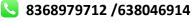


#### **Python Course Content**

#### **Core Python**

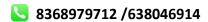
- 1. Introduction to python
- 2. History Of Python
- 3. Advantages & Disadvantages of Python
- 4. Introduction to Interpreter
- 5. Python Installation
- 6. Python Program Execution Flow
- 7. Python Syntaxes
- 8. Introduction to IDE (Eclipse, PyCharam)
- 9. Python Program Development
- 10. Identifiers, keywords
- 11. Basic data types
- 12. Dynamic Typing
- 13. Type conversion functions
- 14. Operators
- 15. Operator Precedence

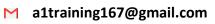






- 16. input/output
- 17. Reading User Input
- 18. Displaying Output
- 19. Formatting Output
- 20. Variables, expressions
- 21. Operators
- 22. Selection control statements (if, if..else)
- 23. range data type, indexing and slicing
- 24. Iterative control statements
- 25. (while, for..in, while...else, for..in .. else)
- 26. Transfer control statements (
  Break & continue)
- 27. Python strings
- 28. List
- 29. Tuple
- 30. **Set**
- 31. Dictionary
- 32. Functions
- 33. Definition and advantages of a function
- 34. Understanding scope

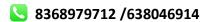


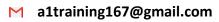






- 35. Default arguments
- 36. Keyword arguments
- 37. Variable length arguments
- 38. Lambda expressions
- 39. Documentation & Annotations
- 40. Types of arguments
- 41. Scope of variables
- 42. Global variables
- 43. Local variables
- 44. Nested functions
- 45. Lambda functions
- 46. Exceptions
- 47. Syntax errors vs Runtime errors
- 48. Handling, Raising exceptions
- 49. Built-in exceptions
- 50. MemoryError
- 51. NameError
- 52. ValueError
- 53. TypeError
- 54. User-defined exceptions
- 55. Clean-up actions



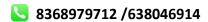






# PYTHON OOPS AND ADVANCED CONCEPTS

- 1. Introduction to OOPS
- 2. OOPS principles
- 3. Classes and Objects
- 4. Instance variables
- 5. Instance methods
- 6. Class variables,
- 7. Class methods
- 8. Constructors
- 9. Inheritance
- 10. Abstract classes
- 11. Inner classes
- 12. Exception handling
- 13. Regular expressions
- 14. File handling
- 15. Accessing (reading / writing) files



M a1training167@gmail.com

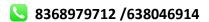




- 16. Serialization
- 17. Multithreading
- 18. Introduction to Database
- 19. Python Database Connectivity
- 20. CRUD Operations with Database
- 21. Unit Testing
- 22. Mini Projects Development
- 23. Debugging

## TRAINIG Django Framework TTJTF

- 1. Introduction to Web Development
- 2. Introduction to Django Framework
- **3. Features of Django**
- 4. Django Installation
- 5. MVC Model
- 6. HTTP Concepts
- 7. Views
- 8. URL Mapping





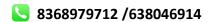


#### **Templates**

- 1. Django Template Language
- 2. Utilities of Templates
- **3. Creating Template Objects**
- 4. Tags, Variables and Filters
- 5. Rendering Templates
- 6. Template Inheritance

# Django Forms (Only Coding)

- 1. Form Handling
- <sub>2.</sub> Form Classes
- 3. Form validations and Error Messages
- 4. Form Display
- 5. Capturing Form data
- 6. Advanced Form Processing







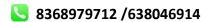
#### **Django Models**

- 1. Django Models
- 2. Model Fields
- 3. Model inheritance
- 4. Primary Keys and the Model
- 5. Dynamic Webpage
- 6. Toggle Hidden Content
- 7. JQuery and Ajax intergreation
- 8. Serialization and Deserialization

### **Templates**



- 2. Utilities of Templates
- 3. Creating Template Objects
- 4. Tags, Variables and Filters
- **5. Rendering Templates**
- 6. Template Inheritance





ly Coding)



#### **Django Forms**

- 1. Form Handling
- 2. Form Classes
- 3. Form validations and Error Messages
- 4. Form Display
- 5. Capturing Form data
- 6. Advanced Form Processing

#### **TOOLS**

1. Git HUB
2. Jenkins
1. TRAINIG
1. INSTITUTE
(Only Coding)

