
Fundamentals of Python Programming

DRAFT

Richard L. Halterman
Southern Adventist University

July 9, 2019

Contents

| | | |
|----------|---|-----------|
| 1 | The Context of Software Development | 1 |
| 1.1 | Software | 2 |
| 1.2 | Development Tools | 2 |
| 1.3 | Learning Programming with Python | 4 |
| 1.4 | Writing a Python Program | 5 |
| 1.5 | The Python Interactive Shell | 9 |
| 1.6 | A Longer Python program | 11 |
| 1.7 | Exercises | 12 |
| 2 | Values and Variables | 13 |
| 2.1 | Integer and String Values | 13 |
| 2.2 | Variables and Assignment | 17 |
| 2.3 | Identifiers | 24 |
| 2.4 | Floating-point Numbers | 26 |
| 2.5 | Control Codes within Strings | 30 |
| 2.6 | User Input | 31 |
| 2.7 | Controlling the print Function | 34 |
| 2.8 | String Formatting | 35 |
| 2.9 | Multi-line Strings | 39 |
| 2.10 | Exercises | 40 |
| 3 | Expressions and Arithmetic | 43 |
| 3.1 | Expressions | 43 |
| 3.2 | Mixed Type Expressions | 49 |
| 3.3 | Operator Precedence and Associativity | 49 |
| 3.4 | Formatting Expressions | 51 |

| | | |
|----------|--|------------|
| 3.5 | Comments | 52 |
| 3.6 | Errors | 53 |
| 3.6.1 | Syntax Errors | 53 |
| 3.6.2 | Run-time Exceptions | 54 |
| 3.6.3 | Logic Errors | 56 |
| 3.7 | Arithmetic Examples | 57 |
| 3.8 | More Arithmetic Operators | 59 |
| 3.9 | Algorithms | 61 |
| 3.10 | Exercises | 62 |
| 4 | Conditional Execution | 67 |
| 4.1 | Boolean Expressions | 67 |
| 4.2 | Boolean Expressions | 68 |
| 4.3 | The Simple if Statement | 69 |
| 4.4 | The if/else Statement | 75 |
| 4.5 | Compound Boolean Expressions | 77 |
| 4.6 | The pass Statement | 80 |
| 4.7 | Floating-point Equality | 82 |
| 4.8 | Nested Conditionals | 83 |
| 4.9 | Multi-way Decision Statements | 93 |
| 4.10 | Multi-way Versus Sequential Conditionals | 97 |
| 4.11 | Conditional Expressions | 99 |
| 4.12 | Errors in Conditional Statements | 102 |
| 4.13 | Logic Complexity | 105 |
| 4.14 | Exercises | 107 |
| 5 | Iteration | 113 |
| 5.1 | The while Statement | 113 |
| 5.2 | Definite Loops vs. Indefinite Loops | 121 |
| 5.3 | The for Statement | 122 |
| 5.4 | Nested Loops | 126 |
| 5.5 | Abnormal Loop Termination | 132 |
| 5.5.1 | The break statement | 133 |
| 5.5.2 | The continue Statement | 136 |
| 5.6 | while/else and for/else | 137 |

| | | |
|----------|--|------------|
| 5.7 | Infinite Loops | 139 |
| 5.8 | Iteration Examples | 143 |
| 5.8.1 | Computing Square Root | 143 |
| 5.8.2 | Drawing a Tree | 144 |
| 5.8.3 | Printing Prime Numbers | 146 |
| 5.8.4 | Insisting on the Proper Input | 150 |
| 5.9 | Exercises | 150 |
| 6 | Using Functions | 157 |
| 6.1 | Introduction to Using Functions | 158 |
| 6.2 | Functions and Modules | 162 |
| 6.3 | The Built-in Functions | 164 |
| 6.4 | Standard Mathematical Functions | 167 |
| 6.5 | time Functions | 170 |
| 6.6 | Random Numbers | 173 |
| 6.7 | System-specific Functions | 176 |
| 6.8 | The eval and exec Functions | 176 |
| 6.9 | Turtle Graphics | 179 |
| 6.10 | Other Techniques for Importing Functions and Modules | 185 |
| 6.11 | Exercises | 191 |
| 7 | Writing Functions | 193 |
| 7.1 | Function Basics | 194 |
| 7.2 | Parameter Passing | 209 |
| 7.3 | Documenting Functions | 211 |
| 7.4 | Function Examples | 213 |
| 7.4.1 | Better Organized Prime Generator | 213 |
| 7.4.2 | Command Interpreter | 215 |
| 7.4.3 | Restricted Input | 216 |
| 7.4.4 | Better Die Rolling Simulator | 218 |
| 7.4.5 | Tree Drawing Function | 219 |
| 7.4.6 | Floating-point Equality | 220 |
| 7.5 | Refactoring to Eliminate Code Duplication | 222 |
| 7.6 | Custom Functions vs. Standard Functions | 224 |
| 7.7 | Exercises | 227 |

