**Poem Fiasco**

**1.**

The file **road-not-taken.txt** doesn’t look right at all! Perhaps a rival poet snuck in and changed it while you were getting coffee. Good thing you’ve been committing often.

* Checkout the HEAD version of **road-not-taken.txt** to discard changes made to the working directory.
* Close and re-open the file to see the result.

Stuck? Get a hint

**2.**

Now, finish the poem by adding a line. Here’s a suggestion:

And that has made all the difference.

Then click Save.

**3.**

Take a look at **oven-bird.txt** to see if it has also been tampered with.

Indeed it has! We’ll want to discard changes in the working directory again.

There’s a commonly used shortcut for this command:

git checkout -- filename

It does the same exact thing that git checkout HEAD filename does. Try it with **oven-bird.txt**.

**4.**

Now, finish “Oven Bird” by adding a line. Here’s a suggestion:

Is what to make of a diminished thing.

Then click Save.

**5.**

Click on **fire-and-ice.txt**.

This file has *not* been altered, but just to be sure, check the diff for this file.

Then, add these last two lines to the poem:

Is also okay  
And would suffice.

**6.**

Now that you’ve restored and completed **road-not-taken.txt** and **oven-bird.txt** and added a line to **fire-and-ice.txt**, add all three of the files to the staging area with a single command.

**7.**

**fire-and-ice.txt** could be better. Before you commit, remove this file from the staging area.

**8.**

Now that you’ve removed **fire-and-ice.txt**, make a commit.

**9.**

You get the crazy idea to change your poems in a big way.

Make some drastic changes to each of the three poems. Remember to click Save after each file change.

**10.**

Now add all three files to the staging area.

**11.**

Make a commit.

**12.**

A little later you take a look at the current state of your poems and regret your last commit.

Reset your Git project to the commit before you made those drastic changes.

**13.**

There’s a problem: you reset HEAD to a previous commit, but the changes you want to get rid of are still in the working directory.

What Git backtracking command that you already know can discard changes to the working directory, restoring the files to the way they look in the HEAD commit?

Stuck? Here’s a [hint](https://www.codecademy.com/en/courses/learn-git/lessons/git-backtracking/exercises/git-checkout-1).