

JOHNNY WEI-BING LIN

---

---

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

---

---

[HTTP://WWW.JOHNNY-LIN.COM/PYINTRO](http://www.johnny-lin.com/pyintro)

2012

© 2012 Johnny Wei-Bing Lin.

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

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License (CC BY-NC-SA). To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

**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.

# Contents

<b>Preface</b>	<b>v</b>
<b>Notices</b>	<b>xi</b>
<b>How to Use This Book</b>	<b>xiii</b>
<b>1 What Is and Why Python?</b>	<b>1</b>
1.1 Python: The good and the bad . . . . .	1
1.2 Examples of AOS uses for Python . . . . .	2
<b>2 Using the Python Interpreter and Interactive Development Environment</b>	<b>7</b>
2.1 Getting and installing Python . . . . .	7
2.1.1 The easiest way: EPD . . . . .	8
2.1.2 The mostly easy way, but for Ubuntu 12.04 . . . . .	8
2.1.3 The not as easy way, but it's still free . . . . .	9
2.2 Getting and installing the course files . . . . .	9
2.3 The Python interpreter . . . . .	9
2.4 The command-line environment . . . . .	11
2.5 The IDLE environment . . . . .	12
2.6 Exercises using Python programming environments . . . . .	13
<b>3 Basic Data and Control Structures</b>	<b>17</b>
3.1 Overview of basic variables and operators . . . . .	17
3.2 Strings . . . . .	19
3.3 Booleans . . . . .	20
3.4 NoneType . . . . .	21
3.5 Lists and tuples . . . . .	22
3.6 Exercises with lists and tuples . . . . .	25
3.7 Dictionaries . . . . .	26
3.8 Exercises with dictionaries . . . . .	28

3.9	Functions . . . . .	29
3.10	Logical constructs . . . . .	33
3.11	Looping . . . . .	34
3.11.1	Looping a definite number of times . . . . .	34
3.11.2	Looping an indefinite number of times . . . . .	36
3.12	Exercises on functions, logical constructs, and looping . . .	37
3.13	Modules . . . . .	39
3.14	A brief introduction to object syntax . . . . .	41
3.15	Exercise that includes using a module . . . . .	42
3.16	Exception handling . . . . .	43
3.17	Summary . . . . .	45
<b>4</b>	<b>Array Operations</b>	<b>47</b>
4.1	What is an array and the NumPy package . . . . .	47
4.2	Creating arrays . . . . .	47
4.3	Array indexing . . . . .	50
4.4	Exercises in creating and indexing arrays . . . . .	52
4.5	Array inquiry . . . . .	53
4.6	Array manipulation . . . . .	54
4.7	General array operations . . . . .	58
4.7.1	General array operations: Method 1 (loops) . . . . .	58
4.7.2	General array operations: Method 2 (array syntax) . .	59
4.7.3	Exercise on general array operations . . . . .	62
4.8	Testing inside an array . . . . .	63
4.8.1	Testing inside an array: Method 1 (loops) . . . . .	63
4.8.2	Testing inside an array: Method 2 (array syntax) . .	64
4.8.3	Exercise on testing inside an array . . . . .	70
4.9	Additional array functions . . . . .	71
4.10	Summary . . . . .	72
<b>5</b>	<b>File Input and Output</b>	<b>73</b>
5.1	File objects . . . . .	74
5.2	Text input/output . . . . .	74
5.2.1	Text input/output: Reading a file . . . . .	75
5.2.2	Text input/output: Writing a file . . . . .	75
5.2.3	Text input/output: Processing file contents . . . . .	76
5.2.4	Exercise to read a multi-column text file . . . . .	79
5.3	NetCDF input/output . . . . .	79
5.3.1	NetCDF input/output: Reading a file . . . . .	80
5.3.2	NetCDF input/output: Writing a file . . . . .	84
5.3.3	Exercise to read and write a netCDF file . . . . .	86

