



1. Introduction

- 1.1. What is Python?
- 1.2. Who uses Python?
- 1.3. Why Choose Python?
- 1.4. Should I use Python 2 or Python 3?

2. Numbers

- 2.1. Numeric Types in Python
- 2.2. Integers
- 2.3. Floating-Point Numbers
- 2.4. Complex Numbers

3. Operators

- 3.1. Arithmetic Operators
- 3.2. Assignment Operators
- 3.3. Comparison Operators
- 3.4. Logical Operators
- 3.5. Identity Operators
- 3.6. Membership Operators
- 3.7. Bitwise Operators
- 3.8. Operator Precedence (Order of Operations)

4. List

- 4.1. Create a List
- 4.2. The list() Constructor
- 4.3. Nested List
- 4.4. Access List Items by Index
- 4.5. Negative List Indexing
- 4.6. Access Nested List Items
- 4.7. Slicing a List
- 4.8. Change Item Value
- 4.9. Add items to a list
- 4.10. Combine Lists
- 4.11. Remove items from a list
- 4.12. List Replication
- 4.13. Find List Length
- 4.14. Check if item exists in a list
- 4.15. Iterate through a List
- 4.16. Python List Methods
- 4.17. Built-in Functions with List

5. Nested List

- 5.1. What is Python Nested List?
- 5.2. Create a Nested List
- 5.3. Access Nested List Items by Index
- 5.4. Negative List Indexing In a Nested List
- 5.5. Change Nested List Item Value
- 5.6. Add items to a Nested list
- 5.7. Remove items from a Nested List
- 5.8. Find Nested List Length
- 5.9. Iterate through a Nested List

6. List Slicing

- 6.1. Slicing a List
- 6.2. Slice with Negative Indices
- 6.3. Slice with Positive Negative Indices
- 6.4. Specify Step of the Slicing
- 6.5. Slice at Beginning End
- 6.6. Reverse a List
- 6.7. Modify Multiple List Values
- 6.8. Insert Multiple List Items
- 6.9. Delete Multiple List Items
- 6.10. Clone or Copy a List

7. List Comprehension

- 7.1. What is List Comprehension?
- 7.2. List Comprehension Examples
- 7.3. List Comprehension with if Clause
- 7.4. Nested List Comprehensions
- 7.5. List Comprehension vs map() + lambda
- 7.6. List Comprehension vs filter() + lambda
- 7.7. List Comprehensions and Variable Scope

8. Tuple

- 8.1. What is Python Tuple?
- 8.2. Create a Tuple
- 8.3. Singleton Tuple
- 8.4. The tuple() Constructor
- 8.5. Nested Tuples
- 8.6. Tuple Packing Unpacking
- 8.7. Access Tuple Items
- 8.8. Tuple Slicing



- 8.9. Change Tuple Items
- 8.10. Delete a Tuple
- 8.11. Tuple Concatenation Repetition
- 8.12. Find Tuple Length
- 8.13. Check if item exists in a tuple
- 8.14. Iterate through a tuple
- 8.15. Tuple Sorting
- 8.16. Python Tuple Methods
- 8.17. Built-in Functions with Tuple

9. Set

- 9.1. What is Python Set?
- 9.2. Create a Set
- 9.3. Set constructor
- 9.4. Add Items to a Set
- 9.5. Remove Set Items
- 9.6. Find Set Size
- 9.7. Iterate Through a Set
- 9.8. Check if Item Exists in a Set
- 9.9. Set Operations
- 9.10. Python Frozenset
- 9.11. Built-in Set Functions

10. Dictionary

- 10.1. What is Python dictionary?
- 10.2. Create a Dictionary
- 10.3. The dict() Constructor
- 10.4. Other Ways to Create Dictionaries
- 10.5. Important Properties of a Dictionary
- 10.6. Access Dictionary Items
- 10.7. Add or Update Dictionary Items
- 10.8. Merge Two Dictionaries
- 10.9. Remove Dictionary Items
- 10.10. Get All Keys, Values and Key:Value Pairs
- 10.11. Iterate Through a Dictionary
- 10.12. Check if a Key or Value Exists
- 10.13. Find Dictionary Length
- 10.14. Python Dictionary Methods
- 10.15. Built-in Functions with Dictionary

11. Nested Dictionary

- 11.1. What is Python Nested Dictionary?
- 11.2. Create a Nested Dictionary
- 11.3. The dict() Constructor
- 11.4. Access Nested Dictionary Items
- 11.5. Change Nested Dictionary Items
- 11.6. Add or Update Nested Dictionary Items
- 11.7. Merge Two Nested Dictionaries
- 11.8. Remove Nested Dictionary Items
- 11.9. Iterate Through a Nested Dictionary

12. Dict Comprehension

- 12.1. What is Dictionary Comprehension?
- 12.2. More Examples
- 12.3. Extracting a Subset of a Dictionary
- 12.4. Filter Dictionary Contents
- 12.5. Invert Mapping / Reverse lookup
- 12.6. Dictionary Comprehension with Enumerate
- 12.7. Initialize Dictionary with Comprehension
- 12.8. Dictionary Comprehension with if Clause
- 12.9. Nested Dictionary Comprehension

13. String

- 13.1. What is Python String?
- 13.2. Create a String
- 13.3. Multiline Strings
- 13.4. The str() Constructor
- 13.5. Access by Index
- 13.6. Slicing a String
- 13.7. Modify a String
- 13.8. String Concatenation
- 13.9. Find String Length
- 13.10. Replace Text Within a String
- 13.11. Split and Join a String
- 13.12. String Case Conversion
- 13.13. Check if Substring Contains in a String
- 13.14. Iterate Through a String
- 13.15. Escape Sequence
- 13.16. Raw String
- 13.17. String Formatting
- 13.18. Python String Methods



