| 3. Git-HOL |
| --- |

1. Introduction:

Git branching allows you to work on new features, bug fixes, or experiments without touching the main branch (often called 'main' or 'master'). Once the work is complete, merging integrates the branch back into main. This prevents incomplete work from breaking the stable codebase.

Objectives:

- Create and switch between branches

- Make commits in a feature branch

- Merge changes back into main

- Clean up unused branches

Prerequisites:

- Git installed and configured

- Local repository already initialized (we’re using GitIgnoreDemo here)

-------------------------

PART 1: BRANCHING

-------------------------

Step 1: Create a new branch called 'GitNewBranch'

This creates the branch but does NOT switch to it yet.

git branch GitNewBranch

Step 2: List all local and remote branches

The '\*' marks the branch you’re currently on (should be 'main' at this point).

'git branch -a' shows both local and remote branches.

git branch -a

Step 3: Switch to the new branch

This moves your working directory to point to GitNewBranch so that new commits will be recorded on this branch instead of main.

git switch GitNewBranch

Step 4: Create a new file in this branch

Here we simulate developing a new feature by adding feature.html.

echo "<h1>New Feature Under Development</h1>" > feature.html

Step 5: Stage the file

Adds the file to the staging area so Git knows it should be included in the next commit.

git add feature.html

Step 6: Commit the staged changes

-m flag lets you add a commit message in one line.

git commit -m "Feat: Add feature.html for new development"

Step 7: Check the branch’s status

This should say "working tree clean" since we committed everything.

git status

-------------------------

PART 2: MERGING

-------------------------

Step 1: Switch back to the main branch

git switch main

Step 2: Compare the main branch with GitNewBranch

This shows differences between branches before merging.

You’ll see feature.html listed as a new file in GitNewBranch.

git diff GitNewBranch

Step 3: Merge GitNewBranch into main

This integrates all commits from GitNewBranch into main.

git merge GitNewBranch

Step 4: View commit history in a graph format

--oneline = short commit messages

--graph = ASCII diagram of branches/merges

--decorate = show branch names and tags

git log --oneline --graph --decorate

Step 5: Delete the merged branch

-d removes the branch locally only if it’s fully merged into main.

git branch -d GitNewBranch

Step 6: Confirm deletion by listing branches again

git branch