SevenMentor PVT.LTD

Unleash the Power of
Python Full Stack

Your Path to Mastery Begins Here !

www.sevenmentor.com

Python Full Stack Course Mission

SevenMentor Institute leads the way in providing cutting-edge IT Training and Skill Development across India. We have strived to establish an ideal learning atmosphere at all our training centers. We prepare our students to become dependable future professionals. Our institute aspires to promote universal access to learning for students. To achieve our mission of promoting better learning we invite all students to enroll in our Python Full Stack Course.

Join us today to make a fulfilling career in the Development sector.

Python Full Stack STATISTICS:

INDUSTRY INSIGHTS

84% Web sites have some form of Python Full Stack based Applications

33% Year on Year Growth For All Python Full Stack Jobs

Learn Python Full Stack and Be in Demand Always!

In today's rapidly evolving Tech Industry, being a skilled Python Full Stack Developer is a valuable asset. There is an unprecedented requirement of all rounders and dynamic developers in the website building domain. SevenMentor's comprehensive Python Full Stack Course is designed to equip you with the knowledge and practical skills to handle every aspect of Python Full Stack Developer.

Become an Omnipresent
Python Full Stack!

Becoming really good at using Python Full Stack is a big aim for any web developer. If you want to be super skilled in this newest version of Python Full Stack, you'll have to be dedicated to always learning and becoming really good at this powerful front-end framework. Python Full Stack is famous for being versatile and widely used in web development. Many developers use it to create dynamic, responsive, and feature-rich web applications.

SevenMentor PVT.LTD

Start Learning to Start Applying:

Gain practical Python Full Stack experiences through well designed courses, latest tools and excellent teachers.

Experienced Faculty

Flexible Scheduling

Hands-On Learning

Mock Interview Sessions

Real-World Projects

Career Support

Comprehensive Curriculum

Lifetime Access

1. HTML 4 and HTML 5.

A. Introduction of HTML
B. Tag, Elements and Attributes
C. Basics syntax
D. Table
E. List
F. Forms
G. Structure of HTML4 and HTML5
H. Semantic and non-semantic tags
I. HTML 5 Features
J. New Input type
K. Forms Attribute
L. SVG
M. Canvas
N. Audio, Video Tag

2. CSS (Cascading Style Sheet)

A. Attributes (ID, Class, Style, Title)
B. CSS Types (Inline, Internal, External)
C. Box-model
D. Display Property (Block, Inline, None)
E. Visibility-Hidden
F. Position Property(Static, Relative, Absolute, Fixed)
G. Z-index Property
H. Combinators (Descendant Selector, Child Selector, Adjacent Sibling Selector, General Sibling Selector)
I. CSS Pseudo-classes (Link, Visited, Hover, Active)
J. CSS Pseudo-elements (First Line, First Letter, Before, After ,Selection)
K. Static Web Page
L. Viewport Meta tag

3. CSS

A. Background, Multiple Backgrounds
B. Font Related Features (online fonts)
C. Text-Effect and Box-Effect
D. Gradients-Linear and Radial
E. Transition
F. Transformation G. Animation
H. Media Queries

4. Bootstrap

A. Introduction of Bootstrap (Responsive)
B. Typography
C. Tables
D. Images, Buttons
E. Grid Structure-Type of columns
F. Forms
G. Jumbortron
H. Well
I. Panel
J. Navbar, Nav Tab
K. Carousel
L. Responsive Web Page

5. JavaScript

A. Introduction of JavaScript B. Use of JavaScript
C. Variables
D. Keywords
E. Data Type (Primitive, non-primitive)
F. JS Conditions (if, if-else)
G. Conditional operators & logical operators
H. Loops (for, while, do-while)
I. Switch Case
J. Functions
K. SetTimeout and set Interval Function

