

PYTHON Fullstack Training

Course Duration: 5 months, Weekly 6 days, Each Class: 1 hr.

Trainer: D. Chaitanya (IIT Roorkee Alumni)

a) **Back End Technologies:** ~ 3.5 month

I) **Core Python:**

- Demo:
 - i. Why Python?
 - ii. Who Uses Python today
 - iii. What can we do with Python
 - iv. How Python Developed and Supported
 - v. Python – Technical Strengths.
 - vi. What next?
- Python Interpreter
- Program Execution – programmer's view, Python's view
- Installation
 - i. Python
 - ii. All Related Software: PyCharm, VS Code, Jupyter Notebook
 - iii. Setup, configure Python in Laptop
- 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
 - i. if-else, if-elif-else, if-else ternary expression
 - ii. while and for loops
 - iii. Comprehensions vs regular
 - iv. Parallel Traversals: map and zip functions
 - v. Other important functions: range, len, enumerate
- 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

- i. Arguments and shared references
 - ii. Arguments passing basics
 - iii. Arguments matching basics
 - iv. Arguments matching syntax
- Advanced Function Concepts – Attributes, lambda functions
- Generators and comprehensions
 - i. Generator functions, yield statement
 - ii. Generator expressions
- Python Modules
 - i. Definition, why modules?
 - ii. Typical Python program architecture
 - iii. Import statement - coding/Hands-on
 - iv. How Import works: Find it, Compile it, Run it
 - v. Standard library modules
 - vi. `__pycache__` folder for byte code files
 - vii. Module search path
- Module coding Basics
 - i. Module creation
 - ii. import statement, from statement, from * statement
 - iii. Module Namespaces, Namespace dictionaries: `__dict__`
- Module Packages
 - i. Package import basics
 - ii. Why Package imports?
 - iii. Relative import basics
 - iv. Why relative imports?
- Advanced Module Topics
 - i. Mixed usage modes: `__name__` and `__main__` - coding/Hands-on
 - ii. The as extension for import and from - coding/Hands-on
- Introduction to Python Classes - coding/Hands-on
 - i. Why Classes?
 - ii. Classes, constructors and Instances
 - iii. Method calls
 - iv. Attribute inheritance search
 - v. OOP is about code reuse
 - vi. Subclassing by Inheritance

- vii. Polymorphism in Action
- viii. Class vs instance attributes
- ix. Coding with Classes – coding/Hands-on
- x. Abstract Super classes
- xi. Nested classes

- Operator Overloading - coding/Hands-on
 - i. Constructors: `__init__`
 - ii. Indexing, Slicing: `__getitem__` and `__setitem__`
 - iii. Attribute Access: `__getattr__` and `__setattr__`
 - iv. String Representation: `__repr__` and `__str__`
 - v. Right side and In-Place Uses: `__radd__` and `__iadd__`
 - vi. Call Expressions: `__call__`
 - vii. Comparisons: `__lt__`, `__gt__` and others
 - viii. Boolean Tests: `__bool__` and `__len__`
 - ix. Destructors: `__del__`

- Special features of Classes - coding/Hands-on
 - i. Inheritance: “IS-a” relationship
 - ii. Composition: “HAS-a” relationship
 - iii. Class objects

- Advanced Class Topics - coding/Hands-on
 - i. “New style” class model
 - ii. Diamond inheritance change
 - iii. MRO: Method Resolution Order
 - iv. Static and Class methods
 - v. The “super” built-in function

- Exception Basics - coding/Hands-on
 - i. Why Exceptions?
 - ii. Default Exception handler
 - iii. Catching Exceptions
 - iv. Raising Exceptions

- Coding Exceptions - coding/Hands-on
 - i. The try/except/else statement
 - ii. try/finally statement
 - iii. raise statement

- iv. assert statement
- v. with/as context managers
- vi. Nesting Exception Handlers

- Exception Objects
 - i. Class based exceptions
 - ii. Why Exception hierarchies?
 - iii. Built-in Exception Classes

