



1. About Python

- 1.1. What is Python?
- 1.2. Python Feature
- 1.3. Python History & Versions
- 1.4. Python Applications
- 1.5. How to install Python?
- 1.6. Programming on an interactive Shell
- 1.7. Pycharm IDLE
- 1.8. Python First Program
- 1.9. Python print() Function
- 1.10. Python input() Function
- 1.11. Indention, Escape sequence
- 1.12. Type Casting

2. Variables

- 2.1. Identifiers
- 2.2. Keywords
- 2.3. Literals
- 2.4. Tokens
- 2.5. Data Type

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. If...Else...Elif

- 4.1. The if Statement
- 4.2. Significance of Indentation
- 4.3. Nested if Statement
- 4.4. The else Statement
- 4.5. The elif (else if) Statement
- 4.6. Substitute for Switch Case
- 4.7. Multiple Conditions
- 4.8. One Line if Statement
- 4.9. Conditional Expressions (ternary operator)
- 4.10. Check If Item Present in a Sequence

5. While Loop

- 5.1. Syntax
- 5.2. Basic Examples
- 5.3. Break in while Loop
- 5.4. Continue in while Loop
- 5.5. Else in While Loop
- 5.6. Infinte Loop (while true)

6. For Loop

- 6.1. Syntax
- 6.2. Basic Examples
- 6.3. Break in for Loop
- 6.4. Continue in for Loop
- 6.5. Else in for Loop
- 6.6. range() function in for loop
- 6.7. Nested for Loop
- 6.8. Access Index in for Loop
- 6.9. Unpacking in a for loop
- 6.10. Modify List While Iterating
- 6.11. Looping Through Multiple Lists

7. List

- 7.1. Create a List
- 7.2. The list() Constructor
- 7.3. Nested List
- 7.4. Access List Items by Index
- 7.5. Negative List Indexing
- 7.6. Access Nested List Items
- 7.7. Slicing a List
- 7.8. Change Item Value
- 7.9. Add items to a list
- 7.10. Combine Lists
- 7.11. Remove items from a list
- 7.12. List Replication
- 7.13. Find List Length
- 7.14. Check if item exists in a list
- 7.15. Iterate through a List
- 7.16. Python List Methods
- 7.17. Built-in Functions with List

8. Nested List

- 8.1. What is Python Nested List?
- 8.2. Create a Nested List



- 8.3. Access Nested List Items by Index
- 8.4. Negative List Indexing In a Nested List
- 8.5. Change Nested List Item Value
- 8.6. Add items to a Nested list
- 8.7. Remove items from a Nested List
- 8.8. Find Nested List Length
- 8.9. Iterate through a Nested List

9. List Slicing

- 9.1. Slicing a List
- 9.2. Slice with Negative Indices
- 9.3. Slice with Positive Negative Indices
- 9.4. Specify Step of the Slicing
- 9.5. Slice at Beginning End
- 9.6. Reverse a List
- 9.7. Modify Multiple List Values
- 9.8. Insert Multiple List Items
- 9.9. Delete Multiple List Items
- 9.10. Clone or Copy a List

10. List Comprehension

- 10.1. What is List Comprehension?
- 10.2. List Comprehension Examples
- 10.3. List Comprehension with if Clause
- 10.4. Nested List Comprehensions
- 10.5. List Comprehension vs map() + lambda
- 10.6. List Comprehension vs filter() + lambda
- 10.7. List Comprehensions and Variable Scope

11. List Methods

- 11.1. append(), insert(), extend()
- 11.2. remove(), pop(), clear(), sort()
- 11.3. copy(), count(), index(), reverse()
- 11.4. all(), any(), enumerate(), len(), list()
- 11.5. max(), min(), sorted(), sum()

12. Tuple

- 12.1. What is Python Tuple?
- 12.2. Create a Tuple
- 12.3. Singleton Tuple
- 12.4. The tuple() Constructor
- 12.5. Nested Tuples
- 12.6. Tuple Packing Unpacking
- 12.7. Access Tuple Items

- 12.8. Tuple Slicing
- 12.9. Change Tuple Items
- 12.10. Delete a Tuple
- 12.11. Tuple Concatenation Repetition
- 12.12. Find Tuple Length
- 12.13. Check if item exists in a tuple
- 12.14. Iterate through a tuple
- 12.15. Tuple Sorting
- 12.16. Python Tuple Methods
- 12.17. Built-in Functions with Tuple

13. Tuple Methods

- 13.1. Count(), index()
- 13.2. all(), any(), enumerate(), len(),
- 13.3. max(), min(), sorted(), sum(), tuple()

14. Set

- 14.1. What is Python Set?
- 14.2. Create a Set
- 14.3. Set constructor
- 14.4. Add Items to a Set
- 14.5. Remove Set Items
- 14.6. Find Set Size
- 14.7. Iterate Through a Set
- 14.8. Check if Item Exists in a Set
- 14.9. Set Operations
- 14.10. Python Frozenset
- 14.11. Built-in Set Functions