L. HTML DOM

M. Use the document object to access and manipulate HTML

N. Changing HTML Elements

O. Adding and Deleting Elements

P. Array

Q. Objects

R. How to access Objects (Dot Notation, bracket Notation)

S. Object Creation (Empty Object, Literal Way, Constructor Way) T. Prototype

U. Validations

V. Events

6. jQuery

1. Introduction

A. JQuery Library

B. First JQuery Example

C. The Document Ready Function

D. How to escape a special character

2. Selectors

A. Basic selectors

B. Precise Selectors

C. Combination of Selectors

D. Hierarchy Selectors

E. Selection Index Filters

F. Visibility Filters

G. Forms Selectors

H. Forms Filters

3. Find Dropdown Selected Item

4. Document Traversal

A. Getting a specific DOM element

5. Event

A. Events Helpers

B. Attach Event

C. Detaching Events

D. Events Triggering

6. HTML Manipulation

7. AJAX with jQuery

8.Use and Benefits JSON

7. JSON Introduction

8. HTML Manipulation

Getting Started with React Course

A. Intro to ReactJS

B.Environment Setup and folder structure JSX

C. Components

D. State

E. Props Overview

F. Props Validation

G. Component API

H. Component Life Cycle

I. Forms

J. Events

K. Refs

L. Keys

M. Router

N. Animations

O. Higher-order Components

P. What is Virtual DOM

Q. A basic setup of Bootstrap for Styling

Virtual DOM Introduction

A. Module Introduction

B. Using types Classes Interface

C. Generic Wrap up & Modules

E. Deep dive into Virtual DOMs

The Basics of React

A. Module Introduction

B. How a React App gets loaded and started

C. Components Creating new Component

D. Analysis of the functionality of

App Module & Component Declaration

E. Templates Understanding Component Styles

F. Understanding Component Selector

Data binding & DOM concepts

A. Creating a basic custom attribute DOM

B. Exploring Renderer to create a DOM

C. Understanding Host Listener and Host

Binding Binding to DOM Properties Building

D. Structural DOMs Exploring ng Switch

Understanding Services & Implementing Dependency Injection

A. Module Introduction

B. Why Services are required

C. Understanding Dependency Hierarchical

D. Injector Injecting a Service into other services

Observables in React

A. Module Introduction

B. Understanding built-in React Observable

C. Creating Observables Using a Custom

observable

D. Understanding Observable Operators

Forms in React

A. Module Introduction

B. Need to React's help Template Driven

vs Reactive Approach

Making HTTP requests In React

A. Introduction & how HTTP request

B. Works in SPAs Sending POST Request

C. Adjusting request headers Sending

GET requests

Templated Driven Approach

A. TD: Submitting and Using the form

B. TD: Understanding Form's State

C. TD: Accessing Form using @ViewChild

D. TD: Adding Validation to user input

E. TD: Understanding Form State

F.TD: Outputting Validation Error Messages

G. TD: Using ngModel and two-way binding

H. TD: Grouping Form Controls

I. TD: Using Form Data

Making HTTP Requests in MERN Stack

A. Introduction &how HTTP request

B. Works in SPAs Sending POST Request

C. Adjusting request headers Sending GET requests

Authentication & Route Protection in Angular Apps

A. Module Introduction How Authentication works in SPAs

B. Introduction to JWT

C. Firebase SDK Signing users up

D. Signing users in Token

Reactive Form Approach

A. Reactive: Creating a Form in Code

B. Reactive: Submitting the form

C. Reactive: Adding Validation

D. Reactive: Getting Access to Controls

E. Reactive: Grouping Controls

F. Reactive: FormArray

G. Reactive: Custom Validators

H. Reactive: Exploring Error Codes

I. Reactive: Custom Async Validator

Contents of SQL

Introduction to SQL

What is SQL?, History and evolution of SQL, Relational databases and data modelling, Uses and applications of SQL

SQL Functions and Operators

Aggregate functions, Mathematical operators, String functions, Date and time functions

Basic SQL Syntax

Structured Query Language (SQL), Data Definition Language (DDL), Data Manipulation Language (DML), Retrieving data with SELECT statements, Filtering data with WHERE clauses, Sorting data with ORDER BY clauses

SQL Transactions and Locks

Transactions, ACID properties, Locks, Concurrency control

SQL Joins and Subqueries

Overview of joins and subqueries, Inner joins, Outer joins, Subqueries

SQL Constraints and Indexes

Primary keys, Foreign keys, Unique constraints, Indexes

Advanced SQL Concepts

Stored procedures, Triggers, Views, User-defined functions

SQL for Data Analysis

Aggregating data with GROUP BY clauses,
Filtering data with HAVING clauses,
Analyzing data with window functions

SQL for Data Manipulation

Updating data with UPDATE statements,
Inserting data with INSERT statements
Deleting data with DELETE statements