5.4	Summary . . . . .	87
<b>6</b>	<b>A “Real” AOS Project: Putting Together a Basic Data Analysis Routine</b>	<b>89</b>
6.1	The assignment . . . . .	89
6.2	Solution One: Fortran-like structure . . . . .	90
6.3	Solution Two: Store results in arrays . . . . .	92
6.4	Solution Three: Store results in dictionaries . . . . .	93
6.5	Solution Four: Store results and functions in dictionaries . .	94
6.6	Exercises on using dictionaries and extending your basic data analysis routine . . . . .	95
6.7	Summary . . . . .	96
<b>7</b>	<b>An Introduction to OOP Using Python: Part I—Basic Principles and Syntax</b>	<b>97</b>
7.1	What is object-oriented programming . . . . .	97
7.1.1	Procedural vs. object-oriented programming . . . . .	98
7.1.2	The nuts and bolts of objects . . . . .	98
7.2	Example of how objects work: Strings . . . . .	99
7.3	Exercise on how objects work: Strings . . . . .	100
7.4	Example of how objects work: Arrays . . . . .	101
7.5	Exercise on how objects work: Arrays . . . . .	103
7.6	Defining your own class . . . . .	104
7.7	Exercise on defining your own class . . . . .	107
7.8	Making classes work together to make complex programming easier . . . . .	110
7.9	Case study 1: The bibliography example . . . . .	110
7.9.1	Structuring the Bibliography class . . . . .	110
7.9.2	What <code>sort_entries_alpha</code> illustrates about OOP .	112
7.9.3	Exercise in extending the Bibliography class . . .	113
7.9.4	What the <code>write_bibliog_alpha</code> method illustrates about OOP . . . . .	115
7.10	Case study 2: Creating a class for geosciences work—Surface domain management . . . . .	115
7.11	Summary . . . . .	119
<b>8</b>	<b>An Introduction to OOP Using Python: Part II—Application to Atmospheric Sciences Problems</b>	<b>121</b>
8.1	Managing metadata and missing values . . . . .	121
8.1.1	What are masked arrays and masked variables? . . .	122
8.1.2	Constructing and deconstructing masked arrays . . .	126

8.1.3	Exercise using masked arrays . . . . .	129
8.2	Managing related but unknown data: Seeing if attributes are defined . . . . .	130
8.3	Exercise to add to the <code>Atmosphere</code> class . . . . .	135
8.4	Dynamically changing subroutine execution order (optional)	137
8.5	Summary . . . . .	141
<b>9</b>	<b>Visualization: Basic Line and Contour Plots</b>	<b>143</b>
9.1	What is matplotlib? . . . . .	143
9.2	Basic line plots . . . . .	144
9.2.1	Controlling line and marker formatting . . . . .	145
9.2.2	Annotation and adjusting the font size of labels . . .	146
9.2.3	Plotting multiple figures and curves . . . . .	150
9.2.4	Adjusting the plot size . . . . .	152
9.2.5	Saving figures to a file . . . . .	152
9.3	Exercise on basic line plots . . . . .	153
9.4	Basic contour plots . . . . .	154
9.5	Exercise on basic contour plots . . . . .	156
9.6	Superimposing a map . . . . .	158
9.7	Exercise on superimposing a map . . . . .	161
9.8	Summary . . . . .	162
<b>10</b>	<b>What Next?</b>	<b>165</b>
10.1	What Python topics would be good to cover next? . . . . .	165
10.2	Some packages of interest to AOS users . . . . .	167
10.3	Additional references . . . . .	168
10.4	A parting invitation . . . . .	169
	<b>Glossary</b>	<b>171</b>
	<b>Acronyms</b>	<b>175</b>
	<b>Bibliography</b>	<b>177</b>
	<b>Index</b>	<b>179</b>

# Preface

## Why this book and who it is for

There are many good books out there to help people learn the Python programming language but none that I know of that are focused on atmospheric and oceanic sciences (AOS) users. For the highly motivated “early adopter” user, this is not a problem, but for most of us, more general resources are not as useful for two reasons:

- We do not learn programming languages as ends in and of themselves but to help us to better do our AOS work. We are not interested in the nature of object-oriented programming (OOP), but rather we are interested in whether we can use (and how to use) OOP to help us make better model runs, analyze reanalysis data, etc.
- Just because computer scientists are scientists does not mean AOS users will find computer science explanations illuminating. Different disciplines not only speak different languages but they also have different thinking processes. This is even true between the different physical sciences. Thus, learning a language is easier if the explanations and examples are geared to how AOS users think.

