

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

1. Commits **MUST** be prefixed with a type, which consists of a verb, `feat`, and a space.
2. The type `feat` **MUST** be used when a commit adds a new feature to the codebase.
3. The type `fix` **MUST** be used when a commit represents a bug fix for the codebase.
4. An optional scope **MAY** be provided after a type. A scope is a phrase denoting a segment of the codebase enclosed in parenthesis, e.g., `fix(parser)`.
5. A description **MUST** immediately follow the type/scope prefix. The description **MUST** be a complete sentence. For example, `fix: array parsing issue when multiple in string.`
6. A longer commit body **MAY** be provided after the short description. The body **MUST** be a complete sentence. The body **MUST** be on a new line after the description.
7. A footer **MAY** be provided one blank line after the body. The footer **MUST** contain information about the pull-request (such as the issues it fixes, e.g., `fixes #123`).
8. Breaking changes **MUST** be indicated at the very beginning of the footer. A breaking change **MUST** consist of the uppercase text `BREAKING CHANGE:` followed by a space.
9. A description **MUST** be provided after the `BREAKING CHANGE:`, describing the change to the API, e.g., `BREAKING CHANGE: environment variables not in config files.`
10. Types other than `feat` and `fix` **MAY** be used in your commit messages.

