




# Python for Atmosphere and Ocean Scientists

Damien B Irving<sup>1</sup>

<sup>1</sup> Commonwealth Scientific and Industrial Research Organisation (CSIRO)

DOI: [10.21105/jose.00036](https://doi.org/10.21105/jose.00036)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

**Submitted:** 22 October 2018

**Published:** 12 November 2018

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC-BY).

## Summary

Python is rapidly emerging as the programming language of choice for data analysis in the atmosphere and ocean sciences. By consulting online tutorials and help pages, most researchers in this community are able to pick up the basic syntax and programming constructs (e.g. loops, lists and conditionals). This self-taught knowledge is sufficient to get work done, but it often involves spending hours to do things that should take minutes, reinventing a lot of wheels, and a nagging uncertainty at the end of it all regarding the reliability and reproducibility of the results. To help address these issues, the [Python for Atmosphere and Ocean Scientists](#) educational materials cover a suite of programming and data management best practices that are not so easy to glean from a quick Google search.

The skills covered in the materials are presented in the context of a typical data analysis task: creating a command line program that plots the average rainfall for any given month, so that the output from two different global climate models can be compared visually. The raster (or “gridded”) output from weather, climate and/or ocean models is typically stored using the Network Common Data Form (netCDF) file format and follows a strict “climate and forecasting” metadata convention. In addition to covering the basics of working with netCDF files, the materials cover the PyAOS stack (i.e. the ecosystem of libraries used in the atmosphere and ocean sciences), how to manage and share a software environment, how to write modular/reusable code, how to write scripts that behave like other command line programs, version control, defensive programming strategies and how to capture and record the provenance of data files and figures.

## Statement of need

The inspiration for the materials was the world-renowned Software Carpentry initiative (G. Wilson, 2014). In 2013, the author was involved in organising the first ever Software Carpentry workshops outside of Europe and North America, one of which was held alongside the annual conference of the Australian Meteorological and Oceanographic Society (AMOS) in Melbourne, Australia. He then trained up as a Software Carpentry instructor and taught workshops alongside the AMOS conference from 2014-2017, as well as other ad hoc workshops in various meteorology and oceanography departments. While these workshops were very popular and well received, there was clearly demand for a workshop designed specifically for atmosphere and ocean scientists. Instead of teaching generic skills in the hope that people would figure out how to apply them in their own context (i.e. in the context of netCDF files, the PyAOS stack, etc), such a workshop would teach programming (G. Wilson et al., 2014), data management and reproducible research (Irving, 2016) skills in the atmosphere and ocean science context. This idea of discipline (or data-type) specific workshops was the driving force behind the establishment of the Data Carpentry initiative, so it was with their assistance that these materials were developed.

The first pilot workshops were held in 2018 at the AMOS Conference in Sydney and at Woods Hole Oceanographic Institution in Massachusetts. The materials have since been formally endorsed and released for use by the wider Data Carpentry instructor community.

Carpentries workshops (i.e. Data Carpentry and Software Carpentry) are taught by volunteer instructors, trained in pedagogy (G. Wilson, 2018), who focus on creating a motivating and engaging environment for learners. The one-day Python for Atmosphere and Ocean Scientists workshop is no different, and incorporates a number of teaching practices that make it effective and enjoyable. These include a code of conduct to ensure the workshop is welcoming, live coding by the instructor, peers teaching peers, continuous opportunity for feedback, collaborative note taking, a lesson design that avoids cognitive overload, regular challenges/quizzes and learners work in a familiar environment by using their own laptop (G. Wilson, 2014). Of course, the materials can also be adapted and used in other forums. The first few modules, for instance, are being used in a third year undergraduate subject at the University of Wollongong.

## Future directions

Going forward, it is hoped that these materials can be collaboratively updated and improved (at the associated [GitHub repository](#)) by the atmosphere and ocean science community (Devenyi et al., 2018). Additional topics that could be covered include strategies and best practices for remapping data from one grid to another, processing very large data arrays (i.e. to reduce processing time and to avoid memory errors) and workflow management. These topics are especially relevant for the oceanography community, as ocean models tend to output large data arrays on complex grids.

## References

- Devenyi, G. A., Emonet, R., Harris, R. M., Hertweck, K. L., Irving, D., Milligan, I., & Wilson, G. (2018). Ten simple rules for collaborative lesson development. *PLOS Computational Biology*, 14, e1005963. doi:[10.1371/journal.pcbi.1005963](https://doi.org/10.1371/journal.pcbi.1005963)
- Irving, D. (2016). A minimum standard for publishing computational results in the weather and climate sciences. *Bulletin of the American Meteorological Society*, 97, 1149–1158. doi:[10.1175/BAMS-D-15-00010.1](https://doi.org/10.1175/BAMS-D-15-00010.1)
- Wilson, G. (2014). Software Carpentry: lessons learned. *F1000Research*, 3, 62. doi:[10.12688/f1000research.3-62.v2](https://doi.org/10.12688/f1000research.3-62.v2)
- Wilson, G. (2018). *Teaching tech together* (p. 300). lulu.com. Retrieved from <http://teachtogether.tech/>
- Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., Haddock, S. H. D., et al. (2014). Best practices for scientific computing. *PLoS Biology*, 12, e1001745. doi:[10.1371/journal.pbio.1001745](https://doi.org/10.1371/journal.pbio.1001745)