This book is a Python introduction for the “rest of us”: for researchers, graduate students, and advanced undergraduates who are interested in learning Python but who want to learn first and foremost how Python will help them in their *own* work. The computer science will just come along for the ride.

## What we’ll cover

I assume that readers will have had some background in procedural programming (perhaps Fortran, IDL, or Matlab) and so already understand variables,

arrays, looping, conditionals (*if/then*), simple input/output, and subroutines/functions. I also assume that in their work, most readers use a procedural programming methodology, writing programs that are broken up into subroutines and functions where input is passed in and out using argument lists (or common blocks or modules). As a result, I start out by showing how you can use Python as a more-or-less drop-in replacement for whatever traditionally procedural programming language you use (e.g., Fortran, IDL, Matlab, etc.). If this is all you want, the first half of the book (Chs. 1–6) should be all you need.

However, while Python can be used solely procedurally,<sup>1</sup> and for many AOS purposes, procedural programming works very well, Python is natively object-oriented, and so users who never tap into Python’s object-oriented nature will end up missing one of the key benefits of moving to Python. To put it another way, if you’re going to write Fortran code in Python, why not just write it in Fortran? As a result, while the first half of the book (mainly) discusses procedural programming in Python, starting in Ch. 7, I provide a gentle introduction to object-oriented programming and how object-oriented programming can be *applied* to AOS problems to enable us to write analysis and modeling programs that are more reliable, easier to maintain, and enable more science to be done.

## Is this the only Python book I’ll ever need?

If you’re looking to buy and/or use only one Python book, this is not the book for you. If you can learn from a reference book, something like Martelli (2006) would be a great book to use instead of this one: it’s comprehensive, concise, authoritative, and clearly written.

When I was learning Python, there was nothing written for a person like me, so I had to use application programming interface (API) documentation, etc. to learn the language. Because I am first an atmospheric scientist, then a programmer, this was very slow going for me. I needed someone to explain to me why Python was structured the way it was, in language that *I* (not a computer scientist) could understand. This is what this book aims to achieve.

After the introduction provided by this book, you should be better able to make full use of reference works like Martelli (2006). In Ch. 10, I provide a list of topics and resources you will want to explore after you finish this book.

---

<sup>1</sup>One of Python’s strengths is that it is a multi-paradigm language that can be used procedurally, in an object-oriented way, and/or functionally.



## Software you'll need

I assume that you have a (later) version of Python 2.x (preferably 2.7) installed on your computer, along with the NumPy, matplotlib (with Basemap), and ScientificPython packages. Although Python has now moved on to version 3.x, almost all scientific users still use a version of 2.x. Please see Section 2.1 for help on installing Python and the needed packages.

## Typesetting and coloring conventions

Throughout the book, I use different forms of typesetting and coloring to provide additional clarity and functionality (text coloring is available in only some versions of the book; later on in this preface I discuss the different versions of the book). Some of the special typesetting conventions I use include:

- Source code: Typeset in a serif, non-proportional font, as in `a = 4`.
- Commands to type on your keyboard or printed to the screen: Typeset in a serif, non-proportional font, as in `print 'hello'`.
- Generic arguments: Typeset in a serif, proportional, italicized font, in between a less than sign and a greater than sign, as in *<condition>*.
- File, directory, and executable names: Typeset in a serif, proportional, italicized font, as in */usr/bin*.

