



Python Distilled



David M. Beazley
Author of *Python Essential Reference*

Python Distilled

Python Distilled

Table of Contents

Cover

Half Title

Title Page

Copyright Page

Contents

Preface

1 Python Basics

1.1 Running Python

1.2 Python Programs

1.3 Primitives, Variables, and Expressions

1.4 Arithmetic Operators

1.5 Conditionals and Control Flow

1.6 Text Strings

1.7 File Input and Output

1.8 Lists

1.9 Tuples

1.10 Sets

1.11 Dictionaries

1.12 Iteration and Looping

1.13 Functions

1.14 Exceptions

1.15 Program Termination

Table of Contents

1.16 Objects and Classes

1.17 Modules

1.18 Script Writing

1.19 Packages

1.20 Structuring an Application

1.21 Managing Third-Party Packages

1.22 Python: It Fits Your Brain

2 Operators, Expressions, and Data Manipulation

2.1 Literals

2.2 Expressions and Locations

2.3 Standard Operators

2.4 In-Place Assignment

2.5 Object Comparison

2.6 Ordered Comparison Operators

2.7 Boolean Expressions and Truth Values

2.8 Conditional Expressions

2.9 Operations Involving Iterables

2.10 Operations on Sequences

2.11 Operations on Mutable Sequences

2.12 Operations on Sets

2.13 Operations on Mappings

2.14 List, Set, and Dictionary Comprehensions

2.15 Generator Expressions

2.16 The Attribute (.) Operator

2.17 The Function Call () Operator

2.18 Order of Evaluation

Table of Contents

2.19 Final Words: The Secret Life of Data

3 Program Structure and Control Flow

3.1 Program Structure and Execution

3.2 Conditional Execution

3.3 Loops and Iteration

3.4 Exceptions

3.4.1 The Exception Hierarchy

3.4.2 Exceptions and Control Flow

3.4.3 Defining New Exceptions

3.4.4 Chained Exceptions

3.4.5 Exception Tracebacks

3.4.6 Exception Handling Advice

3.5 Context Managers and the with Statement

3.6 Assertions and `__debug__`

3.7 Final Words

4 Objects, Types, and Protocols

4.1 Essential Concepts

4.2 Object Identity and Type

4.3 Reference Counting and Garbage Collection

4.4 References and Copies

4.5 Object Representation and Printing

4.6 First-Class Objects

4.7 Using None for Optional or Missing Data

4.8 Object Protocols and Data Abstraction

4.9 Object Protocol

4.10 Number Protocol

4.11 Comparison Protocol

Table of Contents

- 4.12 Conversion Protocols
- 4.13 Container Protocol
- 4.14 Iteration Protocol
- 4.15 Attribute Protocol
- 4.16 Function Protocol
- 4.17 Context Manager Protocol
- 4.18 Final Words: On Being Pythonic

5 Functions

- 5.1 Function Definitions
- 5.2 Default Arguments
- 5.3 Variadic Arguments
- 5.4 Keyword Arguments
- 5.5 Variadic Keyword Arguments
- 5.6 Functions Accepting All Inputs
- 5.7 Positional-Only Arguments
- 5.8 Names, Documentation Strings, and Type Hints
- 5.9 Function Application and Parameter Passing
- 5.10 Return Values
- 5.11 Error Handling
- 5.12 Scoping Rules
- 5.13 Recursion
- 5.14 The lambda Expression
- 5.15 Higher-Order Functions
- 5.16 Argument Passing in Callback Functions
- 5.17 Returning Results from Callbacks
- 5.18 Decorators

