

LEARNING MADE EASY



2nd Edition

# Python<sup>®</sup>

ALL-IN-ONE

for  
**dummies**<sup>®</sup>  
A Wiley Brand



John C. Shovic, PhD  
Alan Simpson





# Python<sup>®</sup>

ALL-IN-ONE

2nd Edition

by John C. Shovic, PhD  
Alan Simpson

for  
**dummies**<sup>®</sup>  
A Wiley Brand

## Python® All-in-One For Dummies®, 2nd Edition

Published by: **John Wiley & Sons, Inc.**, 111 River Street, Hoboken, NJ 07030-5774, [www.wiley.com](http://www.wiley.com)

Copyright © 2021 by John Wiley & Sons, Inc., Hoboken, New Jersey

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at [www.wiley.com/go/permissions](http://www.wiley.com/go/permissions).

**Trademarks:** Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and may not be used without written permission. Python is a registered trademark of Python Software Foundation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit <https://hub.wiley.com/community/support/dummies>.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

Library of Congress Control Number: 2021932818

ISBN 978-1-119-78760-0 (pbk); ISBN 978-1-119-78761-7 (ebk); ISBN 978-1-119-78762-4 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

# Contents at a Glance

---

<b>Introduction</b> .....	1
<b>Book 1: Getting Started</b> .....	7
CHAPTER 1: Starting with Python. ....	9
CHAPTER 2: Interactive Mode, Getting Help, and Writing Apps. ....	29
CHAPTER 3: Python Elements and Syntax .....	53
CHAPTER 4: Building Your First Python Application .....	63
<b>Book 2: Understanding Python Building Blocks</b> .....	85
CHAPTER 1: Working with Numbers, Text, and Dates .....	87
CHAPTER 2: Controlling the Action .....	127
CHAPTER 3: Speeding Along with Lists and Tuples. ....	149
CHAPTER 4: Cruising Massive Data with Dictionaries .....	171
CHAPTER 5: Wrangling Bigger Chunks of Code. ....	195
CHAPTER 6: Doing Python with Class .....	217
CHAPTER 7: Sidestepping Errors .....	251
<b>Book 3: Working with Libraries</b> .....	269
CHAPTER 1: Working with External Files. ....	271
CHAPTER 2: Juggling JSON Data .....	307
CHAPTER 3: Interacting with the Internet. ....	327
CHAPTER 4: Libraries, Packages, and Modules .....	343
<b>Book 4: Using Artificial Intelligence</b> .....	357
CHAPTER 1: Exploring Artificial Intelligence. ....	359
CHAPTER 2: Building a Neural Network .....	369
CHAPTER 3: Doing Machine Learning .....	399
CHAPTER 4: Exploring AI .....	421
<b>Book 5: Doing Data Science</b> .....	433
CHAPTER 1: Understanding the Five Areas of Data Science .....	435
CHAPTER 2: Exploring Big Data .....	441
CHAPTER 3: Using Big Data from Google Cloud .....	457

<b>Book 6: Talking to Hardware</b> .....	475
CHAPTER 1: Introducing Physical Computing .....	477
CHAPTER 2: No Soldering! Using Grove Connectors for Building .....	493
CHAPTER 3: Sensing the World .....	509
CHAPTER 4: Making Things Move .....	541
<b>Book 7: Building Robots</b> .....	569
CHAPTER 1: Introducing Robotics .....	571
CHAPTER 2: Building Your First Python Robot .....	579
CHAPTER 3: Programming Your Robot Rover .....	601
CHAPTER 4: Using Artificial Intelligence in Robotics .....	631
<b>Index</b> .....	667

# Table of Contents

<b>INTRODUCTION</b> .....	1
About This Book .....	1
Foolish Assumptions .....	2
What to Buy .....	2
Icons Used in This Book .....	4
Beyond the Book .....	4
Where to Go from Here .....	5
<b>BOOK 1: GETTING STARTED</b> .....	7
<b>CHAPTER 1: Starting with Python</b> .....	9
Why Python Is Hot .....	10
Choosing the Right Python .....	11
Tools for Success .....	13
Introducing Anaconda and VS Code .....	14
Installing Anaconda and VS Code .....	15
Writing Python in VS Code .....	19
Choosing your Python interpreter .....	21
Writing some Python code .....	22
Getting back to VS Code Python .....	23
Using Jupyter Notebook for Coding .....	23
<b>CHAPTER 2: Interactive Mode, Getting Help, and Writing Apps</b> .....	29
Using Python's Interactive Mode .....	29
Opening Terminal .....	30
Getting your Python version .....	32
Going into the Python Interpreter .....	32
Entering commands .....	33
Using Python's built-in help .....	33
Exiting interactive help .....	35
Searching for specific help topics online .....	36
Lots of free cheat sheets .....	36
Creating a Python Development Workspace .....	37
Creating a Folder for Your Python Code .....	39
Typing, Editing, and Debugging Python Code .....	41
Writing Python code .....	42
Saving your code .....	43
Running Python in VS Code .....	44
Learning simple debugging .....	45
Using the VS Code Python debugger .....	46

