LAKSHMAN_JAVA_IT_SCHOOL

Core Python

1. Origin of Python

2. Introduction to Python and what is a Python

3. What can we do by using Python

4. Features and versions of Python

5. Different languages used to develop Python

6. Interactive mode and Script mode

7. Interpreter vs Compiler

8. Scripting vs Programming Languages

9. Reasons to learn or work Python

10. Python Indentation

11. Comments and Quotations

12. Python Identifiers and Keywords

13. Variables

a. Assigning values to variables in different ways
b. Print(), type() and id()

14. Reading data from user

15. Working with input function

16. Python data types

17. Type conversions and eval()

Assignment - 1

18. Introduction to Data Structures

19. Stringdata Structure

a. Different ways to create a string
b. String indexing and string slicing
c. string concatenation and string multiplication
d. string unpacking
e. splitting the data in different parts as per user
f. capitalize() and tittle() and split()
g. del, count(), find(), swapcase()
h. reverse(),replace() and sort()
i. string immutable

20. List Data Structure:

a. different ways to create a list
b. creating and working with homogeneous lists
c. creating an working with heterogeneous lists
d. list indexing and list slicing
e. list concatenation and list multiplication
f. generating list by using range function
g. list unpacking and list mutable
h. creating nested lists and indexing nested lists
i. python range() and xrange() functions
j. python insert, append andextend
k. remove, pop and clear
l. python list ascending and descending
m. converting given string data structure into list
n. converting given list data structure into string
o. creating list from user values

21. Tuple Data Structure

a. creating a tuple in different ways
b. creating and working with homogeneous tuple

c. creating and working with heterogeneous tuple
d. tuple indexing and tuple slicing
e. tuple concatenation and tuple multiplication
f. tuple unpacking and tuple immutable
g. all, any, len and sort
h. del keyword
i. python tuple ascending and descending
j. creating and working with nested tuples
k. Conversions:
i. converting given string data structure into tuple
ii. converting given list data structure into tuple
iii. converting given tuple data structure into string
iv. converting given tuple data structure into list
l. advantages of tuple over list data structure

22. Set Data Structure

a. Creating and working with set data structure in different ways
b. Normal sets and frozen sets
c. Set mutable and unpacking set data structure
d. Creating and working with sets with homogeneous elements
e. Creating and working with sets with heterogeneous elements
f. Creating empty sets and modifying the empty sets
g. Why sets not support indexing and slicing
h. Add, remove and discard the elements to set data structure
i. Issubset, issuperset and isdisjoint
j. Union, intersection and defference
k. Intersection_update and defference_update
l. Symmetric_difference and symmetric_difference_update
m. Conversions:
i. Converting given string data structure into set
ii. Converting given list data structure into set
iii. Converting given tuple data structure into set
iv. Converting given set data structure into string
v. Converting given set data structure into list
vi. Converting given set data structure into tuple

23. Dictionary Data Structure

a. Creating and working with dictionary data structure in different ways
b. Creating empty dictionary and working with empty dictionary
c. Working with key and value pairs

d. Dictionary mutable and unpacking dictionary
e. Adding and deleting key and value pairs to the existing data structure
f. Difference between pop and popitem operations
g. Extracting only keys from the existing data structure
h. Extracting only values from the existing data structure
i. Clear and pop methods
j. Del keyword and pop method
k. Creating a dictionary from existing another data structure like tuple
l. FAQs on all Data Structures

Assignment – 2

24. Operators

a. Arithmetic operators
b. Logical operators
c. Assignment operators
d. Comparison operators
e. Bitwise operators
f. Identity operators
g. Membership operators

25. Python Functions and Arguments

a. Defining functions and working with functions
b. Using def keyword for functions
c. Called functions and function definition and calling functions
d. Formal arguments and actual arguments
e. Working with named arguments and keyword arguments
f. Default arguments and positional arguments
g. Working with default arguments and normal arguments
h. *args and **kwargs arguments
i. Argument unpacking
j. Variable length arguments
k. Using data structures to function definitions
l. Nested functions
m. Dir() and Format() functions
n. Enumerate function
o. FAQs on functions and Arguments

26. Lambda Functions

a. Creating functions by using lambda keyword
b. Difference between def and lambda functions
c. Working with filter functions
d. Working with map functions
e. Working with reduce functions

27. Control Statements

a. Simple If statement
b. If else statement
c. Elif statement
d. Nested if statement
e. Membership test for string
f. Membership test for tuple
g. Membership test for list
h. Membership test for set
i. Membership test for dictionary
j. FAQs on control statements