15. Set Comprehension

- 15.1. What is Set Comprehension?
- 15.2. SetCompConditional
- 15.3. SetCompNestedConditional
- 15.4. SetCompIfElse
- 15.5. NestedSetComp

16. Set Methods

- 16.1. union(), update(), intersection()
- 16.2. intersection_update(), difference()
- 16.3. difference_update(), symmetric_difference()
- 16.4. symmetric_difference_update()
- 16.5. isdisjoint(), issubset(), issuperset()
- 16.6. all(), any(), enumerate(), len()
- 16.7. max(), min(), sorted(), sum()



17. Dictionary

- 17.1. What is Python dictionary?
- 17.2. Create a Dictionary
- 17.3. The dict() Constructor
- 17.4. Other Ways to Create Dictionaries
- 17.5. Important Properties of a Dictionary
- 17.6. Access Dictionary Items
- 17.7. Add or Update Dictionary Items
- 17.8. Merge Two Dictionaries
- 17.9. Remove Dictionary Items
- 17.10. Get All Keys, Values and Key:Value Pairs
- 17.11. Iterate Through a Dictionary
- 17.12. Check if a Key or Value Exists
- 17.13. Find Dictionary Length
- 17.14. Python Dictionary Methods
- 17.15. Built-in Functions with Dictionary

18. Nested Dictionary

- 18.1. What is Python Nested Dictionary?
- 18.2. Create a Nested Dictionary
- 18.3. The dict() Constructor
- 18.4. Access Nested Dictionary Items
- 18.5. Change Nested Dictionary Items
- 18.6. Add or Update Nested Dictionary Items
- 18.7. Merge Two Nested Dictionaries
- 18.8. Remove Nested Dictionary Items
- 18.9. Iterate Through a Nested Dictionary

19. Dictionary Comprehension

- 19.1. What is Dictionary Comprehension?
- 19.2. More Examples
- 19.3. Extracting a Subset of a Dictionary
- 19.4. Filter Dictionary Contents
- 19.5. Invert Mapping / Reverse lookup
- 19.6. Dictionary Comprehension with Enumerate
- 19.7. Initialize Dictionary with Comprehension
- 19.8. Dictionary Comprehension with if Clause
- 19.9. Nested Dictionary Comprehension

20. Dictionary Methods

- 20.1. Clear(), copy(), fromkeys(), get(), items()
- 20.2. Keys(), pop(), popitem(), setdefault(), update()
- 20.3. Values(), all(), any(), len(), sorted()

21. String

- 21.1. What is Python String?
- 21.2. Create a String
- 21.3. Multiline Strings
- 21.4. The str() Constructor
- 21.5. Access by Index
- 21.6. Slicing a String
- 21.7. Modify a String
- 21.8. String Concatenation
- 21.9. Find String Length
- 21.10. Replace Text Within a String
- 21.11. Split and Join a String
- 21.12. String Case Conversion
- 21.13. Check if Substring Contains in a String
- 21.14. Iterate Through a String
- 21.15. Escape Sequence
- 21.16. Raw String
- 21.17. String Formatting
- 21.18. Python String Methods

22. String Slicing

- 22.1. Slicing a String
- 22.2. Slice with Negative Indices
- 22.3. Slice with Positive & Negative Indices
- 22.4. Specify Step of the Slicing
- 22.5. Slice at Beginning & End
- 22.6. Reverse a String

23. String Methods

- 23.1. CStyle Formatting, FString
- 23.2. capitalize(), casefold(), center()
- 23.3. count(), encode(), endswith()
- 23.4. expandtabs(), find(), format(), index()

24. Regex & Range Function

- 24.1. What is Regex?
- 24.2. Regex Functions
- 24.3. Forming a regular expression
- 24.4. Use of meta character, sequence, sets
- 24.5. Use of findall() function
- 24.6. Use of match object
- 24.7. What is Range Function?



25. Functions

- 25.1. Syntax
- 25.2. Create a Function
- 25.3. Call a Function
- 25.4. Pass Arguments
- 25.5. Types of Arguments
- 25.6. Positional Arguments
- 25.7. Keyword Argument
- 25.8. Default Argument Value
- 25.9. Variable Length Argument (*args and **kwargs)
- 25.10. Return Value
- 25.11. Return Multiple Values
- 25.12. Docstring
- 25.13. Composition
- 25.14. Nested Functions
- 25.15. Recursion
- 25.16. Assigning functions to variables
- 25.17. Python Function Executes at Runtime

26. Variables Scope

- 26.1. Local Scope
- 26.2. Global Scope
- 26.3. Modifying Globals Inside a Function
- 26.4. Enclosing Scope
- 26.5. Scoping Rule - LEGB Rule

