



PYTHON ROCKET SCIENCE

karan singh garewal

Python Rocket Science
Karan Singh Garewal
Copyright © 2018-2022, Karan Singh Garewal
version 1.8

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 or not 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.

Changelog

date	version	
28/02/2022	v. 1.8	Improvements to improve reinforcement learning. Added chapters: (1) Object Persistence With Pickle (2) Concurrent Python Programs (3) Asynchronous Python (4) Remote Procedure Calls With Python (5) Operating System Shell Command Execution
05/05/2020	v. 1.7	Complete rewrite of all chapters
09/25/2019	v. 1.6	Various Fixes
06/13/2019	v.1.5	Fix some code
12/20/2018:	v.1.0	Initial version 1.0

Cover Image Design by Note Thanun, unsplash.com

Table of Contents

Changelog.....	2
Preface.....	10
Part I: Essential Python.....	13
Introduction.....	14
Lets Get Started.....	14
System Shell.....	15
Python Installation On Linux.....	15
The Python Shell.....	16
Comments.....	17
Bytecode Execution.....	17
Getting The Most Out Of Python Rocket Science.....	18
Python Strings.....	19
Introduction.....	19
Variables And Types.....	19
String Operations.....	21
String Index Operations.....	21
Splicing Strings.....	22
String Addition.....	23
Immutability Of Data Types.....	23
Searching Strings.....	23
Global Find And Replace.....	24
Split A String On A Delimiter.....	24
Joining Sequences Of Strings.....	24
Remove Whitespace In A String.....	25
Multi-line Strings.....	25
Various String Functions.....	25
Quotation Marks In Strings.....	27
Combining Integers And Strings.....	28
Convert A String To A Number.....	28
Converting A String Number To An Integer.....	28
String Interpolation.....	29
Convert A Variable To A String.....	29
Raw Strings And Escape Sequences.....	29
Python Lists.....	31
Introduction.....	31
List Index Operations.....	31
Length Of A List.....	32
Sublists.....	32
Adding Lists.....	33
List Multiplication.....	33
Basic List Operations.....	33
Append To A List.....	33
Pop From A Certain Index Position In A List.....	34

Insert An Element Into A List.....	34
Find An Element In A List.....	35
Sorting Lists.....	35
Deleting List Values.....	35
Removing List Items.....	36
Bounds Checking.....	36
Matrices.....	36
Various List Operations.....	37
Iterating Through A List.....	37
List Comprehensions.....	38
Enumerate.....	39
Create A Deep Copy Of A List.....	39
Python Tuples.....	41
Accessing The Elements In A Tuple.....	41
Tuple Assignment.....	41
Tuple Concatenation And Multiplication.....	42
Tuple Functions.....	42
Python Dictionaries.....	43
Introduction.....	43
Retrieving Dictionary Values.....	43
Adding Key Values To A Dictionary.....	44
Listing Keys And Values.....	44
Sorting A Dictionary.....	45
Pop Method.....	45
Deleting Dictionary Keys.....	46
Merging Two Dictionaries.....	46
Testing Whether A Key Exists In A Dictionary.....	46
Iterating Over A Dictionary.....	47
Making An Independent Copy Of A Dictionary.....	47
Python Files.....	48
Introduction.....	48
Opening And Writing To A File.....	48
Reading A File.....	49
Testing For File Existence.....	49
File Operation Functions.....	50
File Seek.....	51
Length Of A File.....	51
Closing Files.....	52
Python File Modes.....	52
Python Sets.....	54
Introduction.....	54
Creating A Set From A String.....	54
Creating A Set From Literals.....	54
Creating A Set From A List.....	55
Set Operations.....	55
Set Intersection: logical <i>and</i> operation on two sets:.....	55
Set Union: logical <i>or</i> operation on two sets:.....	55
Set Difference: elements in set x which are not in set y:.....	55