28. Loopings

a. For loop
b. While loop
c. Pass, continue and break statements
d. Iterating over list, tuple, set and dictionary

29. Advanced Concepts on Data Structures

a. List comprehension
b. Dictionary comprehension
c. Nested data structures

Assignment – 3

Advanced Python

30. File Handling

a. Creating a file in a directory
b. Open the file in the python
c. Different ways to open the file in Python

d. Writing to the file
e. Appending the data to the existing file
f. Modes of operations
g. Seek and tell methods
h. Readline and readlines
i. Working with words and characters in the file
j. Real-time scenarios on files
k. Interview based questions on the file

31. OOPS Concepts

a. Class and object
b. Class variables and instance variables
c. Constructor
d. Data hiding
e. Method overloading and overriding
f. Abstraction
g. Inheritance
h. Polymorphism
i. Encapsulation

32. Modules

a. What is module and purpose of modules
b. Different types of modules
c. Different ways to import modules
d. Standard modules and user modules
e. From … import *
f. Creating own modules
g. Using modules in other modules
h. Working with some standard modules
i. MATH, DATETIME, CALENDAR, SYS, OS Modules

33. Exception Handling in Python

a. What is an exception
b. Handling exceptions
c. Try and except block
d. Handling multiple exceptions using multiple excepts
e. Handling multiple exceptions using single except
f. Working with default except

g. Handling exceptions with else and finally blocks
h. Using assert for handling exceptions

34. Logging in python

a. What is logging and purpose of logging
b. Creating a log file
c. Storing runtime events in log file
d. Different modes to store the data in log file
e. DEBUG, INFO, WARNING, ERROR, CRITICAL

35. Iterators, generators and decorators

a. Working with yield keyword
b. Difference between yield and return
c. Decorating a function with another function

36. Unit Testing in Python

a. Importing unittest module
b. Calling all unit test cases
c. Calling specific unit test case
d. assertEqual, assertTrue and assertFalse

37. Regular Expressions

a. Basics of regular expressions
b. Findall function
c. Search function
d. Match methods
i. Group
ii. Groups
e. Matching and searching
f. Compile and sub functions
g. Mobile numbers verifications
h. Email ids verifications
i. Web scrapping

38. Command Line Arguments

a. Reading command line arguments
b. Using command line arguments

39. Working with Database Connection

a. Connecting to database from Python application
b. Creating connection to the database from Python application
c. Creating database and tables from Python applications to the database
d. Fetching data and updating data in the entities.
e. Using cursor to execute SQL command in Python application
f. Using Fetchall and Fetchone methods

Assignment - 4

Advanced Concepts to Python

40. Introduction to Django framework

a. Introduction to Django framework
b. Creating a project and application
c. Urls, models, templates and views files,
d. Introduction to web development
e. Introduction to Pycharm

41. Numpy

a. Main advantages of Numpy arrays over Python lists
b. Creating normal arrays
c. Creating multi-dimensional arrays
d. Creating float type arrays, complex type arrays
e. Creating arrays with placeholders
f. Reshaping existing arrays
g. Creating linspace arrays

42. Scipy

a. Introduction to Scipy

DJANGO

1. Introduction to Python

2. Introduction to Web Development

3. Introduction to HTML

4. Introduction to DJANGO

5. Installing DJANGO

6. DJANGO Architecture

7. MVC and MVT architectures

8. Introduction and Installing PyCharm

9. Creating a Projects and Applications

10. Using DJANGO's Admin

11. Working with Views

12. URL mappings

13. Managing DJANGO's Settings

14. Creating and Activating Models

15. Working with Models

16. Setting up database and filtering database results

17. Creating templates and adding forms to template.

18. Working with User Interface.

19. Working with Admin Interface.

20. Working with Page Redirection.

21. Sending Emails.

22. Form Processing

23. Files and Images uploading

24. Cookies Handling

25. Sessions and Caching

REST API

(DJANGO Rest Framework)

1. Introduction to Django Web Framework

2. Introduction to REST and API

3. Introduction to Restful Webservices

4. Django Rest Framework(DRF)

5. DRF Advantages and Constraints

6. Difference between Django and REST API

7. REST API pre-requisites

8. REST API Configuration and Installation

9. Serialization and Deserialization

10. JSON and XML

11. Action and handler methods

12. Function based and class based views

13. API Views

14. Mixins

a. ListModelMixin

b. CreateModelMixin

c. UpdateModelMixin

d. DestroyModelMixin