# Exercise 1 Git-HOL

# Step 1: Configure Git

Before you can use Git effectively, you need to tell Git who you are. This information (name and email) will be recorded in every commit you make. Without this, Git won’t be able to associate your changes with your identity.

1. Check if Git is installed and see the version:  
git --version  
# This confirms Git is installed. Example output: "git version 2.43.0"

2. Set your name for all Git projects on this computer:  
git config --global user.name "Your Name"  
# Replace "Your Name" with your real full name.  
# The --global flag means it applies to all repositories on this system.

3. Set your email for all Git projects on this computer:  
git config --global user.email "your.email@example.com"  
# Replace with the same email you use for GitHub or your Git hosting service.

4. Verify that the name and email were saved correctly:  
git config --list  
# This lists all Git configuration settings. Look for 'user.name' and 'user.email'.

## Step 2: Set Default Editor (Optional but Recommended)

Git sometimes needs to open an editor — for example, if you run 'git commit' without the -m flag, or if you need to resolve merge conflicts. By default, Git might use Vim or Nano, which may be unfamiliar to beginners. You can change it to an editor you are comfortable with.

1. On Windows (Notepad++):  
git config --global core.editor "notepad++ -multiInst -notabbar -nosession -noPlugin"

2. On Linux/Mac (VS Code):  
git config --global core.editor "code --wait"

3. Check current editor:  
git config core.editor

## Step 3: Create a Local Repository and Add a File

A Git repository is basically a folder with your project files plus a hidden '.git' folder that stores the entire history of your changes.

1. Create folder:  
mkdir GitDemo

2. Move into folder:  
cd GitDemo

3. Initialize Git:  
git init

4. Verify '.git' folder:  
ls -a

5. Create file:  
echo "Welcome to the GitDemo project" > welcome.txt

6. View file:  
cat welcome.txt

7. Check status:  
git status

8. Stage file:  
git add welcome.txt

9. Check status again:  
git status

10. Commit file:  
git commit -m "Add welcome.txt"

11. Final status:  
git status

## Step 4: Link to a Remote Repository and Synchronize

A remote repository is a copy of your Git repository stored on a server (e.g., GitHub, GitLab, Bitbucket). This lets you share your code with others and keep it backed up online.

1. Link to remote:  
git remote add origin <URL>

2. Push to remote:  
git push origin master

3. Pull from remote:  
git pull origin master