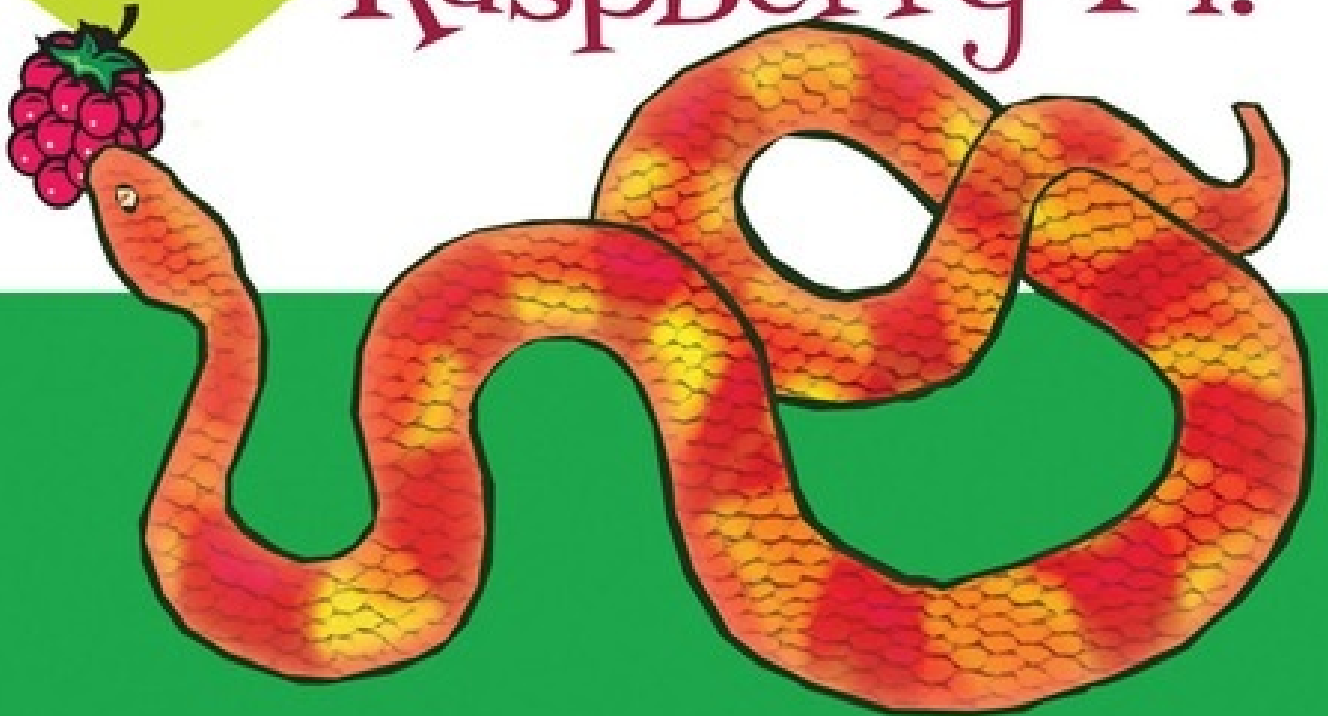


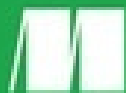
Python programming for kids and other beginners

Hello

Raspberry Pi!™



Ryan Heitz



MANNING

Hello Raspberry Pi!: Python programming for kids and other beginners

Ryan Heitz



Copyright

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact:

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

©2016 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

♻️ Recognizing the importance of preserving what has been written, it is Manning's policy to have all books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without elemental chlorine.



Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Development editor: Dan Maharry
Copyeditor: Tiffany Taylor
Proofreader: Alyson Brener
Technical proofreader: Romin Irani
Typesetter: Marija Tudor
Cover designer: Leslie Haimes

ISBN: 9781617292453

Printed in the United States of America

1 2 3 4 5 6 7 8 9 10 – MAL – 18 17 16 15 14 13

Dedication

To Juliana, Daniel, and John

Brief Table of Contents

[Copyright](#)

[Brief Table of Contents](#)

[Table of Contents](#)

[Preface](#)

[Acknowledgments](#)

[About this book](#)

[1. Getting started](#)

[Chapter 1. Meet Raspberry Pi](#)

[Chapter 2. Exploring Python](#)

[2. Playing with Python](#)

[Chapter 3. Silly Sentence Generator 3000: creating interactive programs](#)

