**1.Git-HOL**

**Step 1 – Configure Git**

1. **Check if Git is installed**

git --version

If you see something like git version 2.x.x, you’re good.

1. **Set your name and email for commits**

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

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

1. **Check your settings**

git config --list

**Step 2 – Get Notepad++ Working in Git Bash**

I’ve got E:\notepad++.exe — Git Bash has no clue what to do with notepad++ as a command, so let’s fix it.

1. **Make an alias**  
   In Git Bash, run:

echo "alias notepadpp='/e/notepad++.exe'" >> ~/.bashrc

source ~/.bashrc

Now you can just type:

notepadpp

and Notepad++ will launch.

1. **Tell Git to use Notepad++ as the default editor**

**git config --global core.editor "'/e/notepad++.exe' -multiInst -nosession"**

1. **Verify**

git config --global -e

If it opens in Notepad++, you nailed it.

**Step 3 – Create a Repo and Add a File**

1. **Make a project folder**

mkdir GitDemo

cd GitDemo

1. **Initialize Git**

git init

1. **Make a file**

echo "Welcome to Git!" > welcome.txt

1. **Check it exists**

ls

cat welcome.txt

1. **See Git status**

git status

1. **Stage it**

git add welcome.txt

1. **Commit it**

git commit

1. **Double-check**

git status

**Step 4 – Push to Remote**

1. **Make a repo on GitHub called GitDemo**
2. **Link it**

git remote add origin https://gitlab.com/userid/GitDemo.git

1. **Pull (optional)**

git pull origin master

1. **Push your changes**

git push -u origin master