



Ms. S. S. Shanthi is an Assistant Professor in the P.G. Department of Computer Science with over 20 years of teaching experience in reputed institutions across Tamil Nadu. Her expertise spans Software Testing, Artificial Intelligence and Machine Learning, Linux, Full Stack Web Development, and Python Programming. She has published in UGC Care-listed journals, authored books and book chapters, and presented papers at national and international conferences. She also served as a Resource Person for the Young Student Scientist Programme (YSSP) supported by the Tamil Nadu State Council for Science and Technology, mentoring young learners in science and research.



Ms. G. Angayarkanni is an Assistant Professor at NGM College, Pollachi, with over 10 years of teaching experience in higher education. Her research interests include Machine Learning, Deep Learning, Artificial Intelligence in Healthcare, and Data Science, focusing on predictive modeling and data-driven solutions. She has published 20+ research articles in reputed indexed journals and presented papers at national and international conferences. A recipient of the Best Presentation Award, she has also participated in 50+ workshops and FDPs. She possesses strong programming expertise in Java and Python and is committed to academic excellence and research innovation.

Learning Python



Ms. S. S. Shanthi
Ms. G. Angayarkanni

Ariviyam Publication

Coimbatore

Email us:

ariviyampublication@gmail.com

editor@ariviyampublication.com

www.sjpublication.com

www.ariviyampublication.com

ISBN 978-93-5602-360-4



9 789356 023604

ARIVIYAM PUBLICATION

(A Unit of Ariviyam Groups)

LEARNING PYTHON



அந்நியம்

ISBN 978-93-5602-360-4



All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the author, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law.

Copyright © 2026 Ms. S.S. Shanthi and Ms. G. Angayarkanni

Year:2025

Rs. 499/-

Published by:

Ariviyam Publication (Ariviyam Groups)

(Formerly SJ Publication)

Business Address:

14, Mahalakshmi Garden, Press Colony,

Coimbatore, TN- 641019

Email us:

editor@ariviyampublication.com sjpublicationedu@gmail.com

Website: www.sjpublication.com , www.ariviyampublication.com

Edited by: Mr. Jagadheesh S M.Sc (CS), M.A., B.Ed.,

Disclaimer:

The authors and publisher are not responsible for any errors or consequences arising from the use of the information in this book. The views expressed are those of the authors.

Printing & Binding:

Printed and bound in India.

Table of Content

Part I – Introduction to Programming and Python

Chapter 1: Introduction to Programming **01 to 15**

- 1.1 What is Programming?
- 1.2 Problem Solving and Algorithms
- 1.3 Flowcharts and Pseudocode
- 1.4 Programming Languages: Low-level and High-level
- 1.5 Interpreters vs Compilers
- 1.6 Overview of Popular Programming Languages

Chapter 2: Introduction to Python **16 to 32**

- 2.1 History and Features of Python
- 2.2 Applications of Python
- 2.3 Python Versions and Implementations
- 2.4 Installing Python
- 2.5 Python IDEs (IDLE, VS Code, PyCharm, Jupyter)
- 2.6 Running Python Programs
 - 2.6.1. Running Python Programs in Interactive Mode
 - 2.6.2 Running Python Programs in Script Mode
 - 2.6.3 Running Python Programs Using Command Prompt / Terminal
 - 2.6.4 Running Python Programs in VS Code
 - 2.6.5 Running Python Programs in PyCharm
 - 2.6.6 Running Python Programs in Jupyter Notebook

Chapter 3: Basics of Python Syntax **33 to 41**

- 3.1 Python Keywords and Identifiers
- 3.2 Indentation and Code Blocks
- 3.3 Comments and Documentation Strings
- 3.4 Basic Input and Output
- 3.5 Python Coding Style and PEP 8

Part II - Data Types, Variables and Operators

Chapter 4: Variables and Assignments Variables and Naming Rules

42 to 46

- 4.1 Dynamic Typing in Python
- 4.2 Multiple Assignments
- 4.3 Type Checking and Type Conversion

Chapter 5: Built-in Data Types

47 to 57

- 5.1 Numeric Types (int, float, complex)
- 5.2 Boolean Type
- 5.3 Strings
- 5.4 Type Casting
- 5.5 None Type

Chapter 6: Operators in Python

58 to 69

- 6.1 Arithmetic Operators
- 6.2 Relational Operators
- 6.3 Logical Operators
- 6.4 Assignment Operators
- 6.5 Bitwise Operators
- 6.6 Membership and Identity Operators
- 6.7 Operator Precedence

Part III - Control Flow and Functions

Chapter 7: Conditional Statements

70 to 76

- 7.1 if Statement
- 7.2 if-else Statement
- 7.3 if-elif-else Ladder
- 7.4 Nested Conditions
- 7.5 Short-hand if

Chapter 8: Looping Statements

77 to 76

- 8.1 while Loop
- 8.2 for Loop
- 8.3 range() Function
- 8.4 Loop Control Statements (break, continue, pass)
- 8.5 Nested Loops