[Chapter 4. Norwegian Blue parrot game: adding logic to programs](#)

[Chapter 5. Raspi's Cave Adventure](#)

[3. Pi and Python projects](#)

[Chapter 6. Blinky Pi](#)

[Chapter 7. Light Up Guessing Game](#)

[Chapter 8. DJ Raspi](#)

[Appendix A. Raspberry Pi troubleshooting](#)

[Appendix B. Raspberry Pi ports and legacy boards](#)

[Appendix C. Solutions to chapter challenges](#)

[Appendix D. Raspberry Pi projects](#)

[Index](#)

[List of Figures](#)

[List of Listings](#)

Table of Contents

[Copyright](#)

[Brief Table of Contents](#)

[Table of Contents](#)

[Preface](#)

[Acknowledgments](#)

[About this book](#)

[1. Getting started](#)

[Chapter 1. Meet Raspberry Pi](#)

[What is the Raspberry Pi?](#)

[Exploring your Raspberry Pi's parts: hardware](#)

[Giving your Pi a cozy home: Pi cases](#)

[The brain of your Pi: system on a chip](#)

[Connecting a keyboard and mouse: USB ports](#)

[Storing memories: your Pi gets a memory card](#)

[Connecting a TV or monitor: HDMI port](#)

[Other ports and connections](#)

[Powering your Pi: microUSB power port](#)

[It's alive! Plugging in the Pi](#)

[Getting your Pi running: software](#)

[Installing the Raspbian operating system](#)

[Configuring the operating system: making it yours](#)

[Saving your configuration and rebooting](#)

[Getting around: learning Raspbian](#)

[Finding and opening applications on your Raspberry Pi](#)

[Your files and folders](#)

[Writing code](#)

[Fruit Picker Extra: shopping at the Pi Store](#)

[Challenge](#)

[Scavenger hunt](#)

[Summary](#)

Playing with Python

Discovering Python's mathematical operators

Adding and subtracting

Multiplying and dividing

Figuring out whole numbers and remainders

Exponents

Square roots

Challenge: stacking Pis!

Storing information using variables

Creating variables and assigning values

Displaying variable values

Storing strings in variables

Changing the value of variables

Displaying text on a screen

Using the print function

Troubleshooting

Creating programs

Writing Python programs with IDLE

Starting a new program

Saving programs

Python interpreting the program

Fruit Picker Extra: creating documents

Writing silly things and saving them

Challenges

The matrix

Building a brick wall

Pi electrons

Summary

2. Playing with Python

Chapter 3. Silly Sentence Generator 3000: creating interactive programs

Creating a welcome message

[Starting a new program](#)

[Saving the program](#)

Adding notes in your code

[Using hashtags for comments](#)

Getting and storing information

Joining strings

[Using more than one input](#)

[Building the sentence](#)

[Troubleshooting](#)

Completing the program: displaying the silly sentence

Fruit Picker Extra: Minecraft Pi

[What's Minecraft?](#)

[Launching Minecraft Pi](#)

[Python programming interface to Minecraft Pi](#)

Challenges

[Knight's Tale Creator 3000](#)

[Subliminal messages](#)

Summary

Chapter 4. Norwegian Blue parrot game: adding logic to programs

Displaying the game introduction

[Creating the game welcome message and instructions](#)

Collecting input from the player

Using if statements to respond to users in different ways

[Practicing if statements](#)

Using while loops to repeat things

[A closer look at while loops](#)

[Breaking out of a while loop](#)

[Practicing while loops](#)

[Using Python code libraries to generate random numbers](#)

[Fruit Picker Extra: Scratch](#)

[Challenges](#)

[Summary](#)

[Chapter 5. Raspi's Cave Adventure](#)

[Project introduction: Raspi's Cave Adventure](#)

[Left cave](#)

[Right cave](#)

[Hey wait, you need a plan \(flow diagrams\)](#)

[Which way should Raspi go? \(checking input\)](#)

[Handling unexpected input](#)

[Turning flow diagrams into code](#)

[Simplify! Making your own functions](#)

[Finishing the left cave](#)

[Exploring the right cave](#)

[Troubleshooting](#)

[Fruit Picker Extra: playing video](#)

[Live streaming: exploring from your Pi](#)

[Challenges](#)

[Introducing dramatic pauses](#)

[Random demise](#)

[Play again?](#)

[Scream!](#)

[Summary](#)

[3. Pi and Python projects](#)

[Chapter 6. Blinky Pi](#)

[Setting up your Pi for physical computing](#)

[GPIO pins](#)

[Breaking out the GPIO pins to a breadboard](#)

Breadboard basics

Building the LED circuit

Step 1. Connect the jumper from GPIO pin 21

Step 2. Add the red LED

Step 3. Connect a resistor

Software: blinkLED program

Running the program

blinkLED: how it works

Adding more LEDs

Building the circuit

Multiple LEDs: program it!

