



Python Projects for Beginners

A Ten-Week Bootcamp Approach to
Python Programming

—
Connor P. Milliken

Apress®

Python Projects for Beginners

**A Ten-Week Bootcamp Approach to
Python Programming**

Connor P. Milliken

Apress®

Python Projects for Beginners

Connor P. Milliken
Derry, NH, USA

ISBN-13 (pbk): 978-1-4842-5354-0
<https://doi.org/10.1007/978-1-4842-5355-7>

ISBN-13 (electronic): 978-1-4842-5355-7

Copyright © 2020 by Connor P. Milliken

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Nikhil Karkal
Development Editor: Rita Fernando
Coordinating Editor: Divya Modi

Cover designed by eStudioCalamar

Cover image designed by Pixabay

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-5354-0. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

This book is dedicated to my girlfriend Jess.

Ever since we first met, you changed my life forever.

*There's so much that I wish to tell you each day,
like how beautiful you are, how you inspire me, or how I would
give anything just to be with you every second of the day.*

*Your smile lights up my whole world and you make me so
unbelievably happy.*

Anytime I have a bad day, I know you'll always be there for me.

*I thought that I would only find you in my dreams, but here you are,
standing in front of me, looking beautiful as ever.*

From the day I met you, I knew I wanted to give you everything.

*You're smart, motivated, beautiful, and resemble all that is
right with this world.*

If I only do one thing right in life, I'd like it to be you.

*I promise to always push you to be better, always support
you in times of need, and always be there with a Werther's
candy to help you study.*

Your dreams have become my dreams, and whatever you want in life,

I want to be there to celebrate and help guide you.

I will always love you, past forever, with all my heart and soul.

So I have only one question left for you...

(turn the page)

Will You Marry Me?

Table of Contents

About the Author	xxi
About the Technical Reviewer	xxiii
Acknowledgments	xxv
Chapter 1: Getting Started	1
Monday: Introduction	2
What Is Python?	2
Why Python?	3
Why This Book?	4
Who This Book Is For?	4
What You'll Learn	5
Tuesday: Setting Up Anaconda and Python	6
Cross-Platform Development	6
Installing Anaconda and Python for Windows	6
What Is Anaconda?	8
What Is Jupyter Notebook?	8
Wednesday: How to Use the Terminal	9
Changing Directories	9
Checking the Directory	10
Making Directories	10
Creating Files	10
Checking a Version Number	11
Clearing the Terminal Output	11
Using the Python Shell	12
Writing Your First Line of Python	12
Exiting the Python Shell	13

TABLE OF CONTENTS

- Thursday: Using Jupyter Notebook 13
 - Opening Jupyter Notebook 14
 - Creating a Python File 14
 - Jupyter Notebook Cells..... 15
- Friday: Creating Your First Program 17
 - Line Numbers Introduced 17
 - Creating the Program 18
 - Final Output 19
- Weekly Summary 20
- Weekly Challenges 20
- Chapter 2: Python Basics 21**
- Monday: Comments and Basic Data Types 22
 - What Are Comments and Why Use Them?..... 22
 - Writing Comments..... 23
 - What Are Data Types? 24
 - The Print Statement..... 24
 - Integers 25
 - Floats..... 25
 - Booleans..... 25
 - Strings 26
- Tuesday: Variables..... 27
 - How They Work..... 27
 - Handling Naming Errors 28
 - Integer and Float Variables 28
 - Boolean Variables 29
 - String Variables 29
 - Using Multiple Variables 29
 - Using Operators on Numerical Variables 30
 - Overwriting Previously Created Variables..... 30
 - Whitespace 31

