

Roughly speaking, "best practices" can be divided into two categories - mechanics and design. Good Python mechanics adhere to [PEP 8](#) - the style guide for Python code - and [PEP 257](#) - docstring conventions. These address rules for naming, spacing, commenting, and several common programming patterns. Additionally, good general programming mechanics that you are expected to follow include removing starter code markings, removing extraneous print statements, and documenting your code. Python design refers to the higher-level organization of your code - the interfaces and implementation boundaries defined by your code objects.

- Review core Pythonic design, including *object-oriented systems thinking*
- Write consistent, scalable code using the [PEP8](#) style guide and the [Pycodestyle](#) linter
- Create well-documented code using *Docstrings*, and *Doctests*

Criteria	Meets Specifications
Write code that is PEP compliant and follows common programming best practices.	<p>The code adheres to the PEP 8 style guide and follows common best practices, including:</p> <ul style="list-style-type: none"> • Variable and function names are clear.
Create basic documentation.	<p>All included docstrings and comments adhere to PEP 8 standards</p> <p>A README file is included in the project root and includes:</p> <ul style="list-style-type: none"> • an overview of the project. • instructions for setting up and running the program. • a brief description of the roles-and-responsibilities of all
Consume public libraries using virtual environments.	<p>The code makes use of public libraries using a virtual environment.</p> <p>All required dependencies are listed in the root <code>requirements.txt</code> file which was created using the <code>\$ pip freeze > requirements.txt</code> bash command.</p> <p>The program runs with no errors.</p>
Implement basic Python exception handling.	<p>If the program encounters a common error case (e.g. attempting to load an incompatible filetype), it throws an exception.</p> <p>All exceptions include a human-readable message.</p> <p>(Optional) Make your exception handling even more awesome:</p> <ul style="list-style-type: none"> • Define custom exception classes for different types of exceptions—for things like <i>*Invalid File</i>, <i>Invalid Text Input</i> (e.g. too long)

Create Python modules.	Classes are organized into multiple directories, with related classes being placed together.
------------------------	--

Style (Mechanics)

Criteria	Meets Specifications
Produce Python code that	Submitted code follows the guidelines of PEP 8 - the Style Guide for Python.
Write Python comments that satisfy PEP 257.	Submitted code follows the docstring conventions of PEP 257. Each module contains a module-level comment describing the purpose of the module. Complex functions, classes, and methods include a docstring annotating the primary action of the callable in the imperative mood, any additional
Submit code free from starter	There are no <code># TODO</code> comments left in the submitted code. Portions of comments that say <code># ELABORATE</code> have been filled in with a description of the corresponding code.

Style (Design)

Criteria	Meets Specifications
Divide attributes appropriately among the <code>NearEarthObject</code> class and the <code>CloseApproach</code> class.	Attributes of <code>NearEarthObjects</code> and <code>CloseApproaches</code> are captured in the constructor from the supplied arguments. Instances of <code>NearEarthObject</code> don't have attributes of individual close approaches.
Attach functionality to classes only when appropriate.	Standalone functions are used when the functional operation doesn't depend on external state and does not conceptually
Design a coherent system to filter objects by user-specified criteria.	The logic backing the filter system is consistent and doesn't contain excess duplicated code.
Consume and produce streams of data as iterables.	Represents concrete data (buffered file contents, static collections of NEOs or close approaches, auxiliary data structures) as concrete. Represents streaming data (close approaches that match