

A commit message: is descriptive text that is added to the commit object by the developer who made the commit. It has a title line, and an optional body.

1. Capitalization and Punctuation: Capitalize the first word and do not end in punctuation. If using Conventional Commits, remember to use all lowercase.

2. Mood: Use imperative mood in the subject line. **Example – Add fix for dark mode toggle state.** Imperative mood gives the tone you are giving an order or request.

- This convention aligns with commit messages generated by commands like `git merge` and `git revert`
- Consistency enhances speed of reading comprehension
- Tends to be more concise than the other moods

3. Add a title: Less than 50 characters

Use Title case (i.e. "Add Logging" instead of "add logging")

4. Add a body (optional):

Less than 100 characters

Explain WHAT the change is, but especially WHY the change was needed

Leave a blank line between the title and body

Separate paragraphs in the body with blank lines

Use a hyphen (-) for bullet points if needed

Use hanging indents if needed

5. **Type of Commit:** Specify the type of commit. It is recommended and can be even more beneficial to have a consistent set of words to describe your changes. Example: Bugfix, Update, Refactor, Bump, and so on. See the section on Conventional Commits below for additional information.
6. **Content:** Be direct, try to eliminate filler words and phrases in these sentences (examples: though, maybe, I think, kind of). Think like a journalist.

Examples:

```
FEAT(logger): Support JSONL log output format
```

```
FIX: Avoid deadlocking when network drops
```

```
TEST(crypt): Increase test coverage of crypt function
```

```
BREAK(users): Remove getFriends API endpoint
```

```
In accordance with new privacy policy, remove the getFriends API endpoint  
(The endpoint will continue to exist for a while and return a deprecated response)
```

```
Implements #2105
```