



Codevensiva Academy | www.academy.evensiva.com

Fullstack Python With Agentic & Generative AI

COMPLETE AI & LLM ENGINEERING BOOTCAMP

Master **Python, Generative AI, LLM Engineering, RAG Systems, AI Agents, LangChain, LangGraph, Multi-Modal AI, and MCP (Model Context Protocol)** through an intensive **3-month industry-focused bootcamp**.



Duration

3 Months Intensive Program (12 Weeks)



Mode

Live Mentor-Led + Hands-On Projects



Level

Beginner to Intermediate/Advanced



Prerequisites

Basic computer knowledge (No prior AI experience required)



Codevensiva
Academy

Course Overview

This accelerated program is designed to take you from **Python fundamentals to building production-ready AI applications**, helping you understand and implement technologies behind **ChatGPT, Gemini, Claude, and modern AI systems**.

The syllabus is optimized for **fast learning, practical implementation, and real-world AI project building**.

Python & Development Tools

From syntax to async programming, Git, Docker, FastAPI & Pydantic

AI & LLM Foundations

Transformer architecture, tokenization, embeddings & attention mechanisms

Prompt Engineering & Agents

Chain-of-thought, LangChain, autonomous agents & tool calling

RAG, LangGraph & MCP

Retrieval-augmented generation, graph workflows, memory systems & Model Context Protocol

What You'll Build

8 production-grade projects across 3 months:

- Python Utility Management System
- AI-ready Backend API (FastAPI)
- Tokenizer from Scratch
- Local Ollama + FastAPI AI App
- Python CLI Coding Agent
- Document RAG Pipeline
- Graph-based AI Workflow System
- MCP-Powered AI Server



Python, Development Tools & AI Foundations

The first month builds your technical foundation — from Python fundamentals to containerized deployment, API development, and a deep understanding of how modern AI and LLMs actually work under the hood.

1

Module 1: Python Programming Essentials

Syntax, data types, functions, OOP, decorators, generators, async programming, and error handling. **Project:** Python Utility Management System

2

Module 2: Git, GitHub & Docker Essentials

Repository management, branching, pull requests, Docker images, containers, and Docker Compose. **Project:** Containerized Python Application with GitHub Workflow

3

Module 3: Pydantic & FastAPI Basics

Data validation, type safety, API development, request validation, and AI-ready backend setup. **Project:** AI-ready Backend API using FastAPI

4

Module 4: AI & LLM Foundations

Generative AI, LLMs, transformer architecture, tokenization, embeddings, attention mechanism, and the "Attention Is All You Need" paper. **Project:** Build a Basic Tokenizer



Python & AI Foundations: What You'll Learn

Python Programming Essentials

- Installing Python & Development Environment Setup
- Python Syntax, Variables, Data Types & Operators
- Conditional Statements, Loops & Iterations
- Functions, Lambda Functions, Scope & Closures
- Exception Handling & File Handling
- Classes, Objects, Inheritance, Polymorphism & Encapsulation
- Decorators, Generators & Iterators, Type Hinting
- Async Programming Basics

Git, GitHub & Docker

- Git Installation, Repository Management, Commits & Branching
- Merge & Pull Requests, Professional Git Workflow
- Containers vs Virtual Machines
- Docker Images, Containers, Dockerfile Basics & Docker Compose

Pydantic & FastAPI

- Data Validation, Models & Schemas, Type Safety, Structured Outputs
- API Development Basics, Request Validation, Building Python APIs

AI & LLM Foundations

- Evolution of AI, Understanding Generative AI, What are LLMs?
- How GPT Works, Tokens & Tokenization, Embeddings
- Attention Mechanism, Positional Encoding
- Research: "Attention Is All You Need" paper



Prompt Engineering, LLM Integration & AI Agents

Month 2 is where AI becomes real. You'll master prompting techniques used in production systems, integrate OpenAI, Gemini, and local LLMs into your applications, and build autonomous AI agents that can reason, plan, and execute tasks.

Module 5: Prompt Engineering Mastery

Zero-shot, one-shot, few-shot, chain-of-thought, and persona-based prompting. ChatML, Alpaca, and LLaMA formats. JSON outputs using Pydantic for reliable AI responses.