Writing Code in a Jupyter Notebook . . . . .	47
Creating a folder for Jupyter Notebook . . . . .	47
Creating and saving a Jupyter notebook . . . . .	48
Typing and running code in a notebook . . . . .	49
Adding Markdown text . . . . .	49
Saving and opening notebooks . . . . .	51
<b>CHAPTER 3: Python Elements and Syntax . . . . .</b>	<b>53</b>
The Zen of Python . . . . .	53
Introducing Object-Oriented Programming . . . . .	56
Discovering Why Indentations Count, Big Time . . . . .	57
Using Python Modules . . . . .	59
Understanding the syntax for importing modules . . . . .	61
Using an alias with modules . . . . .	62
<b>CHAPTER 4: Building Your First Python Application . . . . .</b>	<b>63</b>
Opening the Python App File . . . . .	64
Typing and Using Python Comments . . . . .	64
Understanding Python Data Types . . . . .	66
Numbers . . . . .	67
Words (strings) . . . . .	68
Booleans . . . . .	70
Working with Python Operators . . . . .	71
Arithmetic operators . . . . .	71
Comparison operators . . . . .	72
Boolean operators . . . . .	73
Creating and Using Variables . . . . .	74
Creating valid variable names . . . . .	75
Creating variables in code . . . . .	75
Manipulating variables . . . . .	76
Saving your work . . . . .	78
Running your Python app in VS Code . . . . .	78
Understanding What Syntax Is and Why It Matters . . . . .	79
Putting Code Together . . . . .	84
<b>BOOK 2: UNDERSTANDING PYTHON BUILDING BLOCKS . . . . .</b>	<b>85</b>
<b>CHAPTER 1: Working with Numbers, Text, and Dates . . . . .</b>	<b>87</b>
Calculating Numbers with Functions . . . . .	87
Still More Math Functions . . . . .	90
Formatting Numbers . . . . .	93
Formatting with f-strings . . . . .	93
Showing dollar amounts . . . . .	94



Formatting percent numbers . . . . .	95
Making multiline format strings . . . . .	97
Formatting width and alignment. . . . .	98
Grappling with Weirder Numbers. . . . .	100
Binary, octal, and hexadecimal numbers. . . . .	100
Complex numbers. . . . .	101
Manipulating Strings. . . . .	103
Concatenating strings. . . . .	103
Getting the length of a string. . . . .	104
Working with common string operators . . . . .	105
Manipulating strings with methods . . . . .	107
Uncovering Dates and Times. . . . .	110
Working with dates . . . . .	110
Working with times . . . . .	114
Calculating timespans. . . . .	116
Accounting for Time Zones . . . . .	120
Working with Time Zones. . . . .	122
<b>CHAPTER 2: Controlling the Action . . . . .</b>	<b>127</b>
Main Operators for Controlling the Action . . . . .	127
Making Decisions with if. . . . .	129
Adding else to your if logic. . . . .	132
Handling multiple else statements with elif . . . . .	133
Ternary operations . . . . .	135
Repeating a Process with for. . . . .	136
Looping through numbers in a range . . . . .	136
Looping through a string . . . . .	138
Looping through a list. . . . .	139
Bailing out of a loop . . . . .	140
Looping with continue . . . . .	141
Nesting loops . . . . .	142
Looping with while . . . . .	143
Starting while loops over with continue. . . . .	145
Breaking while loops with break . . . . .	146
<b>CHAPTER 3: Speeding Along with Lists and Tuples. . . . .</b>	<b>149</b>
Defining and Using Lists. . . . .	149
Referencing list items by position . . . . .	150
Looping through a list. . . . .	151
Seeing whether a list contains an item. . . . .	152
Getting the length of a list . . . . .	153
Adding an item to the end of a list . . . . .	153
Inserting an item into a list . . . . .	154
Changing an item in a list. . . . .	155

	Combining lists . . . . .	155
	Removing list items . . . . .	156
	Clearing out a list . . . . .	158
	Counting how many times an item appears in a list . . . . .	159
	Finding an list item's index . . . . .	160
	Alphabetizing and sorting lists . . . . .	161
	Reversing a list . . . . .	164
	Copying a list . . . . .	164
	What's a Tuple and Who Cares? . . . . .	165
	Working with Sets . . . . .	167
<b>CHAPTER 4:</b>	<b>Cruising Massive Data with Dictionaries</b> . . . . .	<b>171</b>
	Understanding Data Dictionaries . . . . .	172
	Creating a Data Dictionary . . . . .	174
	Accessing dictionary data . . . . .	175
	Getting the length of a dictionary . . . . .	177
	Seeing whether a key exists in a dictionary . . . . .	177
	Getting dictionary data with get() . . . . .	178
	Changing the value of a key . . . . .	179
	Adding or changing dictionary data . . . . .	180
	Looping through a Dictionary . . . . .	182
	Data Dictionary Methods . . . . .	183
	Copying a Dictionary . . . . .	184
	Deleting Dictionary Items . . . . .	185
	Having Fun with Multi-Key Dictionaries . . . . .	188
	Using the mysterious fromkeys and setdefault methods . . . . .	190
	Nesting dictionaries . . . . .	193
<b>CHAPTER 5:</b>	<b>Wrangling Bigger Chunks of Code</b> . . . . .	<b>195</b>
	Creating a Function . . . . .	196
	Commenting a Function . . . . .	197
	Passing Information to a Function . . . . .	198
	Defining optional parameters with defaults . . . . .	200
	Passing multiple values to a function . . . . .	201
	Using keyword arguments (kwargs) . . . . .	203
	Passing multiple values in a list . . . . .	205
	Passing in an arbitrary number of arguments . . . . .	207
	Returning Values from Functions . . . . .	208
	Unmasking Anonymous Functions . . . . .	209
<b>CHAPTER 6:</b>	<b>Doing Python with Class</b> . . . . .	<b>217</b>
	Mastering Classes and Objects . . . . .	217
	Creating a Class . . . . .	220
	Creating an Instance from a Class . . . . .	221

