

4. Writing Structured Programs

4.1 Back to the Basics

- 1.1 Assignment
- 1.2 Equality
- 1.3 Conditionals

4.2 Sequences

- 2.1 Operating on Sequence Types
- 2.2 Combining Different Sequence Types
- 2.3 Generator Expressions

4.3 Questions of Style

- 3.1 Python Coding Style
- 3.2 Procedural vs Declarative Style
- 3.3 Some Legitimate Uses for Counters

4.4 Functions: The Foundation of Structured Programming

- 4.1 Function Inputs and Outputs
- 4.2 Parameter Passing
- 4.3 Variable Scope
- 4.4 Checking Parameter Types
- 4.5 Functional Decomposition
- 4.6 Documenting Functions

4.5 Doing More with Functions

- 5.1 Functions as Arguments
- 5.2 Accumulative Functions
- 5.3 Higher-Order Functions
- 5.4 Named Arguments

4.6 Program Development

- 6.1 Structure of a Python Module
- 6.2 Multi-Module Programs
- 6.3 Sources of Error
- 6.4 Debugging Techniques
- 6.5 Defensive Programming

4.7 Algorithm Design

- 7.1 Recursion
- 7.2 Space-Time Tradeoffs
- 7.3 Dynamic Programming

4.8 A Sample of Python Libraries

- 8.1 Matplotlib
- 8.2 NetworkX
- 8.3 csv
- 8.4 NumPy
- 8.5 Other Python Libraries

4.9 Summary

4.10 Further Reading

4.11 Exercises