

O'REILLY®

# Introducing Python

Python入门 (影印版)



東南大學出版社

Bill Lubanovic 著

---

# Python入门 (影印版)

## Introducing Python

*Bill Lubanovic* 著

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

**O'REILLY**®

O'Reilly Media, Inc. 授权东南大学出版社出版

南京 东南大学出版社

## 图书在版编目(CIP)数据

Python 入门:英文/(美)卢布诺维克(Lubanovic,  
B.)著. —影印本. —南京:东南大学出版社,2015.9

书名原文:Introducing Python

ISBN 978-7-5641-5913-9

I. ①P… II. ①卢… III. ①软件工具—程序设计—英文 IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2015)第 165747 号

江苏省版权局著作权合同登记

图字:10-2015-238 号

© 2015 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2015. Authorized reprint of the original English edition, 2015 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2015。

英文影印版由东南大学出版社出版 2015。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有,未得书面许可,本书的任何部分和全部不得以任何形式复制。

## Python 入门(影印版)

---

出版发行:东南大学出版社

地 址:南京四牌楼 2 号 邮编:210096

出 版 人:江建中

网 址:<http://www.seupress.com>

电子邮件:[press@seupress.com](mailto:press@seupress.com)

印 刷:常州市武进第三印刷有限公司

开 本:787 毫米×980 毫米 16 开本

印 张:30

字 数:588 千字

版 次:2015 年 9 月第 1 版

印 次:2015 年 9 月第 1 次印刷

书 号:ISBN 978-7-5641-5913-9

定 价:89.00 元

---

*To Mary, Karin, Tom, and Roxie.*

---

# Preface

This book will introduce you to the Python programming language. It's aimed at beginning programmers, but even if you've written programs before and just want to add Python to your list of languages, *Introducing Python* will get you started.

It's an unhurried introduction, taking small steps from the basics to more involved and varied topics. I mix cookbook and tutorial styles to explain new terms and ideas, but not too many at once. Real Python code is included early and often.

Even though this is an introduction, I include some topics that might seem advanced, such as NoSQL databases and message-passing libraries. I chose these because they can solve some problems better than standard solutions. You'll download and install external Python packages, which is good to know when the “batteries included” with Python don't fit your application. And it's fun to try something new.

I also include some examples of what *not* to do, especially if you've programmed in other languages and try to adapt those styles to Python. And I won't pretend that Python is perfect; I'll show you what to avoid.



Sometimes, I'll include a note such as this when something might be confusing or there's a more appropriate *Pythonic* way to do it.

## Audience

This book is for anybody interested in learning what seems to be emerging as the world's most popular computing language, whether or not you have learned any programming before.

# Outline

The first seven chapters explain Python's basics, and you should read them in order. The later chapters show how Python is used in specific application areas such as the Web, databases, networks, and so on; read them in any order you like. The first three appendices showcase Python in the arts, business, and science. Then, you see how to install Python 3 if you don't have it. Next are answers to the end-of-chapter exercises, and then finally, a few cheat sheets of useful things.

## Chapter 1

Programs are like directions for making socks or grilling potatoes. Some real Python programs give a little demonstration of the language's look, capabilities, and uses in the real world. Python fares well when compared with other languages, but it has some imperfections. An older version of Python (Python 2) is giving way to a newer one (Python 3). If you have Python 2, install Python 3 on your computer. Use the interactive interpreter to try examples from this book yourself.

## Chapter 2

This chapter shows Python's simplest data types: booleans, integers, floating-point numbers, and text strings. You also learn the basic math and text operations.

## Chapter 3

We step up to Python's higher-level built-in data structures: lists, tuples, dictionaries, and sets. You use these as you would Legos to build much more complicated structures. You learn how to step through them by using *iterators* and *comprehensions*.

## Chapter 4

In Chapter 4, you weave the data structures of the previous chapters with code structures to compare, choose, or repeat. You see how to package code in *functions* and handle errors with *exceptions*.

## Chapter 5

This chapter demonstrates how to scale out to larger code structures: modules, packages, and programs. You see where to put code and data, get data in and out, handle options, tour the Python Standard Library, and take a glance at what lies beyond.

## Chapter 6

If you've done object-oriented programming in other languages, Python is a bit more relaxed. Chapter 6 explains when to use objects and classes, and when it's better to use modules or even lists and dictionaries.

