**Rebase:** The purpose is maintaining linear history.

**Cherry-pick**

* Cherry-picking is where you require one or sequence of commits from one branch to another
* Consider the following graph
* In the master branch we need changes from commit id 9c3cc1b
* Now the history looks as shown below
* Articles:
  + merge: <https://www.atlassian.com/git/tutorials/using-branches/git-merge>
  + fast forward merge <https://www.atlassian.com/git/tutorials/using-branches/git-merge#:~:text=Fast%20Forward%20Merge&text=However%2C%20a%20fast%2Dforward%20merge,tie%20together%20the%20two%20histories>.
  + rebase <https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase>
  + cherry pick <https://www.atlassian.com/git/tutorials/cherry-pick>
  + merge vs rebase <https://www.atlassian.com/git/tutorials/merging-vs-rebasing>

**Find out what these expressions mean in git**

* diff examples <https://www.atlassian.com/git/tutorials/saving-changes/git-diff>
* git diff ..
* git diff ^..
* HEAD~5
* HEAD~1

**Rewriting History:** We can re-write history by interactive rebase command.

* We have used interactive rebasing to
  + change commit messages
  + combine commits
  + remove commit
  + edit commits
* <https://git-scm.com/book/en/v2/Git-Tools-Rewriting-History> for re writing history

**Find a way using cli to show**

1) all the commits done on a file: Use **git log --all <filename>** to view the commits influencing <filename> in all branches.

2) show the files in a specific commit git show --pretty="" --name-only <sha1-commit-hash>

3) find all the commits between some dates git log --since=<date-from> --until=<date-to>

4) GIT graph: to represent commit in a GUI formate.

--continue: Restart the rebasing process after having resolved a merge conflict.

--abort: Abort the rebase operation and reset HEAD to the original branch. If <branch> was provided when the rebase operation was started, then HEAD will be reset to <branch>. Otherwise, HEAD will be reset to where it was when the rebase operation was started.

--skip: Restart the rebasing process by skipping the current patch.

How we can do the change commit history?

Answer: by interactive rebasing