

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

Commit message structure
<pre><type>[optional scope]: <description> [optional body] [optional footer]</pre>
<p>A commit that has the text <code>BREAKING CHANGE:</code> at the beginning of its optional body or footer section introduces a breaking API change</p>

- ### Specification
- Commits **MUST** be prefixed with a type, which consists of a verb, `feat`, and a space.
 - The type `feat` **MUST** be used when a commit adds a new feature to the codebase.
 - The type `fix` **MUST** be used when a commit represents a bug fix for the codebase.
 - An optional scope **MAY** be provided after a type. A scope is a phrase describing the codebase enclosed in parenthesis, e.g., `fix(parser)`.
 - A description **MUST** immediately follow the type/scope prefix. The description **MUST** be present in the pull request, e.g., `fix: array parsing issue when multiple items in string`.
 - A longer commit body **MAY** be provided after the short description. The body **MUST** be present in the pull request, one line after the description.
 - A footer **MAY** be provided one blank line after the body. The footer **MUST** be present in the pull request. The footer **MUST** contain information about the pull-request (such as the issues it fixes, e.g., `fixes #123`).
 - 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.
 - A description **MUST** be provided after the `BREAKING CHANGE:`, describing the breaking change, e.g., `BREAKING CHANGE: environment variables not supported in config files`.
 - Types other than `feat` and `fix` **MAY** be used in your commit message.

