







# Python Full Stack Development Training Program

---

**Duration: 6 Months**

## Course Objectives:

-  Gain the essential skills to successfully clear the **Python certification exam** on your **first attempt**.
-  Boost your **resume** with a recognized certification that enhances career opportunities.
-  Develop strong **Python programming expertise**, increasing job prospects and potential salary growth.
-  Build a **strong professional presence on platforms like LinkedIn**, positioning yourself as a sought-after candidate in the industry.

## Course Description:

Python is a **powerful, high-level programming language** known for its simplicity, versatility, and efficiency. Its ease of use makes it ideal for both beginners and experienced developers, allowing seamless execution across different platforms.

With tools like **Py2exe** and **PyInstaller**, Python code can be converted into standalone applications. The **Django framework**, built on the **MVC architecture**, is widely used for developing robust web applications, supported by the **Django Software Foundation**.

Additionally, frameworks such as **Flask**, **Web2py**, and **Pyramid** offer efficient solutions for building scalable applications. Technologies like **Pyjamas** and **IronPython** extend Python's capabilities to client-side **AJAX-based applications**, making it a preferred choice in modern development.

## Bright Mentors Training Program Deliverables:

- **Recorded Sessions** – Get access to high-quality **video and voice recordings** for convenient learning.
- **Hands-on Learning** – Work on **real-world projects, coding exercises, and assignments** for a deeper understanding.

## Additional Program Benefits

- ◆ **Mock Certification Tests** – Access structured **practice exams** to boost confidence and ensure success in the **final certification exam**.
- ◆ **Industry Case Studies** – Learn through **real-world case studies**, enhancing **critical thinking and problem-solving** abilities.
- ◆ **Project-Based Learning** – Work on **live projects** and **real-time industry applications** to apply concepts practically.
- ◆ **Hands-On Training** – Experience an **interactive learning approach** with coding exercises and project work for deeper understanding.
- ◆ **Guidance from Experts** – Learn from **seasoned IT professionals** with **10+ years of industry experience** in software development and technology.
- ◆ **Career Support & Resume Building** – Get expert guidance on **resume writing, interview preparation, and job placement assistance**

## Curriculum Overview:

### **Module 1: Introduction to Web Technologies**

---

- ✓ Understanding **What is the Web?**
- ✓ Exploring **Key Features of the Web**
- ✓ Overview of **W3C and Its Role in Web Standards**
- ✓ Introduction to **WHATWG and Modern Web Development**

### **Module 2: HTML Fundamentals**

---

This module provides a comprehensive understanding of the core concepts of HTML, its structure, and key components essential for web development.

- ✓ **Introduction to HTML** – Overview of HTML and its role in web development
- ✓ **Structure of an HTML Document** – Understanding the fundamental parts of an HTML document

- ✓ **HTML Versioning** – Evolution of HTML and different versions
- ✓ **Head Section** – Key elements within the <head> section
- ✓ **Meta Information** – Importance of metadata for SEO and browser compatibility
- ✓ **Favicons** – Implementing website icons for branding
- ✓ **Body Section** – Organizing and structuring content within the <body> tag
- ✓ **HTML Forms** – Creating interactive forms for user input and data collection
- ✓ **Lists and Tables** – Structuring content using ordered/unordered lists and tables
- ✓ **Multimedia Elements** – Embedding audio, video, and other media formats
- ✓ **Anchors & Images** – Adding hyperlinks and integrating images for enhanced user experience

## **Module 3: Advanced HTML5**

---

This module explores the modern capabilities of HTML5, including new structural elements, enhanced multimedia support, and semantic improvements.

