



Python FullStack Course



FRONTLINES EDUTECH PRIVATE LIMITED



ABOUT US

Frontlines Edutech Private Limited envisions bridging the gap between academia and industry. We understand the pain of unemployment and are trying to wipe off the words of impossible and unemployment from the brains of youth by training and turning them into diamonds. We provide tailor-made, hands-on, and need-based programs on par with industry standards. We believe that the youth of today is the future of tomorrow. And so our courses are tutored by experts from top industries who are passionate about mentoring the youth. Since its inception, Frontlines Edutech has earned the trust and gained recognition from thousands of learners across the state.



**FRONTLINES
EDUTECH**
Building Trust & Careers

FOCUSED ON REFINING TALENT AND TURNING LEARNERS INTO HIGHLY SKILLED PROFESSIONALS SOUGHT AFTER BY TOP COMPANIES.



OUR MISSION AND VISION

At Frontlines Edutech Private Limited, our mission is to bridge the gap between academia and industry by offering practical, customized training programs that equip youth with the skills needed for easy employment. We aim to eliminate unemployment by fostering a mindset of possibility and transforming learners into skilled professionals. With expert industry mentorship, we empower the next generation with the tools to succeed in today's competitive world.

It is in our vision that each young person be prepared with the skill, confidence, and opportunity to succeed in their career. We want to lead an online skills development platform that provides focus on determining the needs of industries through education, so that it would be well-prepared in nurturing professionals who will shape the future of employment. Innovative training eliminates unemployment and helps in unlocking the potential of thousands of learners.





WHY CHOOSE US

We are Success Trainers, Mentors, and Motivators, passionate about helping you reach your full potential with a blend of warmth, wisdom, positivity, and proactivity. Our goal is always your highest good and greatest growth, offering support that is both caring and daring, fun and focused. We ensure learning is not only powerful but also enjoyable, making every step of your growth journey meaningful and engaging.





WHY LEARN THIS COURSE

- This course is for those who want to start their coding journey, whether a beginner or an expert.
- No prior programming knowledge is required.
- This course covers basics to advanced concepts straightforwardly and systematically.
- We take special care of Non-IT students
- Easy to learn
- Beginner Friendly
- The diversity of our learners adds richness to course discussions and interactions.





DELIVERABLES

8.PLACEMENT UPDATES

7.Q&A SESSIONS

6.COURSE COMPLETION CERTIFICATE

5.INTERVIEW GUIDANCE

4.LINKEDIN PROFILE BUILDING

2.RESUME BUILDING

3.DAILY ASSIGNMENTS

9.ON-DEMAND VIDEO COURSE CONTENT

10.DOWNLOADABLE RESOURCES

1.FROM SCRATCH TO MASTER LEVEL TRAINING



TOP COMPANIES HIRING

Infosys

tcs TATA
CONSULTANCY
SERVICES

accenture

cognizant

Capgemini

wipro

IBM

BOX8

KPMG



TOP CITIES FOR PYTHON FULLSTACK DEVELOPER DEMAND

- BANGALORE (BENGALURU)
- GURGAON (GURUGRAM)
- HYDERABAD
- PUNE
- MUMBAI
- CHENNAI
- NOIDA

KEY SECTORS DRIVING DEMAND FOR PYTHON FULLSTACK DEVELOPER

- TECHNOLOGY AND SOFTWARE DEVELOPMENT
- E-COMMERCE HEALTHCARE
- FINANCE AND FINTECH
- HEALTHCARE
- EDUCATION AND EDTECH
- DATA SCIENCE AND ANALYTICS
- ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING
- MEDIA AND ENTERTAINMENT
- STARTUPS AND TECH INCUBATORS



