

Quick examples

- `feat: new feature`
- `fix(scope): bug in scope`
- `feat!: breaking change` / `feat(scope)!: rework API`
- `chore(deps): update dependencies`

Commit types

- `build` : Changes that affect the build system or external dependencies (example scopes: gulp, broccoli, npm)
- `ci` : Changes to CI configuration files and scripts (example scopes: Travis, Circle, BrowserStack, SauceLabs)
- `chore` : Changes which doesn't change source code or tests e.g. changes to the build process, auxiliary tools, libraries
- `docs` : Documentation only changes
- `feat` : A new feature
- `fix` : A bug fix
- `perf` : A code change that improves performance
- `refactor` : A code change that neither fixes a bug nor adds a feature
- `revert` : Revert something
- `style` : Changes that do not affect the meaning of the code (white-space, formatting, missing semi-colons, etc)
- `test` : Adding missing tests or correcting existing tests

Reminders

- Put newline before extended commit body
- More details at conventionalcommits.org