

PYTHON[®] PROGRAMMING FOR TEENS

KENNETH A. LAMBERT

Cengage Learning PTR



Professional • Technical • Reference

Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States



Professional • Technical • Reference

Python® Programming for Teens
Kenneth A. Lambert

Publisher and General Manager,
Cengage Learning PTR: Stacy L. Hiquet

Associate Director of Marketing:
Sarah Panella

Manager of Editorial Services:
Heather Talbot

Senior Marketing Manager:
Mark Hughes

Senior Product Manager: Mitzi Koontz

Project/Copy Editor: Karen A. Gill

Technical Reviewer: Zach Scott

Interior Layout Tech: MPS Limited

Cover Designer: Mike Tanamachi

Indexer: Sharon Shock

Proofreader: Gene Redding

© 2015 Cengage Learning PTR.

CENGAGE and CENGAGE LEARNING are registered trademarks of Cengage Learning, Inc., within the United States and certain other jurisdictions.

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced, transmitted, stored, or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the publisher.

For product information and technology assistance, contact us at
Cengage Learning Customer & Sales Support, 1-800-354-9706.

For permission to use material from this text or product, submit all requests online at cengage.com/permissions.

Further permissions questions can be emailed to
permissionrequest@cengage.com.

Python is a registered trademark of the Python Software Foundation.

All other trademarks are the property of their respective owners.

All images © Cengage Learning unless otherwise noted.

Library of Congress Control Number: 2014939193

ISBN-13: 978-1-305-27195-1

ISBN-10: 1-305-27195-5

Cengage Learning PTR

20 Channel Center Street

Boston, MA 02210

USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at: international.cengage.com/region.

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

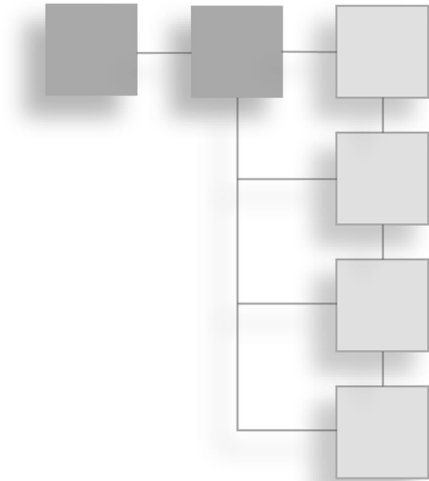
For your lifelong learning solutions, visit cengageptr.com.

Visit our corporate website at cengage.com.

Printed in the United States of America

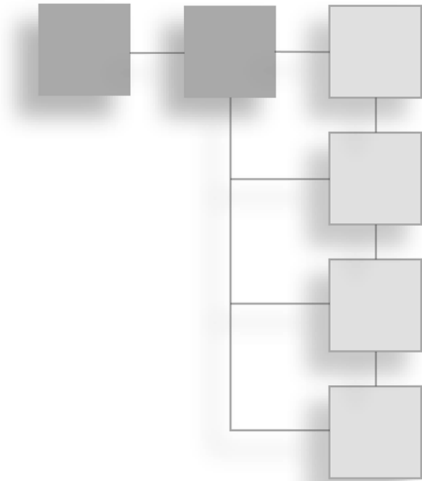
1 2 3 4 5 6 7 16 15 14

ABOUT THE AUTHOR



Kenneth A. Lambert is a professor of computer science and the chair of that department at Washington and Lee University. He has taught introductory programming courses for 29 years and has been an active researcher in computer science education. Lambert has authored or coauthored 25 textbooks, including a series of introductory C++ textbooks with Douglas Nance and Thomas Naps, a series of introductory Java textbooks with Martin Osborne, and a series of introductory Python textbooks. His most recent textbook is *Fundamentals of Python: Data Structures*.

