

Practice 2: Practice Exercise

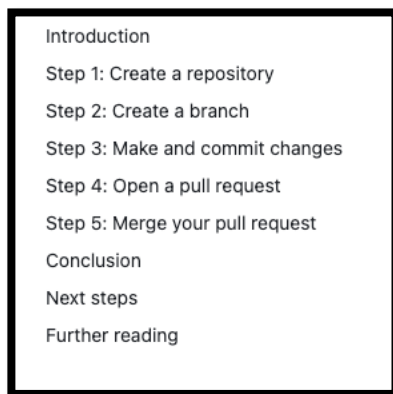
1-Create a GitHub account if you don't have one, follow instructions here:

<https://docs.github.com/en/get-started/start-your-journey/creating-an-account-on-github>

2-Complete exercise "Hello World":

<https://docs.github.com/en/get-started/start-your-journey/hello-world>

You should have completed the following:



3-Mini Git Command-line Exercise

-Prerequisites:

- Git installed (verify in your terminal `git --version`, to see if you have it installed already), otherwise installed from: <https://git-scm.com/download/win>
- A GitHub account
- A terminal (Command Prompt, PowerShell, or macOS/Linux terminal)

Follow steps below, **the code to type in terminal is in italic, open your terminal**:

Navigate to your Desktop folder (follow instructor for this step!)

Step 1: Create a new local repository named git-demo

mkdir git-demo

cd git-demo

git init

You have just created an empty Git repository.

Step 2: Add a new file

```
echo "Hello Git!" > hello.txt
```

```
git status
```

You will see hello.txt as “untracked”.

Step 3: Stage and commit your change

```
git add hello.txt
```

```
git commit -m "Add hello.txt file"
```

You have saved your first snapshot (commit).

Step 4: Create a new branch

```
git branch feature-update
```

```
git checkout feature-update
```

Now you’re on a new branch called feature-update.

Step 5: Make a change in your branch

```
echo "This is a new feature." >> hello.txt
```

```
git add hello.txt
```

```
git commit -m "Add new feature line"
```

You made a change only in your branch.

Step 6: Merge the feature back into main

```
git branch (you should see 2 branches)
```

```
git checkout main (to switch to master branch)
```

```
git merge feature-update (you should see 1 file changed, 0 insertions(+), 0 deletions(-))
```

Your feature branch’s work is now merged into main.

Step 7: View your history

```
git log --oneline --graph
```

You will see both commits and the merge.

```
PS C:\Users\JessicaBenkaroun\Desktop\git-demo> git log --oneline --graph
* 9608649 (HEAD -> master, feature-update) Add new feature line
* dbabdc9 add hello.txt file
```

-Your feature branch (feature-update) was successfully merged into master (or fast-forwarded).

-There is no divergence between branches; they both share the same latest commit.

-The graph shows a simple linear history (no merge conflicts, no extra commits).

(Optional) Step 8: Push to GitHub

If you want to push it online:

Create **git-demo** repository on GitHub first.

```
git remote add origin https://github.com/<your-username>/git-demo.git
```

```
git push -u origin master
```

What you practiced:

Concept	Command Used
Initialise repo	git init
Track files	git add
Save changes	git commit
Branching	git branch, git checkout
Merging	git merge
History	git log