

- ☒ 1. General Introduction
 - ☒ 1.1. The Way of the Program
 - ☒ 1.2. Algorithms
 - ☒ 1.3. The Python Programming Language
 - ☒ 1.4. Executing Python in this Book
 - ☒ 1.5. More About Programs
 - ☒ 1.6. What is Debugging?
 - ☒ 1.7. Syntax errors
 - ☒ 1.8. Runtime Errors
 - ☒ 1.9. Semantic Errors
 - ☒ 1.10. Experimental Debugging
 - ☒ 1.11. Formal and Natural Languages
 - ☒ 1.12. A Typical First Program
 - ☒ 1.13. Comments
 - ☒ 1.14. Glossary
 - ☒ 1.15. Exercises
- ☒ 2. Simple Python Data
 - ☒ 2.1. Variables, Expressions and Statements
 - ☒ 2.2. Values and Data Types
 - ☒ 2.3. Type conversion functions
 - ☒ 2.4. Variables
 - ☒ 2.5. Variable Names and Keywords
 - ☒ 2.6. Statements and Expressions
 - ☒ 2.7. Operators and Operands
 - ☒ 2.8. Input
 - ☒ 2.9. Order of Operations
 - ☒ 2.10. Reassignment
 - ☒ 2.10.1. Developing your mental model of How Python Evaluates
 - ☒ 2.11. Updating Variables
 - ☒ 2.12. Glossary
 - ☒ 2.13. Exercises
- ☒ 3. Debugging Interlude 1
 - ☒ 3.1. How to be a Successful Programmer
 - ☒ 3.2. How to Avoid Debugging
 - ☒ 3.3. Beginning tips for Debugging
 - ☒ 3.4. Know Your Error Messages
 - ☒ 3.4.1. `ParseError`
 - ☒ 3.4.2. `TypeError`
 - ☒ 3.4.3. `NameError`
 - ☒ 3.4.4. `ValueError`

- ☒ 3.5. Summary
- ☒ 3.6. Exercises
- ☒ 4. Python Turtle Graphics
 - ☒ 4.1. Hello Little Turtles!
 - ☒ 4.2. Our First Turtle Program
 - ☒ 4.3. Instances — A Herd of Turtles
 - ☒ 4.4. The for Loop
 - ☒ 4.5. Flow of Execution of the for Loop
 - ☒ 4.6. Iteration Simplifies our Turtle Program
 - ☒ 4.7. The range Function
 - ☒ 4.8. A Few More turtle Methods and Observations
 - ☒ 4.9. Summary of Turtle Methods
 - ☒ 4.10. Glossary
 - ☒ 4.11. Exercises
- ☒ 5. Python Modules
 - ☒ 5.1. Modules and Getting Help
 - ☒ 5.2. More About Using Modules
 - ☒ 5.3. The math module
 - ☒ 5.4. The random module
 - ☒ 5.5. Creating Modules
 - ☒ 5.6. Glossary
 - ☒ 5.7. Exercises
- ☒ 6. Functions
 - ☒ 6.1. Functions
 - ☒ 6.2. Functions that Return Values
 - ☒ 6.3. Unit Testing
 - ☒ 6.3.1. assert with for loops
 - ☒ 6.3.2. Return Value Tests
 - ☒ 6.4. Variables and Parameters are Local
 - ☒ 6.5. The Accumulator Pattern
 - ☒ 6.5.1. The General Accumulator Pattern
 - ☒ 6.5.2. A Variation on the Accumulator Pattern
 - ☒ 6.6. Functions can Call Other Functions
 - ☒ 6.7. Flow of Execution Summary
 - ☒ 6.8. Using a Main Function
 - ☒ 6.9. Program Development
 - ☒ 6.10. Composition
 - ☒ 6.11. A Turtle Bar Chart
 - ☒ 6.12. Glossary
 - ☒ 6.13. Exercises