Please note that general references to application, library, module, and package names are not typeset any differently from regular text. Thus, references to the matplotlib package are typeset just as in this sentence. As most packages have unique names, this should not be confusing. In the few cases where the package names are regular English words (e.g., the time module), references to the module will hopefully be clear from the context.

Usually, the first time a key word is used and/or explained, it will be bold in the text **like this**. Key words are found in the glossary, and when useful, occurrences of those words are hyperlinked to the glossary (if the document has hyperlinks). Many acronyms are hyperlinked to the acronym list (again, if the document has hyperlinks). The glossary and acronym lists start on p. 171.

All generic text is in black. All hyperlinks (whether to locations internal or external to the document), if provided, are in blue. All margin notes are in magenta (if the version supports colors).

# Updates, additional resources, and versions of this book

Updates and additional resources are available at the book's website, <http://www.johnny-lin.com/pyintro>. This includes a link to a list of addenda and errata, the latest edition of which you can also access directly by going here: [http://www.johnny-lin.com/pyintro/addenda\\_errata.shtml](http://www.johnny-lin.com/pyintro/addenda_errata.shtml).

Some of the examples and/or exercises in this book make use of files that are available online for free at the book's website. You'll want to download these files before you start the book; the link to these course files is at the bottom of the book's web page, or you can go directly to [http://www.johnny-lin.com/pyintro/course\\_files.tar](http://www.johnny-lin.com/pyintro/course_files.tar) (which takes you to the version of these files for the latest edition of the book). In the book, these files are referred to as the files that are found in the *course\_files* directory.

There are three versions of the book available. While the textual content is the same in all of them, you should know about the differences, so you can find the one that works for you:

- Print version: Black-and-white text and margin notes; black-and-white figures; no hyperlinks; all chapters bound together; margins formatted for binding; includes a cover; costs money.
- Full PDF version: Black-and-white text and colored margin notes; color figures; colored and enabled hyperlinks; all chapters in a single file; margins formatted without binding; includes a cover; costs money.
- Free PDF version: Black-and-white text and colored margin notes; color figures; no hyperlinks; each chapter in a separate file; margins formatted without binding; does not include a cover; available for free at the book's website.

Although the PDF versions are not bound, the pagination and the formatting are kept consistent with the printed version, i.e., the same paper size and as if printed on left-right opening two-sided pages. Thus, all text, figures, and notes will be in the same relative places on the page, whether it is PDF or printed; the only difference is that with the PDF versions the margins are different to reflect the lack of a binding offset.

Information on how to obtain each version of the book is found on the copyright page (the page right after the title page). Links to access all versions of the book are also given at the book's website, <http://www.johnny-lin.com/pyintro>. Note, as a special gift for purchasing the printed version, the

original owners of a print copy of this book can download a copy of the latest edition of the Full PDF version of the book for free. Please see the last page of the printed book (the Colophon page) for details.

## Personal Acknowledgments

While I often use first person throughout this book, I am acutely aware of the debt I owe to family, friends, and colleagues who, over many years, generously nurtured many of the ideas in this book: Indeed, we all do stand on the shoulders of giants, as Newton said. All praise I happily yield to them; any mistakes and errors are my own.

Much of this book came from a series of short courses I taught at the 2011 and 2012 American Meteorological Society (AMS) Annual Meetings. I want to thank Charles Doutriaux and Dean Williams who were co-teachers in those courses and my students from those courses whose questions and comments helped refine my teaching of this material.

