

Python by Example

Learning to Program in 150 Challenges

Python is today's fastest growing programming language. This engaging and refreshingly different guide breaks down the skills into clear step-by-step chunks and explains the theory using brief easy-to-understand language. Rather than bamboozling readers with pages of mind-numbing technical jargon, this book includes 150 practical challenges, putting the power in the reader's hands. Through creating programs to solve these challenges the reader will quickly progress from mastering the basics to confidently using subroutines, a graphical user interface, and linking to external text, csv and SQL files. This book is perfect for anyone who wants to learn how to program with Python. In particular, students starting out in computer science and teachers who want to improve their confidence in Python will find here a set of ready-made challenges for classroom use.

NICHOLA LACEY is Director of Nichola Wilkin Ltd. She is a trusted source for teaching resources, having sold thousands of resources to schools around the world. As one of the most popular authors on TES, Nichola enjoys an extremely high review rating with hundreds of thousands of downloads. She was a programmer before moving into corporate training and then retraining as a teacher, and she gained a unique skill set of programming and practical classroom experience after being promoted to head of computer science in a private boys' school.

PYTHON BY EXAMPLE

Learning to Program in 150 Challenges

NICHOLA LACEY
Nichola Wilkin Ltd





Shaftesbury Road, Cambridge CB2 8EA, United Kingdom
One Liberty Plaza, 20th Floor, New York, NY 10006, USA
477 Williamstown Road, Port Melbourne, VIC 3207, Australia
314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India
103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment,
a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of
education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781108716833

DOI: 10.1017/9781108591942

© Nicola Lacey 2019

This publication is in copyright. Subject to statutory exception and to the provisions
of relevant collective licensing agreements, no reproduction of any part may take
place without the written permission of Cambridge University Press & Assessment.

First published 2019 (version 6, November 2024)

A catalogue record for this publication is available from the British Library

ISBN 978-1-108-71683-3 Paperback

Cambridge University Press & Assessment has no responsibility for the persistence
or accuracy of URLs for external or third-party internet websites referred to in this
publication and does not guarantee that any content on such websites is, or will
remain, accurate or appropriate.

Contents

<i>Image Credits</i>	vi
<i>Introduction</i>	1
<i>Downloading Python</i>	4
<i>Some Tips</i>	6

Part I: Learning Python

Challenges 1 - 11: The Basics	11
Challenges 12 - 19: If Statements	17
Challenges 20 - 26: Strings	24
Challenges 27 - 34: Maths	31
Challenges 35 - 44: For Loop	35
Challenges 45 - 51: While Loop	40
Challenges 52 - 59: Random	45
Challenges 60 - 68: Turtle Graphics	51
Challenges 69 - 79: Tuples, Lists and Dictionaries	58
Challenges 80 - 87: More String Manipulation	67
Challenges 88 - 95: Numeric Arrays	72
Challenges 96 - 103: 2D Lists and Dictionaries	79
Challenges 105 - 110: Reading and Writing to a Text File	86
Challenges 111 - 117: Reading and Writing to a .csv File	91
Challenges 118 - 123: Subprograms	99
Challenges 124 - 132: Tkinter GUI	110
Challenges 133 - 138: More Tkinter	124
Challenges 139 - 145: SQLite	134

Part II: Chunky Challenges

Challenge 146: Shift Code	150
Challenge 147: Mastermind	153
Challenge 148: Passwords	156
Challenge 149: Times Table (GUI)	161
Challenge 150: Art Gallery	164
<i>What Next?</i>	169
<i>Glossary</i>	170
<i>Index</i>	182



Image Credits

Animal Drawings:

Pages 1, 9, 11, 31, 35, 37, 41, 45, 92, 125, 127, 138, 150, 165: HelenField/Shutterstock.com

Pages 16, 20, 94 and 141(bottom): Victoria Novak/Shutterstock.com

Pages 27 and 157: Dimonika/Shutterstock.com

Pages 46, 58, 73, 93, 95, 103, 128, 135, 147, 169: mart/Shutterstock.com

Pages 59, 68, 151, 154: lynea/Shutterstock.com

Page 80: MoreVector/Shutterstock.com

All other animal drawings: Olga_Angelloz/Shutterstock.com

Other decorative icons:

MG Drachal/Shutterstock.com

Mila Petkova/Shutterstock.com

Nikolaeva/Shutterstock.com

Tiwat K/Shutterstock.com

Introduction



If you have ever picked up a programming manual and felt your forehead go clammy and your eyes cross as you attempt to make sense of the long-winded explanations, this is the guide for you.

