

Types		
feat	Features	A new feature
fix	Bug Fixes	A bug fix
docs	Documentation	Documentation only changes
style	Styles	Changes that do not affect the meaning of the code (white-space, formatting, missing semi-colons, etc)
refactor	Code Refactoring	A code change that neither fixes a bug nor adds a feature
perf	Performance Improvements	A code change that improves performance
test	Tests	Adding missing tests or correcting existing tests
build	Builds	Changes that affect the build system or external dependencies (example scopes: gulp, broccoli, npm)
ci	Continuous Integrations	Changes to our CI configuration files and scripts (example scopes: Travis, Circle, BrowserStack, SauceLabs)
chore	Chores	Other changes that don't modify src or test files
revert	Reverts	Reverts a previous commit

Commit message structure

<type>[optional scope]: **<description>**

[optional body]

[optional footer]

A commit that has the text `BREAKING CHANGE:` at the beginning of its optional body or footer section introduces a breaking API change

Specification

- Commits **MUST** be prefixed with a type, which consists of a verb, `feat`, `fix`, etc., followed by a colon and a space.
- The type `feat` **MUST** be used when a commit adds a new feature to your application or library.
- The type `fix` **MUST** be used when a commit represents a bug fix for your application.
- An optional scope **MAY** be provided after a type. A scope is a phrase describing a section of the codebase enclosed in parenthesis, e.g., `fix(parser):`
- A description **MUST** immediately follow the type/scope prefix. The description is a short description of the pull request, e.g., `fix: array parsing issue when multiple spaces were contained in string.`
- A longer commit body **MAY** be provided after the short description. The body **MUST** begin one blank line after the description.
- A footer **MAY** be provided one blank line after the body. The footer **SHOULD** contain additional meta-information about the pull-request (such as the issues it fixes, e.g., `fixes #13, #5`).
- Breaking changes **MUST** be indicated at the very beginning of the footer or body section of a commit. A breaking change **MUST** consist of the uppercase text `BREAKING CHANGE:`, followed by a colon and a space.
- A description **MUST** be provided after the `BREAKING CHANGE:`, describing what has changed about the API, e.g., `BREAKING CHANGE: environment variables now take precedence over config files.`
- Types other than `feat` and `fix` **MAY** be used in your commit messages.