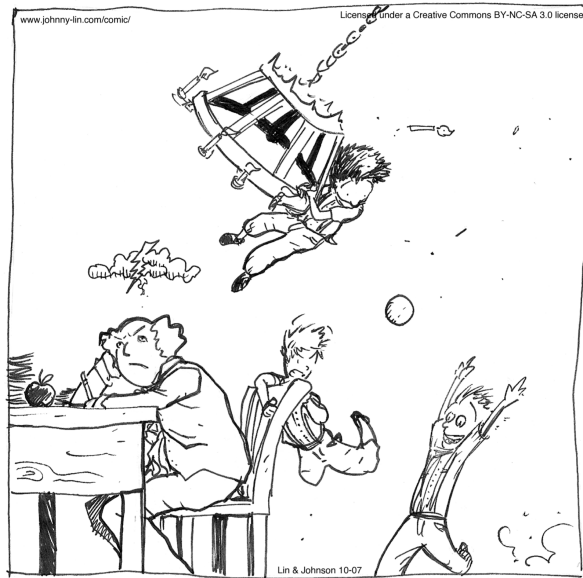
Interactions with and/or support of the following are very much appreciated: Nick Barnes, Dan Braithwaite, Dave Brown, Rodrigo Caballero, June Chang, Laura Clemente, Christian Dieterich, Tyler Erickson, Chih-Yung Feng, Mike Fiorino, Zech Gelzer, Mary Haley, Rob Jacob, Linda McDonald, Tim Olusanya, Ray Pierrehumbert, Ricky Rood, Mike Steder, and Kirsten Trout.

Thanks too for ideas and reference assistance from: Yun-Lan Chen and her colleagues at the Central Weather Bureau (Taiwan) and PyAOS commenter “Neil”<sup>2</sup> (Ch. 6); Peter Caldwell (Ch. 8); Mary Haley and the online and/or interpreter PyNGL, matplotlib, and Basemap documentation (Ch. 9); and the online and/or interpreter Python, NumPy, and CDAT documentation (multiple chapters of the book).

I am personally grateful for those who gave me permission to use material they created: These are acknowledged in the Notices section starting on p. xi and in the captions of the included or adapted figures. And my overflowing thanks to those who provided feedback on drafts of the book: Yun-Lan Chen and her colleagues at the Central Weather Bureau, Taiwan, Alex Decaria, Karen Lin, Grant Petty, and Scott Sellars.

---

<sup>2</sup>See the first comment on <http://pyaos.johnny-lin.com/?p=755> (accessed August 16, 2012).



The Real Reason Newton Couldn't  
Solve the Three Body Problem

My father and mother, sister Joann and brother-in-law Cary, and nephew and niece Tyler and Claire, have given me so much of themselves over the years. My sons Timothy and James fill my life with their infectious laughter (the above comic is firmly tongue-in-cheek ☺), and my wife Karen's love and partnership with me in this book is beyond praise. I started working at home on this book, in earnest, the same week we began to toilet train our eldest son; that alone says volumes about how sacrificially she has supported me in bringing this book to fruition. Finally, I thank my Lord and Savior Jesus Christ for giving me the one fundamental prerequisite for writing this book, life itself, both physically and spiritually: "... I have come that they may have life, and have it to the full" (John 10:10b, NIV).

Johnny Wei-Bing Lin  
Chicago, Illinois  
August 18, 2012

# Notices

## Trademark Acknowledgments

ArcGIS is a registered trademark of Environmental Systems Research Institute, Inc. Debian is a registered trademark of Software in the Public Interest, Inc. IDL is a registered trademark of Exelis Corporation. Linux is a trademark owned by Linus Torvalds. Mac, Mac OS, and OS X are registered trademarks of Apple Inc. Mathematica is a trademark of Wolfram Research, Inc. Matlab and MathWorks are registered trademarks of The MathWorks, Inc. Perl is a registered trademark of Yet Another Society. Python is a registered trademark of the Python Software Foundation. Solaris is a trademark of Oracle. Swiss Army is a registered trademark of Victorinox AG, Ibach, Switzerland and its related companies. Ubuntu is a registered trademark of Canonical Ltd. Windows is a registered trademark of Microsoft Corporation. All other marks mentioned in this book are the property of their respective owners. Any errors or omissions in trademark and/or other mark attribution are not meant to be assertions of trademark and/or other mark rights.

## Copyright Acknowledgments

Scripture taken from the HOLY BIBLE, NEW INTERNATIONAL VERSION®. Copyright © 1973, 1978, 1984 Biblica. Used by permission of Zondervan. All rights reserved. The “NIV” and “New International Version” trademarks are registered in the United States Patent and Trademark Office by Biblica. Use of either trademark requires the permission of Biblica.