Giving an Object Its Attributes . . . . .	222
Creating an instance from a class . . . . .	223
Changing the value of an attribute . . . . .	226
Defining attributes with default values . . . . .	227
Giving a Class Methods . . . . .	228
Passing parameters to methods . . . . .	230
Calling a class method by class name . . . . .	231
Using class variables . . . . .	232
Using class methods . . . . .	234
Using static methods . . . . .	236
Understanding Class Inheritance . . . . .	238
Creating the base (main) class . . . . .	240
Defining a subclass . . . . .	241
Overriding a default value from a subclass . . . . .	243
Adding extra parameters from a subclass . . . . .	243
Calling a base class method . . . . .	246
Using the same name twice . . . . .	247
<b>CHAPTER 7: Sidestepping Errors . . . . .</b>	<b>251</b>
Understanding Exceptions . . . . .	252
Handling Errors Gracefully . . . . .	254
Being Specific about Exceptions . . . . .	255
Keeping Your App from Crashing . . . . .	257
Adding an else to the Mix . . . . .	259
Using try . . . except . . . else . . . finally . . . . .	261
Raising Your Own Exceptions . . . . .	263
 <b>BOOK 3: WORKING WITH LIBRARIES . . . . .</b>	 <b>269</b>
<b>CHAPTER 1: Working with External Files . . . . .</b>	<b>271</b>
Understanding Text and Binary Files . . . . .	271
Opening and Closing Files . . . . .	273
Reading a File's Contents . . . . .	279
Looping through a File . . . . .	281
Looping with readlines() . . . . .	281
Looping with readline() . . . . .	283
Appending versus overwriting files . . . . .	284
Using tell() to determine the pointer location . . . . .	285
Moving the pointer with seek() . . . . .	286
Reading and Copying a Binary File . . . . .	287
Conquering CSV Files . . . . .	290
Opening a CSV file . . . . .	292
Converting strings . . . . .	293

Converting to integers .....	295
Converting to date .....	295
Converting to Boolean .....	297
Converting to floats .....	297
Converting from CSV to Objects and Dictionaries .....	299
Importing CSV to Python objects .....	300
Importing CSV to Python dictionaries .....	303
<b>CHAPTER 2: Juggling JSON Data .....</b>	<b>307</b>
Organizing JSON Data .....	307
Understanding Serialization .....	310
Loading Data from JSON Files .....	312
Converting an Excel date to a JSON date .....	313
Looping through a keyed JSON file .....	314
Converting Firebase timestamps to Python dates .....	317
Loading unkeyed JSON from a Python string .....	318
Loading keyed JSON from a Python string .....	319
Changing JSON data .....	320
Removing data from a dictionary .....	321
Dumping Python Data to JSON .....	322
<b>CHAPTER 3: Interacting with the Internet .....</b>	<b>327</b>
Seeing How the Web Works .....	327
Understanding the mysterious URL .....	328
Exposing the HTTP headers .....	329
Opening a URL from Python .....	331
Posting to the web with Python .....	333
Scraping the web with Python .....	334
Parsing part of a page .....	337
Storing the parsed content .....	337
Saving scraped data to a JSON file .....	340
Saving scraped data to a CSV file .....	341
<b>CHAPTER 4: Libraries, Packages, and Modules .....</b>	<b>343</b>
Understanding the Python Standard Library .....	343
Using the dir() function .....	344
Using the help() function .....	345
Exploring built-in functions .....	347
Exploring Python Packages .....	347
Importing Python Modules .....	349
Making Your Own Modules .....	352