SQL for Performance Tuning

Query optimization B. Index tuning, Database design for performance

Python Syllabus

1. Introduction to Python

A. What is Python and history of Python?
B. Unique features of Python
C. Python-2 and Python-3 differences
D. Install Python and Environment Setup
E. First Python Program
F. Python Identifiers, Keywords and Indentation
G. Comments and document interlude ni Python
H. Command line arguments
I. Getting User Input
J. Python Data Types
K. What are variables?
L. Python Core objects and Functions
M. Number and Maths
N. Week 1Assignments

2.Control Statements

A. if-else
B. if-elif-else
C. while loop
D. for loop
E. break
F. continue
G. assert
H. pass
I. return

3. List, Ranges &Tuples in Python

A. Introduction
B. Lists in Python
C. More about Lists
D. Understanding Iterators
E. Generators, Comprehensions and Lambda Expressions
F. Introduction
G. Generators and Yield

H. Next and Ranges

I. Understanding and using Ranges

J. More About Ranges

K. Ordered Sets with tuples

4. Python Dictionaries and Sets

A. Introduction to the section

B. Python Dictionaries

C. More on Dictionaries

D. Sets

E. Python Sets Examples

5.Input and Output in Python

A. Reading and writing text files

B. Writing Text Files

C. Appending to Files and Chalenge

D. Writing Binary Files Manually

E. Using Pickle ot Write Binary Files

6.Python built in function

A. Python user defined functions

B. Python packages functions

C. Defining and calling Function

D. The anonymous Functions

E. Loops and statement in python

F Python Modules &Packages

7. Python Object Oriented

A. Overview of OOP

B. The self variable

C. Constructor

D. Types Of Variables

E. Namespaces

F. Creating Classes and Objects

G. Inheritance

H. Types of Methods

I. Instance Methods Static Methods

Class Methods

J. Accessing attributes

K. Built-In Class Attributes

L. Destroying Objects

M. Abstract classes and Interfaces

N. Abstract Methods and Abstract class

O. Interface in Python

P. Abstract classes and Interfaces

8. Exceptions

A. Errors in Python

B. Compile-Time Errors

C. Runtime Errors

D. Logical Errors

E. What si Exception?

F. Handling an exception

G. try...except...else

H. try-finally clause
I. Argument of an Exception
J. Python Standard Exceptions
K. Raising an exceptions
L. User-Defined Exceptions

8. Exceptions

A. Errors in Python

B. Compile-Time Errors

C. Runtime Errors

D. Logical Errors

E. What si Exception?

F. Handling an exception

G. try...except...else

H. try-finally clause

I. Argument of an Exception

J. Python Standard Exceptions

K. Raising an exceptions

L. User-Defined Exceptions

9. Python Regular Expressions

A. What are regular expressions?

B. The match Function

C. The search Function

D. Matching vs searching

E. Search and Replace

F. Extended Regular Expressions

G. Wildcard

10. Python Multithreaded Programming

A. What si multithreading?

B. Difference between a Process and Thread

C. Concurrent Programming and GIL

D. Uses of Thread

E. Starting a New Thread

F. The Threading Module

G. Thread Synchronization

H. Locks

I. Semaphore

J. Deadlock of Threads

K. Avoiding Deadlocks

L. Daemon Threads

11. Using Databases in Python

A. Python MySQL Database Access

B. Install the MySQL db and other Packages

C. Create Database Connection

D. CREATE, INSERT, READ Operation

E. DML and DL Operation with Database

12. Graphical User Interface

A. GUI in Python

B. Buton Widget

C. Label Widget

D. Text Widget

Django Prerequisite:

- HTML, CSS, Bootstrap, Python,SQL

Django Architectur

- Django Project MVT Architecture & MVC
- Difference Between MVT and MVC

Routing

- Understanding urls.py and the purpose of it
- Creating urls with names and namespaces
- Building dynamic urls
- Mapping of the urls with the view functions
- Linking the urls of an app in a project with the the root app of the project using 'include

Introduction to Django

- Installation of Django
- Creating a virtual environment for better dependency management
- Setup Django environment
- Create your first sample Django project
- Understanding the project structure of a newly created django project
- Django Admin, Commands and Shell
- Difference between a App and a Project
- Role of Flask and Django

Django views

- Function Based Views
- Django CRUDE operation using function
- What is Render and relative
- Import URL names as Links

Interacting with databases

- What is ORM?
- Benefits of ORM
- Connecting a django project to a database
- Django migrations
- Visualize business data as django models
- Django model Field and field types
- Relationship between django models - One to One,