- ☒ 7. Selection
 - ☒ 7.1. Boolean Values and Boolean Expressions
 - ☒ 7.2. Logical operators
 - ☒ 7.2.1. Logical Opposites
 - ☒ 7.3. Precedence of Operators
 - ☒ 7.4. Conditional Execution: Binary Selection
 - ☒ 7.5. Omitting the else Clause: Unary Selection
 - ☒ 7.6. Nested conditionals
 - ☒ 7.7. Chained conditionals
 - ☒ 7.8. Boolean Functions
 - ☒ 7.8.1. More Unit Testing
 - ☒ 7.9. Glossary
 - ☒ 7.10. Exercises
- ☒ 8. More About Iteration
 - ☒ 8.1. Iteration Revisited
 - ☒ 8.2. The for loop revisited
 - ☒ 8.3. The while Statement
 - ☒ 8.4. Randomly Walking Turtles
 - ☒ 8.5. The $3n + 1$ Sequence
 - ☒ 8.6. Newton's Method
 - ☒ 8.7. The Accumulator Pattern Revisited
 - ☒ 8.8. Other uses of while
 - ☒ 8.8.1. Sentinel Values
 - ☒ 8.8.2. Validating Input
 - ☒ 8.9. Algorithms Revisited
 - ☒ 8.10. Simple Tables
 - ☒ 8.11. 2-Dimensional Iteration: Image Processing
 - ☒ 8.11.1. The RGB Color Model
 - ☒ 8.11.2. Image Objects
 - ☒ 8.11.3. Image Processing and Nested Iteration
 - ☒ 8.12. Image Processing on Your Own
 - ☒ 8.13. Glossary
 - ☒ 8.14. Exercises
- ☒ 9. Strings
 - ☒ 9.1. Strings Revisited
 - ☒ 9.2. A Collection Data Type
 - ☒ 9.3. Operations on Strings
 - ☒ 9.4. Index Operator: Working with the Characters of a String
 - ☒ 9.5. String Methods
 - ☒ 9.5.1. String Format Method

- ☒ 9.6. Length
- ☒ 9.7. The Slice Operator
- ☒ 9.8. String Comparison
- ☒ 9.9. Strings are Immutable
- ☒ 9.10. Traversal and the for Loop: By Item
- ☒ 9.11. Traversal and the for Loop: By Index
- ☒ 9.12. Traversal and the while Loop
- ☒ 9.13. The in and not in operators
- ☒ 9.14. The Accumulator Pattern with Strings
- ☒ 9.15. Turtles and Strings and L Systems
- ☒ 9.16. Looping and Counting
- ☒ 9.17. A find function
- ☒ 9.18. Optional parameters
- ☒ 9.19. Character classification
- ☒ 9.20. Summary
- ☒ 9.21. Glossary
- ☒ 9.22. Exercises
- ☒ 10. Lists
 - ☒ 10.1. Lists
 - ☒ 10.2. List Values
 - ☒ 10.3. List Length
 - ☒ 10.4. Accessing Elements
 - ☒ 10.5. List Membership
 - ☒ 10.6. Concatenation and Repetition
 - ☒ 10.7. List Slices
 - ☒ 10.8. Lists are Mutable
 - ☒ 10.9. List Deletion
 - ☒ 10.10. Objects and References
 - ☒ 10.11. Aliasing
 - ☒ 10.12. Cloning Lists
 - ☒ 10.13. Repetition and References
 - ☒ 10.14. List Methods
 - ☒ 10.15. The Return of L Systems
 - ☒ 10.16. Append versus Concatenate
 - ☒ 10.17. Lists and for loops
 - ☒ 10.18. The Accumulator Pattern with Lists
 - ☒ 10.18.1. Accumulating the Max Value
 - ☒ 10.18.2. Accumulating a String Result
 - ☒ 10.19. Using Lists as Parameters
 - ☒ 10.20. Pure Functions