<b>BOOK 4: USING ARTIFICIAL INTELLIGENCE</b> .....	357
<b>CHAPTER 1: Exploring Artificial Intelligence</b> .....	359
AI Is a Collection of Techniques .....	360
Neural networks .....	360
Machine learning .....	365
TensorFlow — A framework for deep learning .....	366
Current Limitations of AI .....	367
<b>CHAPTER 2: Building a Neural Network</b> .....	369
Understanding Neural Networks .....	370
Layers of neurons .....	371
Weights and biases .....	372
The activation function .....	373
Loss function .....	373
Building a Simple Neural Network in Python .....	374
The neural-net Python code .....	375
Using TensorFlow for the same neural network .....	385
Installing the TensorFlow Python library .....	386
Building a Python Neural Network in TensorFlow .....	387
Loading your data .....	388
Defining your neural-network model and layers .....	388
Compiling your model .....	388
Fitting and training your model .....	388
Evaluating the model .....	388
Breaking down the code .....	390
Checking the results .....	392
Changing to a three-layer neural network in TensorFlow and Keras .....	395
<b>CHAPTER 3: Doing Machine Learning</b> .....	399
Learning by Looking for Solutions in All the Wrong Places .....	400
Creating a Machine-Learning Network for Detecting Clothes Types .....	401
Setting up the software environment .....	402
Getting the data from the Fashion-MNIST dataset .....	403
Training the network .....	404
Testing our network .....	404
Breaking down the code .....	405
Results of the training and evaluation .....	407
Testing a single test image .....	408
Testing on external pictures .....	409
The results, round 1 .....	411

	The CNN model code .....	412
	The results, round 2 .....	414
	Visualizing with Matplotlib .....	415
	Learning More Machine Learning .....	419
<b>CHAPTER 4:</b>	<b>Exploring AI .....</b>	<b>421</b>
	Limitations of the Raspberry Pi and AI .....	421
	Adding Hardware AI to the Raspberry Pi .....	423
	AI in the Cloud .....	425
	Google Cloud .....	427
	Amazon Web Services .....	427
	IBM Cloud .....	427
	Microsoft Azure .....	428
	AI on a Graphics Card .....	428
	Where to Go for More AI Fun in Python .....	430
	<b>BOOK 5: DOING DATA SCIENCE .....</b>	<b>433</b>
<b>CHAPTER 1:</b>	<b>Understanding the Five Areas of Data Science ...</b>	<b>435</b>
	Working with Big, Big Data .....	436
	Volume .....	436
	Variety .....	437
	Velocity .....	437
	Managing volume, variety, and velocity .....	437
	Cooking with Gas: The Five-Step Process of Data Science .....	438
	Capturing the data .....	438
	Processing the data .....	438
	Analyzing the data .....	439
	Communicating the results .....	440
	Maintaining the data .....	440
<b>CHAPTER 2:</b>	<b>Exploring Big Data .....</b>	<b>441</b>
	Introducing NumPy, Pandas, and Matplotlib .....	442
	NumPy .....	442
	Pandas .....	443
	Matplotlib .....	444
	Doing Your First Data Science Project .....	444
	Diamonds are a data scientist's best friend .....	444
	Breaking down the code .....	447
	Visualizing the data with Matplotlib .....	449
<b>CHAPTER 3:</b>	<b>Using Big Data from Google Cloud .....</b>	<b>457</b>
	What Is Big Data? .....	457
	Understanding Google Cloud and BigQuery .....	458
	Google Cloud Platform .....	458
	BigQuery from Google .....	458

Computer security on the cloud . . . . .	459
Signing up for BigQuery . . . . .	460
Reading the Medicare Big Data . . . . .	460
Setting up your project and authentication . . . . .	460
The first big-data code . . . . .	463
Breaking down the code . . . . .	466
Doing a bit of analysis . . . . .	467
Payment percent by state . . . . .	470
Now some visualization . . . . .	471
Looking for the Most Polluted City in the World on an Hourly Basis . . . . .	473
<b>BOOK 6: TALKING TO HARDWARE . . . . .</b>	<b>475</b>
<b>CHAPTER 1: Introducing Physical Computing . . . . .</b>	<b>477</b>
Physical Computing Is Fun . . . . .	478
What Is a Raspberry Pi? . . . . .	478
Building Projects That Move and Sense the Environment . . . . .	480
Sensing the Environment with the Raspberry Pi . . . . .	482
GPIO pins . . . . .	482
GPIO libraries . . . . .	482
Buying and assembling the hardware for “Hello World” . . . . .	483
Controlling an LED with Python . . . . .	487
But Wait, There’s More . . . . .	489
<b>CHAPTER 2: No Soldering! Using Grove Connectors for Building . . . . .</b>	<b>493</b>
Working with the Grove System . . . . .	494
Selecting a Grove base unit . . . . .	494
Error-proofing with a Grove connector . . . . .	496
Grove Connectors . . . . .	498
Grove digital — All about those 1s and 0s . . . . .	498
Grove analog: When 1s and 0s aren’t enough . . . . .	499
Grove UART (or serial) — bit-by-bit transmission . . . . .	500
Grove I2C — Using I2C to make sense of the world . . . . .	502
Connecting with Grove Cables . . . . .	503
An example of the power of the patch! . . . . .	505
Second example: The Adafruit Ultimate GPS . . . . .	506
<b>CHAPTER 3: Sensing the World . . . . .</b>	<b>509</b>
Understanding I2C . . . . .	509
Enabling I2C on the Raspberry Pi . . . . .	511
The hardware for reading temperature and humidity . . . . .	512
Reading temperature and humidity from an I2C device using Python . . . . .	515
Breaking down the program . . . . .	518

