★ *Become a git pro overnight!*

6 GIT HACKS YOU SHOULD KNOW!

Modify Recent Commit

Find Branch

Remove File from commit

Rename branches locally

What changed

Time travelling using Git

Challange

Learn all in one place
→

MODIFY THE MOST RECENT COMMIT

```
git commit --amend
```

1

The **git commit --amend** command is a convenient way to modify the most recent commit. It **lets you combine staged changes** with the previous commit instead of creating an entirely new commit.

FIND BRANCHES

Find branche(s) by commit

```
git branch --contains<commit>
```

2

If you **want to track** the remote-tracking branches you should add the **-r option.** For tracking both the local and remote-tracking branches, use the **-a option.**

REMOVE FILES
→

Remove file from last commit

```
git rm --cached <file-to-remove>
//than
git commit --amend
```

3

Let's say you committed a file by mistake. You can **quickly remove** that file from the last commit by combining the **rm** and **commit --amend** commands.

RENAME BRANCHES
→

4

Rename branches locally

```
git branch -m old-name new-name
```

For more information type
git help branch

WHAT CHANGED →

What changed in a span of time

```
git whatchanged --since = "2 weeks ago"
```

5

This command will **show a log with changes** introduced by each commit from the **last two weeks.**

BONUS + MY FAVORITE

→

Time travelling using GIT

6

```
git reflog
```

It shows you the list of all the things you've done so far.

```
af4d728 (HEAD -> master) HEAD@{0}:
069103b HEAD@{1}: reset: moving to
addcba3 HEAD@{2}: commit (amend):
584da33 HEAD@{3}: commit: :bug: aa
069103b HEAD@{4}: commit: :ambulan
8b5593b HEAD@{5}: commit: :bug:fix
4de68c9 HEAD@{6}: commit: :lipstic
aa0be7a HEAD@{7}: commit (initial)
```

```
af4d728 (HEAD -> master) HEAD@{0}:
069103b HEAD@{1}: reset: moving to
addcba3 HEAD@{2}: commit (amend):
584da33 HEAD@{3}: commit: :bug: aa
069103b HEAD@{4}: commit: :ambulan
8b5593b HEAD@{5}: commit: :bug:fix
4de68c9 HEAD@{6}: commit: :lipstic
aa0be7a HEAD@{7}: commit (initial)
```

Index Value

If you want to go back to any point in **the history,** run the below command, replacing **{index}** with that reference, **eg aa0be7a.**

```
git reset HEAD@{index}
```

CHALLANGE

@developers_community_._   X   @developers_society

I have a #challange for you

Share **two unique git commands** in the comment section & explain them in **simple english.**

Let's see how many people participate

SAVE & SWIPE
→