- ✓ **Introduction to HTML5** – Overview of the latest version of HTML and its importance
- ✓ **History and Evolution of HTML5** – Understanding the transition from HTML4 to HTML5
- ✓ **Significance of HTML5** – Why HTML5 is essential for modern web applications
- ✓ **Key Features and Functional Enhancements** – Overview of major improvements in HTML5
- ✓ **Structure of an HTML5 Document** – Understanding the new and optimized document structure
- ✓ **Advanced Features of HTML5** – Exploring APIs, geolocation, local storage, and offline capabilities
- ✓ **Semantic HTML & Block-Level Elements** – Enhancing accessibility and SEO with semantic tags

- ✓ **New Form Elements and Attributes** – Improved input types and validation techniques
- ✓ **Multimedia Integration** – Implementing <audio> and <video> elements for seamless media playback
- ✓ **Canvas and SVG Graphics** – Creating dynamic graphics and animations using HTML5

## **Module 4: Core CSS – Fundamentals of Styling**

---

This module provides a foundational understanding of CSS (Cascading Style Sheets) and its role in enhancing the appearance and layout of web pages.

- ✓ **Introduction to CSS** – Overview of CSS and its significance in web design
- ✓ **CSS Basics** – Fundamental concepts, including inline, internal, and external styles
- ✓ **CSS Syntax & Structure** – Understanding CSS rules, properties, & values
- ✓ **Evolution of CSS** – Overview of different CSS versions and their advancements
- ✓ **Selectors, IDs, and Classes** – Targeting HTML elements effectively for styling
- ✓ **CSS Styling Techniques** – Applying styles to various elements for a visually appealing UI
- ✓ **Background Styling** – Configuring background colors, images, gradients, and patterns
- ✓ **Text Styling** – Customizing text alignment, spacing, shadows, and other attributes
- ✓ **Font Customization** – Implementing different fonts, Google Fonts, and typography techniques
- ✓ **CSS Borders & Outlines** – Enhancing elements with borders, rounded corners, and shadows
- ✓ **Box Model & Layouts** – Understanding padding, margins, borders, and positioning techniques

## **Module 5: Advanced CSS – Modern Styling Techniques**

---

This module delves deeper into advanced CSS concepts, enabling developers to create dynamic, responsive, and visually engaging websites.

- ✓ **Introduction to Advanced CSS** – Exploring the power of modern CSS techniques
- ✓ **CSS3 Modules** – Understanding the modular nature of CSS3 for improved development
- ✓ **Advanced Selectors** – Utilizing attribute selectors, pseudo-classes, and pseudo-elements
- ✓ **Box Model in Depth** – Mastering content-box vs. border-box and flexible layouts
- ✓ **Enhanced Backgrounds & Borders** – Applying advanced gradients, border effects, and patterns
- ✓ **Text Effects & Typography** – Implementing shadowing, word wrapping, and custom fonts
- ✓ **2D & 3D Transformations** – Rotating, scaling, skewing, and translating elements for interactive UI
- ✓ **Core & Advanced Animations** – Utilizing keyframes, transitions, and animations for smooth effects
- ✓ **Multi-Column Layouts** – Structuring content with CSS grid and flexbox for responsive designs
- ✓ **User Interface Enhancements** – Designing intuitive UI elements, hover effects, and dynamic interactions
- ✓ **Media Queries & Responsive Design** – Creating adaptable layouts for different screen sizes

## **Module 6: Core JavaScript – Fundamentals of Client-Side Scripting**

---

This module provides a comprehensive introduction to JavaScript, covering its fundamental concepts and essential programming techniques used for web development.

- ✓ **Introduction to Scripting** – Understanding scripts and their types (client-side vs. server-side)
- ✓ **Overview of JavaScript** – Role and significance of JavaScript in web development
- ✓ **JavaScript Syntax & Structure** – Writing clean and efficient JavaScript code

- ✓ **Comments in JavaScript** – Implementing single-line and multi-line comments for code readability
- ✓ **Popup Boxes & Alerts** – Using alert(), confirm(), and prompt() for user interactions
- ✓ **Variables & Data Types** – Declaring variables and understanding primitive and reference data types
- ✓ **Operators in JavaScript** – Arithmetic, comparison, logical, and assignment operators
- ✓ **Functions & Events** – Defining functions, function expressions, and handling events
- ✓ **Conditional Statements** – Implementing if, else if, switch-case statements for decision making
- ✓ **Looping Constructs** – Using for, while, and do-while loops for iteration control
- ✓ **Working with Arrays** – Creating and manipulating arrays for efficient data handling

## **Module 7: Advanced JavaScript – Enhancing Web Interactivity**

---

This module focuses on advanced JavaScript concepts, including error handling, object-oriented programming, and browser interactions, enabling developers to build more interactive and dynamic applications.