Chapter 9: Functions

84 to 90

- 9.1 Defining Functions
- 9.2 Function Arguments
 - 9.2.1 Positional Arguments
 - 9.2.2 Keyword Arguments
 - 9.2.3 Default Arguments
 - 9.2.4 Variable-length Arguments
- 9.3 Return Statement
- 9.4 Lambda Functions
- 9.5 Recursion

Chapter 10: Iterators, Generators, and Functional Programming

91 to 102

- 10.1 Introduction**
 - 10.2 Iterators and Iterable Objects**
 - 10.2.1 Iterable Objects
 - 10.2.2 Iterator Objects
 - 10.2.3 Difference Between Iterable and Iterator
 - 10.2.4 Creating Custom Iterators
 - 10.3 The iter() and next() Functions**
 - 10.3.1 The iter() Function
 - 10.3.2 The next() Function
 - 10.3.3 Handling StopIteration Exception
 - 10.4 Generators and the yield Keyword**
 - 10.4.1 Concept of Generators
 - 10.4.2 Generator Functions
 - 10.4.3 Advantages of Generators
 - 10.4.4 Generator vs Normal Functions
 - 10.5 Generator Expressions**

- 10.5.1 Syntax and Usage
- 10.5.2 Generator Expressions vs List Comprehensions
- 10.5.3 Memory Efficiency Using Generators
- 10.6 Functional Programming Concepts**
 - 10.6.1 Introduction to Functional Programming
 - 10.6.2 Characteristics of Functional Programming
 - 10.6.3 Higher-Order Functions
 - 10.6.4 Lambda Functions
- 10.7 Functional Programming Tools in Python**
 - 10.7.1 The map() Function
 - 10.7.2 The filter() Function
 - 10.7.3 The reduce() Function
- 10.8 The functools Module**
 - 10.8.1 Introduction to functools
 - 10.8.2 reduce() Function
 - 10.8.3 partial() Function
 - 10.8.4 lru_cache() Function
- 10.9 Practical Applications and Examples**
 - 10.9.1 Generators for Data Processing
 - 10.9.2 Functional Programming with Lambda Expressions
 - 10.9.3 Real-Time Memory Optimization Examples

Part IV – Core Data Structures

Chapter 11: Strings and String Operations **103 to 110**

- 11.1 Creating Strings
- 11.2 Indexing and Slicing
- 11.3 String Methods
- 11.4 Formatting Strings
- 11.5 Escape Characters

Chapter 12: Lists **111 to 116**

- 12.1 Creating and Accessing Lists
- 12.2 List Operations
- 12.3 List Methods
- 12.4 List Comprehension
- 12.5 Nested Lists

Chapter 13: Tuples and Sets

117 to 122

- 13.1 Tuples: Creation and Operations
- 13.2 Tuple Packing and Unpacking
- 13.3 Sets: Creation and Properties
- 13.4 Set Operations and Methods

Chapter 14: Dictionaries

123 to 127

- 14.1 Creating Dictionaries
- 14.2 Accessing and Modifying Elements
- 14.3 Dictionary Methods
- 14.4 Iterating over Dictionaries
- 14.5 Applications of Dictionaries

Part V – Files, Modules and Exceptions

Chapter 15: File Handling

128 to 133

- 15.1 Types of Files
- 15.2 Opening and Closing Files
- 15.3 Reading and Writing Files
- 15.4 File Methods
- 15.5 Working with CSV and Text Files

Chapter 16: Modules and Packages

134 to 140

- 16.1 Importing Modules
- 16.2 Built-in Modules
- 16.3 User-defined Modules
- 16.4 Packages
- 16.5 Virtual Environments

Chapter 17: Error and Exception Handling

141 to 146

- 17.1 Types of Errors
- 17.2 Exception Hierarchy
- 17.3 try, except, else, finally
- 17.4 Raising Custom Exceptions

17.5 Assertions

Part VI – Object-Oriented Programming

Chapter 18: Introduction to OOP Concepts 147 to 153

- 18.1 Object-Oriented Paradigm
- 18.2 Classes and Objects
- 18.3 Attributes and Methods
- 18.4 Constructors and Destructors

Chapter 19: Advanced OOP Features 154 to 160

- 19.1 Inheritance
- 19.2 Polymorphism
- 19.3 Encapsulation
- 19.4 Abstraction
- 19.5 Method Overriding
- 19.6 Operator Overloading

Part VII – Data Handling and Applications

Chapter 20: Working with NumPy and Pandas 161 to 165

- 20.1 Introduction to NumPy
- 20.2 Arrays and Operations
- 20.3 Introduction to Pandas
- 20.4 Series and DataFrames
- 20.5 Reading and Writing Data Files

Chapter 21: Python for Web and Automation 166 to 169

- 21.1 Working with APIs
- 21.2 Introduction to Web Frameworks
- 21.3 Automation with Python Scripts
- 21.4 Case Studies and Mini Projects

Appendices 170

Appendix A: Python Keywords and Built-in Functions

Appendix B: Common Errors and Debugging Tips
Appendix C: Practice Programs and Exercises
Appendix D: Glossary of Terms

Sample Python Programs

175