II) **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

III) **Django Framework:**

- Django overview, installation, Setup
 - i. Creating and setup a virtual environment
 - ii. Django Project, Apps, Create App
 - iii. Django Server, run Server
 - iv. URL Mapping
- URL Routings & Views
 - i. Creating views
 - ii. Return String response
 - iii. Return Http response
 - iv. Static redirect, Dynamic Redirect
- Templates & Static Files
 - i. Template system
 - ii. Return HTML content
 - iii. Return HTML file – render_to_string, render
- Django Template Language (DTL)
 - i. DTL Filters
 - ii. DTL Tags

- Templates
 - i. Template basics
 - ii. Template inheritance
 - iii. Partial templates
 - iv. 404 template
 - v. Static files
 - vi. Adding App specific static files to template
 - vii. Adding Global static files to template
- Data & Models
 - i. Accessing DB without Django
 - ii. Accessing DB with Django
 - iii. Project settings (settings.py)
 - iv. Models, Create Model
 - v. Model Field Types
 - vi. Migrate Model
 - vii. Django official Documentation
 - viii. Basic CRUD operations
 - ix. Model Field Lookups
 - x. Changing Models
 - xi. Query performance
 - xii. Bulk CRUD operations
 - xiii. Aggregations, ordering
- Admin interface
 - i. Django Admin Panel
 - ii. Super User, Login to Admin Panel
 - iii. Manage DB data from Admin panel
 - iv. Configuring filters, display in Admin panel
- DB Relationships
 - i. One to Many relationship
 - ii. One to one relationship
 - iii. One to many relationship
 - iv. Manage all relationships in Admin panel
- Forms
 - i. GET, POST requests
 - ii. Submit HTML Form data
 - iii. Django Forms
 - iv. Create, validate Django Forms
 - v. Form Validation
 - vi. Insert data into DB with Form Submission

- vii. Model Forms
- viii. Create, update Model Forms
- ix. Manage all relationships in Admin panel
- Views
 - i. Class based views
 - ii. Template View
 - iii. List View
 - iv. Detail View
 - v. Form View
 - vi. Create View
- File Uploads, Emails, React JS
 - i. Upload file in HTML form, Django Form, Model Form
 - ii. Access Files
 - iii. Sending mails without Django
 - iv. Sending mails with Django
 - v. Setup Django to React JS
 - vi. React JS request, Django Response
- Django REST Framework
 - i. Django REST Framework – Install and configure
 - ii. HTML based vs React based web applications
 - iii. Create and Run React JS Project
 - iv. Develop and Use API
 - v. GET, POST, PUT, DELETE requests
 - vi. Mixins, API Views, Viewsets and Routers
 - vii. Authentication, tokens
 - viii. Register User (with base model)
 - ix. Register User (with customization)
 - x. User Login, User Model

IV) Database:

- MySQL Installation, Heidi SQL 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 DB with Python using python libraries.
- Installation of Python libraries for DB access.

b) **Front End Technologies:** ~ 1.5 month

I) **HTML5:**

- Introduction to HTML
- Basic Structure of HTML
- HTML Editors
- HTML Tags
- Paragraphs, Headings and Text
- Formatting Tags
- HTML Lists
- HTML Images
- HTML Tables
- HTML Forms
- HTML Media

II) **CSS3:**

- Introduction to CSS
- Types of CSS
- CSS Properties, Selectors and Values
- Applying CSS to HTML
- CSS colors
- CSS Box Model, Margins, Padding,
- Borders
- CSS Text & Font Formats
- CSS Advanced Topics (Effects, Animations, Shadows, Etc.,)

III) **Javascript:**

- Introduction to JavaScript
- How to Apply JavaScript

- Displaying Output in JavaScript
- Understanding JavaScript Syntax
- Variables & Datatypes
- Operators
- Math and String Manipulations
- Conditional and looping Statements
- Functions
- Validations
- Events
- ES6 Features, http API call

IV) Bootstrap:

- Introduction to Bootstrap
- Bootstrap Setup
- Bootstrap Containers
- Bootstrap Grids
- Bootstrap Tables
- Bootstrap Buttons, Navbars, Alerts
- Bootstrap Carousel
- Bootstrap Forms

V) React JS:

- Introduction to React
- Components
- Props and State
- Handling Events
- Conditional Rendering
- Hooks
- Routing and Navigation
- API Integration
- State Management (Redux)
- Styling in React
- Optimizing Performance
- Forms and Validation
- Testing React Applications
- Deployment

- Advanced Topics (Optional)
- Real-world Projects
- Best Practices and Code Quality