Symmetric Difference: elements which belong to one of the two sets only:.....	55
Superset Test.....	56
Adding An Element To A Set.....	56
Removing An Element From A Set.....	56
Set Comprehensions.....	56
Testing For Set Inclusion.....	57
Boolean Types And None.....	58
None.....	58
Boolean Types.....	58
Equality Tests.....	58
bool Function.....	59
Python Numbers.....	61
Introduction.....	61
Floor Division.....	61
Big Numbers.....	61
Bitwise Operators.....	62
Decimal Type.....	62
Fraction Data Type.....	63
Making Numeric Literals Readable.....	63
Python Operators.....	64
Introduction.....	64
Arithmetic Operators.....	64
Relational Operators.....	65
Logical Operators.....	65
Identity Operators.....	66
Python Statements.....	67
Variable Names.....	67
Statement Blocks.....	67
Multiple Statements.....	69
Assignment.....	69
Multiple Assignment.....	69
Tuple Assignments.....	70
List Assignments.....	70
Sequence Assignments.....	70
Extended Sequence Unpacking.....	70
Short Form Assignment.....	71
The Print Function.....	72
JSON.....	73
Introduction.....	73
Storing And Transmitting JSON Objects.....	75
Python Control Flow.....	76
The if Statement.....	76
if Continuations.....	77
Ternary Expressions.....	77
Python Loops.....	78
Basic Loop Constructs.....	78
For Loops.....	79
Counter Loops And Range Functions.....	80

Python Function Basics.....	81
Introduction.....	81
Basic Function Syntax.....	81
Calling A Function.....	82
Function Name Assignment.....	82
Local Variables.....	82
Python Scope.....	84
Introduction.....	84
Lexical Scope.....	84
Name Resolution.....	85
global keyword.....	85
Scope Of Nonlocal Variables.....	86
Python Function Arguments.....	87
Introduction.....	87
Positional Arguments.....	88
Passing Function Arguments By Keyword.....	88
Default Argument Values.....	89
Pass A Variable Number Of Arguments.....	89
Pass A Variable Number Of Keyword Arguments.....	90
Argument Order.....	90
Unpacking Function Call Arguments.....	90
Pass By Reference And Pass By Value.....	91
Advanced Function Topics.....	92
Recursive Functions.....	92
Function Objects.....	92
Function Attributes.....	93
Anonymous Functions: lambda.....	94
Type Annotations.....	95
Introduction.....	95
Type Annotation Syntax.....	95
Variable Annotations.....	95
Function Annotations.....	96
Python Modules.....	97
Introduction.....	97
Standard Library Modules.....	98
Importing Modules.....	98
The Module Search Path.....	99
Viewing The Module Search Paths.....	100
View The File Path For A Module.....	100
Viewing The Attributes Of A Module.....	101
Optimized byte code files.....	101
Shorthand import Statement.....	101
The from Statement.....	101
The import * Statement.....	102
Conditional Imports.....	102
Mutating Attributes In Modules.....	102
Module Namespace.....	103
Reloading Modules.....	103

Hot Swapping Code.....	103
Python Documentation.....	105
Introduction.....	105
Docstrings.....	105
pydoc help Function.....	106
pydoc In Browser Mode.....	106
Packages.....	108
Introduction.....	108
Advanced Python.....	110
Python Classes.....	111
Introduction.....	111
Python Classes.....	111
Instance Methods.....	113
Class Inheritance.....	114
Multiple Inheritance.....	114
Abstract Classes.....	115
Constructors.....	116
Self.....	117
Calling Base Class Constructors.....	117
Class Variables.....	117
Overloading print.....	118
Special Class Attributes.....	119
Name Mangling.....	119
Object Persistence With Pickle.....	120
Introduction.....	120
Pickle.....	120
Shelve.....	121
Iterators And Comprehensions.....	122
Introduction.....	122
Python Iterators.....	122
iter and next.....	122
Dictionaries.....	123
Map Reduce.....	123
Filter.....	124
Reduce.....	124
Operator Overloading.....	126
Introduction.....	126
Advanced Class Topics.....	128
Extending Built-In Types With Subclasses.....	128
The “New Style” Class Model.....	128
Slots.....	128
Static and Class Methods.....	129
Python Exceptions.....	131
Introduction.....	131
Catchall Exception Handler.....	133
Standard Exception Classes.....	133
Raising Exceptions.....	134
Print An Exception.....	135

