

8. Advanced Git Operations

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

Cherry-pick in Git means selecting specific commits from one branch and applying them to another branch.

Unlike merge, it does not bring the whole branch — it only picks the commits you choose. This is useful when you want only particular changes, not everything from another branch.

Execution steps :

Step 1: Create repo + file + commit

```
mkdir cherrypick_demo
cd cherrypick_demo
git init
echo "line0" > demo.txt
git add demo.txt
git commit -m "Initial commit on main"
```

Step 2: Create source-branch and make commits

```
git checkout -b source-branch
echo "change1" >> demo.txt
git add .
git commit -m "Feature commit 1"
```

```
echo "change2" >> demo.txt
git add .
git commit -m "Feature commit 2"
```

```
echo "change3" >> demo.txt
git add .
git commit -m "Feature commit 3"
```

Step 3 : Go back to main (ready to cherry-pick)

git checkout main

At this moment

- main has **1 commit**
- source-branch has **4 commits (1 base + 3 new)**

Step 4 : See commits in source-branch and copy the hashes

git log --oneline source-branch

Example output (hashes will differ)

```
f9e8d7c Feature commit 3
d4c3b2a Feature commit 2
b2a1f0e Feature commit 1
xxxxxxx Initial commit on main
```

Note: For an inclusive range, pick oldest = b2a1f0e and newest = f9e8d7c.

Step 5 : Cherry-pick range into main

Correct format (must use two dots ..) **git cherry-pick <oldest>^..<newest>**

Example (replace with your hashes):

git cherry-pick b2a1f0e^..f9e8d7c

If a conflict occurs during cherry-pick:

1. Check the conflicting files - **git status**
2. Open the file and manually resolve the conflict markers
(<<<<<<<, =====, >>>>>>>)
3. After fixing the file, stage the changes

git add .

4. Continue the cherry-pick process

git cherry-pick --continue

If needed:

- To abort the entire cherry-pick:

git cherry-pick --abort

- To skip the current problematic commit:

git cherry-pick --skip

Step 6 : verify the result

```
git log --oneline  
cat demo.txt
```

You should see the picked commits on top of the log and the file content updated with the changes.