

Git Commit Message Format

Structure

```
<type>: <message>
<body>
```

Allowed `<type>`

- **feat**: new feature for the user, not a new feature for build script
- **fix**: bug fix for the user, not a fix to a build script
- **docs**: changes to the documentation
- **style**: formatting, missing semi colons, etc; no production code change
- **refactor**: refactoring production code, eg. renaming a variable
- **test**: adding missing tests, refactoring tests; no production code change
- **chore**: updating grunt tasks etc; no production code change

`<message>` details

1. Write in the imperative: the commit message should always be able to complete the following sentence: "If applied, this commit will *your subject line here*."
2. It is not a sentence — do not begin with a capital or end with a period.
3. Limit to a total commit message length of 80 characters.

`<body>` details

1. This section is not mandatory and can usually be left out.
2. Explain the motivation behind the change and how it compares to the previous version.
3. Continue to use the imperative tense.
4. Use `-` for bulleted lists.

Sources

- [Karma: Git Commit Msg](#)
- [Code Like a Girl: Useful Tips for writing better Git commit messages](#)