# Different Features of Git Commit with Explanations and Examples

## 1. Message Description

Every commit has a message that describes the changes made. Commit messages help you and your team understand the purpose of each change.

Example:

git commit -m "Fixed bug in login feature"

## 2. Amend Previous Commit

Modify the most recent commit. If you forgot to add files or want to correct the commit message, you can amend the last commit.

Example:

git commit --amend -m "Updated login bug fix with additional tests"

## 3. Staging Partial Changes

Commit only selected changes in a file. You can stage and commit specific parts of a file instead of committing the entire file.

Example:

git add -p  
git commit -m "Fixed typo in the homepage footer"

## 4. Sign-Off (Verified Commits)

Add a sign-off to your commit. Used in projects that require contributors to certify their work (e.g., Developer Certificate of Origin).

Example:

git commit -s -m "Added unit tests for user authentication"

## 5. Committing with Author Identity

Commit changes as a specific author. Useful when multiple contributors work on a shared machine or during code reviews.

Example:

git commit --author="Jane Doe <jane.doe@example.com>" -m "Updated API documentation"

## 6. Empty Commits

Create commits with no file changes. Useful for testing or marking a point in history without modifying files.

Example:

git commit --allow-empty -m "Trigger CI pipeline"

## 7. Squashing Commits

Combine multiple commits into one. Makes your commit history cleaner by consolidating changes.

Example:

git rebase -i HEAD~3

## 8. Interactive Rebase

Modify, reorder, or squash previous commits. Allows you to refine your commit history before sharing it with others.

Example:

git rebase -i HEAD~5

## 9. Diff Viewing Before Commit

View changes before committing. Helps verify changes before finalizing the commit.

Example:

git diff --staged

## 10. GPG-Signed Commits

Sign commits with a GPG key. Ensures the authenticity of the commit and proves it was made by you.

Example:

git commit -S -m "Added encryption feature"

## 11. Commit Templates

Use a predefined template for commit messages. Helps enforce a consistent message format.

Example:

git config commit.template ~/.gitmessage.txt  
git commit

## 12. Skipping Hooks

Bypass Git hooks during commit. Use this if you want to skip pre-commit checks for specific commits.

Example:

git commit --no-verify -m "Quick fix for deployment"

## 13. Commits with Date Specification

Set a custom commit date. Useful for backdating commits or when you forgot to commit earlier.

Example:

GIT\_AUTHOR\_DATE="2025-02-20T12:34:56" GIT\_COMMITTER\_DATE="2025-02-20T12:34:56" git commit -m "Backdated bug fix"