CONTENTS



	Introduction	xiii
Chapter 1	Getting Started with Python	1
	Taking Care of Preliminaries	1
	Downloading and Installing Python	1
	Launching and Working in the IDLE Shell	2
	Obtaining Python Help	3
	Working with Numbers	4
	Using Arithmetic	4
	Working with Variables and Assignment	6
	Using Functions	9
	Using the math Module	9
	Detecting Errors	11
	Working with Strings	12
	String Literals	12
	The len, str, int, and float Functions	14
	Input and Output Functions	15
	Indexing, Slicing, and Concatenation	16
	String Methods	18
	Working with Lists	19
	List Literals and Operators	19
	List Methods	20
	Lists from Other Sequences	22
	Lists and the random Module	23
	Tuples as Immutable Lists	24

	Working with Dictionaries	25
	Dictionary Literals	25
	Dictionary Methods and Operators	26
	Summary	27
	Exercises	29
Chapter 2	Getting Started with Turtle Graphics	31
	Looking at the Turtle and Its World	31
	Using Basic Movement Operations	34
	Moving and Changing Direction	35
	Drawing a Square	36
	Drawing an Equilateral Triangle	38
	Undoing, Clearing, and Resetting	39
	Setting and Examining the Turtle’s State	40
	The Pen Size	40
	The Shape	40
	The Speed	42
	Other Information About the Turtle’s State	42
	Working with Colors	43
	The Pen Color and the Background Color	43
	How Computers Represent Colors	43
	Filled Shapes	45
	Drawing Circles	46
	Drawing Text	48
	Using the Turtle’s Window and Canvas	49
	Using a Configuration File	51
	Summary	51
	Exercises	52
Chapter 3	Control Structures: Sequencing, Iteration, and Selection	55
	Repeating a Sequence of Statements: Iteration	56
	The for Loop	56
	Nested Loops	57
	How the range Function Works with a for Loop	58
	Loops with Strings, Lists, and Dictionaries	59
	Asking Questions: Boolean Expressions	60
	Boolean Values	61
	Comparisons	61
	Logical Operations	61

	Making Choices: Selection Statements	63
	The One-Way if Statement	63
	The Two-Way if Statement	65
	Probable Options with random.randint	66
	The Multiway if Statement	67
	Using Selection to Control Iteration	68
	The while Loop	69
	Random Walks in Turtle Graphics	70
	Summary	73
	Exercises	74
Chapter 4	Composing, Saving, and Running Programs	75
	Exploring the Program Development Process	75
	Composing a Program	77
	Program Edits	77
	Program Structure	78
	Docstrings and End-of-Line Comments	79
	import Statements	79
	The main Function	79
	The if main == "__main__" Idiom	80
	The mainloop Function	81
	Running a Program	81
	Using a Turtle Graphics Configuration File	81
	Running a Program from an IDLE Window	82
	Running a Program from a Terminal Window	83
	Using the sys Module and Command-Line Arguments	83
	Looking Behind the Scenes: How Python Runs Programs	85
	Computer Hardware	86
	Computer Software	88
	Summary	90
	Exercises	91
Chapter 5	Defining Functions	93
	Basic Elements of Function Definitions	94
	Circles and Squares	94
	Docstrings	95
	The return Statement	96
	Testing Functions in a Program	97
	Optional, Default, and Keyword Arguments	98

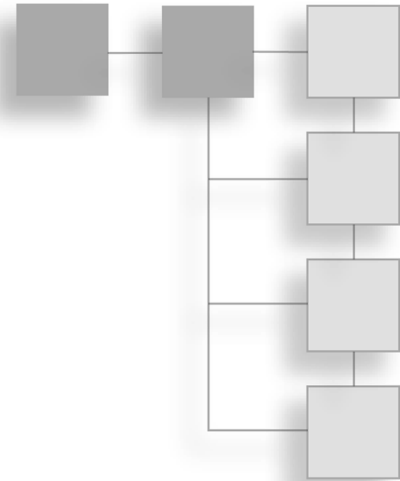
	Functions as General Solutions to Problems	99
	Regular Polygons	99
	Functions as Arguments	101
	Building Functions with lambda Expressions.	102
	Modules as Libraries of Functions	103
	Math Topic: Graphing Functions	103
	Functions in Mathematics	103
	Graphing Functions in Turtle Graphics	105
	Refactoring a Program with Functions	106
	Simplifying the Code for the Random Walk	107
	The atEdge Function	108
	The randomForward Function	109
	The randomTurn Function	109
	Another Version of the Random Walk	111
	Summary	112
	Exercises	112
Chapter 6	User Interaction with the Mouse and the Keyboard	115
	Using Dialog-Based Input.	115
	Input Dialogs in Turtle Graphics	116
	Input Dialogs for Text	117
	Input Dialogs for Numbers.	117
	Responding to Mouse Events.	118
	Drawing Line Segments with Mouse Clicks.	118
	How Event Handling Works.	120
	Freehand Drawing by Dragging the Mouse	120
	Responding to Keyboard Events	122
	The onkey Function	123
	A Complete Retro Drawing Program.	124
	Using Module Variables	125
	Initializing and Using Module Variables	125
	Tracking the History of Turtle Positions	126
	Using Two Mouse Buttons	128
	Adding an Event-Handling Function for the Right Button.	128
	Example 1: Simple Drawing with Random Colors.	129
	Example 2: Drawing and Moving.	129
	Example 3: Drawing, Moving, and Random Colors.	130
	Example 4: Dialogs for Shape Properties.	130
	Summary	133
	Exercises	134

