pre-commit-crocodile

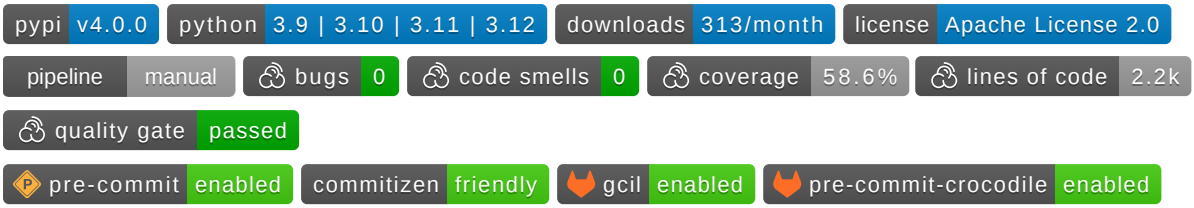
Author: Adrian DC

Description: Git hooks intended for developers using pre-commit

Date: 02/02/2025

Version: `4.0.0-7-g582f84f`

▸ Commits



Git hooks intended for developers using pre-commit and commitizen.

Table of contents

Commits commands

- `git commit` **:** Create a commit with the title prepared by `prepare-commit-message`
- `git commit -s` **:** Create a signed commit with the title prepared by `prepare-commit-message`
- `cz commit` **:** Create a commit interactively using `commitizen` by answering specific questions

---

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]
```

Commit example

```
feat(scope): implement feature ABC in sources

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

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

pre-commit-crocodile