

2. Code Best Practices

Last updated by | Harish Nedunuri | 21 Mar 2023 at 08:45 GMT

Child Pages

- [2.1 Analytics Testing Plan](#)

Best practices

- **Readability:** agree on code conventions, use auto-formatting and linters
- **OOPS:** Leverage Object-Oriented code structure to maximize reusability and keep code sparse
- **Refactoring:** Functions should be generalised and suitable for unit tests (avoid nested functions etc). Identify technical debt and address depending on the priority.
- **Fewer custom functions:** Leverage standard functions/methods from PYPI libraries as much as possible.
- **Redundancies:** remove unused imports, variables and unused functions to avoid code smells, avoid hardcoding paths or any variables in the code
- **Documentation:** Follow naming conventions, doc strings, release notes, code tags, templates
- **Debugging procedure:** Bugs can be fixed in time if they can be reproduced. Any edge case should be incorporated into the code with an appropriate unit test.
- **Align with the team,** question your memory, and ask for help.