Wednesday: Working with Strings	31
String Concatenation	32
Formatting Strings.....	32
String Index	34
String Slicing	36
Thursday: String Manipulation	37
.title()	37
.replace().....	37
.find().....	38
.strip()	38
.split().....	39
Friday: Creating a Receipt Printing Program.....	39
Final Design.....	40
Initial Process.....	40
Defining Our Variables	41
Creating the Top Border.....	42
Displaying the Company Info.....	42
Displaying the Product Info	43
Displaying the Total	44
Displaying the Ending Message	44
Displaying the Bottom Border.....	45
Weekly Summary	45
Challenge Question Solution	45
Weekly Challenges.....	46
Chapter 3: User Input and Conditionals.....	47
Monday: User Input and Type Converting.....	48
Accepting User Input	48
Storing User Input.....	48
What Is Type Converting?	49
Checking the Type	49

TABLE OF CONTENTS

Converting Data Types..... 49

Converting User Input..... 50

Handling Errors..... 51

Code Blocks and Indentation..... 52

Tuesday: If Statements..... 52

 How They Work..... 53

 Writing Your First If Statement 53

 Comparison Operators..... 54

 Checking User Input 54

 Logical Operators 55

 Membership Operators..... 56

Wednesday: Elif Statements 58

 How They Work..... 58

 Writing Your First Elif Statement..... 59

 Checking Multiple Elif Conditions 59

 Conditionals Within Conditionals 60

 If Statements vs. Elif Statements 60

Thursday: Else Statements 62

 How They Work..... 62

 Writing Your First Else Statement..... 62

 Complete Conditional Statement..... 63

Friday: Creating a Calculator..... 64

 Final Design..... 65

 Step #1: Ask User for Calculation to Be Performed 65

 Step #2: Ask for Numbers, Alert Order Matters 66

 Step #3: Set Up Try/Except for Mathematical Operation..... 66

 Final Output..... 67

Weekly Summary..... 69

Challenge Question Solution 69

Weekly Challenges..... 69

Chapter 4: Lists and Loops	71
Monday: Lists	72
What Are Lists?.....	72
Declaring a List of Numbers	72
Accessing Elements Within a List.....	73
Declaring a List of Mixed Data Types.....	73
Lists Within Lists.....	74
Accessing Lists Within Lists	74
Changing Values in a List.....	75
Variable Storage	76
Copying a List.....	77
Tuesday: For Loops	78
How Loops Work.....	78
Writing a For Loop	78
Range()	80
Looping by Element.....	80
Continue Statement.....	81
Break Statement.....	82
Pass Statement	82
Wednesday: While Loops.....	83
Writing a While Loop.....	84
While vs. For.....	84
Infinite Loops	84
Nested Loops.....	85
Thursday: Working with Lists.....	86
Checking Length.....	87
Slicing Lists	87
Adding Items	88
Removing Items.....	88
Working with Numerical List Data	90
Sorting a List	90

TABLE OF CONTENTS

- Conditionals and Lists 91
- Loops and Lists..... 92
- Friday: Creating Hangman..... 93
 - Final Design..... 94
 - Previous Line Symbols Introduced 94
 - Adding Imports 95
 - Declaring Game Variables..... 96
 - Generating the Hidden Word..... 96
 - Creating the Game Loop 97
 - Outputting Game Information 97
 - Checking a Guess 98
 - Clearing Output..... 98
 - Creating the Losing Condition 99
 - Handling Correct Guesses 99
 - Creating a Winning Condition 100
 - Outputting Guessed Letters 101
 - Adding Guessed Letters..... 101
 - Handling Previous Guesses 102
 - Final Output 102
- Weekly Summary 103
- Challenge Question Solution 103
- Weekly Challenges 104
- Chapter 5: Functions 105**
 - Monday: Creating and Calling Functions..... 106
 - What Are Functions?..... 106
 - Function Syntax..... 107
 - Writing Your First Function 107
 - Function Stages..... 108
 - UDF vs. Built-in..... 109
 - Performing a Calculation 109