- ☒ 10.21. Which is Better?
- ☒ 10.22. Functions that Produce Lists
- ☒ 10.23. List Comprehensions
- ☒ 10.24. Nested Lists
- ☒ 10.25. Strings and Lists
- ☒ 10.26. list Type Conversion Function
- ☒ 10.27. Tuples and Mutability
- ☒ 10.28. Tuple Assignment
- ☒ 10.29. Tuples as Return Values
- ☒ 10.30. Glossary
- ☒ 10.31. Exercises
- ☐ 11. **Files**
 - ☒ 11.1. Working with Data Files
 - ☒ 11.2. Finding a File on your Disk
 - ☒ 11.3. Reading a File
 - ☒ 11.4. Iterating over lines in a file
 - ☒ 11.5. Alternative File Reading Methods
 - ☒ 11.6. Writing Text Files
 - ☒ 11.7. With Statements
 - ☒ 11.8. Fetching Something From The Web
 - ☒ 11.9. Glossary
 - ☒ 11.10. Exercises
- ☒ 12. Dictionaries
 - ☒ 12.1. Dictionaries
 - ☒ 12.2. Dictionary Operations
 - ☒ 12.3. Dictionary Methods
 - ☒ 12.4. Aliasing and Copying
 - ☒ 12.5. Sparse Matrices
 - ☒ 12.6. Glossary
 - ☒ 12.7. Exercises
- ☒ 13. Exceptions
 - ☒ 13.1. What is an exception?
 - ☒ 13.2. Exception Handling Flow of control
 - ☒ 13.2.1. Raising and Catching Errors with try and except
 - ☒ 13.3. Runtime Stack and raise command
 - ☒ 13.4. Summary
 - ☒ 13.5. Standard Exceptions
 - ☒ 13.6. Principles for using Exceptions
 - ☒ 13.7. Exceptions Syntax
 - ☒ 13.7.1. Catch All Exceptions

- ☒ 13.7.2. Catch A Specific Exception
- ☒ 13.7.3. Catch Multiple Specific Exceptions
- ☒ 13.7.4. Clean-up After Exceptions
- ☒ 13.7.5. An Example of File I/O
- ☒ 13.8. The finally clause of the try statement
- ☒ 13.9. Glossary
- ☒ 13.10. Exercises
- ☐ 14. Web Applications
 - ☒ 14.1. Web Applications
 - ☒ 14.2. How the Web Works
 - ☒ 14.3. How Web Applications Work
 - ☒ 14.4. Web Applications and HTML Forms
 - ☒ 14.5. Writing Web Applications With Flask
 - ☒ 14.6. More About Flask
 - ☒ 14.7. Input For A Flask Web Application
 - ☒ 14.8. Web Applications With a User Interface
 - ☒ 14.9. Glossary
- ☐ 15. GUI and Event Driven Programming
 - ☐ 15.1. Graphical User Interfaces
 - ☐ 15.2. GUI Programming
 - ☐ 15.3. GUI Programming Options
 - ☐ 15.4. Tkinter
 - ☐ 15.5. Tkinter Pre-programmed Interfaces
 - ☐ 15.6. Tkinter Custom Interfaces
 - ☐ 15.7. Hello World
 - ☐ 15.8. Tkinter Standard Dialog Boxes
 - ☐ 15.8.1. Messages
 - ☐ 15.8.2. Yes/No Questions
 - ☐ 15.8.3. Single Value Data Entry
 - ☐ 15.8.4. File Chooser
 - ☐ 15.8.5. Color Chooser
 - ☐ 15.9. GUI Widgets
 - ☐ 15.10. Creating Widgets
 - ☐ 15.11. Layout Mangers
 - ☐ 15.12. Specifying Dimensions
 - ☐ 15.13. Place Layout Manager
 - ☐ 15.14. Grid Layout Manager
 - ☐ 15.15. Pack Layout Manager
 - ☐ 15.15.1. Summary
 - ☐ 15.16. Widget Groupings