Challenges

Wave pattern

Simon Says

Random blinking

Summary

Chapter 7. Light Up Guessing Game

Guessing Game design

Hardware: building the circuit

Numbers, numbers, numbers!

Wiring an RGB LED

Circuit sketch

Software: LEDGuessingGame program

Setting up the GPIO pins for the RGB LED

Main game loop and logic

Guessing Game Loop and logic

Adding the Play Again Loop and logic

Playing the game

Troubleshooting

Challenges

[Game winner](#)

[Easter egg](#)

[Warmer and colder](#)

[Darth Vader surprise](#)

[Summary](#)

[Chapter 8. DJ Raspi](#)

[Project overview](#)

[Setting up your Pi to play sounds](#)

[OMXPlayer and MP3s](#)

[Troubleshooting](#)

[Hardware: building the circuit](#)

[Wiring a button](#)

[Circuit sketch](#)

[Adding the second button](#)

[Software: the DJ Raspi program](#)

[Setting up the Pi: initializing the buttons](#)

[Getting a list of sounds](#)

[Getting a value of an item stored in a list](#)

[Getting the length of a list](#)

[Building a list of sound files with the os library](#)

[Playing a sound when a button is pressed](#)

[Functions!](#)

[Testing: your first gig as DJ Raspi](#)

[Troubleshooting](#)

[Challenges](#)

[Double button press surprise](#)

[Yoda Magic 8 Ball](#)

[Continuing to explore](#)

[Summary](#)

[Appendix A. Raspberry Pi troubleshooting](#)

[Making sure your Pi has power](#)

[Checking the connection to your TV or monitor](#)

[Pi starts booting up but then stops](#)

[Making your Pi a new SD card](#)

[Reformatting your SD card](#)

[Problems not covered here](#)

[Appendix B. Raspberry Pi ports and legacy boards](#)

[Raspberry Pi ports](#)

[Connecting to a wireless network](#)

[3.5 mm audio/video port](#)

[Camera Serial Interface: connecting a camera](#)

[Ethernet port](#)

[TV or monitor connection options](#)

[Legacy boards](#)

[Raspberry Pi 1 Model B](#)

[Raspberry Pi 1 Model B+](#)

[Other boards](#)

[Appendix C. Solutions to chapter challenges](#)

[Chapter 1](#)

[Chapter 2](#)

[The Matrix](#)

[Building a brick wall](#)

[Pi electrons](#)

[Chapter 3](#)

[Knight's Tale Creator 3000](#)

[Subliminal messages](#)

[Chapter 4](#)

[Rock, Paper, Scissors!](#)

[Chapter 5](#)

[Introducing dramatic pauses](#)

[Random demise](#)

[Play again?](#)

[Chapter 6](#)

[Wave pattern](#)

[Simon says](#)

[Random blinking](#)

[Chapter 7](#)

[Game winner](#)

[Easter egg](#)

[Warmer and colder](#)

[Darth Vader surprise](#)

[Chapter 8](#)

[Double button press surprise](#)

[Yoda Magic 8 Ball](#)

[Appendix D. Raspberry Pi projects](#)

[Halloween heads](#)

[Time-lapse photography](#)

[Raspberry Pi robot](#)

[Index](#)

[List of Figures](#)

[List of Listings](#)

Preface

In 2013, a parent and friend of mine asked if I would teach a Python course to middle school students at a local school. My friend gently asked if I could somehow use the Raspberry Pi computer in the course. I love learning new things and I had been reading a lot about the Raspberry Pi. So as you can imagine, I was tremendously excited at the opportunity of using it and emphatically said “Yes!” That event began my journey of developing a course for kids on programming in Python and using the Raspberry Pi and later, this book.

Quickly, as I worked with the Raspberry Pi, I became a disciple of the Raspberry Pi inventors: the best way for kids to learn programming is by giving them an affordable, ready-to-program computer. It was the perfect platform to learn how to program.

As a teacher of computer science, I grew to deeply appreciate Python. I became convinced that it was not only a great programming language, but its focus on readability and simplicity made it perfect for kids to learn as their first programming language.