✅ **Project:** Production-grade Prompt Workflow

Module 6: Running & Integrating LLMs

OpenAI & Gemini API integration, authentication, and model usage. Running models locally with Ollama + Docker. Hugging Face open-source LLMs and fine-tuned models. FastAPI + LLM integration for AI backend services.

✅ **Project:** Local Ollama + FastAPI AI Application

Module 7: Building AI Agents

What are AI agents? Tool calling, agent workflows, autonomous systems. CLI-based coding assistants and Claude-style agent workflows. LangChain basics: chains, tools, and agents.

✅ **Project:** Python CLI Coding Agent



Prompt Engineering & AI Agents: Techniques & Tools

Prompting Techniques

- Zero-Shot Prompting
- One-Shot Prompting
- Few-Shot Prompting
- Chain-of-Thought Prompting
- Persona-Based Prompting

Prompt Frameworks

- ChatML Format
- Alpaca Format
- LLaMA Prompting

Structured Outputs

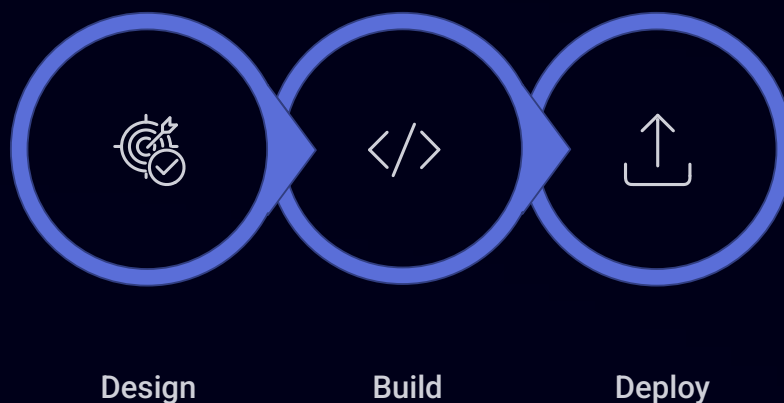
- JSON Outputs using Pydantic
- Reliable AI Responses

LLM Integration Stack

- OpenAI & Gemini APIs — authentication, model usage
- Ollama Setup — local AI systems with Docker
- Hugging Face — open-source and fine-tuned LLMs
- FastAPI + LLM — AI-powered backend services

AI Agent Fundamentals

- Tool Calling & Agent Workflows
- Autonomous Systems Design
- CLI-Based Coding Assistant
- Claude-style Agent Workflows
- LangChain: Chains, Tools & Agents



From designing agent goals to deploying with Ollama or FastAPI APIs, this workflow guides you through building production-ready autonomous AI systems.



RAG, LangGraph, Memory Systems & MCP

The final month brings everything together. You'll build Retrieval-Augmented Generation pipelines that answer questions from your own documents, design graph-based AI workflows with LangGraph, implement memory systems for persistent context, and explore the cutting-edge Model Context Protocol (MCP).

1

Module 8: Retrieval-Augmented Generation (RAG)

Document indexing, chunking strategies, embeddings, retrieval mechanisms, and response generation. LangChain document loaders, text splitters, retrievers, and vector stores. FastAPI + RAG for production APIs and high-performance systems.

✓ **Project:** Document RAG Pipeline

2

Module 9: LangGraph & AI Memory Systems

Nodes, edges, state management, and graph-based AI workflows. Short-term and long-term memory, context persistence, and vector databases. Neo4j basics and knowledge graphs for graph memory.

✓ **Project:** Graph-based AI Workflow System

3

Module 10: Conversational AI, Multi-Modal AI & MCP

AI voice agents, human-like conversations, Speech-to-Text (STT), Text-to-Speech (TTS), image understanding, and text+image AI systems. Model Context Protocol (MCP): architecture, STDIO & SSE, and building MCP servers with Python.