## Chapter 7

Learn to manage data like a pro. This chapter is all about text and binary data, the joy of Unicode characters, and I/O.

## Chapter 8

Data needs to go somewhere. In this chapter, you begin with basic flat files, directories, and filesystems. Then, you see how to handle common file formats such as CSV, JSON, and XML. You also explore how to store and retrieve with relational databases, and even some recent NoSQL data stores.

## Chapter 9

The Web gets its own chapter, which covers clients, servers, scraping, APIs, and frameworks. In Chapter 9, you work up a real website with request parameters and templates.

## Chapter 10

This is the hard-core system chapter. In this one, you learn to manage programs, processes, and threads; deal with dates and times; and automate some system administration tasks.

## Chapter 11

Networking is the subject here: services, protocols, and APIs. Examples range from low-level TCP sockets, to messaging libraries and queuing systems, to cloud deployment.

## Chapter 12

This chapter contains tips for Python developers, including installing, using IDEs, testing, debugging, logging, source control, and documentation. Chapter 12 also helps you to find and install useful third-party packages, package your own code for reuse, and learn where to get more information. Good luck.

## Appendix A

The first appendix delves into what people are doing with Python in the arts: graphics, music, animation, and games.

## Appendix B

Python has specific applications for business: data visualization (plots, graphs, and maps), security, and regulation.

## Appendix C

Python has a strong presence in science: math and statistics, physical science, bioscience, and medicine. Appendix C features NumPy, SciPy, and Pandas.

## Appendix D

If you don't already have Python 3 on your computer, this appendix shows you how to install it, no matter if you're running Windows, Mac OS/X, Linux, or Unix.

## Appendix E

This has the answers to the end-of-chapter exercises. Don't peek here until you've tried the exercises yourself.

## Appendix F

This appendix contains cheat sheets to use as a quick reference.

# Python Versions

Computer languages change over time as developers add features and fix mistakes. The examples in this book were written and tested while running Python version 3.3. Version 3.4 was released as this book was being edited, and I'll talk about a few of its additions. If you want to know what was added to Python and when, try the What's New in Python (<https://docs.python.org/3/whatsnew/>) page. It's a technical reference; a bit heavy when you're just starting with Python, but may be useful in the future if you ever have to get programs to work on computers with different Python versions.

# Conventions Used in This Book

The following typographical conventions are used in this book:

### *Italic*

Indicates new terms, URLs, email addresses, filenames, and file extensions.

### Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variables, functions, and data types.

### **Constant width bold**

Shows commands or other text that should be typed literally by the user.

### *Constant width italic*

Shows text that should be replaced with user-supplied values or by values determined by context.



This icon signifies a tip, suggestion, or general note.





This icon indicates a warning or caution.

## Using Code Examples

The substantial code examples in this book—although not the exercises, which are challenges for the reader—are available online for you to download (<https://github.com/madscheme/introducing-python>). This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you’re reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O’Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product’s documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*Introducing Python* by Bill Lubanovic (O’Reilly). Copyright 2015 Bill Lubanovic, 978-1-449-35936-2.”

If you feel your use of code examples falls outside fair use or the permission given here, feel free to contact us at [permissions@oreilly.com](mailto:permissions@oreilly.com).

## Safari® Books Online



*Safari Books Online* is an on-demand digital library that delivers expert content in both book and video form from the world’s leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of plans and pricing for enterprise, government, education, and individuals.

Members have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O’Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill,

Jones & Bartlett, Course Technology, and hundreds more. For more information about Safari Books Online, please visit us online.

## How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472  
800-998-9938 (in the United States or Canada)  
707-829-0515 (international or local)  
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at [http://bit.ly/introducing\\_python](http://bit.ly/introducing_python).

To comment or ask technical questions about this book, send email to [bookquestions@oreilly.com](mailto:bookquestions@oreilly.com).

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

## Acknowledgments

Thanks go to the many people who read and commented on my draft. I'd like to particularly mention the careful reviews by Eli Bessert, Henry Canival, Jeremy Elliott, Monte Milanuk, Loïc Pefferkorn, and Steven Wayne.

