Recover Your Repository

See how to effortlessly recover the previous state of a Git repository.

We'll cover the following

- The git reset command
  - Making a “mistake”
  - Restoring the previous state
  - The --hard flag

The `git reset` command

You can use `git reset` to recover the state of the Git repository in various ways.

By default, Git will recover whatever has been added to the index/staging area and place it in your working directory.

By contrast, a `git reset --hard` will blitz all local and added changes, reverting your checkout to a just-cloned and committed state.

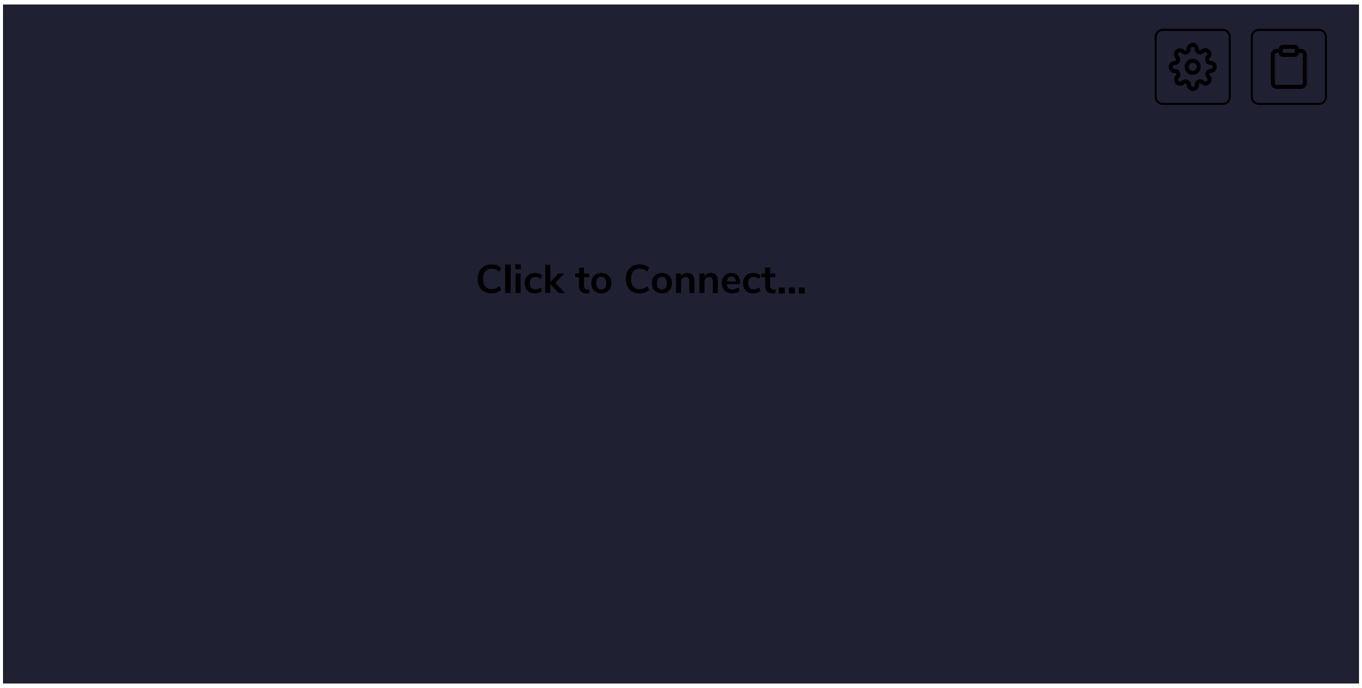
Back to our friend `git status`:

```
git status
```

What does this tell you about the state you’re in?

Terminal 1

Terminal



Making a “mistake”

Now you’re going to make another mistake:

```
git add .
```

Oh, dear. You’ve not only deleted your files, but you’ve `add`ed the deletions to the staging/index area of the Git repository.

Help me, `git status`!

