

6) Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge

ans)

a) Create a branch called feature-branch

`git branch feature-branch`

b) to get into that branch use

`git checkout feature-branch`

add one file here and commit it by using `git add` and `git commit`
and push to remote repository by using

`git push origin feature-branch`

To merge this feature branch into main branch use below command

`git checkout main`

`git merge feature-branch`

`git push origin main`

7 Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.

Ans)

To create a lightweight Git tag named v1.0 for a commit in your local repository, use the following command:

`git tag v1.0`

This command creates a lightweight tag for the current commit.

And to show the tag on git hub use

`git push origin v1.0`

8 Write the command to cherry-pick a range of commits from "source-branch" to the current branch.

Ans)> Create one branch called feature branch with some commits

`git checkout main`

you will get Switched to branch 'main'

use `git log feature-branch`

To show all the logs of the feature branch as shown below

commit 1777dfc0045ca4824c6388276fc71fb322e83751 (feature-branch)

Author: w3schools-test <test@w3schools.com>

Date: Tue Nov 19 20:23:11 2024 +0530

Add change in feature branch

Cherry pick command syntax like shown below

`git cherry-pick commit hash`

ex:

Use `git cherry-pick 1777dfc0045ca4824c6388276fc71fb322e83751`(commit hash) to move this commit to main branch

9) Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message
ans)

To view the details of a specific commit, including the author, date, and commit message, use the following Git command:

`git show <commit-id>`

This command will display:

- The commit ID
- Author name and email
- Commit date
- Commit message
- A diff of the changes introduced by the commit