Johnny Wei-Bing Lin

A Hands-On Introduction to Using Python in the Atmospheric and Oceanic Sciences

http://www.johnny-lin.com/pyintro

2012

 Printed version: ISBN 978-1-300-07616-2. PDF versions: No ISBNs are assigned.

Who would *not* want to pay money for this book?: if you do not need a black-and-white paper copy of the book, a color PDF copy with functional hyperlinks, have limited funds, or are interested in such a small portion of the book that it makes no sense to buy the whole thing. The book's web site (http://www.johnny-lin.com/pyintro) has available, for free, PDFs of every chapter as separate files.

Who would want to pay money for this book?: if you want a black-and-white paper copy of the book, a color PDF copy with functional hyperlinks, or you want to help support the author financially. You can buy a black-and-white paper copy of the book at http://www.johnny-lin.com/pyintro/buypaper.shtml and a hyperlink-enabled color PDF copy of the book at http://www.johnny-lin.com/pyintro/buypdf.shtml.

A special appeal to instructors: Instruction at for-profit institutions, as a commercial use, is not covered under the terms of the CC BY-NC-SA, and so instructors at those institutions should not make copies of the book for students beyond copying permitted under Fair Use. Instruction at not-for-profit institutions is not a commercial use, so instructors may legally make copies of this book for the students in their classes, under the terms of the CC BY-NC-SA, so long as no profit is made through the copy and sale (or Fair Use is not exceeded). However, most instruction at not-for-profit institutions still involves payment of tuition: lots of people are getting paid for their contributions. Please consider also paying the author of this book something for his contribution.

Regardless of whether or not you paid money for your copy of the book, you are free to use any and all parts of the book under the terms of the CC BY-NC-SA.

Glossary

attribute data bound to an object that are designed to be acted on by methods also bound to that object.

calling execute or run a function.

class the template or "pattern" all instances of that class follow.

data coordinates a coordinate system for a plot where locations are specified by the values of the *x*- and *y*-axes data ranges.

delimit show where a sequence or collection begins and ends.

development environment an application that facilitates software development, often by providing coding, documentation, debugging, and execution tools in one place.

docstring a triple-quote delimited string that goes right after the `def` statement (or similar construct) and which provides a "help"-like description of the function.

dynamically typed variables take on the type of whatever value they are set to when they are assigned.

exception an error state in the program that cannot be processed by the current scope.

immutable a variable/object that cannot be changed.

import compile a module or package and make what is in the module or package accessible to the Python program that is doing the importing.

inherit incorporate the attribute and method definitions of another class into a definition of a new class of objects.

inheritance dealing with inheriting attribute and method definitions of another class into a definition of a new class of objects.

instance an object that is the specific realization of a class of objects.

instantiate create an instance of a class.

instantiating creating an instance of a class.

instantiation the act of creating an instance of a class.

interpreter the execution environment for Python commands.

iterable a data structure that one can go through, one element at a time; in such a structure, after you've looked at one element of it, it will move you on to the next element.

iterator used nearly interchangably with the noun form of "iterable".

method functions bound to an object that are designed to act on the data also bound to that object.

module an importable Python source code file that typically contains function, class, and variable object definitions.

multi-paradigm language a computer language that supports multiple programming methodologies, for instance, object-oriented programming and procedural programming.

mutable a variable/object that can be changed.

namespace a set of function, variable, class, etc. names; these names can be stored inside an object variable and referenced via that variable.

newline character a special text code that specifies a new line; the specific code is operating system dependent.

object a "variable" that has attached to it both data (attributes) and functions designed to act on that data (methods).

object file for a compiled language, this is a file produced by the compiler after compiling the source code file; this is *not* an object in the sense of object-oriented programming.

package a directory of importable Python source code files (and, potentially, subpackages) that typically contains function, class, and variable object definitions.

package manager a program that streamlines the installation of tools and applications as part of an operating system or distribution; this is not to be confused with a Python package, which is not, in general, an operating system or distribution package.

procedural programming a programming paradigm where a program is broken up into discrete procedures or subroutines, each of which do a specified task and communicate with the rest of the program solely (ideally) through input and output variables that are passed in argument lists and/or return values..

PyAOS a web community whose goal is to support the use of Python in the atmospheric and oceanic sciences; see http://pyaos.johnny-lin.com.

rank the number of dimensions in an array; thus, a 2-D array has rank 2.

runtime when some code or a program is actually executing.

shape a tuple whose elements are the number of elements in each dimension of an array; in Python, the elements are arranged so the fastest varying dimension is the last element in the tuple and the slowest varying dimension is the first element in the tuple.

terminal window a text window in which you can directly type in operating system and other commands.

typecode a single character string that specifies the type of the elements of a NumPy array.

Acronyms

AMS American Meteorological Society.

AOS atmospheric and oceanic sciences.

API application programming interface.

CDAT Climate Data Analysis Tools.

cdms Climate Data Management System.

CISL Computational Information Systems Laboratory.

dpi dots per inch.

EPD Enthought Python Distribution.

GCM general circulation model.

GUI graphical user interface.

HOPS Hyperslab OPerator Suite.

i/o input/output.

IDL Interactive Data Language.

LLNL Lawrence Livermore National Laboratory.

NCAR National Center for Atmospheric Research.

NGL NCAR Graphics Language.

NRCC Northeast Regional Climate Center.

OO object-oriented.

OOP object-oriented programming.

PCMDI Program for Coupled Model Diagnostics and Intercomparison.

UV-CDAT Ultrascale Visualization-Climate Data Analysis Tools.

vcs Visualization Control System.

Bibliography

Basili, V. R. and Selby, R. W. (1987). Comparing the effectiveness of software testing strategies. *IEEE Trans. Software Eng.*, SE-13(12):1278–1296.

Curtis, B. (1995). Objects of our desire: Empirical research on object-oriented development. *Human-Computer Interaction*, 10:337–344.

Lin, J. W.-B. (2009). qtcm 0.1.2: a Python implementation of the Neelin-Zeng Quasi-Equilibrium Tropical Circulation Model. *Geosci. Model Dev.*, 2:1–11, doi:10.5194/gmd–2–1–2009.

Lin, J. W.-B. (2012). Why Python is the next wave in earth sciences computing. *Bull. Amer. Meteor. Soc.*, (submitted).

Martelli, A. (2006). *Python in a Nutshell*. O'Reilly Media, Sebastopol, CA, 2nd edition.

Index

About the Author

Johnny Wei-Bing Lin graduated from Stanford University with a B.S. in Mechanical Engineering and an M.S. in Civil Engineering-Water Resources. After working as an environmental engineer, he returned to school and received his Ph.D. in Atmospheric Sciences from UCLA. His atmospheric sciences research is focused on stochastic convective parameterizations, ice-atmosphere interactions in the Arctic, and simple frameworks for modularizing climate models. He has chaired the AMS Python Symposiums and has taught or co-taught some of the AMS Python short courses. Johnny also helps coordinate the PyAOS mailing list and blog (http://pyaos.johnny-lin.com), an effort at building up the atmospheric and oceanic sciences Python community. Currently, he is a Professor of Physics at North Park University in Chicago.

Colophon

This book was written using PDFLaTeX (PDFTeX 3.1415926-1.40.10-2.2) and the Vim editor, running on an Ubuntu 12.04 GNU/Linux system. Times-like fonts are provided by the TX Fonts package (http://www.ctan.org/pkg/txfonts). The title page and examples environment are based on the "titleTMB" example from Peter Wilson's July 13, 2010 work *Some Examples of Title Pages* (http://www.ctan.org/tex-archive/info/latex-samples/TitlePages).