Built In Django Apps And Web

- Development Concerns
- Authentication system
- Messages framework
- Emailing system

The Django Template System

- Basic Using Template System
- Need for templates
- How to Configure Template
- Template Loading
- Using conditionals and looping in django templates
- Django convention of storing the static assets like images, css and javascript and referencing
- them from the template
- Handling different kind of media files Template Inheritance

The Django Admin Dashboard

- Activating the admin interface
- Customize admin dashboard
- Creating superuser for accessing the backend admin dashboard
- Customizing django admin templates look and feel (perform CRUD Operations)

Session Management

- HTTP as a stateless protocol
- Django session management built in app
- Concept of session management and how it internally uses cookies and session id
- Storing attributes in a session and retrieving from the same session
- Destroying a session using 'flush’

Django Class Based Generic Views

- Introduction Generic Class Based Views
- How to create generic class based views
- Class based CRUDE operation
- Create views
- Retrieve views
- Update views
- Delete views

Rest Web Services

- Web Services and REST
- Django REST framework
- What is serializers?
- Creating Views
- Creating urls
- Update and delete records using
- serializer/django rest freamwork
- POSTMAN details and how to check
- API's in POSTMAN

Your Success Is Our Priority:

Every student at **SevenMentor** gets personalized guidance, Mentorship, and ample opportunities to address individual questions and concerns. All our sessions are designed to be engaging, interactive, and tailored to your learning pace, ensuring you grasp each concept with clarity.

Sarika kore

Web Full Stack & Web Development Trainer

(SevenMentor & Training Pvt. Ltd.)

She have total 15 years of experience. she is currently working as a Technical Trainer. she did MCA and she havetrained more than 3000 students for various courses. she have also worked in Net Protector Anti Virus Company for 2 years.

Sonal Vanarse

(Web Full Stack Trainer)

Professional Software developer trainer with over 9 years of industry experience in Full stack technologies (MERN, MEAN, UI/UX). Have strong Jeal of staying up to date with the latest technology trends across the industries. Her unique teaching pedagogy helps the student to learn the best practices being followed in software industries. Trained more than 3000 students and contributed into their career success.

Anil Giri

Java full stack & .Net Full stack Trainer

(SevenMentor & Training Pvt. Ltd.)

Professional IT trainer with over decade and eight years of experience in Java and .NET Full stack technologies. Worked for many corporates and delivered corporate training PAN India also managed team of 50 trainers and has many accolades under his belt due to his excellence at work.
An expert at designing training courses with technical content, Effective at imparting knowledge through his superior verbal communication skills, active listening style and unique teaching pedagogy. He has proven success in utilizing cutting-edge technology to devise training exercises for all types of learners.

Dharmaraj Pawale

Sr. Technical Trainer

(SevenMentor & Training Pvt. Ltd.)

Dharmaraj has expertise in retail as well as corporatetraining in Java Full Stack, MEAN Stack, MERN Stack, Hadoop-Big Data, Spark. Hehas extensive 19+ years experience in both Software Training and Development.He has trained more than 10000+ students and corporate professionals in India aswell as overseas. He is also Salesforce Certified Administrator. His area ofexpertise lies in both Frontend as well as Backend Technologies. Frontend technologiesinclude HTML5, CSS3, Bootstrap, JavaScript, React & Redux, Angular andBackend Technologies include Java, Servlet & JSP, RESTful Web Service, HibernateJPA, Spring, Spring Boot, MicroService, Node JS, Express JS, RDBMS Databasessuch as MySQL, Oracle SQL, PL/SQL, NoSQL Databases such as MongoDB

THINGS THAT SET US APART FROM THE REST:

SevenMentor prepares you well so that you can embark on a rewarding career in Python Full Stack through our training.

Hands-On Projects: Gain practical experience by working on real-world projects, building a robust portfolio that will impress potential employers.

Flexibility: Our flexible schedule options allow you to learn at your own pace, making it perfect for both beginners and experienced developers looking to upskill.

Career Support: We're dedicated to your success! Benefit from career guidance, resume building, interview prep, and job placement

Community: Join a vibrant community of like-minded learners, where you can collaborate, share ideas, and network with peers.