Fast forward in time—after teaching Python using the Raspberry Pi to many classes of kids, I had developed a set of engaging and funny projects that the kids enjoyed. Just as important, the students learned! The feedback from the kids and the parents was fantastic! Imagine kids rushing to take part in a programming class. It was wonderful!

A few months after developing my course, Nicole Butterfield and Robin de Jongh of Manning Publications contacted me about turning it into a book. I was thrilled at the prospect of bringing the activities and projects from the computer lab into the hands of kids everywhere. What is more, this book would fill an important gap. What I had found when I originally started teaching my course was that there were no books on the Raspberry Pi and programming in Python that were designed for kids. Since the main reason for inventing the Raspberry Pi was to get more kids programming, I was enthusiastic to work on this project.

Nearly two years later, and several versions of the Raspberry Pi later, I’m proud to present this book to the kids and other beginners who want to learn to program. I hope you enjoy using this book and it starts you on your own journey in computer science!

Acknowledgments

Thank you to my wife, Juliana, and our two children, Daniel and John, for their endless support and patience through the long days, nights, and weekends I needed to write this book.

I'd also like to thank Manning Publications for having the vision to pursue this project. In particular, thanks to Robin de Jongh and Nicole Butterfield who kicked off this project by finding and encouraging me; to publisher Marjan Bace for his commitment to me and to this book; to Ozren Harlovic for orchestrating the book review process; to Kevin Sullivan and Mary Piergies for overseeing production; to Chuck Larson for the wonderful work on the graphics; to Tiffany Taylor for her outstanding copyediting; to Alyson Brener for her thorough proofreading; to Candace Gillhoolley and Ana Romac for promoting the book; to technical development editors Donald Bailey, Joel Kotarski, Jeanne Boyarsky, and John Hyaduck; and to Romin Irani, technical proofreader.

This book was significantly improved by my editor at Manning, Dan Maharry, who helped to develop and edit the book from concept to finished product. I'd like to thank Dan for his excellent insights, support, encouragement, and guidance throughout the process.

A big thank you to all the technical reviewers who read the manuscript at various stages of its development and contributed invaluable feedback: Adam Hinden, Antonio Mas Rodriguez, Betsy Hoofnagle, Catherine Freytag, Dr. Christian Mennerich, Dan Kacendar, David Kerns, Ema Battista, Fanick Atchia, Grace Kacendar, Henry Freytag, Jaqueline Currie, John Pentakalos, Keenan Hom, Kevin Adjaho Atchia, Matthew Giblin, Nathan Sperry, Odysseas Pentakalos, Sam Kerns, Richard Freytag, Savannah Wilson, and Scott M. King.

Thank you also to all the readers who bought and read the MEAP (Manning Early Access Program) versions of the chapters and who took the time to post comments in the Author Online forum. You helped make this a better book!

The Raspberry Pi Foundation, original inventors, and community deserve a special mention. Thank you for designing something that is helping children to learn computer science. I'd also like to thank Guido van Rossum, the inventor of Python; the Python Software Foundation; and the Python user community, for creating and maintaining a simple and useful programming language for everyone.

About this book

The Raspberry Pi is a small, low-cost computer invented in the U.K. by the Raspberry Pi Foundation. It provides an easy-to-use tool for learning to program in Python. The Raspberry Pi, with its companion memory card, is preloaded with all the software you need to jump into programming in Python. The Raspberry Pi is made for you to learn to code by playing with it. It includes many input and output ports to give you flexibility in how you connect it. Much like a desktop computer, you need to connect a keyboard, mouse, monitor, and power cable to get started.

This book will teach you how to set up your Raspberry Pi, to write programs in Python, and to use your Raspberry Pi and Python to complete some projects. We'll cover the basics of Python: displaying text, gathering input, repeating commands, creating logic, as well as using the input and output pins of your Raspberry Pi for projects.

This book does not cover advanced Python topics, nor act as a comprehensive reference for Python. Since it is a book for beginners, these topics have been left out for clarity and brevity. If you'd like to learn more Python, there are links to online resources throughout the book.

This book is for kids and other beginners who would like to learn to program. It's also for kids who have a Raspberry Pi and want to learn what they can do with it. We'll introduce you to your Raspberry Pi and teach you Python in a natural, playful way, introducing topics and giving you activities to do using your Raspberry Pi. You don't need to have any prior programming experience. As long as you know how to use a mouse and open up programs by clicking on icons or menu items, you'll do great.

