



GIT Commands

Let's understand how git **cherry-pick** command works,

In order to test the scenario let follow certain steps and understand how it works.

- 1) Create files as shown in the below image and perform indexing on the files by running a stage command. Commit the files to the local repo such that there's a separate commit code generated for each file addition to the repository.

```
$ echo "1st file" > file1.txt
$ echo "2nd file" > file2.txt
$ echo "3rd file" > file3.txt
$ echo "4th file" > file4.txt
$ echo "5th file" > file5.txt
```

```
$ git add file1.txt && git commit -m "1st file added"
[master (root-commit) 5c981ef] 1st file added
1 file changed, 1 insertion(+)
create mode 100644 file1.txt
```

```
$ git add file2.txt && git commit -m "2nd file added"
[master c787a34] 2nd file added
1 file changed, 1 insertion(+)
create mode 100644 file2.txt
```

```
$ git add file3.txt && git commit -m "3rd file added"
[master 84d955d] 3rd file added 1 file changed, 1 insertion(+) create mode
100644 file3.txt
$ git add file4.txt && git commit -m "4th file added"
[master 759d914] 4th file added
1 file changed, 1 insertion(+)
create mode 100644 file4.txt
```

```
$ git add file5.txt && git commit -m "5th file added"
[master 33b6f3f] 5th file added
1 file changed, 1 insertion(+)
create mode 100644 file5.txt
```

Now if we run git log command, we get below status of the repository

```
$ git log
commit 33b6f3f89192d32ed51a41c57eal456bac694a0
Author: ganesh <ganesh@automationfactory.in>
Date: Sun Dec 9 17:53:43 2018 +0000
```

5th file added

```
commit 759d9146a9d812b6119497cadelc3ddad642221
Author: ganesh <ganesh@automationfactory.in>
Date: Sun Dec 9 17:53:29 2018 +0000
```

4th file added

```
commit 84d955d4767f17295dcc9ad333fd3afcbce91f468
Author: ganesh <ganesh@automationfactory.in>
```



GIT Commands

Date: Sun Dec 9 17:53:14 2018 +0000

3rd file added

```
commit c787a34bf7e97d707550a9a5896c2fc14b82e4ef
Author: ganesh <ganesh@automationfactory.in>
Date: Sun Dec 9 17:52:58 2018 +0000
```

2nd file added

```
commit 5c981efb4a4251a09a1b3b955b9f66450acabadb
Author: ganesh <ganesh@automationfactory.in>
Date: Sun Dec 9 17:51:53 2018 +0000
```

1st file added

Now delete all files and commit the change to the repository.

```
$ rm -f *.txt

$ git add --all

$ git commit -m "all files deleted"
```

Now let's have a look at the log to know about the SHA1 code.

```
$ git reflog

d378560 HEAD@{0}: commit: all files deleted
33b6f3f HEAD@{1}: commit: 5th file added
759d914 HEAD@{2}: commit: 4th file added
84d955d HEAD@{3}: commit: 3rd file added
c787a34 HEAD@{4}: commit: 2nd file added
5c981ef HEAD@{5}: commit (initial): 1st file added
```

Now to get the files back, but in a selective way, we can use git cherry-pick command as shown below,

```
$ git cherry-pick 33b6f3f
```

In order to cherry-pick selective commits, we can use below command as,

```
$ git cherry-pick master~4 master~2
[devel 4891f24] 3rd file added
1 file changed, 1 insertion(+)
create mode 100644 file3.txt
[devel a5197e1] 5th file added
1 file changed, 1 insertion(+)
create mode 100644 file5.txt
```