Portions of Ch. 1 are taken and/or adapted from Lin (2012), © Copyright 2012 American Meteorological Society (AMS) and are used by permission. Permission to use figures, tables, and brief excerpts from Lin (2012) in scientific and educational works is hereby granted provided that the source is acknowledged. Any use of material in Lin (2012) that is determined to be “fair use” under Section 107 of the U.S. Copyright Act or that satisfies the

conditions specified in Section 108 of the U.S. Copyright Act (17 USC §108, as revised by P.L. 94-553) does not require the AMS's permission. Reproduction, systematic reproduction, posting in electronic form, such as on a web site or in a searchable database, or other uses of Lin (2012), except as exempted by the above statement, requires written permission or a license from the AMS. Additional details are provided in the AMS Copyright Policy, available on the AMS Web site located at <http://www.ametsoc.org> or from the AMS at 617-227-2425 or [copyright@ametsoc.org](mailto:copyright@ametsoc.org). Lin (2012) has been submitted for publication and thus copyright in that work may be transferred without further notice, and that version may no longer be accessible.

Sprinkled throughout the book are comics from *Under the Microscope*, a whimsical look at the world of geoscience, academia, and the meaning of life, by Dan Johnson and Johnny Lin, and are used by permission. The comics are available online at <http://www.johnny-lin.com/comic>. In some cases, the comics I've included are related to the section at hand; in others, they're just for fun.

All figures not created by myself are used by permission and are noted either in this acknowledgments section or in the respective figure captions. Use in this book of information from all other resources is believed to be covered under Fair Use doctrine.

## Other Usage Acknowledgments

The cover image (if a cover is provided) was resized and cropped from an image taken by the crew of Expedition 23 on-board the International Space Station (ISS) on May 25, 2010 and shows an edge-on view of the Earth's atmosphere over the Indian Ocean (outer space is to the upper right). The image is provided by the ISS Crew Earth Observations experiment and Image Science and Analysis Laboratory, at NASA's Johnson Space Center. The image is available online at <http://earthobservatory.nasa.gov/IOTD/view.php?id=44267> and is not copyrighted.

The dataset of monthly mean surface/near-surface air temperature from the NCEP/NCAR Reanalysis 1 was taken from NCEP Reanalysis data provided by the NOAA/OAR/ESRL PSD, Boulder, Colorado, USA from their web site at <http://www.esrl.noaa.gov/psd>.

# How to Use This Book

## First, the bad news: How not to use this book

Because I wrote this book to help teach atmospheric and oceanic scientists brand-new to Python how to use the language, there are a lot of things the book does not do well, and there are a lot of topics the book does not cover.

**Things the book does not do well:** This book is a bad Python reference. The order in the book matters, because the topics build on each another. This book also does a poor job of telling you how to install Python. I touch on it in Section 2.1, but that isn't the focus of the book.

**Topics that are not covered:** This book is geared for beginners so I had to leave out a lot of topics that, while important, are not central to helping beginners use *and* understand Python. Topics I have left out include: object inheritance, operating system commands, environment customization, wrapping Fortran routines, and many worthy visualization packages. I describe some of these topics in Section 10.1, but only enough to motivate the reader to look into those issues after they have a handle on the concepts I do cover.

**How not to use the book:** Along the lines of using this book as a reference, you will find the book to be sorely lacking if you ignore the exercises. Yes, every author says “do the exercises,” but in this book, the exercises are intimately interwoven with the pedagogy. Skipping the exercises is not just skipping breakfast, but more like skipping air. (I talk about this more below.)

## How to use this book

I recommend you don't think of this book as a text you're reading but as a course you're taking. (In fact, the source code and sample data files are in a directory called *course\_files* and much of the content comes from the 2012 AMS Annual Meeting Beginner's Python short course I taught.) As a course you're taking (either a self-study course or a course with an instructor), this will involve not just reading but a lot of doing.