- ✓ **Understanding JavaScript Errors** – Types of errors (syntax, runtime, logical) and debugging techniques
- ✓ **Exception Handling** – Using try, catch, finally, and throw statements for robust error management
- ✓ **JavaScript Objects & Prototypes** – Creating and manipulating objects, constructors, and prototypes
- ✓ **Browser Object Model (BOM)** – Interacting with browser properties, history, navigator, and screen objects
- ✓ **Document Object Model (DOM) Manipulation** – Selecting, modifying, and updating HTML elements dynamically
- ✓ **Event Handling in JavaScript** – Adding event listeners for user interactions (click, hover, keydown, etc.)

- ✓ **JavaScript Form Validations** – Implementing client-side form validation to enhance user input security
- ✓ **Timers & Intervals** – Utilizing `setTimeout()` and `setInterval()` for executing functions at intervals
- ✓ **Asynchronous JavaScript** – Understanding callback functions, promises, and `async/await` for non-blocking operations

## **Module 8: Introduction to Python**

---

This module provides a comprehensive introduction to Python, covering its history, features, installation process, and basic programming concepts.

- ✓ **Introduction to Python** – Understanding Python and its significance in modern programming
- ✓ **What is Python?** – Overview of Python as a high-level, interpreted, and versatile programming language
- ✓ **History and Evolution of Python** – Development timeline and key milestones in Python's growth
- ✓ **Python Versions** – Exploring different versions of Python and their enhancements
- ✓ **Key Features of Python** – Simplicity, readability, cross-platform compatibility, and extensibility
- ✓ **Installing Python** – Step-by-step guide to installing Python on various operating systems
- ✓ **Setting Up Python with Different IDEs** – Installing and configuring Python with various Integrated Development Environments (IDEs)
- ✓ **Writing Your First Python Program** – Basics of Python scripting and running a simple program
- ✓ **Printing Output to the Screen** – Using the `print()` function for displaying output
- ✓ **Reading User Input from the Keyboard** – Capturing input using the `input()` function
- ✓ **Executing Python Programs via Command Prompt and GUI-Based IDEs** – Running Python scripts in different environments

## Module 9: Different Execution Modes in Python

---

This module explores various modes of running Python scripts, working with IDEs, and debugging techniques.

- ✓ **Executing Python Scripts** – Understanding different ways to run Python programs
- ✓ **Interactive Mode Execution** – Running Python commands interactively in the Python shell
- ✓ **Script Mode Execution** – Writing and executing Python scripts from a file
- ✓ **Python Comments** – Using single-line and multi-line comments for code documentation
- ✓ **Cross-Platform Compatibility** – Running Python on Unix, Linux, Windows, macOS, and Android
- ✓ **Exploring New Python IDEs** – Overview of modern Integrated Development Environments for Python
- ✓ **Working with PyCharm IDE** – Setting up and using PyCharm for efficient Python development
- ✓ **Components of PyCharm** – Understanding the user interface and key features of PyCharm
- ✓ **Debugging in PyCharm** – Using breakpoints, inspecting variables, and troubleshooting errors
- ✓ **Sublime Text IDE for Python** – Configuring Sublime Text as a lightweight Python development tool
- ✓ **Introduction to PIP** – Understanding and using Python's package manager (pip) for installing and managing libraries

## Module 10: Variables and Data Handling in Python

---

This module covers the fundamentals of variables, constants, data types, operators, and string manipulation in Python.

- ✓ Understanding Variables and Constants in Python
- ✓ Standard Data Types and Type Conversion
- ✓ Operators and Operands in Python
- ✓ Swapping Variables Efficiently
- ✓ String Handling Techniques

## **Module 11: Conditional Statements in Python**

---

This module explores Python's decision-making structures, enabling efficient control flow in programs.

