Master the Fundamentals of Python

by
Teddy Petrou

© 2022 Teddy Petrou All Rights Reserved

Contents

1 Operators 9
1.1 Arithmetic operators . . . 9
1.2 Operation Arguments . . . 10
1.3 Operator Precedence . . . 11
1.4 Comparison Operators . . . 12
1.5 Unary Plus/Minus Operators . . . 14
1.6 Boolean Operators . . . 15
1.7 Assigning values to variable names . . . 17
1.8 Comments . . . 19
1.9 Augmented assignment statements . . . 20
1.10 Other operators . . . 20
1.11 Exercises for Module 1 - Operators . . . 20

2 What is Python? 23
2.1 Multiple Implementations . . . 24
2.2 Language Specification . . . 24
2.3 Specification vs Implementation . . . 25
2.4 The original Python implementation . . . 25
2.5 Python Syntax . . . 25
2.6 Tokens . . . 26
2.7 Other syntax rules . . . 27
2.8 An Interactive language . . . 29
2.9 Running Python Interactively . . . 29
2.10 Executing entire programs . . . 30
2.11 Why use Python? . . . 30
2.12 Exercises for Module 2 - What is Python? . . . 31

3 Objects and Types 33
3.1 Every ‘value’ in Python is an ‘object’ . . . 33
3.2 Types in Python . . . 33
3.3 Integer type . . . 34
3.4 Boolean type . . . 35
3.5 Float type . . . 35
3.6 Complex type . . . 36
3.7 None type . . . 36
3.8 Variables in the `type` function . . . 36
3.9 Object identity . . . 37
3.10 Dynamic Typing . . . 38
3.11 Built-in types . . . 38
3.12 Attributes and methods . . . 38
3.13 Dot Notation . . . 39

3.14 What isn't an object . . . 40
3.15 Exercises for Module 3 - Objects and Types . . . 40

4 Strings 43
4.1 Built-in Types . . . 43
4.2 String definition . . . 44
4.3 Operators with strings . . . 47
4.4 String methods . . . 49
4.5 Method Chaining . . . 54
4.6 Find the length of a string . . . 55
4.7 String Interpolation . . . 56
4.8 String formatting . . . 57
4.9 Selecting substrings . . . 64
4.10 Changing the characters of a string . . . 68
4.11 More String Functionality . . . 69
4.12 Test for a substring . . . 69
4.13 Exercises for Module 4 - Strings . . . 70

5 Lists 75
5.1 List creation . . . 75
5.2 Selecting list items . . . 76
5.3 Mutating Lists . . . 78
5.4 The `is` operator . . . 79
5.5 Variable names are references to objects . . . 79
5.6 Discovering list methods . . . 82
5.7 Using operators with lists . . . 86
5.8 Reassigning slices of lists . . . 87
5.9 Lists of lists . . . 89
5.10 Creating a string from a list . . . 90
5.11 Exercises for Module 5 - Lists . . . 90

6 Ranges and Constructors 95
6.1 Ranges . . . 95
6.2 Viewing the sequence defined by `range` . . . 96
6.3 Constructors . . . 97
6.4 The `bool` constructor . . . 97
6.5 The `int` constructor . . . 99
6.6 The `float` constructor . . . 99
6.7 The `complex` constructor . . . 100
6.8 The `str` constructor . . . 100
6.9 More range functionality . . . 100
6.10 Exercises for Module 6 - Ranges and Constructors . . . 101

7 Module 7 - Conditional Statements 103
7.1 Conditional Statements . . . 103
7.2 Python `if` statements . . . 104
7.3 Ternary conditional operator . . . 107
7.4 Implied truth values . . . 109
7.5 Exercises for Module 7 - Conditional Statements . . . 110

8 Module 8 - Writing Entire Programs 113
8.1 Computer programs . . . 113
8.2 Running a Python Program . . . 114

8.3 Source-Code Editors 115
8.4 Downloading and Installing Visual Studio Code 115
8.5 Setting up VS Code 115
8.6 Opening up Python files in VS Code 116
8.7 Creating a New Python File 116
8.8 Creating a trivia game 116
8.9 Building a Choose Your Own Adventure Game 118
8.10 Game details 118
8.11 Implementing the game 119

