

Exercise Python Solutions

by
Ted Petrou

© 2019 Ted Petrou All Rights Reserved

Contents

I	Environment Setup and Jupyter Notebooks	1
1	Installing Python and Setting up an Environment for Data Science	3
1.1	Conda	3
1.2	Miniconda	3
1.3	Python and Conda installation complete	6
1.4	Creating a new environment just for data analysis	9
1.5	Create a new environment	10
1.6	Other Considerations	13
1.7	Summary of Steps	15
1.8	Using the book with Jupyter Notebooks	15
2	Introduction to Jupyter Notebooks	17
2.1	Jupyter Notebook Basics	17
2.2	Getting Started with Jupyter Notebooks	18
2.3	Executing Cells	19
2.4	Keyboard Shortcuts	21
2.5	Completing exercises in this book	22
2.6	Getting help in the notebook	23
3	Markdown Tutorial	25
3.1	Headers	25
3.2	Italics and bold	25
3.3	Code formatting	25
3.4	Lists	26
3.5	Hyperlinks	27
3.6	Images	27
3.7	New lines	27
4	Jupyter Notebooks More	29
4.1	Where Jupyter Notebooks Excel	29
4.2	Where Jupyter Notebooks Fail	29
4.3	On-demand Jupyter Notebooks	30
5	Jupyter Notebook Extensions	31
5.1	The NBextensions tab	31
5.2	The Skip-Traceback Extension	32
5.3	Skip-Traceback in the Notebook	32
II	Python Fundamentals	35
6	Operators	37

6.1	Arithmetic operators	37
6.2	Arguments	38
6.3	Operator Precedence	39
6.4	Comparison Operators	40
6.5	Unary Plus/Minus Operators	42
6.6	Boolean Operators	43
6.7	Assigning values to variables	45
6.8	Comments	47
6.9	Augmented assignment statements	47
6.10	Other operators	48
6.11	Exercises	48
7	What is Python?	51
7.1	Multiple Implementations	52
7.2	Language Specification	52
7.3	Specification vs Implementation	52
7.4	The original Python implementation	53
7.5	Python Syntax	53
7.6	Tokens	53
7.7	Other syntax rules	55
7.8	An Interactive language	56
7.9	Running Python Interactively	56
7.10	Executing entire programs	57
7.11	Why use Python?	57
8	Objects and Types	59
8.1	Every ‘value’ in Python is an ‘object’	59
8.2	Types in Python	59
8.3	Boolean type	60
8.4	Float type	60
8.5	Complex type	61
8.6	None type	61
8.7	Readability with underscores	62
8.8	Variables in the <code>type</code> function	62
8.9	Object identity	62
8.10	Dynamic Typing	63
8.11	Built-in types	64
8.12	Attributes and methods	64
8.13	Dot Notation	65
8.14	What isn’t an object	66
8.15	Exercises	66
9	Strings	67
9.1	Built-in Types	67
9.2	Strings - what are they?	68
9.3	Operators with strings	71
9.4	String methods	73
9.5	Method Chaining	78
9.6	Find the length of a string	80
9.7	String Interpolation	80
9.8	String formatting	82
9.9	Selecting substrings	88

9.10	Changing the characters of a string	92
9.11	More String Functionality	93
9.12	Test for a substring	94
9.13	Exercises	95
10	Lists	99
10.1	List Creation	99
10.2	Selecting list items	100
10.3	Mutating Lists	101
10.4	The <code>is</code> operator	103
10.5	Variables are references to objects	103
10.6	Finding list methods	106
10.7	More list methods	107
10.8	List Concatenation with the addition operator	109
10.9	Repeated concatenation with the multiplication operator	110
10.10	List equality	110
10.11	Reassigning slices of lists	110
10.12	Lists of lists	111
10.13	Check if an item is in a list	112
10.14	Creating a string from a list	112
10.15	Exercises	113
11	Ranges and Constructors	117
11.1	Ranges	117
11.2	Viewing the sequence defined by <code>range</code>	118
11.3	Constructors	119
11.4	The <code>bool</code> constructor	119
11.5	The <code>int</code> constructor	121
11.6	The <code>float</code> constructor	121
11.7	The <code>complex</code> constructor	122
11.8	The <code>str</code> constructor	122
11.9	More range functionality	123
11.10	Exercises	123
12	Control Flow	127
12.1	Conditional Statements	127
12.2	Python <code>if</code> statements	127
12.3	Looping	131
12.4	For-loops	131
12.5	Example for-loops	134
12.6	While-loops	135
12.7	Example while-loops	136
12.8	More on Loops	138
12.9	Nested Loops	139
12.10	Mini-Project: Craps	140
12.11	List comprehensions	142
12.12	Conditional list comprehensions	143
12.13	Ternary conditional operator	145
12.14	Complex list comprehensions	146
12.15	Implied truth values	148
12.16	Exercises	149
13	Functions	155

13.1	Function calling	155
13.2	Built-in functions	155
13.3	Other built-in functions	160
13.4	User-defined functions	165
13.5	Positional and keyword arguments	168
13.6	Keyword-only arguments	169
13.7	Documenting functions with docstrings	170
13.8	Anonymous functions	171
13.9	The built-in <code>map</code> and <code>filter</code> Functions	174
13.10	Rules of Thumb for Functions	174
13.11	Tic-tac-toe project	175
13.12	The <code>enumerate</code> function	175
13.13	Exercises	176
14	Tuples, Sets, and Dictionaries	181
14.1	Tuples	181
14.2	Selecting items from a tuple	183
14.3	Concatenating tuples	183
14.4	Tuple Methods	184
14.5	Sets	184
14.6	Set Membership Checking	186
14.7	Set Operations	187
14.8	Mutating sets	188
14.9	Case study with sets - The Birthday Paradox	189
14.10	Dictionaries	192
14.11	Selecting values in a dictionary	193
14.12	Get just the keys and values separately	195
14.13	Mutating Dictionaries	195
14.14	Iterating through dictionaries	196
14.15	Dictionary Comprehensions	197
14.16	Exercises	198
15	Imports and Modules	201
15.1	Built-in features	201
15.2	Importing a module	201
15.3	Different kinds of imports	203
15.4	Creating our own Python Modules	204
15.5	How does Python find modules?	205
15.6	Exercises	206
15.7	Project: Creating your own array	207
III	Object-Oriented Programming	213
16	Classes	215
16.1	Review	215
16.2	Creating New Types	216
16.3	Classes are blueprints	217
16.4	Creating a new instance	219
16.5	Mutating Attributes	221
16.6	Docstrings for Classes	223
16.7	Object Composition	226
16.8	The default dictionary	228

16.9 Project - Rolling Dice	230
16.10 Public and Private methods	236
16.11 Instance attributes vs local variables within a method	237