Chapter 7	Recursion	135
	Recursive Design	135
	Top-Down Design	135
	Recursive Function Design	137
	Recursive Function Call Tracing	139
	Recursive Functions and Loops	140
	Infinite Recursion	142
	Sequential Search of a List	143
	Binary Search of a List	145
	Recursive Patterns in Art: Abstract Painting	147
	Design of the Program	148
	Performance Tracking	148
	The drawRectangle Function	149
	The mondrian Function	149
	The main Function	151
	The tracer and update Functions	152
	Recursive Patterns in Nature: Fractals	153
	What the Program Does	155
	Design of the Program	155
	Code for the Program	156
	Summary	157
	Exercises	158
Chapter 8	Objects and Classes	159
	Objects, Methods, and Classes in Turtle Graphics	160
	The Turtle Class and Its Methods	160
	A Random Walk with Several Turtles	161
	A New Class: RegularPolygon	162
	Design: Determine the Attributes	163
	Design: Determine the Behavior	164
	Implementation: The Structure of a Class Definition	166
	Implementation: The <code>__init__</code> Method	168
	Implementation: Showing and Hiding	169
	Implementation: Getting and Setting	170
	Implementation: Translation, Scaling, and Rotation	171
	Inheritance: Squares and Hexagons as Subclasses	171
	New Class: Menu Item	172
	Design: Determine the Attributes and Behavior	173
	Implementation and Testing	173

	Response to User Events, Revisited	176
	Whose Click Is It Anyway?	176
	A Grid Class for the Game of Tic-Tac-Toe	179
	Modeling a Grid	179
	Defining a Class for a Tic-Tac-Toe Grid	181
	Using Class Variables	182
	Stretching the Shape of a Turtle	183
	Making a Move.	183
	Defining a Class for the Grid	184
	Laying Out the Grid	184
	Defining Methods for the Game Logic	185
	Coding the Main Application Module.	186
	Summary	187
	Exercises	188
Chapter 9	Animations	189
	Animating the Turtle with a Timer	189
	Using the ontimer Function	190
	Scheduling Actions at Regular Intervals	190
	Animating Many Turtles.	193
	What Is an Animated Turtle?	193
	Using an Animated Turtle in a Shell Session	195
	Defining the AnimatedTurtle Class	195
	Sleepy and Speedy as Animated Turtles	197
	Creating Custom Turtle Shapes	199
	Creating Simple Shapes	199
	Creating Compound Shapes.	202
	Summary	203
	Exercises	204
Appendix A	Turtle Graphics Commands	205
	Turtle Functions	205
	Turtle Motion	205
	Pen Control	207
	Turtle State	209
	Event Handling	210
	Functions Related to the Window and Canvas	211
	Window Functions	211
	Input Functions	212

Appendix B Solutions to Exercises	213
Exercise Solutions for Chapter 1	213
Exercise 1.....	213
Exercise 2.....	213
Exercise Solutions for Chapter 2	213
Exercise 1.....	213
Exercise 2.....	214
Exercise Solutions for Chapter 3	216
Exercise 1.....	216
Exercise 2.....	216
Exercise Solutions for Chapter 4	216
Exercise 1.....	216
Exercise 2.....	217
Exercise Solutions for Chapter 5	218
Exercise 1.....	218
Exercise 2.....	218
Exercise Solutions for Chapter 6	219
Exercise 1.....	219
Exercise 2.....	221
Exercise Solutions for Chapter 7	222
Exercise 1.....	222
Exercise 2.....	222
Exercise Solutions for Chapter 8	224
Exercise 1.....	224
Exercise 2.....	227
Exercise Solutions for Chapter 9	229
Exercise 1.....	229
Exercise 2.....	231
Index	233

INTRODUCTION



Welcome to *Python Programming for Teens*. Whether you're under 20 or just a teenager at heart, this book will introduce you to computer programming. You can use it in a classroom or on your own. The only assumption is that you know how to use a modern computer system with a keyboard, screen, and mouse.