9 Looping **121**
9.1 For-loops 121
9.2 Example for-loops 123
9.3 While-loops 125
9.4 Example while-loops 126
9.5 More on Loops 127
9.6 Nested Loops 129
9.7 Mini-Project: Craps 130
9.8 Exercises for Module 9 - Looping 132

10 List Comprehensions **135**
10.1 Conditional list comprehensions 137
10.2 Ternary conditional operator 138
10.3 Complex list comprehensions 139
10.4 Implied truth values 141
10.5 Exercises for Module 10 - List Comprehensions 142

11 Built-in Functions **145**
11.1 Function calling 145
11.2 Built-in functions 145
11.3 Other built-in functions 150
11.4 Exercises for Module 11 - Built-in Functions 155

12 User-defined functions **157**
12.1 Positional and keyword arguments 160
12.2 Keyword-only arguments 161
12.3 Positional-only arguments 162
12.4 Documenting functions with docstrings 164
12.5 Functions are objects 165
12.6 Function attributes and methods 167
12.7 Some built-in ‘functions’ are not functions 168
12.8 Anonymous functions 168
12.9 The built-in `map` and `filter` 170
12.10Rules of Thumb for Functions 172
12.11Code Refactoring 172
12.12Exercises for Module 12 - User-Defined Functions 172

13 Tic-Tac-Toe **177**
13.1 The functions 177
13.2 Test your functions 177
13.3 Play a game 180
13.4 Solution 180

14 Tuples, Sets, and Dictionaries **183**
14.1 Tuples . . . 183
14.2 Selecting items from a tuple . . . 186
14.3 Using operators with tuples . . . 186
14.4 Tuple Methods . . . 187
14.5 Why use tuples when lists are more flexible? . . . 187
14.6 Sets . . . 188
14.7 Using the `set` constructor . . . 189
14.8 Set membership checking . . . 190
14.9 Mathematical set operations . . . 191
14.10Set methods . . . 193
14.11The birthday paradox . . . 193
14.12Dictionaries . . . 195
14.13Selecting values in a dictionary . . . 197
14.14Dictionary membership checking . . . 198
14.15Dictionary methods . . . 198
14.16Mutating dictionaries . . . 200
14.17Iterating through dictionaries . . . 201
14.18Tuple, set, and dictionary comprehensions . . . 202
14.19Unpacking iterables . . . 203
14.20The `zip` function . . . 207
14.21Exercises for Module 14 - Tuples, Sets, and Dictionaries . . . 208

15 Python Modules **211**
15.1 Importing a module . . . 211
15.2 Different ways to use the `import` statement . . . 213
15.3 Highlights from the standard library . . . 215
15.4 The `re` module - regular expressions . . . 215
15.5 The `datetime` module . . . 216
15.6 The `calendar` module . . . 218
15.7 The `time` module . . . 219
15.8 The `collections` module . . . 219
15.9 The `copy` module . . . 223
15.10The `bisect` module . . . 224
15.11The `math` module . . . 225
15.12The `fractions` module . . . 226
15.13The `statistics` module . . . 226
15.14The `sys` module . . . 227
15.15Third-Party Libraries . . . 227
15.16Exercises for Module 15 - Python Modules . . . 227

16 User-Defined Python Modules **229**
16.1 Modular Programming . . . 229
16.2 Creating your own Python Modules . . . 229
16.3 How does Python find modules? . . . 230
16.4 “Private” variable names . . . 232
16.5 Importing modules within modules . . . 233
16.6 The `my_array.py` module . . . 233
16.7 Unit testing . . . 234
16.8 Importing edited modules into a Jupyter Notebook . . . 235

17 Errors and Exceptions **237**

17.1 Syntax Errors . . . 237
17.2 Errors vs Exceptions . . . 238
17.3 NameError . . . 239
17.4 “Raised” exceptions . . . 239
17.5 KeyError . . . 239
17.6 IndexError . . . 240
17.7 TypeError . . . 240
17.8 ValueError . . . 240
17.9 AttributeError . . . 241
17.10Other exceptions . . . 241
17.11Purposefully raising exceptions . . . 241
17.12Raising errors on bad input . . . 242
17.13Handling exceptions . . . 243
17.14Catching multiple different errors . . . 245
17.15The `else` and `finally` blocks . . . 245
17.16More on Exceptions . . . 246
17.17Exercises . . . 246