✓ **Project:** MCP-powered AI Server



RAG Architecture, LangGraph & MCP Explained

Complete RAG Architecture

- Document Indexing & Chunking Strategies
- Embeddings & Retrieval Mechanisms
- Response Generation Pipeline
- LangChain: Document Loaders, Text Splitters, Retrievers, Vector Stores
- FastAPI + RAG for High Performance Production APIs

LangGraph Fundamentals

- Nodes, Edges & State Management
- Graph-Based AI Workflows
- Short-Term & Long-Term Memory
- Context Persistence & Vector Databases
- Neo4j Basics & Knowledge Graphs

Conversational AI

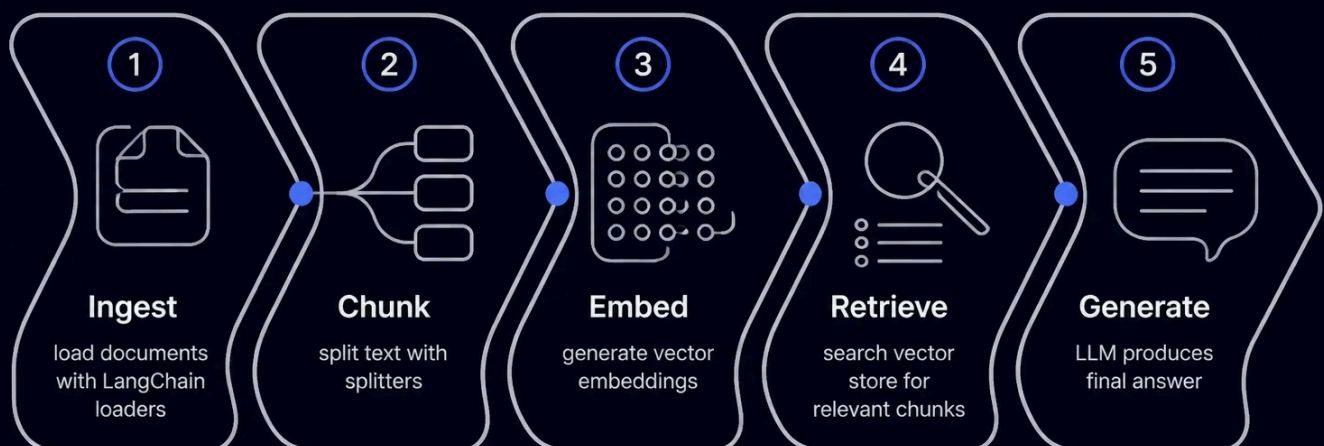
- Building AI Voice Agents
- Human-like Conversations
- Speech-to-Text (STT)
- Text-to-Speech (TTS)

Multi-Modal AI

- Image Understanding
- Text + Image AI Systems

Model Context Protocol (MCP)

- What is MCP?
- MCP Architecture
- STUDIO & SSE
- Build MCP Server with Python



Final Capstone Projects & Tools Covered

By the end of the **3-month intensive program**, students will have built a complete portfolio of production-grade AI applications. Every project is designed to demonstrate real-world skills that employers and clients look for.

Capstone Projects Built

- Python Utility Management System
- AI-ready Backend API using FastAPI
- Tokenizer from Scratch
- Local Ollama + FastAPI AI App
- Python CLI Coding Assistant
- Document RAG Pipeline
- Graph-based AI Workflow System
- MCP-Powered AI Server

Tools & Technologies

Python

Git & GitHub

Docker

FastAPI

Pydantic

OpenAI & Gemini

Ollama

Hugging Face

LangChain

LangGraph

Neo4j

Vector DBs

STT & TTS

MCP



Codevensiva
Academy

Career Outcomes & Who Should Join

After completing this **3-month AI Engineering Bootcamp**, you'll be equipped to apply for some of the most in-demand roles in the tech industry. This course is designed for anyone ready to transition into the AI-powered future.



AI Engineer

Design and deploy intelligent systems at scale



Generative AI Developer

Build applications powered by LLMs and generative models



Python Developer with AI

Combine Python expertise with modern AI capabilities



LLM Engineer

Specialize in large language model integration and optimization



RAG Engineer

Build retrieval-augmented generation pipelines for enterprise AI



Backend AI Developer

Create AI-powered APIs and backend services for production systems



AI Application Developer

Develop end-to-end AI applications from concept to deployment

- Who Should Join?** This course is ideal for beginners entering Python & AI development, developers building AI-powered applications, students preparing for future-ready careers, working professionals transitioning into Generative AI, and backend developers upgrading to AI systems. **By the end of this 3-month program, you won't just understand AI — you'll confidently build real-world AI applications from scratch.**



Codevensiva
Academy