I am a firm believer that the only real way you can learn a programming language is to program in that language—this book has “hands-on” in its title for a reason ☺. As a result, this book is built around examples and exercises for the reader. I will start off by discussing a topic, and interspersed with that discussion will be worked out examples of varying degrees of complexity. *You really have to type in the examples*, even if they are worked out and explained in the text; otherwise, you won’t get much out of the text. After discussion and examples, I provide exercises that implement the material we just discussed. *Those exercises are also crucial to getting the most out of the book*. In fact, for a number of the exercises, later discussion builds off of your attempt (successful or not) at completing the exercise.<sup>3</sup>

All examples and exercises are numbered globally; numbering does not restart with a new chapter.

Before you start using the book, you’ll want to download the supporting files for some of the examples and exercises. In the book, these will be the files referred to as residing in the directory *course\_files*. This directory of files is available online at the book’s website; see p. viii for details on how to obtain the files.

## Additional notes about the way I wrote this book

**Tone:** Because I grew up in Seattle and did my undergraduate and graduate work all in California, I’m a “son of the West” (to misappropriate Tolkien’s phrase), and so I’ve written this book in a more informal, conversational tone. (I use contractions and put in smileys ☺.) I hope you find my voice to be something of a “tour guide” of Python and that you will be encouraged to explore and try things out. Ultimately, I think this will make it both easier to learn Python and more fun.

**Repetition:** In most academic papers and books, the author will talk about a topic once and then (often) assume you remember what was said for the rest of this book. I always found it frustrating when textbook authors followed that convention. Repetition is one of the best ways to teach (and learn); why don’t textbook authors take advantage of this pedagogy? Besides, if I am a brand-new learner, I probably will not fully understand a concept the first time I hear it. Thus, in this book, I repeat myself, repeatedly. If you are a naturally gifted programmer, you may find this annoying, and I apologize for annoying you, but then again, I didn’t write this book for naturally

---

<sup>3</sup>This pedagogy is based on an AMS Board on Continuing Professional Development template derived from a template by Results Through Training <<http://www.RTTWorks.com>>.



Chapter	Subject	Class Hours
1	What Is and Why Python	1
2	Using the Python Interpreter and Interactive Development Environment	1
3	Basic Data and Control Structures	3
4	Array Operations	3
5	File Input and Output	2
6	A “Real” AOS Project: Putting Together a Basic Data Analysis Routine	1
7	An Introduction to OOP Using Python: Part I—Basic Principles and Syntax	3
8	An Introduction to OOP Using Python: Part II—Application to Atmospheric Sciences Problems	2
9	Visualization: Basic Line and Contour Plots	3
10	What Next?	1

Table 1: Breakdown of class time to cover each chapter.

gifted programmers but for people like myself who learned programming by looping through the topic many times.

## A note to instructors

As I mentioned earlier, this book works well for self-study but can also be used as part of a course. When I taught this as short course, we covered nearly 80% of the material over the course of two (long) days. Table 1 gives an estimate of how much class time each chapter will take, in instructor contact hours. This totals 20 class hours. Depending on whether and how you augment the content with additional homework exercises and projects, this book will work for a 2 credit hour, one quarter (10 week) class or a 1–2 credit hour, one semester (15 week) class.

The AMS 2012 Short Courses web page has PDF copies of the presentation slides for the Beginner’s course as well as an Advanced course that was also offered. You’re welcome to download and use the slides under the terms of their license: please see [http://pyaos.johnny-lin.com/?page\\_id=807](http://pyaos.johnny-lin.com/?page_id=807).

## **An online, introductory Python course**

If you're interested in an online, introductory, AOS Python course taught by the author of this book, please email me at [pyintro@johnny-lin.com](mailto:pyintro@johnny-lin.com). The course includes real-time, interactive video lectures (which can also be viewed outside of the scheduled class meeting time), not just online assignments and interaction.