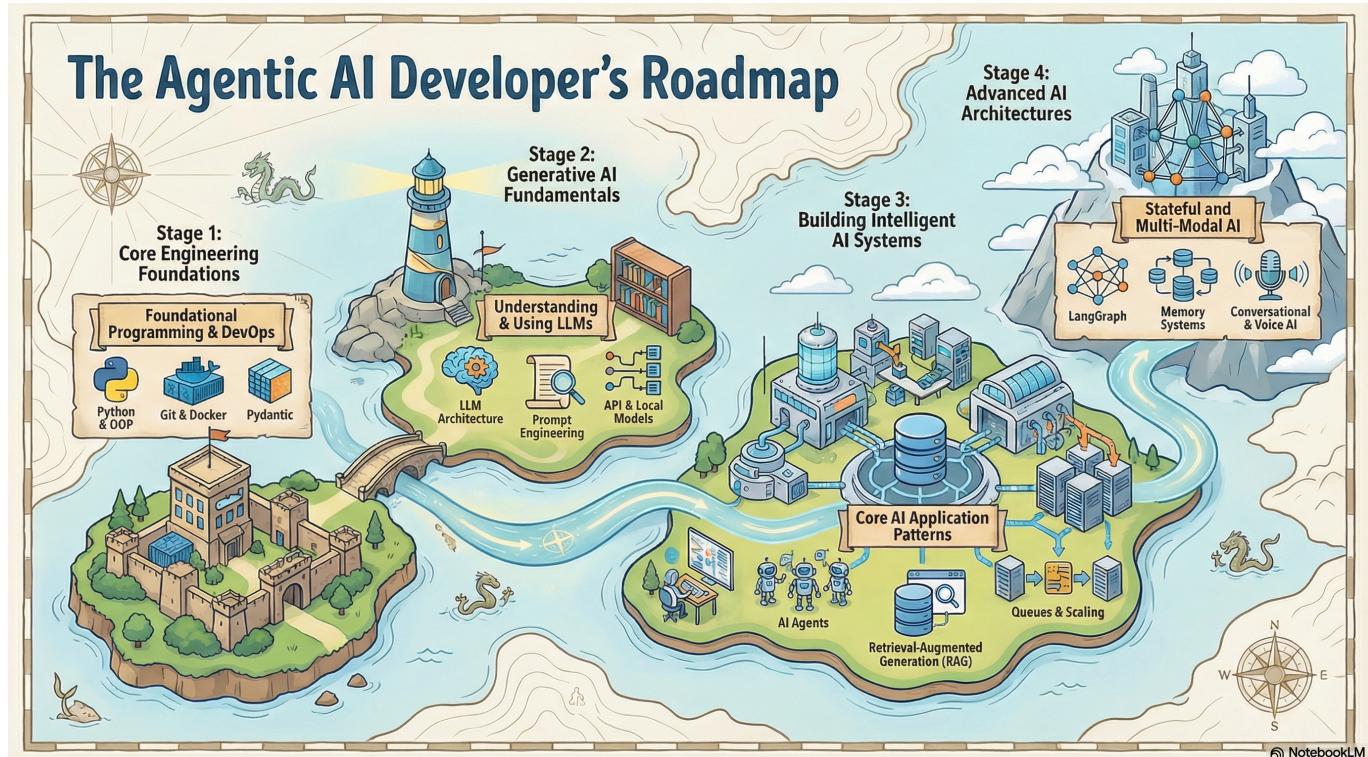


# kintsugi-stack-generative-agentic-ai-python-fullstack

"The problem with automation is that it only works until it doesn't." — Martin Fowler

- Author: [Kintsugi-Programmer](#)



Disclaimer: The content presented here is a curated blend of my personal learning journey, experiences, open-source documentation, and invaluable knowledge gained from diverse sources. I do not claim sole ownership over all the material; this is a community-driven effort to learn, share, and grow together.

## Table of Contents

- [kintsugi-stack-generative-agentic-ai-python-fullstack](#)
  - [Table of Contents](#)
  - [1. Introduction](#)
    - [1.1. Installation of Tools \(VSCode and Python\)](#)
    - [1.2. VS Code Setup \(Extensions and Themes\)](#)

## 1. Introduction

### 1.1. Installation of Tools (VSCode and Python)

- **Course Context**

- This section is the **first section** of the **Agentic AI with Python** course.

