Coding

Contents

Table of contents:

- NumPy arrays
- Fundamentals of Computer Science
- Intro to Python
- Pandas
- Programming tools

Why is programming important?

For an aspiring Earth scientist, programming is a vital tool in understanding the fundamental physics of our planet and its subsystems. Programming teaches how to logically tackle problems in the most effective way, it teaches creative and computational thinking. It aids and complements mathematics and engineering.

Earth scientists often record observations of the world and use data to create a story on the mechanism. Whether we observe earthquake arrivals, flood extent, pollutant dispersion or oil flow in a well, large datasets may be beyond manual exploration. A few lines of code may unravel unexpected behaviours, or even lead to exciting discoveries.

Lastly, programming gives an advantage over other scientists in your field without that skill. It can open many industrial opportunities, as well as new research directions. It also shows your capacity to learn and that you have solution-driven thinking.

Read more:

- "Computational thinking, 10 years later" - blog post published by Microsoft Research corporate vice president.

Why coding style matters?

Python programming language is well known for its high readability and simplicity. It is said that Python is more often read than written. To achieve readability, it is highly recommended to follow coding guidelines, or "Pythonic" ways of coding.

Python Enhancement Proposals (PEPs) are primary mechanisms for adding new features into Python design. PEP 8 in particular, is a document describing recommended style guidelines for Python by Guido van Rossum, Python's designer. Style guidelines involve recommendations such as naming conventions, commenting standards, line length, white space conventions etc. Adhering to these standards makes the code consistent and easier to follow for someone unfamiliar with your work.

A few tips on what makes a good Python script:

- Create meaningful variable names, if used more than once. This makes the code easier to understand for other users.
- Add only meaningful comments to your code. No need to explain what 1 + 2 does, but it is important to explain why last index is omitted from a loop.
- Simple is better than complex, but complex is better than complicated.
- Use functions in your code - do not duplicate your code.
- Try to keep things compact, but usually keep one statement per line.
- Practice the principle of least astonishment - your code's behaviour should not surprise other users. Write it in a way that it behaves in a way most users will expect it to behave.

Good resources for style guidelines:

- Hitchhiker's Guide to Python - very compact, easy to follow guideline
- PEP 8 - the official guidelines published by Python's designer Guido van Rossum
- Google's Python Style Guide - some companies or projects will follow specialised guidelines, here is an example from Google
- NumPy guideline - a bit more specialised guideline for scientific programming package NumPy

Extra resources:

- Essential Software Engineering for Researchers - Imperial College introductory course to Python.

By Imperial College London