To make your learning experience fun and interesting, you will write programs that draw pictures on the screen and allow you to interact with them by using the mouse. Along the way, you will learn the basic principles of program design and problem solving with computers. You will then be able to apply these ideas and techniques to solve problems in almost any area of study. But most of all, you will experience the joy of building things that work *and* look great!

WHY PYTHON?

Computer technology and applications have become increasingly more sophisticated over the past several decades, and so has the computer science curriculum, especially at the introductory level. Today's students learn a bit of programming and problem solving and are then expected to move quickly into topics like software development, complexity analysis, and data structures that, 20 years ago, were reserved for advanced courses. In addition, the ascent of object-oriented programming as a dominant method has led instructors and textbook authors to bring powerful, industrial-strength programming languages such as C++ and Java into the introductory curriculum. As a result, instead of experiencing the rewards and excitement of computer programming, beginning students

often become overwhelmed by the combined tasks of mastering advanced concepts and learning the syntax of a programming language.

This book uses the Python programming language as a way of making the learning experience manageable and attractive for students and instructors alike. Python offers the following pedagogical benefits:

- Python has simple, conventional syntax. Its statements are close to those of ordinary English, and its expressions use the conventional notation found in algebra. Thus, beginners can spend less time learning the syntax of a programming language and more time learning to solve interesting problems.
- Python has safe semantics. Any expression or statement whose meaning violates the definition of the language produces an error message.
- Python scales well. It is easy for beginners to write simple programs. Python also includes all the advanced features of a modern programming language, such as support for data structures and object-oriented software development, for use when they become necessary.
- Python is highly interactive. Expressions and statements can be entered at an interpreter's prompts to allow the programmer to try out experimental code and receive immediate feedback. Longer code segments can then be composed and saved in script files to be loaded and run as modules or standalone applications.
- Python is general purpose. In today's context, this means that the language includes resources for contemporary applications, including media computing and networks.
- Python is free and is in widespread use in the industry. Students can download it to run on a variety of devices. There is a large Python user community, and expertise in Python programming has great resume value.

To summarize these benefits, Python is a comfortable and flexible vehicle for expressing ideas about computation, both for beginners and experts alike. If students learn these ideas well in their first experience with programming, they should have no problems making a quick transition to other languages and technologies needed to achieve their educational or career objectives. Most importantly, beginners will spend less time staring at a computer screen and more time thinking about interesting problems to solve.

ORGANIZATION OF THE BOOK

The approach in this book is easygoing, with each new concept introduced only when you need it.

Chapter 1, “Getting Started with Python,” advises you how to download, install, and start the Python programming software used in this book. You try out simple program commands and become acquainted with the basic features of the Python language that you will use throughout the book.

Chapter 2, “Getting Started with Turtle Graphics,” introduces the basic commands for turtle graphics. You learn to draw pictures with a set of simple commands. Along the way, you discover a thing or two about colors and two-dimensional geometry.

Chapter 3, “Control Structures: Sequencing, Iteration, and Selection,” covers the program commands that allow the computer to make choices and perform repetitive tasks.

Chapters 4, “Composing, Saving, and Running Programs,” shows you how to save your programs in files, so you can give them to others or work on them another day. You learn how to organize a program like an essay, so it is easy for you and others to read, understand, and edit. You also learn a bit about how the computer is able to read, understand, and run a program.

Chapter 5, “Defining Functions,” introduces an important design feature: the function. By organizing your programs with functions, you can simplify complex tasks and eliminate unnecessary duplications in your code.

Chapter 6, “User Interaction with the Mouse and the Keyboard,” covers features that allow people to interact with your programs. You learn program commands for responding to mouse and keyboard events, as well as pop-up dialogs that can take information from your programs’ users.

Chapter 7, “Recursion,” teaches you about another important design strategy called recursion. You write some recursive functions that generate computer art and fractal images.

Chapter 8, “Objects and Classes,” offers a beginner’s guide to the use of objects and classes in programming. You learn how to define new types of objects, such as menu items for choosing colors and grids for board games, and use them in interesting programs.

Chapter 9, “Animations,” concludes the book with a brief introduction to animations. You discover how to get images to move independently and interact in interesting ways.

Two appendixes follow the last chapter. Appendix A, “Turtle Graphics Commands,” provides a reference for the set of turtle graphics commands introduced in the book.