14. String Slicing

- 14.1. Slicing a String
- 14.2. Slice with Negative Indices
- 14.3. Slice with Positive & Negative Indices
- 14.4. Specify Step of the Slicing
- 14.5. Slice at Beginning & End
- 14.6. Reverse a String

15. If...Else...Elif

- 15.1. The if Statement
- 15.2. Significance of Indentation
- 15.3. Nested if Statement
- 15.4. The else Statement
- 15.5. The elif (else if) Statement
- 15.6. Substitute for Switch Case
- 15.7. Multiple Conditions
- 15.8. One Line if Statement
- 15.9. Conditional Expressions (ternary operator)
- 15.10. Check If Item Present in a Sequence

16. While Loop

- 16.1. Syntax
- 16.2. Basic Examples
- 16.3. Break in while Loop
- 16.4. Continue in while Loop
- 16.5. Else in While Loop
- 16.6. Infinte Loop (while true)

17. for Loop

- 17.1. Syntax
- 17.2. Basic Examples
- 17.3. Break in for Loop
- 17.4. Continue in for Loop
- 17.5. Else in for Loop
- 17.6. range() function in for loop
- 17.7. Nested for Loop
- 17.8. Access Index in for Loop
- 17.9. Unpacking in a for loop
- 17.10. Modify List While Iterating
- 17.11. Looping Through Multiple Lists

18. Functions

- 18.1. Syntax
- 18.2. Create a Function
- 18.3. Call a Function
- 18.4. Pass Arguments
- 18.5. Types of Arguments
- 18.6. Positional Arguments
- 18.7. Keyword Argument
- 18.8. Default Argument Value
- 18.9. Variable Length Argument (*args and **kwargs)
- 18.10. Return Value
- 18.11. Return Multiple Values
- 18.12. Docstring
- 18.13. Composition
- 18.14. Nested Functions
- 18.15. Recursion
- 18.16. Assigning functions to variables
- 18.17. Python Function Executes at Runtime

19. Variables Scope

- 19.1. Local Scope
- 19.2. Global Scope
- 19.3. Modifying Globals Inside a Function
- 19.4. Enclosing Scope
- 19.5. Scoping Rule - LEGB Rule

20. Lambda Function

- 20.1. What is a Lambda Function?
- 20.2. Basic Example
- 20.3. Important characteristics
- 20.4. Multiple Arguments
- 20.5. Ways to Pass Arguments
- 20.6. Lambdas With Map, Filter, and Reduce
- 20.7. Return Multiple Values
- 20.8. if else in a Lambda
- 20.9. List Comprehension in a Lambda
- 20.10. Jump Table Using a Lambda
- 20.11. Lambda Key Functions
- 20.12. Decorating a lambda
- 20.13. Lambda Closures



21. Classes and Objects

- 21.1. Create a Class
- 21.2. The `__init__()` Method
- 21.3. The self Parameter
- 21.4. Attributes
- 21.5. Instance Attribute
- 21.6. Class Attribute
- 21.7. Create an Object
- 21.8. Access and Modify Attributes
- 21.9. Methods
- 21.10. Instance Methods
- 21.11. Delete Attributes and Objects
- 21.12. Constructor
- 21.13. Inheritance
- 21.14. Polymorphism
- 21.15. Encapsulation

22. Inheritance

- 22.1. What Is Inheritance?
- 22.2. Analogy
- 22.3. Defining a Base Class
- 22.4. Subclassing
- 22.5. Override a Method
- 22.6. Add a Method
- 22.7. The `super()` Function
- 22.8. Multiple Inheritance

23. Decorators

- 23.1. Python Functions
- 23.2. Simple Decorator
- 23.3. Syntactic Sugar
- 23.4. Decorating Functions that Takes Arguments
- 23.5. Returning Values from Decorated Functions
- 23.6. Preserving Function Metadata
- 23.7. Unwrapping a Decorator
- 23.8. Nesting Decorators
- 23.9. Applying Decorators to Built-in Functions
- 23.10. Real World Examples

24. @Property

- 24.1. Define a Property

- 24.2. `property()` Function
- 24.3. `@property` - Property as a Decorator
- 24.4. Real World Example
- 24.5. Computed Attributes
- 24.6. Extending a Property in a Subclass

25. Exception Handling

- 25.1. What is Exception Handling?
- 25.2. The try and except Block
- 25.3. Catch Multiple Exceptions
- 25.4. The Else Clause
- 25.5. The Finally Clause
- 25.6. Raising an Exception
- 25.7. User-defined Exceptions

26. Read/Write Txt Files

- 26.1. Open a File
- 26.2. Specify File Mode
- 26.3. Read a File
- 26.4. Read Lines
- 26.5. Write a File
- 26.6. Write Multiple Lines
- 26.7. Flush Output Buffer
- 26.8. Close a File
- 26.9. Create a New File
- 26.10. Delete a File
- 26.11. Check if File Exists
- 26.12. Random Access

27. Read/Write CSV Files

- 27.1. Open a CSV File
- 27.2. Specify File Mode
- 27.3. Close a CSV File
- 27.4. Read a CSV File
- 27.5. Write to a CSV File
- 27.6. Read a CSV File Into a Dictionary
- 27.7. Write a CSV File From a Dictionary
- 27.8. Use a different delimiter
- 27.9. Handle Comma Within a Data
- 27.10. Catching and Reporting Errors



28. Built-in Methods
29. List Methods
30. Tuple Methods
31. Set Methods
32. Dictionary Methods
33. Database connection