Tuesday: Parameters.....	110
What Are Parameters?.....	110
Passing a Single Parameter	111
Multiple Parameters	111
Passing a List	112
Default Parameters.....	113
Making Parameters Optional	113
Named Parameter Assignment.....	114
*args.....	114
**kwargs.....	115
Wednesday: Return Statement.....	116
How It Works	116
Using Return.....	117
Ternary Operator.....	118
Thursday: Scope	119
Types of Scope	119
Global Scope Access	119
Handling Function Scope.....	120
In-Place Algorithms	120
Friday: Creating a Shopping Cart	121
Final Design.....	122
Initial Setup	122
Adding Items	123
Removing Items.....	123
Showing the Cart.....	124
Clearing the Cart.....	124
Creating the Main Loop	124
Handling User Input.....	125
Final Output.....	126

TABLE OF CONTENTS

- Weekly Summary 126
- Challenge Question Solution 127
- Weekly Challenges 127
- Chapter 6: Data Collections and Files 129**
- Monday: Dictionaries 129
 - What Are Dictionaries? 130
 - Declaring a Dictionary 130
 - Accessing Dictionary Information..... 131
 - Using the Get Method 131
 - Dictionaries with Lists 132
 - Lists with Dictionaries 132
 - Dictionaries with Dictionaries..... 133
- Tuesday: Working with Dictionaries 134
 - Adding New Information..... 134
 - Changing Information 135
 - Deleting Information 135
 - Looping a Dictionary..... 135
- Wednesday: Tuples, Sets, Frozensets 137
 - What Are Tuples? 137
 - Declaring a Tuple 138
 - What Are Sets? 138
 - Declaring a Set 138
 - What Are Frozensets? 139
 - Declaring a Frozenset..... 139
 - Data Collection Differences 140
- Thursday: Reading and Writing Files..... 140
 - Working with Text Files..... 141
 - Writing to CSV Files 142
 - Reading from CSV Files 142
 - File Modes in Python 143

Friday: Creating a User Database with CSV Files	144
Final Design.....	144
Setting Up Necessary Imports.....	145
Handling User Registration.....	145
Handling User Login	146
Creating the Main Loop	147
Weekly Summary	148
Challenge Question Solution	149
Weekly Challenges.....	149
Chapter 7: Object-Oriented Programming	151
Monday: Creating and Instantiating a Class.....	152
What Is an Object?.....	152
OOP Stages.....	153
Creating a Class.....	153
Creating an Instance.....	154
Creating Multiple Instances.....	154
Tuesday: Attributes.....	156
Declaring and Accessing Attributes.....	156
Changing an Instance Attributes	157
Using the <code>__init__()</code> Method	157
The “self” Keyword.....	158
Instantiating Multiple Objects with <code>__init__()</code>	159
Global Attributes vs. Instance Attributes.....	159
Wednesday: Methods	161
Defining and Calling a Method	161
Accessing Class Attributes in Methods	162
Method Scope.....	162
Passing Arguments into Methods.....	163
Using Setters and Getters.....	164
Incrementing Attributes with Methods	165

TABLE OF CONTENTS

- Methods Calling Methods 166
- Magic Methods 166
- Thursday: Inheritance 168
 - What Is Inheritance? 168
 - Inheriting a Class 168
 - Using the super() Method 169
 - Method Overriding 170
 - Inheriting Multiple Classes 171
- Friday: Creating Blackjack 172
 - Final Design 173
 - Setting Up Imports 174
 - Creating the Game Class 174
 - Generating the Deck 175
 - Pulling a Card from the Deck 175
 - Creating a Player Class 176
 - Adding Cards to the Player’s Hand 177
 - Showing a Player’s Hand 178
 - Calculating the Hand Total 179
 - Handling the Player’s Turn 181
 - Handling the Dealer’s Turn 182
 - Calculating a Winner 183
 - Final Output 184
- Weekly Summary 184
- Challenge Question Solution 185
- Weekly Challenges 185
- Chapter 8: Advanced Topics I: Efficiency 187**
 - Monday: List Comprehension 188
 - List Comprehension Syntax 188
 - Generating a List of Numbers 189
 - If Statements 190
 - If-Else Statements 190