PYTHON FULLSTACK SYLLABUS

BACK END TECHNOLOGIES

CORE PYTHON

- Course Introduction
 - Why Python?
 - Who Uses Python Today?
 - What Can We Do with Python?
 - How Python Developed and Supported
 - Python – Technical Strengths
 - What's Next?
- Python Interpreter
- Program Execution – programmer's view, Python's view
- Installation
 - Python
 - Related Software: PyCharm, Anaconda
 - Setup and Configure Python on Laptop
 - IDLE: UI, Usage, Features
- All Numeric types in Python, coding/Hands-on
- Python Variables, objects, References, Shared References, coding/Hands-on
- Garbage Collection of objects
- All built-in types in Python: Strings, Lists, Dictionaries, Tuples, Sets, Files
- Python Statements - coding/Hands-on
- Assignments, Expressions and Prints
 - if-else, if-elif-else, if-else ternary expression
 - while and for loops
 - Comprehensions vs regular
 - Parallel Traversals: map and zip functions
- Iterations and Comprehensions-coding/Hands-on
- Python online Documentation
- Python Functions – def, nested functions
- Variable Scopes – basics, LEGB rules, global, nonlocal
- Function Arguments - coding/Hands-on
 - Arguments and shared references
 - Arguments passing basics
 - Arguments matching basics
 - Arguments matching syntax
 - Multiple Results
- Advanced Function Concepts – Recursive functions, Attributes, Annotations, lambda
- Generators and comprehensions – Generator functions, yield, Generator expression
- Python Modules
 - Definition, why modules?
 - Typical Python program architecture
 - Import statement - coding/Hands-on
 - How Import works: Find it, Compile it, Run it
 - Standard library modules
 - __pycache__ folder for byte code files
 - Module search path



- Module Coding Basics
 - Module creation
 - import statement, from statement, from * statement
 - Module Namespaces, Namespace dictionaries: __dict__
 - Reloading modules
- Module Packages
 - Package import basics
 - Why Package imports?
 - Relative import basics
 - Why relative imports?
 - Package Namespaces
- Advanced Module Topics
 - Data hiding in modules - coding/Hands-on
 - Mixed usage modes: name and main - coding/Hands-on
 - The as extension for import and from - coding/Hands-on
- Introduction to Python Classes- coding/Hands-On
 - Why Classes?
 - Classes, constructors and Instances
 - Method calls
 - Attribute inheritance search
 - OOP is about code reuse
 - Subclassing by Inheritance
 - Polymorphism in Action
 - Class vs instance attributes
 - Storing objects in DB – Pickles & Shelves
- Coding with Classes - coding/Hands-on
 - Abstract super classes
 - Nested classes
 - Classes vs Modules
 - Namespace dictionaries
 - LEGB scopes rule revisited
- Operator Overloading- coding/Hands-on
 - Constructors: __init__
 - Indexing, Slicing: __getitem__ and __setitem__
 - Attribute Access: __getattr__ and __setattr__
 - String Representation: __repr__ and __str__
 - Right side and In-Place Uses: __radd__ and __iadd__
 - Call Expressions: __call__
 - Comparisons: __lt__, __gt__ and others
 - Boolean Tests: __bool__ and __len__
 - Destructors: __del__
- Special Features of Classes
 - Inheritance: “IS-a” relationship
 - Composition: “HAS-a” relationship
 - Pseudo private class attributes
 - Bound and unbound method objects
 - Class objects
 - “Mix-In” classes
- Advanced Class Topics
 - “New style” class model
 - Diamond inheritance change
 - MRO: Method Resolution Order
 - Slots: Attribute Declarations
 - Properties: Attribute Accessors
 - Static and Class methods
 - The “super” built-in function
- Exception Basics- coding/Hands-on
 - Why Exceptions?
 - Default Exception handler
 - Catching Exceptions
 - Raising Exceptions



- Coding Exceptions - coding/Hands-on
 - The try/except/else statement
 - try/finally statement
 - raise statement
 - assert statement
 - with/as context managers
 - Nesting Exception Handlers
- Exception Objects
 - Class based exceptions
 - Why Exception hierarchies?
 - Built-in Exception Classes
 - Custom Exceptions
- Regular Expressions with Python
 - What are regular expressions?
 - regex module in python
 - The match Function
 - The search Function
 - Matching vs searching
 - Search and Replace Meta characters, advanced patterns

DJANGO FRAMEWORK

- Django Overview, Installation
- Web Application Architecture
- Creating and Setting Up a Virtual Environment
- Project and Apps in Django
- First Project Creation and Apps Creation, App Registration
- URL Patterns, Creating URLs, Mapping URLs
- path, include in URL Mappings
- Django Templates, Create Templates
- DTL – Django Template Language
- Variables, Filters, Tags in Django Templates
- Template Inheritance
- Partial Templates