Measuring Oxygen and a Flame . . . . .	521
Analog-to-digital converters (ADC) . . . . .	522
The Grove oxygen sensor . . . . .	522
Hooking up the oxygen experiment . . . . .	524
Breaking down the code . . . . .	527
Interpreting the results . . . . .	528
Building a Dashboard on Your Phone with Blynk . . . . .	530
HDC1080 temperature and humidity sensor redux . . . . .	530
Adding the Blynk dashboard . . . . .	531
The modified temperatureTest.py software for the Blynk app . . . . .	534
Breaking down the code . . . . .	536
Where to Go from Here . . . . .	539
<b>CHAPTER 4: Making Things Move . . . . .</b>	<b>541</b>
Exploring Electric Motors . . . . .	541
Small DC motors . . . . .	542
Servo motors . . . . .	543
Stepper motors . . . . .	543
Controlling a DC Motor . . . . .	544
Grove I2C motor driver . . . . .	545
Python DC motor software . . . . .	548
Running a Servo Motor . . . . .	551
Python servo software . . . . .	555
Breaking down the code . . . . .	556
Making a Stepper Motor Step . . . . .	558
Python stepper software . . . . .	566
Breaking down the code . . . . .	567
<b>BOOK 7: BUILDING ROBOTS . . . . .</b>	<b>569</b>
<b>CHAPTER 1: Introducing Robotics . . . . .</b>	<b>571</b>
A Robot Is Not Always Like a Human . . . . .	571
Not Every Robot Has Arms or Wheels . . . . .	572
The Wilkinson bread-making robot . . . . .	573
Baxter, the coffee-making robot . . . . .	574
The Griffin Bluetooth-enabled toaster . . . . .	575
Understanding the Main Parts of a Robot . . . . .	576
Computers . . . . .	576
Motors and actuators . . . . .	577
Communications . . . . .	577
Sensors . . . . .	577
Programming Robots . . . . .	578



<b>CHAPTER 2:</b>	<b>Building Your First Python Robot</b> .....	579
	Introducing the Mars Rover PiCar-B.....	580
	What you need for the build.....	580
	Understanding the robot components.....	581
	Assembling the Robot.....	590
	Testing Your Robot.....	592
	Calibrating your servos.....	592
	Preparing for running tests on your rover in Python.....	595
	Installing software for the PiCar-B Python test.....	595
	The PiCar-B Python test code.....	596
	Pi camera video testing.....	597
<b>CHAPTER 3:</b>	<b>Programming Your Robot Rover</b> .....	601
	Building a Simple, High-Level Python Interface.....	601
	The motorForward() function.....	602
	The wheelsLeft function().....	602
	The wheelsPercent function().....	603
	Making a Single Move with Python.....	603
	Functions of the RobotInterface Class.....	604
	Front LED functions.....	605
	Pixel strip functions.....	606
	Ultrasonic distance sensor function.....	608
	Main motor functions.....	608
	Servo functions.....	609
	General servo function.....	613
	The Python Robot Interface Test.....	613
	Coordinating Motor Movements with Sensors.....	617
	Making a Python Brain for Our Robot.....	621
	Overview of the Included Adeept Software.....	628
	Where to Go from Here.....	629
<b>CHAPTER 4:</b>	<b>Using Artificial Intelligence in Robotics</b> .....	631
	This Chapter's Projects: Going to the Dogs.....	632
	Setting Up the First Project.....	632
	Machine Learning Using TensorFlow.....	633
	The code.....	635
	How the code works.....	637
	The results.....	640
	Testing the Trained Network.....	642
	The code.....	642
	How the code works.....	644
	The results.....	646

Taking Cats and Dogs to Our Robot .....	648
The code .....	649
How it works .....	652
The results .....	652
Setting Up the Second Project .....	654
The FindAndChaseTheBall.py Python Program .....	655
The structure of the program .....	656
The ultrasonic thread .....	656
The video display thread .....	657
The OpenCV frame analyzer thread .....	657
The Main Program .....	661
The program's configuration .....	661
Setting the ball's color .....	662
Chasing the ball .....	664
Program notes .....	664
AI and the Future of Robotics .....	666
<b>INDEX</b> .....	<b>667</b>

# Introduction

---

**T**he power of Python. The Python language is becoming more and more popular, and in 2017 it became the most popular language in the world according to IEEE Spectrum. The power of Python is real.

Python the number-one language because it's easy to learn and use, due partly to its simplified syntax and natural-language flow but also to the amazing user community and the breadth of applications available.

## About This Book

---

This book is a reference manual to guide you through the process of learning Python and how to use it in modern computer applications, such as data science, artificial intelligence, physical computing, and robotics. If you're looking to learn a little about a lot of exciting things, this is the book for you. It gives you an introduction to the topics that you'll need to explore more deeply.