List Comprehension with Variables	191
Dictionary Comprehension	192
Tuesday: Lambda Functions.....	193
Lambda Function Syntax	193
Using a Lambda.....	193
Passing Multiple Arguments.....	194
Saving Lambda Functions	195
Conditional Statements	195
Returning a Lambda	196
Wednesday: Map, Filter, and Reduce	197
Map Without Lambdas.....	197
Map with Lambdas.....	198
Filter Without Lambdas.....	199
Filter with Lambdas.....	200
The Problem with Reduce	201
Using Reduce.....	201
Thursday: Recursive Functions and Memoization.....	203
Understanding Recursive Functions.....	203
Writing a Factorial Function	204
The Fibonacci Sequence	205
Understanding Memoization.....	206
Using Memoization	207
Using @lru_cache	208
Friday: Writing a Binary Search.....	209
Final Design.....	209
Program Setup.....	211
Step 1: Sort the List.....	211
Step 2: Find the Middle Index	212
Step 3: Check the Value at the Middle Index	213
Step 4: Check if Value Is Greater	213
Step 5: Check if Value Is Less.....	214

TABLE OF CONTENTS

- Step 6: Set Up a Loop to Repeat Steps 214
- Step 7: Return False Otherwise 215
- Final Output 216
- Weekly Summary 217
- Challenge Question Solution 217
- Weekly Challenges 218
- Chapter 9: Advanced Topics II: Complexity 219**
- Monday: Generators and Iterators..... 220
 - Iterators vs. Iterables..... 220
 - Creating a Basic Iterator..... 220
 - Creating Our Own Iterator..... 221
 - What Are Generators?..... 222
 - Creating a Range Generator 222
- Tuesday: Decorators..... 224
 - What Are Decorators?..... 224
 - Higher-Order Functions 225
 - Creating and Applying a Decorator 225
 - Decorators with Parameters..... 226
 - Functions with Decorators and Parameters 226
 - Restricting Function Access 227
- Wednesday: Modules 229
 - Importing a Module 229
 - Importing Only Variables and Functions 230
 - Using an Alias 231
 - Creating Our Own Module 231
 - Using Our Module in Jupyter Notebook..... 232
- Thursday: Understanding Algorithmic Complexity 234
 - What Is Big O Notation?..... 234
 - Hash Tables 236
 - Dictionaries vs. Lists 238
 - Battle of the Algorithms..... 239

Friday: Interview Prep	241
Developer Interview Process	241
What to Do Before the Interview	243
General Questions	245
Whiteboarding and Technical Questions.....	248
End of Interview Questions.....	249
What to Do After the Interview	250
Weekly Summary	251
Challenge Question Solution	252
Weekly Challenges	252
Chapter 10: Introduction to Data Analysis.....	253
Monday: Virtual Environments and Requests Module	254
What Are Virtual Environments?	254
What Is Pip?.....	256
Creating a Virtual Environment.....	256
Activating the Virtual Environment	257
Installing Packages	258
APIs and the Requests Module.....	259
Using the Requests Module.....	259
Tuesday: Pandas	263
What Is Pandas?	263
Key Terms	264
Installing Pandas	265
Importing Pandas	265
Creating a DataFrame.....	265
Accessing Data.....	267
Built-in Methods	268
Filtration	271
Column Transformations.....	272
Aggregations	274