This book requires a Raspberry Pi, cables, and some other parts to complete the projects and activities. These items are needed throughout the book:

- Raspberry Pi 2 Model B
- 8 GB SD memory card, preloaded with the Raspberry Pi Foundation's NOOBS (New Out of the Box Software)
- USB power supply with micro USB cable (must deliver 1.2 A @ 5 V)
- USB keyboard
- USB mouse
- TV or monitor
- Cable to connect to TV or monitor (specific cables for your TV or monitor are discussed in [chapter 1](#))

To complete the projects in [part 3](#), you'll also need these parts:

- Solderless breadboard
- GPIO ribbon cable for the Raspberry Pi 2 Model B (40 pin)
- GPIO breakout board
- 1 dozen jumper wires, male-to-male
- 1 red LED (light-emitting diode)
- 1 green LED
- 1 blue LED

- 1 red, green, blue (RGB) LED
 - ~~3 push buttons~~
 - 3 resistors, 10K ohm
 - 3 resistors, 180 ohm (or between 100 and 300 ohms)
 - Headphones or powered computer speakers
-

You can typically find all these items in a Raspberry Pi starter kit or available individually through online retailers and stores that sell the Raspberry Pi, such as CanaKit, Sparkfun, or Adafruit.

Roadmap

This book is divided into three parts.

[Part 1](#) introduces you to the Raspberry Pi, shows you how to set it up, and provides an introduction to the Python programming language:

- [Chapter 1](#) provides an overview of the Raspberry Pi and how to set it up for the first time.
- [Chapter 2](#) shows you how to write your first Python programs and introduces you to doing math and displaying text with Python.

[Part 2](#) shows you how to build different text-based games while learning how to gather input, display information, make decisions, and repeat instructions in Python:

- [Chapter 3](#) teaches you how to create your first interactive Python game, the Silly Sentence Generator 3000, by asking users to type in something and then displaying funny messages to the screen.
- [Chapter 4](#) explores how to give your programs logic and use repeating loops as you create a Norwegian Blue Guessing Game.
- [Chapter 5](#) demonstrates how to build a Cave Adventure Game, give users multiple choices, check input from users, and create your own Python functions.

[Part 3](#) involves making your Raspberry Pi interact with the world around it:

- [Chapter 6](#) explains setting up your Pi with an electronics breadboard, building a simple circuit, and controlling an LED (light) using your Raspberry Pi and Python.
- [Chapter 7](#) dives into creating an interactive guessing game that uses lights to respond to a player's input, letting them know with different colors whether their answer is right or wrong.
- [Chapter 8](#) teaches you how to listen to your Pi's input pins by making a project that combines light and sound to make your own DJ Raspi sound mixer.

Code conventions and downloads

All source code in this book is in a `fixed-width font like this`, which sets it apart from the surrounding text. In many listings, the code is annotated to point out key concepts. I have tried to format the code so that it fits within the available page space in the book by adding line breaks and using indentation carefully.

The code accompanying this book is hosted at the GitHub repository: <https://github.com/rheitz/hello-raspberry-pi>. It is also available for download as a zip file from the publisher's website at www.manning.com/books/hello-raspberry-pi.

Author Online

Purchase of *Hello Raspberry Pi!* includes free access to a private web forum run by Manning Publications where you can make comments about the book, ask technical questions, and receive help from the author and other users. To access the forum and subscribe to it, point your web browser to www.manning.com/books/hello-raspberry-pi. This Author Online (AO) page provides information on how to get on the forum once you're registered, what kind of help is available, and the rules of conduct on the forum.

Manning's commitment to our readers is to provide a venue where a meaningful dialog among individual readers and between readers and the author can take place. It's not a commitment to any specific amount of participation on the part of the author, whose contribution to the AO remains voluntary (and unpaid). We suggest you try asking the author some challenging questions, lest his interest stray!

The AO forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

About the author

Ryan Heitz is a teacher, programmer, maker, father, and big kid. He is the cofounder of Ideaventions a Science Center for kids, and Ideaventions Academy for Mathematics and Science, a private school focused on science and technology. He specializes in teaching kids how to experience computer science in a fun and engaging way. As a programmer, Ryan has developed software for everything from NASA data collection systems to web mapping applications.

