Reverting a Commit

The main Git commands to revert a previous commit are git revert and git reset. The choice depends on the desired outcome.

- git revert: This command creates a **new commit** that undoes the changes from a specified commit. It's the **safer and preferred method** for shared repositories because it doesn't rewrite the project history. The original commit remains in the history, and the new "revert" commit nullifies its effect.
- git reset: This command moves the branch to a previous commit, **effectively rewriting the commit history**. This is generally **not recommended** for commits that have been pushed to a shared remote repository, as it can cause conflicts for other collaborators.

1. Create a New Repository: On GitHub, create a new public repository named devops-revert-exercise. Do not initialize it with a README.

2. Clone the Repository: On their local machine, open a terminal or command prompt and run:

   ```
   git clone https://github.com/YOUR_USERNAME/devops-revert-exercise.git cd devops-revert-exercise
   ```

3. Add Initial Files: Create a file named index.html with some basic HTML content.

4. Commit and Push: Add, commit, and push these initial changes to GitHub.

5. Modify the File: Open index.html and add a new line of text, simulating a mistake. For example, add This line is a mistake and has a typo. .

6. Commit the Mistake: Commit this change with a clear message indicating it's an error.

7. Verify: check your GitHub repository's commit history to see the new commit.

8. Find the Commit Hash: The learners need to find the unique ID (hash) of the "bad" commit they want to revert. They can do this by running git log.

   They should copy the first few characters of the hash next to the commit message "feat: Add incorrect text line". The hash will look something like d9f78a2.

9. Run the Revert Command:

   ```
   git revert d9f78a2
   ```

Git will open a text editor asking for a commit message. It will pre-populate it with something like "Revert 'feat: Add incorrect text line'". You can save and close the editor. A new commit is now created locally.

10. Push the Revert: Push this new commit to GitHub.

```
git push origin main
```

11. Verify the Result: Ask the learners to check the index.html file on GitHub. The "bad" line of text should be gone. They should also check the commit history, where they will now see three commits: the initial commit, the "bad" commit, and the new "revert" commit. This demonstrates that git revert adds a new commit to undo changes, preserving the project history.