- ☐ 15.17. Command Events
- ☐ 15.18. Hello World Again
- ☐ 15.19. Other Events
- ☐ 15.20. Low-Level Event Processing
- ☐ 15.21. Focus
- ☐ 15.22. Event Binding
- ☐ 15.23. Event Descriptors
- ☐ 15.24. Event Objects
- ☐ 15.25. Event Processing
- ☐ 15.26. The Design of GUI Programs
- ☐ 15.27. Common Widget Properties
- ☐ 15.28. Specific Widget Properties
- ☐ 15.29. Widget Attributes
- ☐ 15.30. Timer Events
- ☐ 15.30.1. Animations and Repeated Tasks
- ☐ 15.30.2. Canceling Timer Events
- ☐ 15.30.3. Multiple Parameters to Timer Callbacks
- ☐ 15.31. A Programming Example
- ☐ 15.31.1. A Whack-a-mole Game
- ☐ 15.31.2. Summary
- ☐ 15.32. Managing GUI Program Complexity
- ☐ 15.32.1. Creating the View
- ☐ 15.32.2. Creating the Model
- ☐ 15.32.3. Creating the Controller
- ☐ 15.33. Exercises
- ☐ 15.34. Glossary
- ☐ 16. Recursion
 - ☐ 16.1. What Is Recursion?
 - ☐ 16.2. Calculating the Sum of a List of Numbers
 - ☐ 16.3. The Three Laws of Recursion
 - ☐ 16.4. Converting an Integer to a String in Any Base
 - ☐ 16.5. Visualizing Recursion
 - ☐ 16.6. Sierpinski Triangle
 - ☐ 16.7. Glossary
 - ☐ 16.8. Programming Exercises
 - ☐ 16.9. Exercises
- ☐ 17. Classes and Objects - the Basics
 - ☐ 17.1. Object-oriented programming
 - ☐ 17.2. A change of perspective

- ☐ 17.3. Objects Revisited
- ☐ 17.4. User Defined Classes
- ☐ 17.5. Improving our Constructor
- ☐ 17.6. Adding Other Methods to our Class
- ☐ 17.7. Objects as Arguments and Parameters
- ☐ 17.8. Converting an Object to a String
- ☐ 17.9. Instances as Return Values
- ☐ 17.10. Glossary
- ☐ 17.11. Exercises
- ☐ 18. Classes and Objects - Digging a Little Deeper
 - ☐ 18.1. Fractions
 - ☐ 18.2. Objects are Mutable
 - ☐ 18.3. Sameness
 - ☐ 18.4. Arithmetic Methods
 - ☐ 18.5. Glossary
 - ☐ 18.6. Exercises
- ☐ 19. Inheritance
 - ☐ 19.1. Pillars of OOP
 - ☐ 19.2. Introduction to Inheritance
 - ☐ 19.3. Extending
 - ☐ 19.4. Reuse Through Composition
 - ☐ 19.5. Class Diagrams
 - ☐ 19.6. Composition vs. Inheritance
 - ☐ 19.7. Case Study: Structured Postal Addresses
 - ☐ 19.7.1. Storing Postal Addresses
 - ☐ 19.7.2. Storing International Addresses
 - ☐ 19.7.3. Inheritance Applied
 - ☐ 19.7.4. A List of Addresses
 - ☐ 19.7.5. Using isinstance
- ☐ 20. Unit Testing
 - ☐ 20.1. Introduction: Unit Testing
 - ☐ 20.2. Checking Assumptions With assert
 - ☐ 20.2.1. Designing Defensive Functions
 - ☐ 20.2.2. The assert Statement
 - ☐ 20.2.3. More on assert and Preconditions
 - ☐ 20.3. Testing Functions
 - ☐ 20.3.1. Automated Unit Tests
 - ☐ 20.3.2. Automated Unit Tests with assert
 - ☐ 20.3.3. Unit Tests can have bugs

- ☐ 20.4. Designing Testable Functions
 - ☐ 20.4.1. Design by Contract
- ☐ 20.5. Writing Unit Tests
 - ☐ 20.5.1. Specification-Based Testing
- ☐ 20.6. Test-First Development
 - ☐ 20.6.1. Benefits of Test-First Development
- ☐ 20.7. Testing with pytest
 - ☐ 20.7.1. Organizing pytest Functions
 - ☐ 20.7.2. Using pytest
 - ☐ 20.7.3. Understanding pytest Failure Reports
 - ☐ 20.7.4. Integrated Unit Testing with pytest
- ☐ 20.8. Glossary
- ☐ 20.9. Exercises
- ☐ 21. Hidden Items