- ✓ Implementing if, if-else, and if-elif-else Statements
- ✓ Handling Multi-Way and Nested Conditions
- ✓ Optimizing Conditional Execution
- ✓ Understanding When the else Condition is Unused

## **Module 12: Iteration and Looping in Python**

---

This module explores Python's looping constructs for efficient repetition and control flow.

- ✓ Implementing while and for loops for iterative execution
- ✓ Utilizing loops for non-numeric sequences
- ✓ Controlling loop behavior with break and continue statements
- ✓ Enhancing loops using the enumerate() function

## **Module 13: Lists in Python – Dynamic Data Handling**

---

This module covers the fundamentals of Python lists, their properties, and essential operations.

- ✓ Understanding list mutability and indexing
- ✓ Iterating through lists efficiently
- ✓ Performing slicing and advanced list operations
- ✓ Exploring built-in list methods and functional tools like map(), filter(), and reduce()

## **Module 14: Working with Tuples in Python**

---

This module delves into tuples, their advantages, and their applications in data structures.

- ✓ Comparing tuples with lists and their immutability benefits
- ✓ Packing, unpacking, and nested tuple structures

- ✓ Using tuples as dictionary keys and membership testing
- ✓ Implementing tuple slicing and deletion techniques

## **Module 15: Python Sets – Unordered Collections & Operations**

---

This module explores Python sets, their properties, and various operations for handling unique data elements.

- ✓ Creating and managing sets in Python
- ✓ Iterating through set elements efficiently
- ✓ Utilizing built-in set methods for manipulation
- ✓ Performing set operations such as union, intersection, and difference
- ✓ Applying built-in functions with sets
- ✓ Understanding and implementing frozen sets for immutable collections

## **Module 16: Python Dictionaries – Key-Value Data Management**

---

This module covers Python dictionaries, their structure, and methods for efficient data handling.

- ✓ Creating and working with dictionaries in Python
- ✓ Understanding hashing and its role in dictionary performance
- ✓ Utilizing dictionary methods for modification and retrieval
- ✓ Copying and updating dictionary entries
- ✓ Removing keys and managing dictionary elements
- ✓ Sorting dictionaries and retrieving items using built-in functions

## **Module 17: Python Functions – Defining, Calling, & Optimizing Code**

---

This module explores the fundamentals of functions in Python, covering their definition, usage, and advanced concepts for efficient programming.

- ✓ Understanding functions and their significance in Python
- ✓ Defining and invoking functions effectively
- ✓ Exploring different types of functions
- ✓ Importance of indentation in Python function blocks

- ✓ Returning values from functions and handling return types
- ✓ Types of function arguments: default, non-default, keyword, and arbitrary arguments
- ✓ Rules and best practices for function definitions
- ✓ Scope and lifetime of variables in functions
- ✓ Lambda (anonymous) functions for concise coding
- ✓ Functional programming with map(), filter(), and reduce()
- ✓ Using docstrings for function documentation
- ✓ Advanced concepts: iterators, generators, closures, and decorators

## **Module 18: Advanced Python – Error Handling and Exception Management**

---

This module delves into Python's error-handling mechanisms to ensure smooth and robust program execution.

- ✓ Understanding Python exceptions and runtime errors
- ✓ Identifying common errors and handling abnormal program termination
- ✓ Exception hierarchy and significance in error resolution
- ✓ Implementing structured exception handling with try-except blocks
- ✓ Using try-except-else for conditional error handling
- ✓ Ensuring final execution with try-finally blocks for resource management

## **Module 19: Python Classes and Objects – Object-Oriented Programming (OOP) Concepts**

---

This module introduces Object-Oriented Programming (OOP) in Python, covering its principles and implementation.

- ✓ Overview of Object-Oriented Programming (OOP)
- ✓ Understanding OOP principles and their significance
- ✓ Core concepts of objects and classes in Python
- ✓ Working with access modifiers for data protection
- ✓ Defining and structuring Python classes
- ✓ Understanding the self variable in class methods
- ✓ Exploring inheritance and its types
- ✓ Implementation and functionality of inheritance in Python

## Module 20: Python Regular Expressions – Pattern Matching & String Processing

---

This module explores the use of regular expressions in Python for text processing and pattern matching.