Termination Actions.....	135
Custom Exceptions.....	136
Context Managers.....	137
with/as Context Managers.....	137
Multiple Context Managers.....	138
Decorators.....	139
Introduction.....	139
Function Decorator Syntax.....	139
Python Loggers And Debuggers.....	141
Assertion Errors.....	141
Loggers.....	141
Using The pdb Debugger.....	142
Check The Memory Footprint Of An Object.....	143
Unit Tests With Pytest.....	144
Introduction.....	144
Running Unit Tests With Pytest.....	144
Writing Pytest Tests.....	145
Floating Point Numbers And Tests For Approximation.....	146
Testing Exceptions.....	147
Setup And Teardown.....	147
Pytest Fixtures.....	148
Parametrized Test Functions.....	149
Mocks.....	150
Operating System Shell Command Execution.....	153
Modern Python.....	154
Concurrent Python Programs.....	155
Introduction.....	155
Thread Implementation in Python.....	155
Race Conditions.....	158
Resource Access With Mutual Exclusion.....	159
Resource Access With Semaphores.....	161
Asynchronous Python.....	163
Introduction.....	163
Remote Procedure Calls With Python.....	166
Introduction.....	166
Making A RPC Server.....	167

Preface

Python Rocket Science has been used in coding bootcamps in the United States, India and the UAE to teach the Python language. Students range from seasoned developers to high school students. Typically they have never had any prior exposure to Python. Having mastered this course, the students then go on to take courses in cutting edge technologies such as machine learning, blockchains and virtual reality.

Time is a very important resource in coding bootcamps since students have to be bootstrapped to high levels of proficiency in a minimum period of time. *Python Rocket Science* is specially designed to meet this objective. It is lucid, economical and effective. It is designed to be absorbed within a period of ten days. This will take you to from zero to the level of an intermediate Python developer.

Python is a remarkably well designed language. When I went to University, some students liked to say: '*I think in C*'. Meaning that the constructs and syntax of the C language fit effortlessly into the brain and the brain likes C. Python is like this, it assimilates into the brain easily. One of the reasons for this is that like C, Python has one way to do one thing. By contrast, the Ruby language has a multitude of ways to do one thing. This just produces clutter and an impedance mismatch in the brain.

A guiding philosophical principle in the design of Python is that clarity is preferred over magic. If you have programmed in Ruby and Rails you will know that a lot of times you are left wondering about how some magical construct works. And you have to dig deeply into the source code to gain an understanding. You won't have this experience in Python. Like C, Python

values clarity of code over magic. In a way Ruby and Rails is like Mandarin, you have to memorize a plethora of constructs. In contrast Python and C do not impose such a heavy cognitive load. Learn Python and then use it like you use the English language. With Python the language will not stand in your way and you are left to concentrate on the logical structure of your software application.

Since Python Rocket Science is a bootcamp course, it can be Kamakazi ingested over a period of about twenty five hours; or a week-end. Thereafter you can use this book as a handy language reference. To maximize learning you must type every example in this book into the Python Shell and play with the example. This is *de rigeur*. If you are starting this book with zero knowledge then the first few pages which talk about interpreters and virtual machines will be the hardest portion of the book. Just suck it in and it will make a lot more sense by the time you complete this book.

On the TIOBE index of the most popular computer languages, Python has ascended into the exalted rank of the five most popular computer languages. A dizzying ascent fueled by the pervasive use of Python in machine learning, blockchains, virtual reality, neural networks and algorithmic trading. All of these are esoteric cutting edge technologies. By learning Python, you are placing yourself in the midst of all of this future tech. Your time spent learning Python will be a very good investment and you will certainly have a lot of fun learning this super language.

This book is divided into three sections: Essential Python, Advanced Python and Modern Python. If you are using this book in conjunction with my book *Practical Blockchains And Cryptocurrencies*, then you will only need to master the *Essential Python* section to absorb the book. *Practical Blockchains And Cryptocurrencies* builds a bitcoin clone in Python. This book thoughly explains the algorithms that underly Bitcoin.

This book is meant to be read sequentially from start to finish. However some of the chapters are marked as optional and you may omit reading these chapters.

OK, lets get down to it. Have fun!