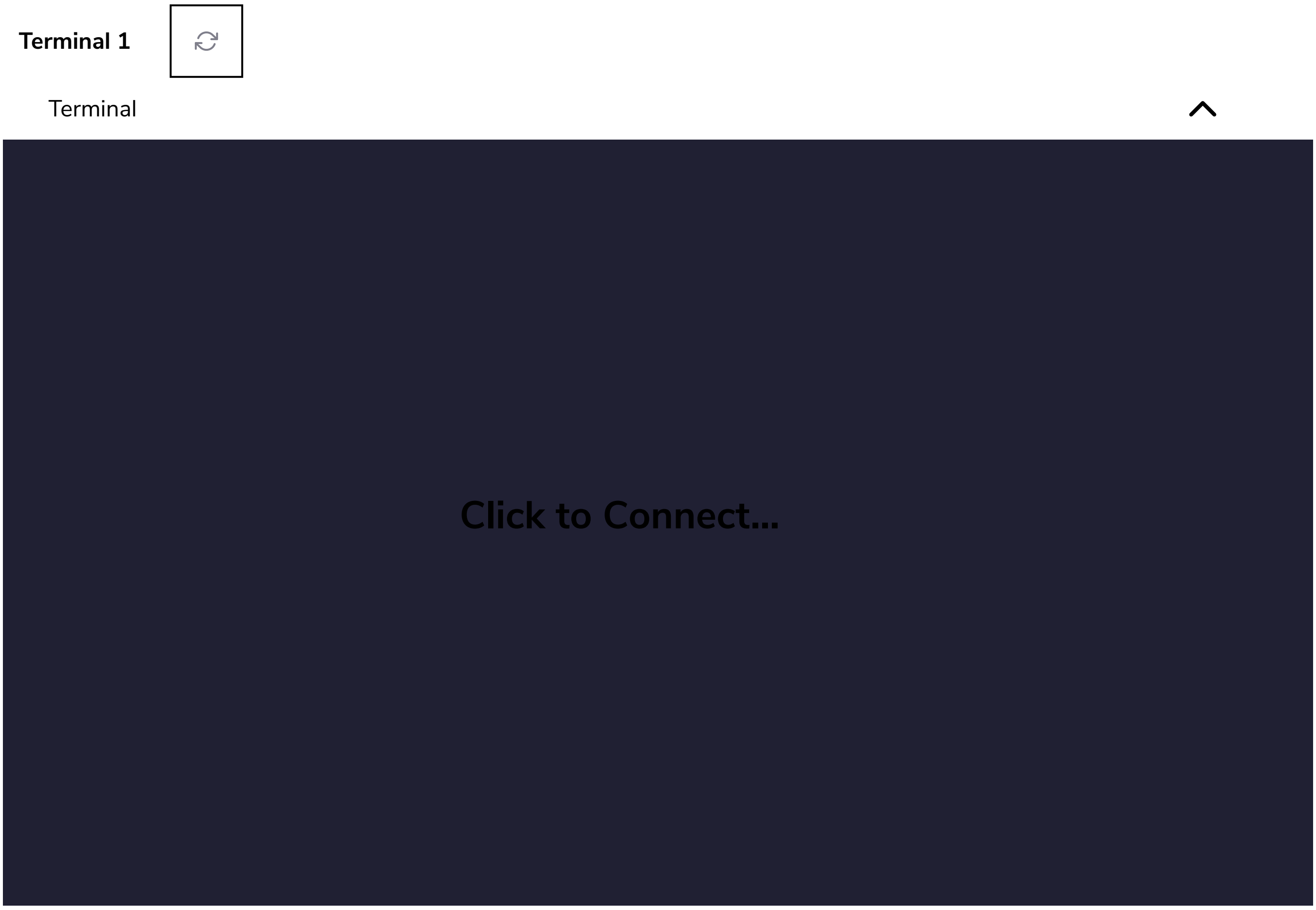
```
git status
```

Restoring the previous state

In most versions, `git status` reports that the deleted files are ready to be committed, but there’s also a helpful message at the top about using `git reset` (this message may vary depending on Git version or it might not even be there at all). If there’s a message like that, then read it carefully and type:

```
git reset --mixed
git status
```

Yes, I know that's not what `git status` told you to do. But it's the same thing. `--mixed` is the default flag that `git reset` uses.



Based on the `git status` output, can you figure out at which point you've reverted with `git reset`?

Show Hint

The `--mixed` flag takes items out of their added status but keeps them altered in the current working folder (or "working tree").

The `--hard` flag

Now, do the same delete/add cycle over but `git reset --hard` this time rather than the default `--mixed`:

```
rm -rf ../cloned_shutit/*
git add .
git reset --hard
git status
```

What does `git status` tell you now? Have the files been returned?

Terminal 1 +

Terminal

Run `git log`. What does it tell you?

So `--hard` not only takes items out of their added status, but they also make the working tree state consistent with what was last committed. You can effectively lose your changes with the `--hard` flag.

What does this tell you about the `.git` folder? How do you think the content was returned to the folder?

Show Hint

What flag do you think should be the default to `git reset`? Why? (There's no right answer by the way. It's worth thinking about why the default was chosen to embed the knowledge).

← Back

Accidental Deletion

Next →

Challenge: Clone a Repository

☑ Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/recover-your-repository__clone-a-repository__learn-git-the-hard-way)