I have been in your position, attempting to learn how to program and having to rely on the traditional style of guides. I know from painful experience how quickly I glaze over and my brain solidifies; after only a few pages the tedium leaves me blindly reading words without any real notion of what they mean any more. Inevitably I give up and the whole process makes me feel like a limp failure, gasping for breath after I surface from drowning in technical jargon.

I hated having to read through pointless drivel and then be presented with a short program telling me exactly what to type in and then spend the next 20 pages reading about what I have just done and the 101 ways I could run it. I hated having no control over trying things out for myself and I hated the way these guides would only contain one or two challenges at the end of a chapter of theory.



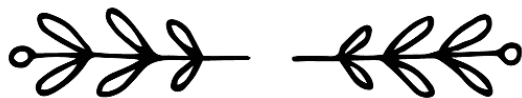
I knew there had to be a better way, and thankfully there is. I wrote it and you are presently reading it, so aren't you lucky? This guide is refreshingly different and helps you learn how to program with Python by using practical examples rather than self-important explanations.



Many programmers learn through experimentation, looking at others' code and working out what method is best for a given situation. This book is a hands-on approach to learning programming. After minimal reading you are set a number of challenges to create the programs. You can explore and experiment with the programming language and look at the example solutions to learn how to think like a programmer. There are no chapters entitled "the architecture of a computer", "the theory of programming" or any other gobbledy-gook other authors like to waste time with. I don't want to baffle you with theory or blind you with overbearing explanations that suck out your enthusiasm for learning to program.

Hopefully, you want to get stuck into creating programs, solving problems and enjoying the sense of accomplishment that you get as you proudly look over your lines of code, knowing that you created something that works. That is great, your eagerness is to be applauded and I salute those who are reading this while already sitting at their computers, fingers poised and ready to get going. If that is the case, that you already have Python open on your screen and are itching to get going, then away you go and I'll see you in the first chapter called "The Basics" on page 11.

For everyone who is still with us and is feeling a little more timid, there are just a few more things to tell you about before you take the plunge.



How to Use This Book

This book builds from very simple programs to more complex ones. If you are new to programming or new to Python, start with "The Basics" and work through the chapters in order.

If you are familiar with Python programming and feel confident with the basics, the theory and logic surrounding programming, then you can just dip in and out of the book to get help on the specifics you need.



The book is split into two sections:

Part I

In Part I, each chapter takes you through some basic programming rules and challenges for you to complete and includes:

- a **simple explanation** giving you pointers, which is useful if you are new to programming in Python;
- **examples of code** with a short explanation, which you can use as a basis to solve the challenges;
- a **list of challenges** for you to work through that get harder as you move through them. Each challenge should only take between a couple of minutes and 20 minutes to solve; however, some of the more complex challenges near the end of Part I will take longer as you build up the techniques you will be using. Don't panic if you take longer than this, as long as you solve the problems without *too* much copying from the suggested solution, you are doing fine;
- code containing a **possible solution** for each challenge; there is often more than one answer available, but we include just a single program as a possible solution that you can refer to if you get stuck on a particular aspect of the code.

Part II

In Part II, you are given some larger challenges which utilize the programming skills you learnt in Part I and allow you to consolidate and reinforce the techniques you have been practising. In this section, you are not given the help and example code that is given in Part I and it will take longer to solve each challenge. After each challenge, you are given one possible answer that you may find useful if you are stuck. However, you may have found another solution that works just as well.



Who Is This Book For?

This book is suitable for anyone who wants to learn how to program with Python. It is an essential tool for teachers and students in Key Stage 3 or those studying computer science who need help and ready-made examples to practise programming techniques and build confidence. It can also be used to help with a computer science programming project resource bank, to help pupils needing additional support or just a quick reminder of the syntax when creating programs.