- ✓ Introduction to regular expressions and their applications
- ✓ Understanding the syntax and structure of regular expressions
- ✓ Exploring different pattern-matching techniques
- ✓ Using literal characters and special sequences for advanced searches

## Module 21: Bootstrap – A Powerful Mobile-First Frontend Framework

---

This module covers Bootstrap, a responsive front-end framework used for creating visually appealing and mobile-friendly web designs.

- ✓ Fundamentals of responsive web design and its importance
- ✓ Implementing Bootstrap's typography features
- ✓ Creating tables, buttons, dropdowns, and navigation bars
- ✓ Handling images and responsive utilities in Bootstrap
- ✓ Using Bootstrap glyph icons for enhanced UI design

## Module 22: Bootstrap Grid System – Structuring Responsive Layouts

---

This module explains the Bootstrap Grid System, essential for designing responsive and adaptive layouts.

- ✓ Introduction to grids and their role in web design
- ✓ Understanding Bootstrap's flexible grid system
- ✓ Implementing the **mobile-first strategy** for responsiveness
- ✓ Structuring layouts with Bootstrap's grid system
- ✓ Utilizing media queries for adaptive design across devices

## **Module 23: Bootstrap Grid System – Advanced Layout Techniques**

This module covers advanced grid options in Bootstrap for creating flexible and responsive web layouts.

- ✓ Managing responsive column resets for dynamic designs
- ✓ Utilizing offset columns for precise alignment and spacing
- ✓ Implementing nested columns for complex grid structures

## **Module 24: Django Web Framework – Building Scalable Web Applications**

This module introduces Django, a high-level Python framework for rapid web development.

- ✓ Understanding frameworks and their role in web development
- ✓ Overview of Django and its core design principles
- ✓ Evolution and history of Django as a robust framework
- ✓ Key features and advantages of using Django
- ✓ Setting up the development environment for Django
- ✓ Configuring a web server for Django applications
- ✓ Understanding the MVC (Model-View-Controller) pattern
- ✓ Comparing MVC and MVT (Model-View-Template) architecture
- ✓ Exploring Django's MVT pattern for efficient web development

## **Module 25: Introduction to Django – Setting Up Your First Project**

This module provides a step-by-step guide to creating and configuring a Django project.

- ✓ Building your first Django project from scratch
- ✓ Integrating the project with Sublime Text for efficient development
- ✓ Understanding the Django project structure and its components
- ✓ Running the development server and managing configurations
- ✓ Handling migrations and resolving common issues
- ✓ Configuring the database for seamless data management
- ✓ Setting up and customizing project settings

## **Module 26: Developing a Django Application – Models, Views & Admin Interface**

---

This module explores the structure of a Django application and its core components.

- ✓ Understanding Django's application structure and workflow
- ✓ Exploring the **Model Layer** and its role in database management
- ✓ Defining models and working with model fields
- ✓ Querying data using Django's **QuerySet API**
- ✓ Configuring and utilizing Django's built-in **Admin Interface**
- ✓ Setting up and managing database migrations
- ✓ Understanding the **Views Layer** and handling HTTP requests
- ✓ Creating simple views, including a basic "Hello World" output
- ✓ Implementing functional views and class-based views for scalable development

## **Module 27: Django URL Mapping – Managing Navigation and Routing**

---

This module focuses on structuring and handling URLs in Django to create a well-organized web application.

- ✓ Structuring and managing URLs efficiently in Django projects
- ✓ Understanding the significance of URLs in web development
- ✓ Configuring and handling working URLs
- ✓ Implementing forms for user input and data submission
- ✓ Passing parameters from URLs to views for dynamic content
- ✓ Introduction to the **Templates Layer** for rendering HTML pages
- ✓ Utilizing the render() function to generate web responses

## **Module 28: Django Template Language (DTL) – Dynamic Content Rendering**

---

This module explores Django's Template Language, enabling the creation of flexible and reusable UI components.

