**Restoring a Revision in a New Local Branch**

As said, using the reset command on your HEAD branch is a quite drastic action: it will remove any commits (on this branch) that came after the specified revision. If you're sure that this is what you want, everything is fine.

However, there is also a "safer" way in case you'd prefer leaving your current HEAD branch untouched. Since "branches" are so cheap and easy in Git, we can easily create a new branch which starts at that old revision:

$ git checkout -b old-project-state 0ad5a7a6

Normally, the *checkout* command is used to just *switch* branches. However, providing the -b parameter, you can also let it **create** a new branch (named "old-project-state" in this example). If you *don't* want it to start at the current HEAD revision, you also need to provide a commit hash - the old project revision we want to restore.

Voilà: you now have a new branch named "old-project-state" reflecting the old version of your project - without touching or even removing any other commits or branches.

In case you are using the [Tower Git client](https://www.git-tower.com/?utm_source=learn-website&utm_campaign=git-faq&utm_medium=easy-in-tower&utm_content=restore-repo-to-previous-revision), you can simply drag & drop the commit you want to return to and create a new branch:

https://www.git-tower.com/learn/git/faq/restore-repo-to-previous-revision