Table of Contents

5.19 Map, Filter, and Reduce

5.20 Function Introspection, Attributes, and Signatures

5.21 Environment Inspection

5.22 Dynamic Code Execution and Creation

5.23 Asynchronous Functions and await

5.24 Final Words: Thoughts on Functions and Composition

6 Generators

6.1 Generators and yield

6.2 Restartable Generators

6.3 Generator Delegation

6.4 Using Generators in Practice

6.5 Enhanced Generators and yield Expressions

6.6 Applications of Enhanced Generators

6.7 Generators and the Bridge to Awaiting

6.8 Final Words: A Brief History of Generators and Looking Forward

7 Classes and Object-Oriented Programming

7.1 Objects

7.2 The class Statement

7.3 Instances

7.4 Attribute Access

7.5 Scoping Rules

7.6 Operator Overloading and Protocols

7.7 Inheritance

7.8 Avoiding Inheritance via Composition

7.9 Avoiding Inheritance via Functions

7.10 Dynamic Binding and Duck Typing

Table of Contents

- 7.11 The Danger of Inheriting from Built-in Types
- 7.12 Class Variables and Methods
- 7.13 Static Methods
- 7.14 A Word about Design Patterns
- 7.15 Data Encapsulation and Private Attributes
- 7.16 Type Hinting
- 7.17 Properties
- 7.18 Types, Interfaces, and Abstract Base Classes
- 7.19 Multiple Inheritance, Interfaces, and Mixins
- 7.20 Type-Based Dispatch
- 7.21 Class Decorators
- 7.22 Supervised Inheritance
- 7.23 The Object Life Cycle and Memory Management
- 7.24 Weak References
- 7.25 Internal Object Representation and Attribute Binding
- 7.26 Proxies, Wrappers, and Delegation
- 7.27 Reducing Memory Use with `__slots__`
- 7.28 Descriptors
- 7.29 Class Definition Process
- 7.30 Dynamic Class Creation
- 7.31 Metaclasses
- 7.32 Built-in Objects for Instances and Classes
- 7.33 Final Words: Keep It Simple

8 Modules and Packages

- 8.1 Modules and the import Statement
- 8.2 Module Caching

Table of Contents

- 8.3 Importing Selected Names from a Module
- 8.4 Circular Imports
- 8.5 Module Reloading and Unloading
- 8.6 Module Compilation
- 8.7 The Module Search Path
- 8.8 Execution as the Main Program
- 8.9 Packages
- 8.10 Imports Within a Package
- 8.11 Running a Package Submodule as a Script
- 8.12 Controlling the Package Namespace
- 8.13 Controlling Package Exports
- 8.14 Package Data
- 8.15 Module Objects
- 8.16 Deploying Python Packages
- 8.17 The Penultimate Word: Start with a Package
- 8.18 The Final Word: Keep It Simple

9 Input and Output

- 9.1 Data Representation
- 9.2 Text Encoding and Decoding
- 9.3 Text and Byte Formatting
- 9.4 Reading Command-Line Options
- 9.5 Environment Variables
- 9.6 Files and File Objects
 - 9.6.1 Filenames
 - 9.6.2 File Modes
 - 9.6.3 I/O Buffering
 - 9.6.4 Text Mode Encoding

Table of Contents

9.6.5 Text-Mode Line Handling

9.7 I/O Abstraction Layers

9.7.1 File Methods

9.8 Standard Input, Output, and Error

9.9 Directories

9.10 The print () function

9.11 Generating Output

9.12 Consuming Input

9.13 Object Serialization

9.14 Blocking Operations and Concurrency

9.14.1 Nonblocking I/O

9.14.2 I/O Polling

9.14.3 Threads

9.14.4 Concurrent Execution with asyncio

9.15 Standard Library Modules

9.15.1 asyncio Module

9.15.2 binascii Module

9.15.3 cgi Module

9.15.4 configparser Module

9.15.5 csv Module

9.15.6 errno Module

9.15.7 fcntl Module

9.15.8 hashlib Module

9.15.9 http Package

9.15.10 io Module

9.15.11 json Module

9.15.12 logging Module

9.15.13 os Module

9.15.14 os.path Module

Table of Contents

- 9.15.15 pathlib Module
- 9.15.16 re Module
- 9.15.17 shutil Module
- 9.15.18 select Module
- 9.15.19 smtplib Module
- 9.15.20 socket Module
- 9.15.21 struct Module
- 9.15.22 subprocess Module
- 9.15.23 tempfile Module
- 9.15.24 textwrap Module
- 9.15.25 threading Module
- 9.15.26 time Module
- 9.15.27 urllib Package
- 9.15.28 unicodedata Module
- 9.15.29 xml Package

9.16 Final Words

10 Built-in Functions and Standard Library

10.1 Built-in Functions

10.2 Built-in Exceptions

- 10.2.1 Exception Base Classes
- 10.2.2 Exception Attributes
- 10.2.3 Predefined Exception Classes

10.3 Standard Library

- 10.3.1 collections Module
- 10.3.2 datetime Module
- 10.3.3 itertools Module
- 10.3.4 inspect Module
- 10.3.5 math Module
- 10.3.6 os Module

Table of Contents

10.3.7 random Module

10.3.8 re Module

10.3.9 shutil Module

10.3.10 statistics Module

10.3.11 sys Module

10.3.12 time Module

10.3.13 turtle Module

10.3.14 unittest Module

10.4 Final Words: Use the Built-Ins

Index