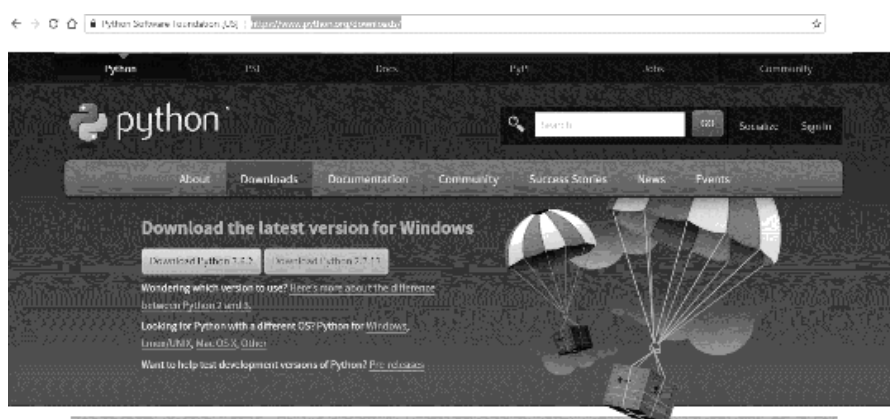
4

Downloading Python

Downloading Python

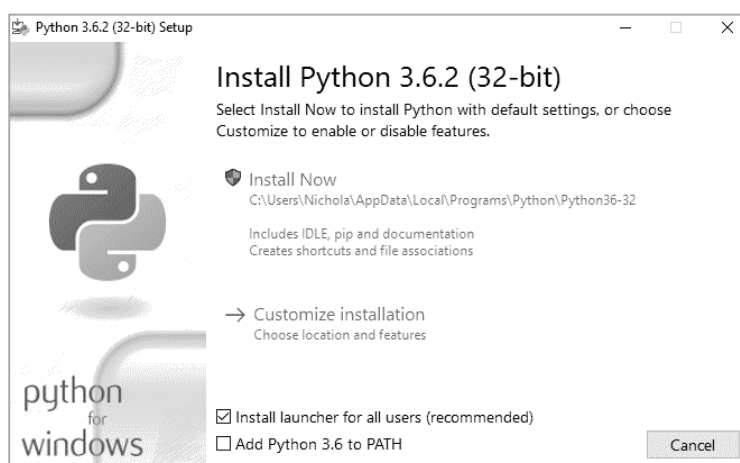
You can download Python for free from the official Python website:

www.python.org/downloads/



Click on the latest version (in the example above, click on the **Download Python 3.6.2** button) to start the installation.

The program will download an executable (.exe) file. When you run this program, you will see an install window like the one shown below.

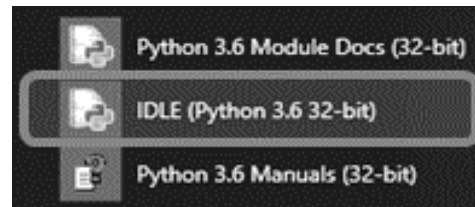
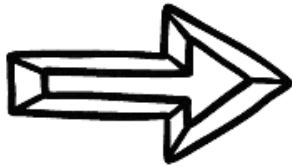


Click the **Install Now** option and the program will start installing Python onto your system.

fff

Running Python

To start Python on a Windows system, click on the **Windows** icon or **Start** menu and select the **IDLE (Python version number)** option as highlighted below.



Some Tips

File Location

On a Windows system, the Python folder is usually found in the C:\ drive and will be named `Python36` (or similar) and the files will automatically be saved in the same location, unless you save them specifically in another location.



Using Comments

Comments are a very useful tool for programmers. They serve two purposes:

- adding an explanation of how the program works;
- stopping parts of the program from working temporarily so you can run and test other parts of the program.



The first, and original, purpose of explaining how a program works is so other programmers can make sense of your programs in case your code needs to be altered and updated in the future and to remind you about why you wrote particular lines of code.

```
print("This is a simple program")  
print() #Outputs a blank line to help with layout  
name = input("Please input your name: ") #Asks for an input  
print("Hello", name) #Joins "Hello" and their name together
```

In this example, comments have been added at the end of the last three lines. They are shown in red and start with the # symbol.

In reality, you would not add comments on lines which contain obvious code as it would clutter the screen; you would only add comments where necessary.

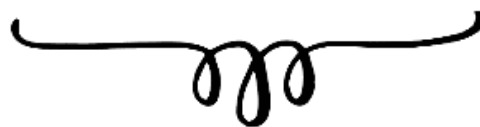
As Python knows to ignore anything after a # symbol, programmers soon started to use # at the start of lines of their code to block out sections they do not want to run so they can focus on and test others.

```
#print("This is a simple program")
print()
name = input("Please input your name: ")
print("Hello", name)
```

In this example, the # has been added to the first line of the program to temporarily stop it from running. To bring it back into the running order simply delete the # and the code will be reactivated.



In this guide, we have not included any comments to the programs so you have to read the code to make sense of it. That way you will really learn how to code! If you are creating programs as part of your coursework you should add comments to explain your programming to the examiner.



Formatting Python

In most versions of Python IDLE it is possible to quickly add comments and indent code using the menus. This way, if you need to block out entire areas using a comment you simply highlight the lines and then select the **Format** menu and select **Comment Out Region**. Similarly, if you need to indent a region (we will look at the reason for indenting code later) then you can also easily do this with the menu.



File	Edit	Format	Run	Options	Windows	Help
		Indent Region			Ctrl+]	
		Dedent Region			Ctrl+[
		Comment Out Region			Alt+3	
		Uncomment Region			Alt+4	

Okay, that is all the “housekeeping” out of the way. No more procrastinating; take a deep breath and away we go...