*Python All-in-One For Dummies*, 2nd Edition guides you through the Python language and then takes you on a tour through some cool libraries and technologies (the Raspberry Pi, robotics, AI, data science, and more) that all revolve around the Python language. When you work on new projects and new technologies, Python is there with a diverse number of libraries just waiting for you to use.

This is a hands-on book, with examples and code throughout. You are expected to enter the code, run it, and then modify it to do what you want. You don't just buy a robot; you build it so you can understand all the pieces and can make sense of the way Python works with the robot to control its motors and sensors. Artificial intelligence is complicated, but Python helps make a significant part of it accessible. Data science is complicated, but Python helps you do data science more easily. Robotics is complicated, but Python gives you the code that controls the robot. And Python even enables you to tie these pieces together and use, say, AI in robotics.

In this book, we take you through the basics of the Python language in small, easy-to-understand steps. After we have introduced you to the language, we step into the world of artificial intelligence, exploring programming in machine

learning and neural networks using Python and TensorFlow and working on real problems and real software, not just toy applications.

After that, we're off to the exciting world of big data and data science with Python. We look at big public data sets such as medical and environmental data.

Finally, you get to experience the magic of what we call physical computing. Using the inexpensive, small, and incredibly popular Raspberry Pi computer, we show you how to use Python to control motors and read sensors. This is a lead-up to the final minibook, "Building Robots," where you build a robot and control it with Python and your own programs, even using artificial intelligence. This is not your mother's RC car.

Python data science, robotics, AI, and fun all in the same book.

This book won't make you understand everything about these fields, but it will give you a great introduction to the terminology and the power of Python in all these fields. Enjoy the book and go forth and learn more afterwards.

## Foolish Assumptions

We assume that you know how to use a computer in a basic way. If you can turn on the computer and use a mouse, you're ready for this book. We assume that you don't know how to program yet, although you will have some skills in programming by the end of the book. If we're wrong and you already know Python (or some other computer language), jump ahead to minibook 4 and dig right into learning something new. Our intent is to guide you through the language of Python and then through some of the amazing technologies and devices that use Python. We provide complete examples. If you get stuck on something, look it up on the web, read a tutorial, and then come back to it.

## What to Buy

To complete the projects in Books 4 through 7, you need a Raspberry Pi 3B+ starter kit at <https://amzn.to/2WzYdoY> or a Raspberry Pi 4B Starter Kit at <https://amzn.to/3nIH8W8>. In addition, you need the items listed in this section, organized by minibook.



TIP

If you want to use a Raspberry Pi 4B in the robot in Book 7, it will dramatically reduce the battery life, and with some types of batteries the robot may not be able to boot the Pi 4B.

## Book 6

For building the projects in Book 6, you need the following:

- » Pi2Grover board at <https://shop.switchdoc.com> or [www.amazon.com](http://www.amazon.com). (You can get \$5.00 off the board at [shop.switchdoc.com](https://shop.switchdoc.com) by using the discount code PI2DUMMIES at checkout.)
- » Grove blue LED module, which includes a Grove cable, at <https://shop.switchdoc.com> or Amazon.
- » A package of Grove male jumper patch cables, specifically the Grove-4-male-pin-to-Grove-conversion cables, at <https://shop.switchdoc.com/products/grove-4-pin-male-jumper-to-grove-4-pin-conversion-cable-5-pcs-per-pack> and <https://amzn.to/3nyGbic>.
- » A package of female-to-Grove patch cables at <https://shop.switchdoc.com/products/grove-4-pin-female-jumper-to-grove-4-pin-conversion-cable-5-pcs-per-pack> and <https://amzn.to/3jhQmXY>.
- » Grove HDC1080 I2C temperature and humidity sensor at <https://store.switchdoc.com> or [www.amazon.com](http://www.amazon.com). The SwitchDoc Labs HDC1080 sensor comes with a Grove connector. If you buy a non-Grove sensor on Amazon, you'll need a female-to-Grove patch cable, as discussed in Chapter 2 of this minibook. You can get a female-to-Grove patch cable at <https://shop.switchdoc.com/products/grove-4-pin-female-jumper-to-grove-4-pin-conversion-cable-5-pcs-per-pack> and <https://amzn.to/3jhQmXY>.
- » Grove oxygen sensor at [www.seeedstudio.com](http://www.seeedstudio.com) or [www.amazon.com](http://www.amazon.com).
- » Pi2Grover Raspberry-Pi-to-Grove converter, <https://shop.switchdoc.com> or [www.amazon.com](http://www.amazon.com). (You can get \$5.00 off the board at [shop.switchdoc.com](https://shop.switchdoc.com) by using the discount code PI2DUMMIES at checkout.)
- » Grove four-channel, 16-bit analog-to-digital converter at <https://store.switchdoc.com> or [www.amazon.com](http://www.amazon.com).
- » Grove I2C motor drive (with a Grove cable) at [www.seeedstudio.com](http://www.seeedstudio.com) or <https://amazon.com>.
- » Two small DC motors at [www.adafruit.com/product/711](http://www.adafruit.com/product/711) or <https://amazon.com>.
- » SG90 micro servo motor at [www.ebay.com](http://www.ebay.com) or <https://amazon.com>. These motors are inexpensive, so you may end up having to buy two or more for under \$10.
- » 28BYJ-48 ULN2003 5V stepper motor at [www.ebay.com](http://www.ebay.com) or <https://amzn.to/2BuNDV1>. This type of motor is inexpensive, so you may end up having to buy five for \$12. Make sure you get the ones with the driver boards (such as the ones at the Amazon.com link).