---

# Table of Contents

<b>Preface.....</b>	<b>xv</b>
<b>1. A Taste of Py.....</b>	<b>1</b>
Python in the Real World	5
Python versus Language X	6
So, Why Python?	9
When Not to Use Python	9
Python 2 versus Python 3	10
Installing Python	11
Running Python	11
Using the Interactive Interpreter	11
Use Python Files	12
What's Next?	13
Your Moment of Zen	13
Things to Do	14
<b>2. Py Ingredients: Numbers, Strings, and Variables.....</b>	<b>15</b>
Variables, Names, and Objects	15
Numbers	19
Integers	19
Precedence	23
Bases	24
Type Conversions	25
How Big Is an int?	26
Floats	27
Math Functions	27
Strings	27
Create with Quotes	28
Convert Data Types by Using str()	30

Escape with \	30
Combine with +	31
Duplicate with *	32
Extract a Character with []	32
Slice with [ <i>start</i> : <i>end</i> : <i>step</i> ]	33
Get Length with len()	35
Split with split()	35
Combine with join()	36
Playing with Strings	36
Case and Alignment	37
Substitute with replace()	38
More String Things	39
Things to Do	39
<b>3. Py Filling: Lists, Tuples, Dictionaries, and Sets.....</b>	<b>41</b>
Lists and Tuples	41
Lists	42
Create with [] or list()	42
Convert Other Data Types to Lists with list()	43
Get an Item by Using [ <i>offset</i> ]	43
Lists of Lists	44
Change an Item by [ <i>offset</i> ]	45
Get a Slice to Extract Items by Offset Range	45
Add an Item to the End with append()	46
Combine Lists by Using extend() or +=	46
Add an Item by Offset with insert()	46
Delete an Item by Offset with del	47
Delete an Item by Value with remove()	47
Get an Item by Offset and Delete It by Using pop()	47
Find an Item's Offset by Value with index()	48
Test for a Value with in	48
Count Occurrences of a Value by Using count()	49
Convert to a String with join()	49
Reorder Items with sort()	49
Get Length by Using len()	50
Assign with =, Copy with copy()	50
Tuples	52
Create a Tuple by Using ()	52
Tuples versus Lists	53
Dictionaries	53
Create with {}	54
Convert by Using dict()	54

Add or Change an Item by [ <i>key</i> ]	55
Combine Dictionaries with <code>update()</code>	56
Delete an Item by Key with <code>del</code>	57
Delete All Items by Using <code>clear()</code>	57
Test for a Key by Using <code>in</code>	58
Get an Item by [ <i>key</i> ]	58
Get All Keys by Using <code>keys()</code>	59
Get All Values by Using <code>values()</code>	59
Get All Key-Value Pairs by Using <code>items()</code>	59
Assign with <code>=</code> , Copy with <code>copy()</code>	59
Sets	60
Create with <code>set()</code>	61
Convert from Other Data Types with <code>set()</code>	61
Test for Value by Using <code>in</code>	62
Combinations and Operators	63
Compare Data Structures	66
Make Bigger Data Structures	66
Things to Do	67
<b>4. Py Crust: Code Structures.....</b>	<b>69</b>
Comment with <code>#</code>	69
Continue Lines with <code>\</code>	70
Compare with <code>if</code> , <code>elif</code> , and <code>else</code>	71
What Is True?	74
Repeat with <code>while</code>	75
Cancel with <code>break</code>	75
Skip Ahead with <code>continue</code>	76
Check <code>break</code> Use with <code>else</code>	76
Iterate with <code>for</code>	77
Cancel with <code>break</code>	78
Skip with <code>continue</code>	78
Check <code>break</code> Use with <code>else</code>	78
Iterate Multiple Sequences with <code>zip()</code>	79
Generate Number Sequences with <code>range()</code>	80
Other Iterators	81
Comprehensions	81
List Comprehensions	81
Dictionary Comprehensions	83
Set Comprehensions	84
Generator Comprehensions	84
Functions	85
Positional Arguments	89