Our Students are at reputed Tech Companies

BAJAJ FINSERV

GT FINANCIAL SERVICES

INFERYX

STRELEMA

pivotchain

LONAR TECHNOLOGIES

Simplify Healthcare

OMFYS

Scymes

KALA Genset Source of Power

Altum Credo Home Finance

INFOWORLD THE SOFTWARE PEOPLE

NABLA INFOTECH

Circulants

SRVmedia Insights | Ideas | Impact

Rabbit&Tortoise TECHNOLOGY SOLUTIONS

NIMAP INFOTECH FOCUS TO REDEFINE

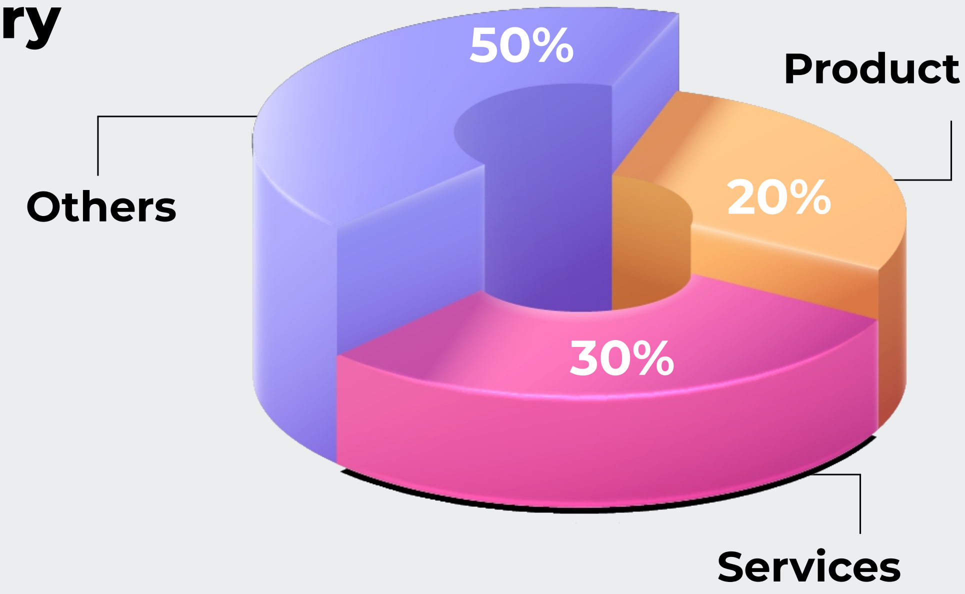
Integs Cloud

Success is around the corner!

Consider the potential earnings and opportunities that a career in Python Full Stack can offer. With Python Full Stack you'll be well-positioned to secure a job that provides you with a stable and potentially prosperous future.

