



SAMPLE TEST PDF

DISCOVER THE HARNESS OF PYTHON PROGRAMMING

Python Programming Language

Purpose :
Learning

Author :
Louveau Simon

24 mars 2021

Table des matières

1	Introduction	5
2	Whetting Your Appetite	6
3	Using the Python Interprete	7
3.1	Invoking the Interpreter	7
3.1.1	Argument Passing	7
3.1.2	Interactive Mode	7
3.2	The Interpreter and Its Environment	7
3.2.1	Source Code Encoding	7
4	An Informal Introduction to Python	8
4.1	Using Python as a Calculator	8
4.1.1	Numbers	8
4.1.2	Strings	8
4.1.3	Lists	8
4.2	First Steps Towards Programming	8
5	More Control Flow Tools	9
5.1	if Statements	9
5.2	for Statements	9
5.3	The range() Function	9
5.4	break and continue Statements, and else Clauses on Loops	9
5.5	pass Statements	9
5.6	Defining Functions	9
5.7	More on Defining Functions	9
5.7.1	Default Argument Values	9
5.7.2	Keyword Arguments	9
5.7.3	Special parameters	9
5.7.4	Positional-or-Keyword Arguments	9
5.7.5	Positional-Only Parameters	9
5.7.6	Keyword-Only Arguments	9
5.7.7	Function Examples	9
5.7.8	Recap	9
5.7.9	Arbitrary Argument Lists	9
5.7.10	Unpacking Argument Lists	9
5.7.11	Lambda Expressions	9
5.7.12	Documentation Strings	9
5.7.13	Function Annotations	9

5.8	Intermezzo : Coding Style	9
6	Data Structures	10
6.1	More on Lists	10
6.1.1	Using Lists as Stacks	10
6.1.2	Using Lists as Queues	10
6.1.3	List Comprehensions	10
6.1.4	Nested List Comprehensions	10
6.2	The del statement	10
6.3	Tuples and Sequences	10
6.4	Sets	10
6.5	Dictionaries	10
6.6	Looping Techniques	10
6.7	More on Conditions	10
6.8	Comparing Sequences and Other Types	10
7	Modules	11
7.1	More on Modules	11
7.1.1	Executing modules as scripts	11
7.1.2	The Module Search Path	11
7.1.3	“Compiled” Python files	11
7.2	Standard Modules	11
7.3	The dir() Function	11
7.4	Packages	11
7.4.1	Importing * From a Package	11
7.4.2	Intra-package References	11
7.4.3	Packages in Multiple Directories	11
8	Input and Output	12
8.1	Fancier Output Formatting	12
8.1.1	Formatted String Literals	12
8.1.2	The String format() Method	12
8.1.3	Manual String Formatting	12
8.1.4	Old string formatting	12
8.2	Reading and Writing Files	12
8.2.1	Methods of File Objects	12
8.2.2	Saving structured data with json	12
9	Errors and Exceptions	13
9.1	Syntax Errors	13
9.2	Exceptions	13

9.3	Handling Exceptions	13
9.4	Raising Exceptions	13
9.5	Exception Chaining	13
9.6	User-defined Exceptions	13
9.7	Defining Clean-up Actions	13
9.8	Predefined Clean-up Actions	13
10	Classes	14
10.1	A Word About Names and Objects	14
10.2	Python Scopes and Namespaces	14
	10.2.1 Scopes and Namespaces Example	14
10.3	A First Look at Classes	14
	10.3.1 Class Definition Syntax	14
	10.3.2 Class Objects	14
	10.3.3 Instance Objects	14
	10.3.4 Method Objects	14
	10.3.5 Class and Instance Variables	14
10.4	Random Remarks	14
10.5	Inheritance	14
	10.5.1 Multiple Inheritance	14
10.6	Private Variables	14
10.7	Odds and Ends	14
10.8	Iterators	14
10.9	Generators	14
10.10	Generator Expressions	14
11	Brief Tour of the Standard Library	15
11.1	Operating System Interface	15
11.2	File Wildcards	15
11.3	Command Line Arguments	15
11.4	Error Output Redirection and Program Termination	15
11.5	String Pattern Matching	15
11.6	Mathematics	15
11.7	Internet Access	15
11.8	Dates and Times	15
11.9	Data Compression	15
11.10	Performance Measurement	15
11.11	Quality Control	15
11.12	Batteries Included	15
12	Brief Tour of the Standard Library — Part II	16

12.1	Output Formatting	16
12.2	Templating	16
12.3	Working with Binary Data Record Layouts	16
12.4	Multi-threading	16
12.5	Logging	16
12.6	Weak References	16
12.7	Tools for Working with Lists	16
12.8	Decimal Floating Point Arithmetic	16
13	Virtual Environments and Packages	17
13.1	Introduction	17
13.2	Creating Virtual Environments	17
13.3	Managing Packages with pip	17
14	What Now ?	18
15	Interactive Input Editing and History Substitution	19
15.1	Tab Completion and History Editing	19
15.2	Alternatives to the Interactive Interpreter	19
16	Floating Point Arithmetic : Issues and Limitations	20
16.1	Representation Error	20
17	Appendix	21
17.1	Interactive Mode	21
17.1.1	Error Handling	21
17.1.2	Executable Python Scripts	21
17.1.3	The Interactive Startup File	21
17.1.4	The Customization Modules	21

