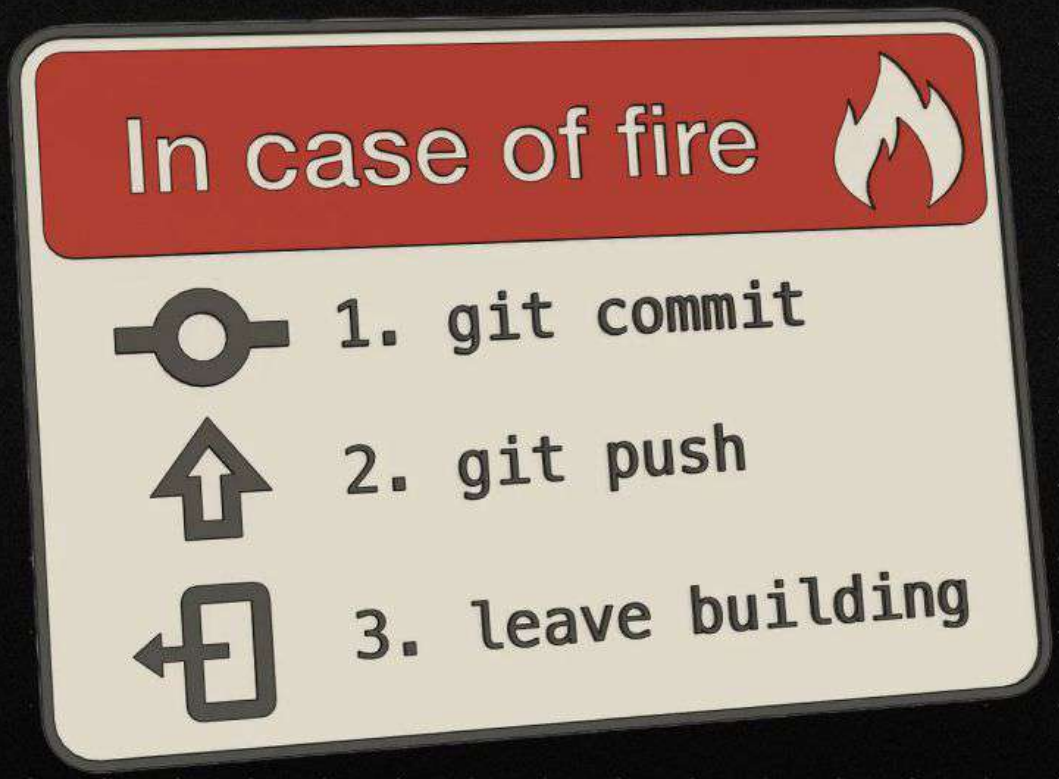
Good Commit vs. Bad Commit

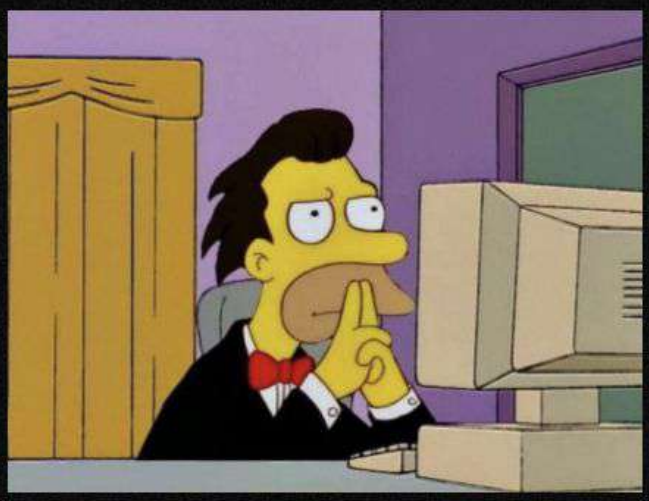


patreon.com/CodewithSloba

Swipe →

What is a Commit?

A commit in Git captures the state of your code at a specific point in time, including metadata like author, timestamp, and a message.



Commits serve to save progress, document changes, and merge contributions from different collaborators.

Keep Commits Atomic & Focused

Each commit should contain only one logical change. Avoid bundling multiple changes together.

✅ Good Commit:

commit.git

```
git commit -m "Add password encryption to user authentication"
```

❌ Bad Commit:

commit.git

```
git commit -m "Added encryption and updated UI styles"
```

Use Clear & Descriptive Messages

A commit message should explain what was changed and why. Avoid vague descriptions.

✅ Good Commit:

commit.git

```
git commit -m "fix(auth): resolve null pointer exception in login"
```

❌ Bad Commit:

commit.git

```
git commit -m "Fixed bug"
```

Follow Conventional Commit Format

Use prefixes like feat, fix, chore, docs, refactor to categorize commits.

✅ Good Commit:

commit.git

```
git commit -m "feat(api): implement JWT-based authentication"
```

❌ Bad Commit:

commit.git

```
git commit -m "Updated API"
```

Ensure Commits Are Tested

Only commit changes that have been tested and verified.

✅ Good Commit:

commit.git

```
git commit -m "fix(payment): resolve timeout issue after validating with test cases"
```

❌ Bad Commit:

commit.git

```
git commit -m "Fix payment issue (not tested yet)"
```

Keep Commit Scope Clear

A commit should affect only relevant files related to the change.

✅ Good Commit:

commit.git

```
git commit -m "refactor(auth): separate login validation logic into a helper function"
```

❌ Bad Commit:

commit.git

```
git commit -m "Refactored auth and made UI adjustments"
```

Swipe →

Write Meaningful Multi-Line Commit Messages

For complex changes, use detailed messages explaining the reason for the change.

✅ Good Commit:

commit.git

```
git commit -m "fix(cache): resolve memory leak issue
- Updated cache cleanup strategy
- Reduced unnecessary object references"
```

❌ Bad Commit:

commit.git

```
git commit -m "Fixed cache issue"
```