

Git Commit Command Guide

1. Basic `git commit`

- **Command:** `git commit -m "<message>"`
- **Use:** Records changes in the staging area with a commit message.
- **Example:**

```
git commit -m "Add new feature"
```

2. Commit all changes

- **Command:** `git commit -a -m "<message>"`
- **Use:** Stages all tracked changes and commits them in one step.
- **Example:**

```
git commit -a -m "Update all modified files"
```

3. Amend previous commit

- **Command:** `git commit --amend -m "<new-message>"`
- **Use:** Modifies the last commit (useful for correcting message or adding changes).
- **Example:**

```
git commit --amend -m "Correct previous commit message"
```

4. Commit with verbose output

- **Command:** `git commit -v`
- **Use:** Shows the diff of changes in the commit message editor.
- **Example:**

```
git commit -v
```

5. Commit only specific files

- **Command:** `git commit <file1> <file2> -m "<message>"`
- **Use:** Commits only the specified files.
- **Example:**

```
git commit file1.txt file2.txt -m "Commit selected files"
```