| | | |
|-----------|--|------------|
| 8 | More on Functions | 233 |
| 8.1 | Global Variables | 233 |
| 8.2 | Default Parameters | 238 |
| 8.3 | Introduction to Recursion | 241 |
| 8.4 | Making Functions Reusable | 246 |
| 8.5 | Functions as Data | 249 |
| 8.6 | Separating Concerns with Pluggable Modules | 253 |
| 8.7 | Lambda Expressions | 273 |
| 8.8 | Generators | 278 |
| 8.9 | Local Function Definitions | 286 |
| 8.10 | Decorators | 292 |
| 8.11 | Partial Application | 303 |
| 8.12 | Exercises | 306 |
| 9 | Objects | 311 |
| 9.1 | Using Objects | 311 |
| 9.2 | String Objects | 312 |
| 9.3 | File Objects | 316 |
| 9.4 | Fraction Objects | 323 |
| 9.5 | Turtle Graphics Objects | 325 |
| 9.6 | Graphics with tkinter Objects | 326 |
| 9.7 | Other Standard Python Objects | 332 |
| 9.8 | Object Mutability and Aliasing | 332 |
| 9.9 | Garbage Collection | 336 |
| 9.10 | Exercises | 337 |
| 10 | Lists | 339 |
| 10.1 | Using Lists | 341 |
| 10.2 | List Traversal | 345 |
| 10.3 | Building Lists | 346 |
| 10.4 | List Membership | 351 |
| 10.5 | List Assignment and Equivalence | 352 |
| 10.6 | List Bounds | 357 |
| 10.7 | Slicing | 358 |
| 10.8 | List Element Removal | 361 |

| | | |
|-----------|---|------------|
| 10.9 | Lists and Functions | 362 |
| 10.10 | List Methods | 363 |
| 10.11 | Prime Generation with a List | 366 |
| 10.12 | Command-line Arguments | 368 |
| 10.13 | List Comprehensions | 369 |
| 10.14 | Multidimensional Lists | 375 |
| 10.15 | Summary of List Creation Techniques | 384 |
| 10.16 | Lists vs. Generators | 385 |
| 10.17 | Exercises | 385 |
| 11 | Tuples, Dictionaries, and Sets | 389 |
| 11.1 | Tuples | 389 |
| 11.2 | Arbitrary Argument Lists | 393 |
| 11.3 | Dictionaries | 398 |
| 11.4 | Using Dictionaries | 402 |
| 11.5 | Counting with Dictionaries | 404 |
| 11.6 | Grouping with Dictionaries | 408 |
| 11.7 | Keyword Arguments | 411 |
| 11.8 | Sets | 414 |
| 11.9 | Set Quantification with <code>all</code> and <code>any</code> | 415 |
| 11.10 | Enumerating the Elements of a Data Structure | 419 |
| 11.11 | Exercises | 421 |
| 12 | Handling Exceptions | 425 |
| 12.1 | Motivation | 425 |
| 12.2 | Common Standard Exceptions | 426 |
| 12.3 | Handling Exceptions | 428 |
| 12.4 | Handling Multiple Exceptions | 431 |
| 12.5 | The Catch-all Handler | 432 |
| 12.6 | Catching Exception Objects | 436 |
| 12.7 | Exception Handling Scope | 437 |
| 12.8 | Raising Exceptions | 446 |
| 12.9 | The <code>try</code> Statement's Optional <code>else</code> Block | 451 |
| 12.10 | <code>finally</code> block | 453 |
| 12.11 | Using Exceptions Wisely | 457 |

| | |
|--|------------|
| 12.12 Exercises | 458 |
| 13 Custom Types | 463 |
| 13.1 Circle Objects | 463 |
| 13.2 Restricting Access to Members | 473 |
| 13.3 Rational Numbers | 474 |
| 13.4 Bank Account Objects | 479 |
| 13.5 Timing Objects | 483 |
| 13.6 Traffic Light Objects | 486 |
| 13.7 Automated Testing | 492 |
| 13.8 Plotting Data | 497 |
| 13.9 Dynamic Content | 502 |
| 13.10 Class Variables | 506 |
| 13.11 Exercises | 507 |
| 14 Class Design: Composition and Inheritance | 513 |
| 14.1 Composition | 513 |
| 14.2 Class Inheritance | 518 |
| 14.3 Composition vs. Inheritance | 537 |
| 14.4 Multiple Inheritance | 540 |
| 14.5 Unit Testing | 558 |
| 14.6 Custom Exceptions | 563 |
| 14.7 Exercises | 565 |
| 15 Algorithm Quality | 567 |
| 15.1 Good Algorithms Versus Bad Algorithms | 567 |
| 15.2 Sorting | 576 |
| 15.3 Flexible Sorting | 579 |
| 15.4 Search | 582 |
| 15.4.1 Linear Search | 582 |
| 15.4.2 Binary Search | 584 |
| 15.5 Recursion Revisited | 593 |
| 15.6 List Permutations | 600 |
| 15.7 Randomly Permuting a List | 609 |
| 15.8 Reversing a List | 614 |

| | |
|--|------------|
| 15.9 Memoization | 615 |
| 15.10 Exercises | 627 |
| 16 Representing Relationships with Graphs | 631 |
| 16.1 Introduction to Graphs | 631 |
| 16.2 Implementing Graphs in Python | 635 |
| 16.3 Path Finding | 635 |
| 16.4 Breadth-first Search | 637 |
| 16.5 Depth-first Search | 647 |
| 16.6 Exercises | 654 |
| Index | 655 |