## Book 7

For the robot in Book 7, purchase the following:

- » Adept Raspberry Pi PiCar-B. Make sure you buy the PiCar-B and *not* the PiCar-A. Look for “Adept Mars Rover PiCar-B.” You can buy the PiCar-B at Amazon.com <https://amzn.to/36dukPU>, [www.ebay.com](http://www.ebay.com), and [www.adept.com](http://www.adept.com).
- » Two 18650 3.7V LiPo 5000mAh batteries at <https://amazon.com> and many other places.

## Icons Used in This Book

What’s a *Dummies* book without icons pointing you in the direction of truly helpful information that’s sure to speed you along your way? Here we briefly describe each icon we use in this book.



TIP

The Tip icon points out helpful information that’s likely to make your job easier.



REMEMBER

This icon marks a generally interesting and useful fact — something you may want to remember for later use.



WARNING

The Warning icon highlights lurking danger. When we use this icon, we’re telling you to pay attention and proceed with caution.



TECHNICAL  
STUFF

When you see this icon, you know that there’s techie-type material nearby. If you’re not feeling technical-minded, you can skip this information.

## Beyond the Book

In addition to the material in the print or e-book you’re reading right now, this product also comes with some access-anywhere goodies on the web. To get this material, simply go to [www.dummies.com](http://www.dummies.com) and search for “Python All-in-One For Dummies cheat sheet” in the Search box. In addition, we provide all the source code for this book at [www.dummies.com/go/pythonaiofd2e](http://www.dummies.com/go/pythonaiofd2e). Click Downloads in the left column, and you’ll see the code links organized by minibook.

# Where to Go from Here

---

*Python All-in-One For Dummies*, 2nd Edition is designed so that you can read a chapter or section out of order, depending on what subjects you're most interested in. Where you go from here is up to you!

Book 1 is a great place to start reading if you've never used Python before. Discovering the basics and common terminology can be helpful when reading later chapters that use the terms and commands regularly!





# 1

## Getting Started

# Contents at a Glance

---

<b>CHAPTER 1: Starting with Python</b> .....	9
Why Python Is Hot .....	10
Choosing the Right Python .....	11
Tools for Success .....	13
Writing Python in VS Code .....	19
Using Jupyter Notebook for Coding .....	23
<b>CHAPTER 2: Interactive Mode, Getting Help, and Writing Apps</b> .....	29
Using Python's Interactive Mode .....	29
Creating a Python Development Workspace .....	37
Creating a Folder for Your Python Code .....	39
Typing, Editing, and Debugging Python Code .....	41
Writing Code in a Jupyter Notebook .....	47
<b>CHAPTER 3: Python Elements and Syntax</b> .....	53
The Zen of Python .....	53
Introducing Object-Oriented Programming .....	56
Discovering Why Indentations Count, Big Time .....	57
Using Python Modules .....	59
<b>CHAPTER 4: Building Your First Python Application</b> .....	63
Opening the Python App File .....	64
Typing and Using Python Comments .....	64
Understanding Python Data Types .....	66
Working with Python Operators .....	71
Creating and Using Variables .....	74
Understanding What Syntax Is and Why It Matters .....	79
Putting Code Together .....	84

- » Discovering why Python is hot
- » Finding the tools for success
- » Writing Python in VS Code
- » Writing Python in Jupyter notebooks

## Chapter **1**

# Starting with Python

**B**ecause you're reading this chapter, you probably realize that Python is a great language to know if you're looking for a good job in programming, or if you want to expand your existing programming skills into exciting cutting-edge technologies such as artificial intelligence (AI), machine learning (ML), data science, or robotics, or even if you're just building apps in general. So we're not going to try to sell you on Python. It sells itself.

Our approach leans heavily toward the hands-on. A common failure in many programming tutorials is that they already assume you're a professional programmer in some language, and they skip over things they assume you already know.

This book is different in that we *don't* assume that you're already programming in Python or some other language. We *do* assume that you can use a computer and understand basics such as files and folders.

We also assume you're not up for settling down in an easy chair in front of the fireplace to read page after page of theoretical stuff about Python, like some kind of boring novel. You don't have that much free time to kill. So we're going to get right into it and focus on *doing*, hands-on, because that's the only way most of us learn. We've never seen anyone read a book about Python and then sit at a computer and write Python like a pro. Human brains don't work that way. We learn through practice and repetition, and that requires being hands-on.

# Why Python Is Hot

We promised we weren't going to spend a bunch of time trying to sell you on Python, and that's not our intent here. But we would like to talk briefly about *why* it's so hot.