Part 1. Getting started

Get ready to explore Python using your Raspberry Pi! You'll need a Raspberry Pi and a few other parts and cables for [part 1](#). Here's your shopping list:

- Raspberry Pi 2 Model B
- 8 GB SD memory card, preloaded with the Raspberry Pi Foundation's NOOBS (New Out Of the Box Software)
- USB power supply with micro USB cable (must deliver 1.2 A @ 5 V)
- USB keyboard
- USB mouse
- TV or monitor
- Cable to connect to TV or monitor (specific cables for your TV or monitor are discussed in [chapter 1](#))

Optional item:

- Raspberry Pi case

[Part 1](#) will get you on your way to using your Raspberry Pi and launch you into programming it with Python. In [chapter 1](#), you'll set up your Raspberry Pi, learn how to start (or boot) it up, and then look around inside the Pi's desktop. [Chapter 2](#) is where you'll start exploring the Python language. You'll create your first programs and learn to give instructions to your Raspberry Pi using Python.

By the end of [part 1](#), you'll know how to get a Raspberry Pi up and running. You'll be able to write a Python program and interact with your Pi to make it do things like figure out the cost of a cheeseburger meal and display silly messages on the screen.

Chapter 1. Meet Raspberry Pi

In this chapter, you'll learn how to

- *Set up your Raspberry Pi*
- *Install an operating system—Raspbian—on your Pi*
- *Find and open applications*
- *Write your first bit of code in Python*

What kinds of things do you think you can do with a Raspberry Pi?

1. Play games.
2. Watch videos.
3. Create a video game.
4. Listen to music.
5. Make a sound mixer for a dance party.
6. Build a robot.

Believe it or not, these are all projects you can do yourself, and if you learn to program in Python, the sky is the limit. You can achieve quite a lot on your Pi, as long as you can write a program to do it. But before we talk about that, let's take a look at a Raspberry Pi and discover what makes it tick.

What is the Raspberry Pi?

The *Raspberry Pi*, sometimes referred to as the *Pi*, is a small, low-cost computer invented in the U.K. by the Raspberry Pi Foundation. It provides an easy-to-use tool to help you learn to code in Python (the *Pi* part of its name came from the focus on using it to code in Python).

About the size of a deck of cards, it isn't as powerful as a laptop or desktop computer; its computing power is more similar to that of a smart phone. But what it lacks in processing power, it makes up for in its many features:

- Its readiness for programming in Python
- The many ways you can use it
- Its small size and cost

The Pi, with its companion memory card, is preloaded with all the software you need to jump into programming in Python. Type in commands, and see what happens. Enter a program you find on the internet or in a magazine, run it, and see how it works. The Pi is made for you to learn to code by playing with it, using it, and interacting with it.

Once you learn to program in Python, you can use your Pi as a base for all sorts of projects—with your imagination, the possibilities are endless! The Pi's small size makes it easy to carry around and include in projects. Hide it on a shelf or mount it on a wall with a camera to make a security system; power it with a rechargeable battery pack if you need it to be portable; or even attach it to a remote-controlled car or helicopter. And if you happen to mess something up, it's simple to recover. Even if you manage to break the Pi, it's pretty cheap to replace.

At its core, the Raspberry Pi is a circuit board that has all the components found in many computers. The next section checks out the components of the Pi and explores what they do. Let's go!

- [download online Start with Why: How Great Leaders Inspire Everyone to Take Action book](#)
- **Marked pdf, azw (kindle), epub**
- [click Schaum's Outline of Applied Physics \(4th Edition\) \(Schaum's Outlines Series\)](#)
- [click TraitÃ©s, 42-44 : Sur les genres de l'Ãatre, I, II et III pdf, azw \(kindle\), epub, doc, mobi](#)
- [read online Heroes: Canadian Champions, Dark Horses and Icons pdf, azw \(kindle\)](#)

- <http://patrickvincitore.com/?ebooks/Start-with-Why--How-Great-Leaders-Inspire-Everyone-to-Take-Action.pdf>
- <http://dadhoc.com/lib/On-the-Origins-of-Classical-Economics--Distribution-and-Value-from-William-Petty-to-Adam-Smith.pdf>
- <http://test.markblaustein.com/library/Schaum-s-Outline-of-Applied-Physics--4th-Edition---Schaum-s-Outlines-Series-.pdf>
- <http://econtact.webschaefer.com/?books/Trait--s--42-44---Sur-les-genres-de-l---tre--l--II-et-III.pdf>
- <http://anvilpr.com/library/Heroes--Canadian-Champions--Dark-Horses-and-Icons.pdf>