TABLE OF CONTENTS

- Pandas Joins 277
- Dataset Pipeline..... 280
- Wednesday: Data Visualization 281
 - Types of Charts 282
 - Installing Matplotlib..... 282
 - Importing Matplotlib 283
 - Line Plot..... 283
 - Bar Plot..... 285
 - Box Plot 286
 - Scatter Plot..... 288
 - Histogram 289
 - Saving the Chart 292
 - Flattening Multidimensional Data..... 293
- Thursday: Web Scraping 295
 - Installing Beautiful Soup..... 295
 - Importing Beautiful Soup..... 295
 - Requesting Page Content 296
 - Parsing the Response with Beautiful Soup 297
 - Scraping Data..... 297
 - DOM Traversal..... 299
- Friday: Web Site Analysis 304
 - Final Design..... 304
 - Importing Libraries 306
 - Creating the Main Loop 307
 - Scraping the Web Site 307
 - Scrape All Text 308
 - Filtering Elements..... 309
 - Filtering Waste..... 310
 - Count Word Frequency 312
 - Sort Dictionary by Word Frequency 313
 - Displaying the Top Word 313

Graphing the Results	314
Final Output	315
Weekly Summary	315
Challenge Question Solution	316
Weekly Challenges	316
Afterword: Post-Course: What to Do Now?	319
Back-End Development with Python.....	319
Full-Stack Development with Python.....	320
Data Analysis with Python.....	320
Data Science with Python	320
Resources	320
Final Message	323
Index.....	325

About the Author



Connor P. Milliken Focused on helping others achieve their goals through education and technology, **Connor P. Milliken** brings a wealth of programming and business experience to his classes.

He graduated with a computer science degree from Daniel Webster College and is pursuing a master’s in computer science with a focus in interactive intelligence from Georgia Tech.

Before becoming an instructor at Coding Temple, he was designing simulators in the video game industry for several years. During that time, he took on a vast number of roles from business to programming that he used to release a total of 11 different titles on PC and co-created an award-winning football card game called “Masters of the Gridiron.”

Connor has experience in more than seven different languages and three frameworks. He focuses primarily in web development and data analytics using Python. When this book was written, he taught for a coding bootcamp in Boston, MA, where students can learn Python, web development, and data analytics over a 10-week full-time course. He is now a software engineer at Hubspot, Inc. in Cambridge, MA.

Github: *Connor-SM*

About the Technical Reviewer



Bharath Thiruveedula currently works for a major telco service provider. He is core reviewer and key contributor to various OpenStack/ONAP projects. Bharath is passionate about open source technologies and is an evangelist who is focused on making his mark in the Cloud/Container domains. He has been working on distributed systems and machine learning for a significant amount of time.

Acknowledgments

I would like to thank the following people for their generosity and help:

Jessica Boucher, who has been my rock this whole time. Your love and support have continued to help me in all my endeavors. I'm truly blessed to have you in my life.

My family, who have supported and believed in me all my life. Without your guidance, none of this would be possible. To have parents and siblings like you all is nothing short of a miracle and I wouldn't have it any other way.

Clay and Dee Dreslough, who gave me an opportunity and mentored me. This book would not be possible without your guidance over the years. It was at Sports Mogul that I had realized my passion of computer programming, thanks to you both.

Derek Hawkins, who mentored and taught me a lot about teaching, programming, Python, and Ping Pong.

Kirsten Arnold, who created all the art within this book. The work you were able to create from my poor drawing skills was exactly what I had imagined.

Ripal Patel, who helped with the interview portion of Week 9. Your expertise in the hiring and interview process has been wonderful for not only me but the students.

My friends, who over the years have been there for me through it all. Whether it was watching my dog, going on adventures, or just hanging out... thank you. I will always make the drive for you all.

My coaches, who taught me about perseverance, hard work, commitment, and teamwork. Whether it was 6 AM practices or triple sessions in the middle of summer, you've played a big part in my life and for that I'm grateful.

ACKNOWLEDGMENTS

The Coding Temple team, who gave me the opportunity and entrusted me to educate those wanting to pursue a career in tech.

The Apress team, who have helped me throughout this entire process with writing, formatting, reviewing, and more.

My students, who helped to show me why teaching is so rewarding.