- ✓ Understanding the role of the **template layer** in Django applications
- ✓ Using filters, tags (if, for), blocks, and extend tags for template customization
- ✓ Implementing comment tags and structuring templates effectively
- ✓ Creating and extending base templates for modular and maintainable UI design

## **Module 29: Django Models – Database Interaction & Data Management**

---

This module covers Django's **ORM (Object-Relational Mapping)** and model management for handling data.

- ✓ Defining and creating database models in Django
- ✓ Performing CRUD (Create, Read, Update, Delete) operations efficiently
- ✓ Establishing relationships between models for structured data management
- ✓ Implementing page redirection in Django for seamless user navigation

## **Module 30: Django Email Integration – Sending and Managing Emails**

---

This module covers email functionality in Django, enabling applications to send automated and customized emails.

- ✓ Sending basic emails using Django's built-in email system
- ✓ Sending bulk emails efficiently with the `send_mass_mail` option
- ✓ Formatting and sending HTML-based emails
- ✓ Attaching files and media to HTML emails

## **Module 31: Django Forms, File Handling & Cookie Management**

---

This module focuses on form processing, file uploads, and session handling in Django.

- ✓ Implementing forms within Django views for data submission
- ✓ Utilizing Django forms for CRUD (Create, Read, Update, Delete) operations

- ✓ Enhancing form styling with Django **Crispy Forms**
- ✓ Managing file uploads, including images, in Django applications
- ✓ Configuring **Django with Apache** for production deployment
- ✓ Handling cookies and sessions for user authentication and tracking
- ✓ Implementing comments functionality in Django-based applications

## **Module 32: Django Admin Panel – Managing Data & Customization**

---

This module explores Django's built-in admin interface for efficient application management.

- ✓ Creating and configuring a **superuser** for admin access
- ✓ Navigating and managing data through Django's admin dashboard
- ✓ Registering models and managing database records via the admin panel
- ✓ Executing queries and displaying data through the command line
- ✓ Customizing the Django **admin interface** for enhanced usability
- ✓ Understanding **Django ORM (Object-Relational Mapping)** for database interaction

## **Module 33: Django API Development – Building and Managing APIs**

---

This module focuses on creating and managing APIs using Django's REST framework for seamless data exchange.

- ✓ Implementing **serializers** for data conversion and validation
- ✓ Developing API views for handling client requests
- ✓ Applying filtering backends for optimized data retrieval
- ✓ Enabling **pagination** for structured API responses
- ✓ Performing CRUD (Create, Read, Update, Delete) operations via API
- ✓ Managing serializer fields for efficient data handling
- ✓ Testing and debugging API endpoints for reliability

## Module 34: Managing Static Files in Django

This module covers working with static assets such as CSS, JavaScript, and images in Django applications.

- ✓ Integrating **CSS files** into Django templates for styling
- ✓ Loading and managing **JavaScript files** in web pages
- ✓ Handling image uploads using Django models
- ✓ Implementing **user authentication** for secure access
- ✓ Building sample projects and real-world web applications

## Module 35: Career & Placement Assistance

This module provides career guidance, interview preparation, and certification insights for job readiness.

- ✓ Strategies to successfully clear technical interviews
- ✓ Common interview questions with expert answers
- ✓ Python and Django-specific interview preparation
- ✓ Resume-building techniques for impactful job applications
- ✓ Guidance on attempting global certification exams
- ✓ Earning industry-recognized credentials and applying for job opportunities

# Learning Management System

## Key Features **Learning Management System**



### **Live Sessions with Class Recordings**

Gain access to interactive live training sessions along with recorded classes to reinforce learning at your own pace.



### **Expert-Led Training Sessions**

Learn from industry experts through structured training sessions designed to enhance your skills and practical knowledge.



### **Earn Your Training Certificate**

Receive an official training certification upon successful course completion to validate your learning and expertise.



### **Experience Certificate for Professionals**

Get an experience certificate based on your hands-on project work and practical assessments.

