HAS Tools:

Writing clean code

November 6, 2024

What makes code “clean”?

- Clean code is easy to read, follows common guidelines, and can be built off of
- Clean code also follows best practices, which are developed/accepted by the community of developers, rather than enforced by the syntax of the programming language itsel

Why make spend time to make/keep your code clean?

Code is read much more often than it is written

Code that is easy to read is easier to develop and maintain

Code is a research artifact - if you’re not sloppy with your lab work or field work you shouldn’t be sloppy with your code either

If others can use your code they are more likely to cite you

Overview of how to maintain best practices

- Python Enhancement Protocols (PEP)
  - PEP 8: Style Guide for Python Code
  - PEP 257: Docstring Conventions
  - PEP 484: Type Hints
- Writing good documentation
- Using standard project organization
- Maintain both good code layout, as well as naming conventions

Code layout

- Some overall basics:
  - Use sensible and intuitive variable & function names:
    fahrenheit_to_celsius > convert_t
  - Give constants names
  - Use comments to explain tricky parts
- PEP8 is the standard python style guide:
  https://peps.python.org/pep-0008/

Naming conventions

TYPE	NAMING CONVENTION	EXAMPLES
Function	Use lowercase words separated by underscores.	function, my_function
Variable	Use lowercase letters or word, or words separated with underscores. (I.e., snake_case)	x, var, my_variable
Class	Start each word with a capital letter. Do not separate words with underscores. (I.e., CamalCase)	Model, MyClass
Method	Use lowercase words separated with underscores.	class_method, method
Constant	Use an uppercase single letter, word, or words separated by underscores.	CONSTANT, MY_CONSTANT
Module	Use short lowercase words separated with underscores.	module.py, my_module.py
Package	Use short lowercase words without underscores.	package, mypackage

Let’s see how this works out in the codespaces