- Static Files (CSS, Images, etc.)
- Including Static Files in Templates
- Django-DB Connection Settings Configuration
- Models, Defining & Migrating Models
- Django Model Fields, Field Options
- Admin Panel, Creating Superuser
- Model Registration to Admin Panel
- Django Python Shell
- All CRUD Operations with Models
- Bulk CRUD Operations with Models
- Types of DB Relationships in Django
- Build One to Many DB Relationship Code
- Build One to One DB Relationship Code
- Build Many to Many DB Relationship Code
- Building HTML Forms, Django Forms, Model Forms
- Validating Fields in All Forms
- Validators in Django
- CSRF Token
- Views in Django
- Function-Based Views
- Class-Based Built-in Views
- File Uploading
- Sending Emails with and without Django

DJANGO REST FRAMEWORK

- Why REST Framework
- REST APIs
- REST API Interaction with React JS
- JSON Data
- DRF Serializers
- DRF Viewsets in Detail
- DRF Routers in Detail



- DRF Permissions
- DRF Mixins in Detail
- DRF Authentication
- Token Authentication
- DRF Sessions
- DRF Settings

DATABASE

- MySQL Installation, HeidiSQL Installation
- What is RDBMS?
- RDBMS Terminology
- Python MySQL Database Access
- Create Database Connection
- DDL Operations with Databases
- Accessing Database
- DQL Operations with Databases
- DML Operations with Databases
- Aggregations in Database
- Table Joins
- Interacting with DB Using Python Libraries
- Installation of Python Libraries for DB Access

FRONTEND SYLLABUS

HTML & CSS SYLLABUS

- Module 1: HTML Learning Objectives
 - HTML Introduction - Basic Tags
 - HTML Heading Tags
 - HTML Image Tags
 - HTML Attributes
 - Input Tags and Attributes
 - HTML List Items
 - Meta Tags

- Module 2: CSS Learning Objectives
 - CSS Styles
 - CSS Box Model
 - CSS Selectors
 - CSS Combinators
 - CSS Specificity
 - CSS Flex
 - CSS Grid
 - CSS Positions
 - CSS Units
 - CSS Media Queries

JAVASCRIPT SYLLABUS

- Module 1: JavaScript Level-1
 - Variables
 - Mathematical Operators
 - Relational Operators
 - Comparison Operators
 - Conditional Statements
 - Logical Operators
 - Ternary Operator
 - While Loops
 - For Loops
 - Arrays Basics
 - Strings Basics
 - Nested Loops
 - Objects – I
 - Objects - II
 - Functions
 - Inbuilt Functions
 - Object Methods
 - Higher Order Functions
 - ES6-I
 - CMD
 - Git and GitHub



- Module 2: JavaScript Level-2
 - DOM Manipulation
 - DOM Traversal
 - DOM Events
 - Event Handling
 - DOM Attributes
 - Local Storage
 - Sorting Techniques
 - Synchronous vs Asynchronous JS
 - How JS Works
 - Web APIs
 - Event Loop
 - Promises
 - Fetch
- Module 3: JavaScript Level-3 Learning Objectives (Advanced Topics)
 - Different Loops
 - Arrow Functions
 - Object Shorthand
 - Statements vs Expressions
 - Lexical Scope
 - let vs var vs const
 - Destructuring Arrays and Objects
 - Hoisting
 - IIFE
- this Keyword
- call vs apply vs bind
- Classical Inheritance
- Prototype Chain
- Creating Our Own Prototype Chains
- Object.assign vs Object.setPrototypeOf
- Chaining with Functions Prototype Object
- Object Creation Using Factory Functions, Constructor Functions & Classes
- Prototypal Inheritance Using Factory Functions, Constructor Functions & Classes
- new Keyword Related Problems
- DOM - CRUD Operations
- DOM - JWT Authentication
- Event Bubbling
- Event Capturing
- DOM - Pagination
- JEST Testing
- Writing Clean Code
- HTTP Methods
- React Intro and Essentials





FRONTLINES EDUTECH

Building Trust & Careers

Follow Us



FRONTLINES MEDIA



FRONTLINES EDUTECH



FRONTLINES MEDIA-IGNITED-MINDS



For Enquiry Contact Us:



📞 +(91)-83330 77727

✉️ media.frontlines@gmail.com

🌐 www.frontlinesedutech.com