Keyword Arguments	89
Specify Default Parameter Values	90
Gather Positional Arguments with *	91
Gather Keyword Arguments with **	92
Docstrings	92
Functions Are First-Class Citizens	93
Inner Functions	95
Closures	95
Anonymous Functions: the lambda() Function	96
Generators	98
Decorators	99
Namespaces and Scope	101
Uses of _ and __ in Names	103
Handle Errors with try and except	103
Make Your Own Exceptions	105
Things to Do	106
<b>5. Py Boxes: Modules, Packages, and Programs.....</b>	<b>109</b>
Standalone Programs	109
Command-Line Arguments	110
Modules and the import Statement	110
Import a Module	110
Import a Module with Another Name	112
Import Only What You Want from a Module	112
Module Search Path	113
Packages	113
The Python Standard Library	114
Handle Missing Keys with.setdefault() and.setdefaultdict()	115
Count Items with Counter()	117
Order by Key with OrderedDict()	118
Stack + Queue == deque	118
Iterate over Code Structures with itertools	119
Print Nicely with pprint()	121
More Batteries: Get Other Python Code	121
Things to Do	122
<b>6. Oh Oh: Objects and Classes.....</b>	<b>123</b>
What Are Objects?	123
Define a Class with class	124
Inheritance	126
Override a Method	127
Add a Method	128

Get Help from Your Parent with super	129
In self Defense	130
Get and Set Attribute Values with Properties	131
Name Mangling for Privacy	133
Method Types	134
Duck Typing	135
Special Methods	137
Composition	140
When to Use Classes and Objects versus Modules	141
Named Tuples	141
Things to Do	143
<b>7. Mangle Data Like a Pro.....</b>	<b>145</b>
Text Strings	145
Unicode	145
Format	152
Match with Regular Expressions	156
Binary Data	163
bytes and bytearray	164
Convert Binary Data with struct	165
Other Binary Data Tools	168
Convert Bytes/Strings with binascii()	169
Bit Operators	169
Things to Do	170
<b>8. Data Has to Go Somewhere.....</b>	<b>173</b>
File Input/Output	173
Write a Text File with write()	174
Read a Text File with read(), readline(), or readlines()	176
Write a Binary File with write()	177
Read a Binary File with read()	178
Close Files Automatically by Using with	178
Change Position with seek()	179
Structured Text Files	181
CSV	181
XML	183
HTML	185
JSON	185
YAML	188
A Security Note	189
Configuration Files	190
Other Interchange Formats	191

Serialize by Using pickle	191
Structured Binary Files	192
Spreadsheets	192
HDF5	192
Relational Databases	193
SQL	194
DB-API	195
SQLite	196
MySQL	198
PostgreSQL	198
SQLAlchemy	199
NoSQL Data Stores	205
The dbm Family	205
Memcached	206
Redis	206
Other NoSQL	215
Full-Text Databases	215
Things to Do	215
<b>9. The Web, Untangled.....</b>	<b>217</b>
Web Clients	218
Test with telnet	219
Python's Standard Web Libraries	220
Beyond the Standard Library: Requests	222
Web Servers	223
The Simplest Python Web Server	223
Web Server Gateway Interface	224
Frameworks	225
Bottle	226
Flask	228
Non-Python Web Servers	232
Other Frameworks	234
Web Services and Automation	236
The webbrowser Module	236
Web APIs and Representational State Transfer	236
JSON	237
Crawl and Scrape	237
Scrape HTML with BeautifulSoup	238
Things to Do	239
<b>10. Systems.....</b>	<b>241</b>
Files	241



Create with open()	241
Check Existence with exists()	242
Check Type with isfile()	242
Copy with copy()	243
Change Name with rename()	243
Link with link() or symlink()	243
Change Permissions with chmod()	243
Change Ownership with chown()	244
Get a Pathname with abspath()	244
Get a symlink Pathname with realpath()	244
Delete a File with remove()	244
Directories	244
Create with mkdir()	245
Delete with rmdir()	245
List Contents with listdir()	245
Change Current Directory with chdir()	246
List Matching Files with glob()	246
Programs and Processes	247
Create a Process with subprocess	247
Create a Process with multiprocessing	249
Kill a Process with terminate()	249
Calendars and Clocks	250
The datetime Module	251
Using the time Module	254
Read and Write Dates and Times	255
Alternative Modules	258
Things to Do	259
<b>11. Concurrency and Networks.....</b>	<b>261</b>
Concurrency	262
Queues	263
Processes	264
Threads	265
Green Threads and gevent	267
twisted	270
asyncio	271
Redis	271
Beyond Queues	275
Networks	276
Patterns	276
The Publish-Subscribe Model	276
TCP/IP	280