1 Introduction

2 Whetting Your Appetite

3 Using the Python Interpreter

3.1 Invoking the Interpreter

3.1.1 Argument Passing

3.1.2 Interactive Mode

3.2 The Interpreter and Its Environment

3.2.1 Source Code Encoding

4 An Informal Introduction to Python

4.1 Using Python as a Calculator

4.1.1 Numbers

4.1.2 Strings

4.1.3 Lists

4.2 First Steps Towards Programming

5 More Control Flow Tools

- 5.1 if Statements
- 5.2 for Statements
- 5.3 The range() Function
- 5.4 break and continue Statements, and else Clauses on Loops
- 5.5 pass Statements
- 5.6 Defining Functions
- 5.7 More on Defining Functions
 - 5.7.1 Default Argument Values
 - 5.7.2 Keyword Arguments
 - 5.7.3 Special parameters
 - 5.7.4 Positional-or-Keyword Arguments
 - 5.7.5 Positional-Only Parameters
 - 5.7.6 Keyword-Only Arguments
 - 5.7.7 Function Examples
 - 5.7.8 Recap
 - 5.7.9 Arbitrary Argument Lists
 - 5.7.10 Unpacking Argument Lists
 - 5.7.11 Lambda Expressions
 - 5.7.12 Documentation Strings
 - 5.7.13 Function Annotations
- 5.8 Intermezzo : Coding Style

6 Data Structures

6.1 More on Lists

6.1.1 Using Lists as Stacks

6.1.2 Using Lists as Queues

6.1.3 List Comprehensions

6.1.4 Nested List Comprehensions

6.2 The del statement

6.3 Tuples and Sequences

6.4 Sets

6.5 Dictionaries

6.6 Looping Techniques

6.7 More on Conditions

6.8 Comparing Sequences and Other Types

7 Modules

7.1 More on Modules

7.1.1 Executing modules as scripts

7.1.2 The Module Search Path

7.1.3 “Compiled” Python files

7.2 Standard Modules

7.3 The `dir()` Function

7.4 Packages

7.4.1 Importing * From a Package

7.4.2 Intra-package References

7.4.3 Packages in Multiple Directories

8 Input and Output

8.1 Fancier Output Formatting

8.1.1 Formatted String Literals

8.1.2 The String `format()` Method

8.1.3 Manual String Formatting

8.1.4 Old string formatting

8.2 Reading and Writing Files

8.2.1 Methods of File Objects

8.2.2 Saving structured data with json

9 Errors and Exceptions

- 9.1 Syntax Errors
- 9.2 Exceptions
- 9.3 Handling Exceptions
- 9.4 Raising Exceptions
- 9.5 Exception Chaining
- 9.6 User-defined Exceptions
- 9.7 Defining Clean-up Actions
- 9.8 Predefined Clean-up Actions

10 Classes

10.1 A Word About Names and Objects

10.2 Python Scopes and Namespaces

10.2.1 Scopes and Namespaces Example

10.3 A First Look at Classes

10.3.1 Class Definition Syntax

10.3.2 Class Objects

10.3.3 Instance Objects

10.3.4 Method Objects

10.3.5 Class and Instance Variables

10.4 Random Remarks

10.5 Inheritance

10.5.1 Multiple Inheritance

10.6 Private Variables

10.7 Odds and Ends

10.8 Iterators

10.9 Generators

10.10 Generator Expressions

11 Brief Tour of the Standard Library

- 11.1 Operating System Interface
- 11.2 File Wildcards
- 11.3 Command Line Arguments
- 11.4 Error Output Redirection and Program Termination
- 11.5 String Pattern Matching
- 11.6 Mathematics
- 11.7 Internet Access
- 11.8 Dates and Times
- 11.9 Data Compression
- 11.10 Performance Measurement
- 11.11 Quality Control
- 11.12 Batteries Included

12 Brief Tour of the Standard Library — Part II

12.1 Output Formatting

12.2 Templating

12.3 Working with Binary Data Record Layouts

12.4 Multi-threading

12.5 Logging

12.6 Weak References

12.7 Tools for Working with Lists

12.8 Decimal Floating Point Arithmetic

13 Virtual Environments and Packages

13.1 Introduction

13.2 Creating Virtual Environments

13.3 Managing Packages with pip

14 What Now ?

15 Interactive Input Editing and History Substitution

15.1 Tab Completion and History Editing

15.2 Alternatives to the Interactive Interpreter

16 Floating Point Arithmetic : Issues and Limitations

16.1 Representation Error

17 Appendix

17.1 Interactive Mode

17.1.1 Error Handling

17.1.2 Executable Python Scripts

17.1.3 The Interactive Startup File

17.1.4 The Customization Modules