Python is hot primarily because it has all the right stuff for the kind of software development that's driving the software development world these days. Machine learning, robotics, artificial intelligence, and data science are the leading technologies today and for the foreseeable future. Python is popular mainly because it already has lots of capabilities in these areas, while many older languages lag behind in these technologies.

Just as there are different brands of toothpaste, shampoo, cars, and just about every other product you can buy, there are different brands of programming languages with names such as Java, C, C++ (*pronounced C plus plus*), and C# (*pronounced C sharp*). They're all programming languages, just like all brands of toothpaste are toothpaste. The main reasons cited for Python's current popularity are

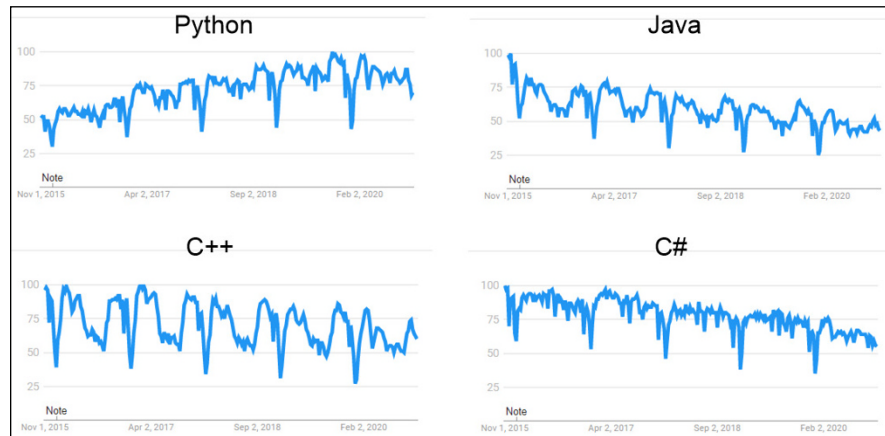
- » Python is relatively easy to learn.
- » Everything you need to learn (and do) in Python is free.
- » Python offers more ready-made tools for current hot technologies such as data science, machine learning, artificial intelligence, and robotics than most other languages.

## HTML, CSS, AND JavaScript

Some of you may have heard of languages such as HTML, CSS, and JavaScript. Those aren't traditional programming languages for developing apps or other generic software. HTML and CSS are specialized for developing web pages. And although JavaScript is a programming language, it is heavily geared to website development and isn't quite in the same category of general programming languages like Python and Java.

If you specifically want to design and create websites, you have to learn HTML, CSS, and JavaScript whether you're already familiar with Python or some other programming language.

Figure 1-1 shows Google search trends over the last five years. As you can see, Python has been gaining in popularity (as indicated by the upward slope of the trend) whereas other languages have stayed about the same or declined. This certainly supports the notion that Python is the language people want to learn right now and for the future. Most people would agree that given trends in modern computing, learning Python gives you the best opportunity for getting a secure, high-paying job in the world of information technology.



**FIGURE 1-1:**  
Google search trends for the last five years or so.



TIP

You can do your own Google trend searches at <https://trends.google.com>.

## Choosing the Right Python

There are different *versions* of Python out roaming the world, prompting many a beginner to wonder things such as

- » Why are there different versions?
- » How are they different?
- » Which one should I learn?

All good questions, and we'll start with the first. A version is kind of like a car year. You can go out and buy a 1968 Ford Mustang, a 1990 Ford Mustang, a 2019 Ford Mustang, or a 2020 Ford Mustang. They're all Ford Mustangs. The only difference is that the one with the highest year number is the most current Ford Mustang. That Mustang is different from the older models in that it has some improvements based on experience with earlier models, as well as features current with the times.

Programming languages (and most other software products) work the same way. But as a rule we don't ascribe year numbers to them because they're not released on a yearly basis. They're released whenever they're released. But the principle is the same. The version with the highest number is the newest, most recent model, sporting improvements based on experience with earlier versions, as well as features relevant to the current times.

Just as we use a decimal point with money to separate dollars from cents, we use decimal points with version numbers to indicate how much the software has changed. When there's a significant change, the entire version number is usually changed. More minor changes are expressed as decimal points. You can see how the version number increases along with the year in Table 1-1, which shows the release dates of various Python versions. We've skipped a few releases because there is little reason to know or understand the differences between all the versions. We present the table only so you can see how newer versions have higher version numbers; that's all that matters.

**TABLE 1-1**

### Examples of Python Versions and Release Dates

Version	When Released
Python 3.9	October 2020
Python 3.8	October 2019
Python 3.7	June 2018
Python 3.6	December 2016
Python 3.5	September 2015
Python 3.4	March 2014
Python 3.3	September 2012
Python 3.2	February 2011
Python 3.1	June 2009
Python 3.0	December 2008
Python 2.7	July 2010
Python 2.6	October 2008
Python 2.0	October 2000
Python 1.6	September 2000
Python 1.5	February 1997
Python 1.0	January 1994