CHAPTER 1

Getting Started

Hello there! Welcome to your first step toward becoming a Python developer. Exciting isn't it? Whether you're just beginning to learn how to program, or have experience in other languages, the lessons taught in this book will help to accelerate your goals. As a Python instructor, I can guarantee you that it's not about where you start, it's about how hard you're willing to work.

At the time of writing this book, my daily job is a coding bootcamp instructor where I teach students how to go from zero programming experience to professional developers in just ten weeks. This book was designed with the intent to bring a bootcamp-based approach to text. This book aims to help you learn subjects that are valuable to becoming a professional developer with Python.

Each subsequent chapter will have an overview and a brief description of what we'll cover that week. This week we'll be covering all the necessary basics to get us jump started. Following the age old saying, "*You must learn to walk before you can run,*" we must understand what our tools are and how to use them before we can begin coding.

Overview

- Understanding why and how this book works
- Installing Python and Anaconda
- Understanding how to use these new tools
- Understanding how to use the terminal
- Writing your first Python program

Without further ado, let's get started, shall we?

Monday: Introduction

Almost every programmer remembers that “Aha!” moment, when everything clicked for them. For me that was when I picked up Python. After years of computer science education, one of the best methods I found to learn was by building applications and applying the knowledge you learn. That’s why this book will have you coding along rather than reading about the theory behind programming. Python makes it simple to pick up concepts otherwise difficult in other languages. This makes it a great language for breaking into the development industry!

You may have already noticed that the structure of this book is different than most. Instead of chapters, we have each topic separated by weeks or days. Notice the current header for the section. This is part of the bootcamp-based approach, so that you may set goals for each day. There will be two ways to follow along this book:

1. Over the course of ten weeks
2. Over the course of ten days

If you’d like to follow the 10-week approach, then think of each chapter as a weekly goal. All chapters are broken up further into daily segments Monday to Friday. The first four days, Monday through Thursday, will introduce new concepts to understand. Friday, or better known as Project Day, is where we will create a program together based on the lessons learned throughout the week. The focus is that you set aside 30–60 minutes each day to complete each daily task.

If you’re eager enough to try the bootcamp style, where you learn all the material in ten days, then think of each chapter as a single day. Granted, you must know that in order to complete this book in ten days, you will need to dedicate around 8 hours per day, which is a typical day for coding bootcamp students. In bootcamps (*like the one I taught*), we go over several concepts daily, and each subsequent day we reiterate the topics learned from previous lessons. This helps to accelerate the process of learning each concept.

What Is Python?

Python is an **interpreted, high-level, general-purpose** programming language. To understand what each of these descriptions mean, let’s make a few comparisons:

- **Low Level vs. High Level:** Refers to whether we program using instructions and data objects at the level of the machine or whether we program using more abstract operations that have been provided by the language designer. Low-level languages (like C, C++) require you to allocate and manage memory, whereas Python manages memory for us.
- **General Purpose vs. Targeted:** Refers to whether the operations of the programming language are widely applicable or are fine-tuned to a domain. For example, SQL is a targeted language that is designed to facilitate extracting information from relational databases, but you wouldn't want to use it to build an operating system.
- **Interpreted vs. Compiled:** Refers to whether the sequence of instructions written by the programmer, called "*source code*," is executed directly (*by an interpreter*) or whether it is first converted (*by a compiler*) into a sequence of machine-level primitive operations. Most applications designed with Python are run through the interpreter, so errors are found at runtime.

Python also emphasizes code readability and uses whitespace to separate snippets of code. We'll learn more about how whitespace in Python works as we get into our lessons, but for now just know that Python is a great first language to break into the computer science industry.

Why Python?

I could go on about why Python is so amazing, but a simple Google search would do that for me. Python is one of the easier languages to learn. Notice I said "*easier*" and not "*easy*"... that's because programming is still difficult, but Python reads closer to the English language than most other languages. This is one of the benefits of learning Python, because concepts that you learn from this book are still applicable to other languages. Python is also one of the most sought-after skills in the technology industry today, used by companies such as Google, Facebook, IBM, etc. It's been used to build applications like Instagram, Pinterest, Dropbox, and much more!