27. Lambda Function

- 27.1. What is a Lambda Function?
- 27.2. Basic Example
- 27.3. Important characteristics
- 27.4. Multiple Arguments
- 27.5. Ways to Pass Arguments
- 27.6. Lambdas With Map, Filter, and Reduce
- 27.7. Return Multiple Values
- 27.8. if else in a Lambda
- 27.9. List Comprehension in a Lambda
- 27.10. Jump Table Using a Lambda
- 27.11. Lambda Key Functions
- 27.12. Decorating a lambda
- 27.13. Lambda Closures

28. Generator

- 28.1. What is Generator?
- 28.2. How to create Generator function?
- 28.3. Yield vs. return
- 28.4. generator vs normal function
- 28.5. Advantage of generator
- 28.6. Pipelining with generator

29. Decorators

- 29.1. Python Functions
- 29.2. Simple Decorator
- 29.3. Syntactic Sugar
- 29.4. Decorating Functions that Takes Arguments
- 29.5. Returning Values from Decorated Functions
- 29.6. Preserving Function Metadata
- 29.7. Unwrapping a Decorator
- 29.8. Nesting Decorators
- 29.9. Applying Decorators to Built-in Functions
- 29.10. Real World Examples

30. @Property

- 30.1. Define a Property
- 30.2. property() Function
- 30.3. @property - Property as a Decorator
- 30.4. Real World Example
- 30.5. Computed Attributes
- 30.6. Extending a Property in a Subclass

31. Classes and Objects

- 31.1. Create a Class
- 31.2. The __init__() Method
- 31.3. The self Parameter
- 31.4. Attributes
- 31.5. Instance Attribute
- 31.6. Class Attribute
- 31.7. Create an Object
- 31.8. Access and Modify Attributes
- 31.9. Methods
- 31.10. Instance Methods
- 31.11. Delete Attributes and Objects
- 31.12. Constructor
- 31.13. Namespace
- 31.14. Nested class



32. Inheritance

- 32.1. What Is Inheritance?
- 32.2. Analogy
- 32.3. Defining a Base Class
- 32.4. Subclass
- 32.5. Override a Method
- 32.6. Add a Method
- 32.7. The super() Function
- 32.8. Multilevel Inheritance
- 32.9. Hierarchical Inheritance
- 32.10. Multiple Inheritance

33. Polymorphism

- 33.1. Duck Typing
- 33.2. Strong Typing
- 33.3. Method Overloading
- 33.4. Method Overriding
- 33.5. Operator Overriding

34. Abstraction

- 34.1. Abstract class
- 34.2. Interface
- 34.3. Package
- 34.4. Module

35. Exception Handling

- 35.1. What is Exception Handling?
- 35.2. The try and except Block
- 35.3. Catch Multiple Exceptions
- 35.4. The Else Clause
- 35.5. The Finally Clause
- 35.6. Raising an Exception
- 35.7. User-defined Exceptions

File Handling

36. Read/Write Txt Files

- 36.1. Open a File
- 36.2. Specify File Mode
- 36.3. Read a File
- 36.4. Read Lines
- 36.5. Write a File
- 36.6. Write Multiple Lines
- 36.7. Flush Output Buffer

- 36.8. Close a File
- 36.9. Create a New File
- 36.10. Delete a File
- 36.11. Check if File Exists
- 36.12. Random Access

37. Read/Write CSV Files

- 37.1. Open a CSV File
- 37.2. Specify File Mode
- 37.3. Close a CSV File
- 37.4. Read a CSV File
- 37.5. Write to a CSV File
- 37.6. Read a CSV File Into a Dictionary
- 37.7. Write a CSV File From a Dictionary
- 37.8. Use a different delimiter
- 37.9. Handle Comma Within a Data
- 37.10. Catching and Reporting Errors

38. Read/Write Excel Files

- 38.1. Open a Excel File
- 38.2. Close a Excel File
- 38.3. Read a Excel File
- 38.4. Write to a Excel File

39. Date & Time

- 39.1. How to get the current time?
- 39.2. Time Tuple
- 39.3. Getting formatted time
- 39.4. Sleep time
- 39.5. The datetime Module
- 39.6. Creating date object
- 39.7. Comparison of two dates
- 39.8. The Calendar Module
- 39.9. Printing the calendar of whole year

40. Thread (Multitasking)

- 40.1. Thread
- 40.2. Thread Synchronization
- 40.3. Thread Communication
- 40.4. Daemon Thread

41. Modules

- 41.1. Collection Module
- 41.2. OS Module



- 41.3. Math Module
- 41.4. Random Module
- 41.5. Statics Module
- 41.6. Sys Module

42. Database connection (MySQL)

- 42.1. Environment Setup
- 42.2. Database Connection
- 42.3. Creating New Database
- 42.4. Creating Tables
- 42.5. Insert Operation
- 42.6. Read Operation
- 42.7. Update operation
- 42.8. Join operation
- 42.9. Performing Transactions