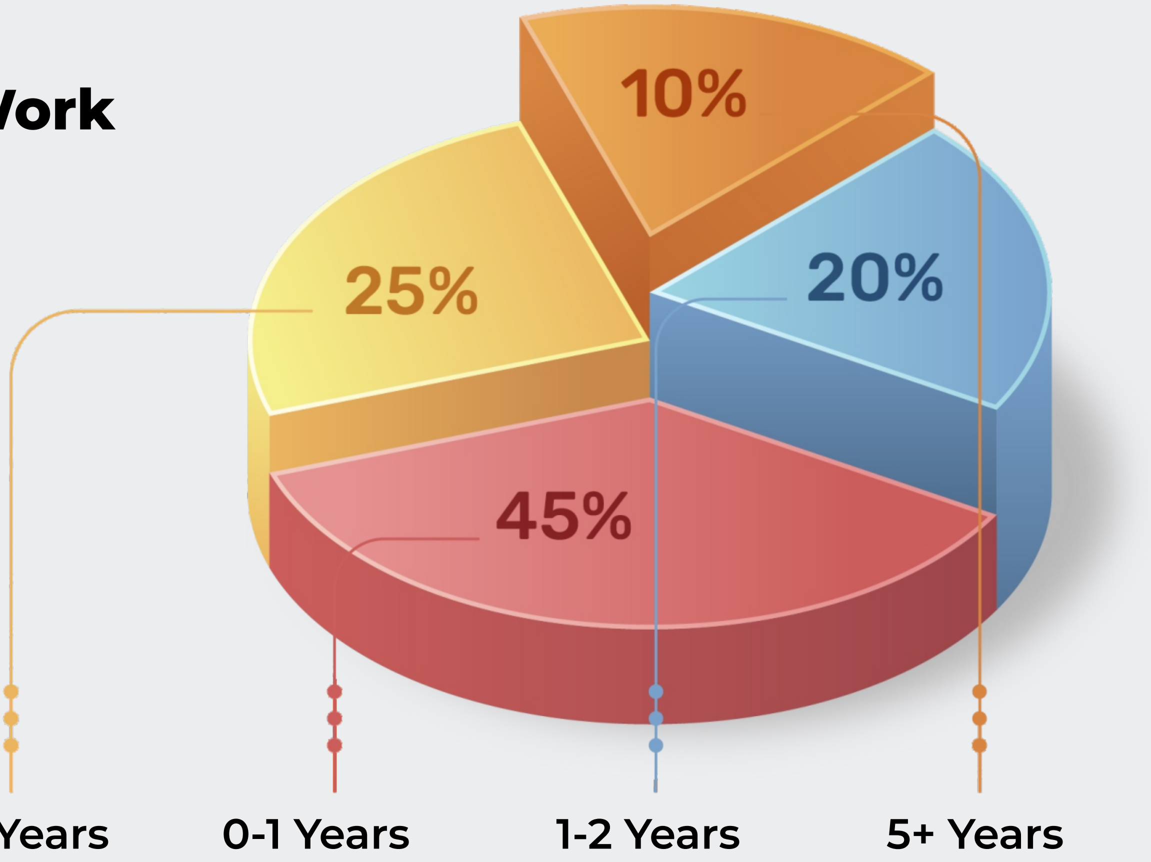
Learners’ Industry Background

🟪 🟧 🟥



Learners’ Work Experience

🟨 🟥 🟦 🟧



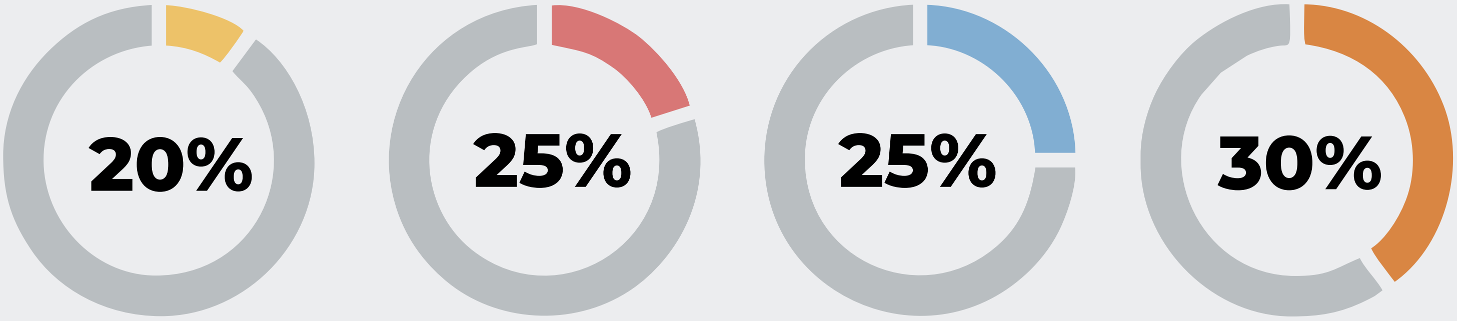
Learners' Expertise

🟧 Non-IT and Others

🟦 Tech

🟥 Python Full Stack

🟨 Professionals



Begin Your Journey Towards Better Future:

The global Python Full Stack vacancies will grow by 55% and generate 4 million new jobs by 2030.

The global vacancies for Python Full Stack will grow by 55% and generate 4 million new jobs by 2030.

Affordable Training without Compromise:

We understand that pursuing your dreams in App and Web Development shouldn't be a financial burden. That's why we've made afford ability a cornerstone of our Python Full Stack. We believe that quality education should be accessible to everyone, and we've structured our program to reflect this commitment. Our Python Full Stack takes you on an exciting journey but at a substantially low price.

How to become a great Python Full Stack ?

- Enroll at **SevenMentor Institute**
- Get hands-on training from **experienced teachers**
- Receive Industry-recognized **Python Full Stack Certification**
- Work for leading **MNCs** through our **on-campus interviews**

Success is Just a Call Away!

So if you are ready to code your way to success? Enroll now at SevenMentor and unlock your potential as a Python Full Stack Developer. Our counsellors are a call away and they will be more than happy to talk with you. Anyday, Everyday, we are there for you!

Request For Call Back

☎ **020 7117 3035**