It's also one of the fastest growing languages in 2019, climbing to the top 3 languages to learn for the future.¹ How well does it pay though? According to Indeed.com, the average salary in 2018 was around **\$117,000 USD!**² That's a lot of monopoly money!

One of the biggest reasons for learning Python, though, must be the use of the language itself. It's used in several different industries: front-end development, back-end development, full-stack, testing, data analytics, data science, web design, etc., which makes it a useful language.

Why This Book?

Let's start with the main reason for wanting to read this book. The material taught throughout this book has a proven track record. I've personally used this exact organization approach to help get my students well-paying positions across a variety of industries. The structure of this curriculum has been repeatedly improved over the years to stick with current industry trends.

One of the next great strengths of this book vs. its competitors is how the concepts are taught. I won't bore you with details; instead we'll build small- and large-scale applications together throughout the course of this book. The best way to learn is often by doing! Especially when it comes to programming, one of the lessons I often tell students is to just try writing the code, and if it breaks, fix it. You won't be able to learn if you don't try to break things!

Lastly, this book will not only teach you how to program but how to think like a programmer. At the beginning of each week, I'll challenge you, and by the end of the lesson, you'll be able to understand the approach you need to take. You can always tell the difference between those who are only able to program and those that are proven developers.

Who This Book Is For?

It's always good to understand what you're getting into before you start reading the book. To want to read a book, you first must realize if the book itself is designed for you. If you can answer yes to any of the following questions, then this book is for you:

¹www.tiobe.com/tiobe-index/

²www.indeed.com/salaries/Python-Developer-Salaries

- Do you have experience in other programming languages but want to pick up a high-level language?
- Have you never programmed before but are eager to learn?
- Did you take computer science courses previously, but they just didn't help you learn how to create applications?
- Do you want to make a career change?
- Have you tried to learn languages previously but couldn't because of the difficulty of the language?
- Have you programmed in Python before but want to improve your abilities and learn new tools?

This book is designed for a wide array of readers, no matter your background. The real question is on you, “**How hard are you willing to work?**” The concepts taught in this book can benefit anyone willing to learn. Even if you've programmed in Python before, this book can still help you become a stronger developer.

What You'll Learn

This book was created to be used for bootcamp classes designed in teaching Python. You can expect to cover necessary information that would be required of you on the job as a Python developer. These concepts will give you the ability to go forward with your education in programming. At the end of each chapter, we'll use the concepts covered to create a variety of real-world applications. After all, we're not just focused on Python here, we're trying to build you up to become a better developer.

Tomorrow, we'll find out how to install the necessary software that this book uses. If you already have Anaconda and Python on your machine, you can skip to Wednesday's lesson.

Tuesday: Setting Up Anaconda and Python

Today, we're going to get our software setup. Throughout this book we'll be using a software platform called **Anaconda**, an **integrated development environment (IDE)** called **Jupyter Notebook**, and the language of Python itself. This book will strictly cover Python 3; however, at times you may see me mention subtle differences between versions 2 and 3. Let's go ahead and download and install these first, then I'll get into what each of them are.

Cross-Platform Development

Python runs on all major operating systems, making it a cross-platform language. This means that you can write code on one operating system and work with someone that uses a completely different machine than you. If both machines have Python installed, they should both be able to run the program.

Installing Anaconda and Python for Windows

Most OS X and Linux operating systems already come with Python installed; however, you still need to download Anaconda. For Windows users, Python usually isn't included, but it gets installed with Anaconda. Use the following steps to install Anaconda properly:

1. Open your browser and type www.anaconda.com/distribution/.
2. Click the download button in the header (see Figure 1-1).



Figure 1-1. Anaconda Download Page

3. Once you are on the next page, make sure you select the proper operating system on the header at the top. Click that button (see Figure 1-2).