

# pre-commit-crocodile

**Author:** Adrian DC

**Description:** Git hooks intended for developers using pre-commit

**Date:** 02/02/2025

**Version:** 4.0.0-7-g582f84f

► **commitizen**

Project metadata and quality metrics:

- python** 3.9 | 3.10 | 3.11 | 3.12
- downloads** 313/month
- license** Apache License 2.0
- pipeline** manual
- bugs** 0
- code smells** 0
- coverage** 58.6%
- lines of code** 2.2k
- quality gate** passed
- pre-commit** enabled
- commitizen** friendly
- gci** enabled
- pre-commit-crocodile** enabled

Git hooks intended for developers using [pre-commit](#) and [commitizen](#).

## Configurations for [commitizen](#)

Sources / `.cz.yaml`

```
---
commitizen:

# commitizen configurations
always_signoff: true
bump_message: 'docs(changelog): regenerate release tag changes history'
changelog_incremental: false
major_version_zero: true
retry_after_failure: false
tag_format: $version
update_changelog_on_bump: false
use_shortcuts: true
version_provider: scm
version_scheme: pep440

# commitizen rules
# - Based upon 'cz_conventional_commits' original rules for Conventional Commits
# - Improved with 'Issue:' detailed messages examples and 'Signed-off-by' footer
# - Commit scope turned mandatory by the rules
# - Commit scope refactored with support for spaces and comma
# - Breaking change reimplemented with the standard '!' scope suffix
# - Questions simplified for professional usage
# - Type 'security' implemented for security related commits
# - Accept temporary commits starting with 'wip' or 'WIP'
name: cz_customize
customize:
  example: |-
    feat(scope): implement feature ABC in sources

    Issue: #123
    Details: Details about the changes
    ---

    Signed-off-by: Firstname LASTNAME <firstname.lastname@example.com>
info: |

## Conventional Commits - Customized specification

The commit contains the following structural elements:

### Commit type

- fix: Bug or issues fixes
- feat: New or extended feature
- chore: Changes to non-production code
- docs: Documentation only changes
- style: Cosmetic changes only (white-space, formatting, etc)
- refactor: Code changes that neither fix a bug nor add a feature
- perf: Code change that improve performance
- security: Changes resolve security related issues
- test: Adding missing or correcting existing tests
- build: Changes to the build system or dependencies (example: makefile, pip, docker)
- ci: Changes to CI configuration files and scripts (example: gitlab-ci)
- improvement: Changes that improve sources without any impact
- wip: Temporary commits meant for local developments and rebases

### Commit scope

A scope has to be provided to a commit's type, to provide additional contextual information,
```

such as the changed file, the containing folder or a globally modified feature, and is contained within parenthesis, e.g. feat(parser): add ability to parse arrays.

### ### Commit breaking changes

BREAKING CHANGE: A commit with '!' before the ':' or that has the text BREAKING CHANGE: at the beginning of its optional body or footer section introduces a breaking API change (correlating with MAJOR in semantic versioning).

A BREAKING CHANGE can be part of commits of any type.

### ### Commit syntax

Notice these types are not mandated by the conventional commits specification, and have no implicit effect in semantic versioning (unless they include a BREAKING CHANGE).

<type>(mandatory scope)[optional !]: <description>

[optional body]

[optional footer]

```
message_template: "\
  {{prefix}}\
  ({{scope}})\
  {% if is_breaking_change %}!{% endif %}\
  : \
  {{subject}}\
  {% if body or footer %}\n{% endif %}\
  {% if body %}\n{{body}}{% endif %}\
  {% if footer %}\n{{footer}}{% endif %}\
  {% if body or footer %}\n---{% endif %}\
"
```

questions:

- type: list
  - name: prefix
  - message: 'Type of the change .'
  - choices:
    - value: fix
      - name: 'fix: Bug or issues fixes'
      - key: x
    - value: feat
      - name: 'feat: New or extended feature'
      - key: f
    - value: chore
      - name: 'chore: Changes to non-production code'
      - key: h
    - value: docs
      - name: 'docs: Documentation only changes'
      - key: d
    - value: style
      - name: 'style: Cosmetic changes only (white-space, formatting, etc)'
      - key: s
    - value: refactor
      - name: 'refactor: Code changes that neither fix a bug nor add a feature'
      - key: r
    - value: perf
      - name: 'perf: Code change that improve performance'
      - key: p
    - value: security
      - name: 'security: Changes that improve sources without any impact'

```

    key: e
  - value: test
    name: 'test: Adding missing or correcting existing tests'
    key: t
  - value: build
    name: 'build: Changes to the build system or dependencies (example: makefile, pip, docker)'
    key: b
  - value: ci
    name: 'ci: Changes to CI configuration files and scripts (example: gitlab-ci)'
    key: c
  - value: improvement
    name: 'improvement: Changes that improve sources without any impact'
    key: i
- type: input
  name: scope
  message: 'Scope of the change :'
  filter: 'required_validator_scope'
  default: ''
- type: input
  name: subject
  message: 'Title of the commit (starting in lower case and without period) :'
  filter: 'required_validator_title_strip'
  default: ''
- type: input
  name: body
  message: 'Additional contextual message (Empty to skip) :'
  default: 'Issue: #...'
  filter: 'multiple_line_breaker'
- type: input
  name: footer
  message: 'Footer information and references (Empty to skip) :'
  default: ''
- type: confirm
  message: 'BREAKING CHANGE to warn ?'
  name: is_breaking_change
  default: False
schema: |-
  <type><scope>: <subject>
  <BLANK LINE>
  <body>
  <BLANK LINE>
  <footer>
schema_pattern: "\
  (?s)\
  ^([Ww][Ii][Pp]).*$\
  |\
  (fix|feat|chore|docs|style|refactor|perf|test|build|ci|improvement|security|revert)\
  (\^[^\n\r]+\r)\
  !?\
  :\
  \
  ([^\n\r]+\r)\
  ((\n\n.*)|(\s*))?\
  (\n[^\n\r]+-by:.*)\
  $\
  "

# commitizen styles
style:
  - ["qmark", "fg:#FF9D00 bold"]

```

```
- ["question", "bold"]
- ["answer", "fg:#FF9D00 bold"]
- ["pointer", "fg:#FF9D00 bold"]
- ["highlighted", "fg:#FF9D00 bold"]
- ["selected", "fg:#CC5454"]
- ["separator", "fg:#CC5454"]
- ["instruction", ""]
- ["text", ""]
- ["disabled", "fg:#858585 italic"]
```

pre-commit-crocodile