensitive or unnecessary files.