

David M. Beazley



# Python

## Essential Reference

Fourth Edition

**Developer's Library**



# Python

ESSENTIAL REFERENCE

---

Fourth Edition

# Python Essential Reference

## Table of Contents

### Table of Contents

#### Introduction

#### I: The Python Language

##### 1 A Tutorial Introduction

Running Python

Variables and Arithmetic Expressions

Conditionals

File Input and Output

Strings

Lists

Tuples

Sets

Dictionaries

Iteration and Looping

Functions

Generators

Coroutines

Objects and Classes

Exceptions

Modules

Getting Help

##### 2 Lexical Conventions and Syntax

Line Structure and Indentation

Identifiers and Reserved Words

# Table of Contents

Numeric Literals

String Literals

Containers

Operators, Delimiters, and Special Symbols

Documentation Strings

Decorators

Source Code Encoding

## 3 Types and Objects

Terminology

Object Identity and Type

Reference Counting and Garbage Collection

References and Copies

First-Class Objects

Built-in Types for Representing Data

Built-in Types for Representing Program Structure

Built-in Types for Interpreter Internals

Object Behavior and Special Methods

## 4 Operators and Expressions

Operations on Numbers

Operations on Sequences

String Formatting

Advanced String Formatting

Operations on Dictionaries

Operations on Sets

Augmented Assignment

The Attribute (.) Operator

The Function Call () Operator

Conversion Functions

Boolean Expressions and Truth Values

# **Table of Contents**

Object Equality and Identity

Order of Evaluation

Conditional Expressions

## **5 Program Structure and Control Flow**

Program Structure and Execution

Conditional Execution

Loops and Iteration

Exceptions

Context Managers and the with Statement

Assertions and `__debug__`

## **6 Functions and Functional Programming**

Functions

Parameter Passing and Return Values

Scoping Rules

Functions as Objects and Closures

Decorators

Generators and yield

Coroutines and yield Expressions

Using Generators and Coroutines

List Comprehensions

Generator Expressions

Declarative Programming

The lambda Operator

Recursion

Documentation Strings

Function Attributes

`eval()`, `exec()`, and `compile()`

## **7 Classes and Object-Oriented Programming**

The class Statement

# Table of Contents

Class Instances

Scoping Rules

Inheritance

Polymorphism Dynamic Binding and Duck Typing

Static Methods and Class Methods

Properties

Descriptors

Data Encapsulation and Private Attributes

Object Memory Management

Object Representation and Attribute Binding

`__slots__`

Operator Overloading

Types and Class Membership Tests

Abstract Base Classes

Metaclasses

Class Decorators

## 8 Modules, Packages, and Distribution

Modules and the `import` Statement

Importing Selected Symbols from a Module

Execution as the Main Program

The Module Search Path

Module Loading and Compilation

Module Reloading and Unloading

Packages

Distributing Python Programs and Libraries

Installing Third-Party Libraries

## 9 Input and Output

Reading Command-Line Options

Environment Variables

# **Table of Contents**

- Files and File Objects
- Standard Input, Output, and Error
- The print Statement
- The print() Function
- Variable Interpolation in Text Output
- Generating Output
- Unicode String Handling
- Unicode I/O
- Object Persistence and the pickle Module

## **10 Execution Environment**

- Interpreter Options and Environment
- Interactive Sessions
- Launching Python Applications
- Site Configuration Files
- Per-user Site Packages
- Enabling Future Features
- Program Termination

## **11 Testing, Debugging, Profiling, and Tuning**

- Documentation Strings and the doctest Module
- Unit Testing and the unittest Module
- The Python Debugger and the pdb Module
- Program Profiling
- Tuning and Optimization

## **II: The Python Library**

### **12 Built-In Functions and Exceptions**

- Built-in Functions and Types
- Built-In Exceptions
- Built-In Warnings

# Table of Contents

future\_builtins

## 13 Python Runtime Services

atexit

copy

gc

inspect

marshal

pickle

sys

traceback

types

warnings

weakref

## 14 Mathematics

decimal

fractions

math

numbers

random

## 15 Data Structures, Algorithms, and Code Simplification

abc

array

bisect

collections

contextlib

functools

heapq

itertools

operator

# **Table of Contents**

## **16 String and Text Handling**

codecs

re

string

struct

unicodedata

## **17 Python Database Access**

Relational Database API Specification

sqlite3 Module

DBM-Style Database Modules

shelve Module

## **18 File and Directory Handling**

bz2

filecmp

fnmatch

glob

gzip

shutil

tarfile

tempfile

zipfile

zlib

## **19 Operating System Services**

commands

configParser, Configparser

datetime

errno

fcntl

io

# Table of Contents

logging

mmap

msvcrt

optparse

os

os.path

signal

subprocess

time

winreg

## 20 Threads and Concurrency

Basic Concepts

Concurrent Programming and Python

multiprocessing

threading

queue, Queue

Coroutines and Microthreading

## 21 Network Programming and Sockets

Network Programming Basics

asynchat

asyncore

select

socket

ssl

SocketServer

## 22 Internet Application Programming

ftplib

http Package

smtplib

# Table of Contents

urllib Package

xmlrpc Package

## 23 Web Programming

cgi

cgitb

wsgiref

webbrowser

## 24 Internet Data Handling and Encoding

base64

binascii

csv

email Package

hashlib

hmac

HTMLParser

json

mimetypes

quopri

xml Package

## 25 Miscellaneous Library Modules

Python Services

String Processing

Operating System Modules

Network

Internet Data Handling

Internationalization

Multimedia Services

Miscellaneous

## III: Extending and Embedding

# **Table of Contents**

## **26 Extending and Embedding Python**

Extension Modules

Embedding the Python Interpreter

ctypes

Advanced Extending and Embedding

Jython and IronPython

## **Appendix: Python 3**

### **Who Should Be Using Python 3?**

### **New Language Features**

Source Code Encoding and Identifiers

Set Literals

Set and Dictionary Comprehensions

Extended Iterable Unpacking

Nonlocal Variables

Function Annotations

Keyword-Only Arguments

Ellipsis as an Expression

Chained Exceptions

Improved super()

Advanced Metaclasses

### **Common Pitfalls**

Text Versus Bytes

New I/O System

print() and exec() Functions

Use of Iterators and Views

Integers and Integer Division

Comparisons

Iterators and Generators

File Names, Arguments, and Environment Variables

# **Table of Contents**

Library Reorganization

Absolute Imports

## **Code Migration and 2to3**

Porting Code to Python 2.6

Providing Test Coverage

Using the 2to3 Tool

A Practical Porting Strategy

Simultaneous Python 2 and Python 3 Support

Participate

## **Index**