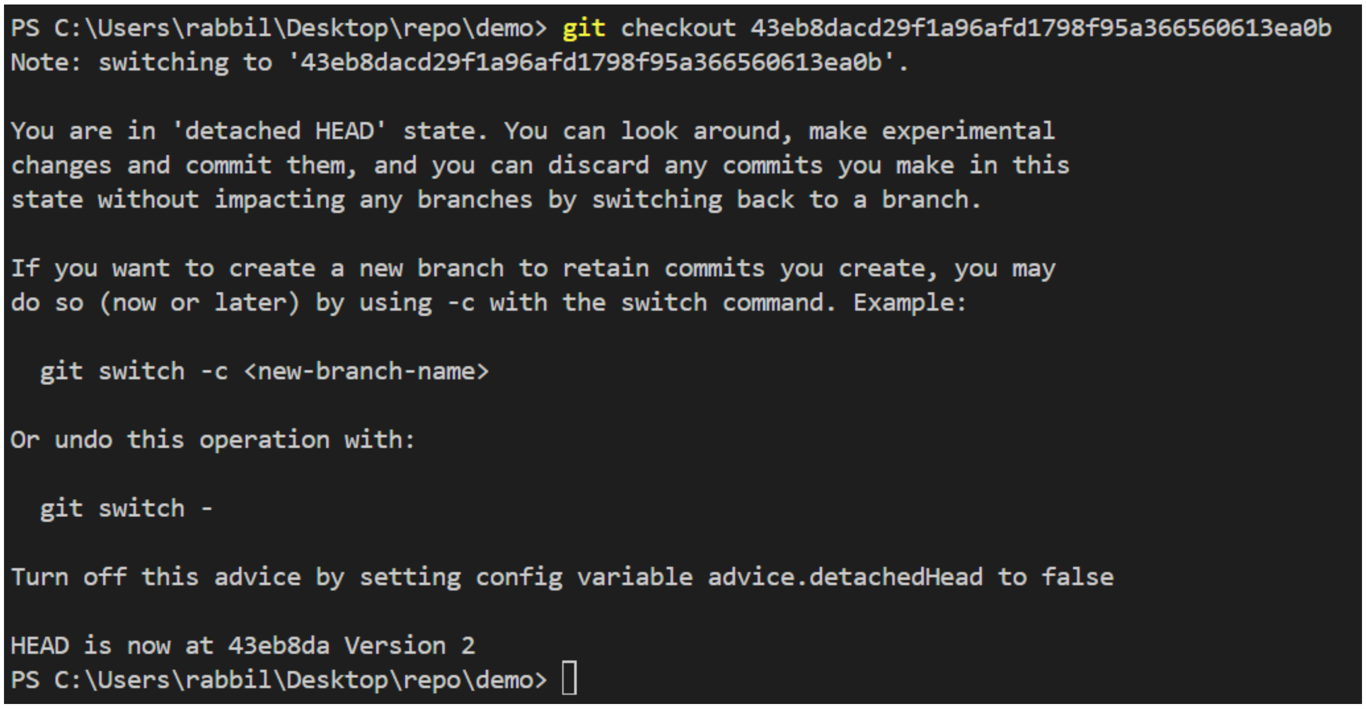
HEAD as the snapshot of your current position within the Git repository's commit history. The **git show head** is used to check the status of the Head. This command will show the location of the Head.

```
git show HEAD
```

Git Detached Head

- GitHub keeps track of all commits or snapshots over time.
- If you check the 'git log' in your terminal, you can show all the previous commits up to the first commit.
- Detached HEAD mode allows you to discover an older state of a repository. It is a natural state in Git.
- When Head doesn't point to most recent commit, such state is called detached Head.
- If you checkout with an older commit, it will stand the detached head condition

```
git checkout 43eb8dacd29f1a96afd1798f95a366560613ea0b
```



```
PS C:\Users\rabbil\Desktop\repo\demo> git checkout 43eb8dacd29f1a96afd1798f95a366560613ea0b
Note: switching to '43eb8dacd29f1a96afd1798f95a366560613ea0b'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 43eb8da Version 2
PS C:\Users\rabbil\Desktop\repo\demo>
```

---

1. `HEAD~1` : The commit before the current one.
2. `HEAD^` : The commit before the current one (same as `HEAD~1`).
3. `HEAD` : The current commit.