- The purpose of this section is to **set up the developer environment** for building **agentic AI systems**.
- The section explains **what tools are required** before starting the course.
- All tools discussed are:
  - **Free**
  - **Very easy to set up**
  - **Most likely already installed** on the learner's machine

- **Objective of This section**

- To explain the **development environment setup** needed for the course.
- To introduce the **basic tools** required before writing any code.
- This section does **not** involve coding yet.
- It prepares learners to follow upcoming tutorials smoothly.

- **Tool 1: IDE (Integrated Development Environment)**

- An IDE is required to **write and manage code**.
- The instructor will use **Visual Studio Code (VS Code)**.
- Learners are **free to use any IDE** they like.
- However, it is **highly recommended** to use **Visual Studio Code** because:
  - It helps learners **follow the tutorial step by step**
  - Screens and instructions will match the instructor's setup

- **Visual Studio Code Details**

- Visual Studio Code is:
  - **Free**
  - Available for **Windows**
  - Available for **Linux**
  - Available for **other platforms**
- Installation steps:
  - Click the **Download** button
  - Choose the appropriate platform
  - Follow on-screen instructions
- VS Code will be used as the **primary IDE** in this course.

- **Tool 2: Python Programming Language**

- Python **must be installed** on the machine.
- The entire course is **based on Python**.
- The instructor **assumes basic Python knowledge**, not advanced expertise.

- **Required Python Knowledge (Basic Level)**

- Understanding:
  - **Variables**
  - **How to define functions**
  - **How to create variables**
  - **Simple mathematical operations**
  - **Simple classes**
- Being a **Python expert is not required**.
- Basic familiarity is **sufficient to continue** with the course.

- **Python Installation Check (Example)**

- On the instructor's machine:
  - Python is already installed.
- To check Python installation:
  - Open the terminal
  - Type:

```
python
```

- Press **Enter**
- Output shows:
  - Python version **3.1 / 3.2** installed
- Conclusion:
  - Python must be installed before proceeding.

- **Python Installation Process**

- Installation is **very simple**
- Steps:
  - Click the **Download** button
  - Click **Next** multiple times
  - Follow on-screen instructions
- Python will be installed successfully after setup.

- **Tool 3: AI and LLM Platforms**

- This is an **Agentic AI course**, so AI platforms are required.

- The course will use:
  - **OpenAI**
  - **Gemini**
  - **Other Large Language Models (LLMs)**
- **OpenAI and ChatGPT Requirement**
  - Learners should:
    - Be aware of **ChatGPT**
    - Have an **OpenAI account**
  - Do not worry if you do not have one yet.
- **Upcoming Dedicated sections**
  - Separate sections will cover:
    - How to **set up an OpenAI account**
    - How to **sign up on OpenAI**
    - How to **add credits** to OpenAI
    - How to **use Gemini**
    - How to work with **other LLM tools**
- **Summary of This section**
  - This section is a **basic setup overview**.
  - It explains:
    - IDE requirement
    - Python installation
    - AI platform awareness
  - Assumption:
    - Most learners already have these tools installed.
  - This setup enables learners to:
    - Follow upcoming sections
    - Start building agentic AI systems smoothly

## 1.2. VS Code Setup (Extensions and Themes)

- **Topic: VS Code Setup, Theme, and Extensions (Optional section)**
  - **Context**
    - One very interesting comment that I usually get on my sections is:
      - What is my VS Code setup?
      - What theme am I using?

- What extensions am I using?
- Many viewers are curious about how my development environment looks and works.
- **Purpose of This section**
  - In this particular section, we discuss:
    - All the tools I use
    - All the themes I use
    - All the extensions I use in my VS Code
  - This is shared for people who:
    - Are interested in my setup
    - Want to make their development environment look exactly like mine
    - Want to follow this course comfortably with the same setup
- **Optional Nature of This section**
  - This section is **absolutely optional**
  - If you want, you can skip this section
  - Skipping this section will not affect learning the main course content

---

- **Overview of My VS Code Environment**

- Here is my VS Code
- These are a few extensions that I currently have installed
- My primary work area:
  - I work **a lot on TypeScript**
  - So I have TypeScript-related extensions installed

---

- **Installed Extensions (General)**

- I have:
  - Some **.NET extensions** installed
  - Some **container-related tools** for Docker work

---

- **Theme Used**

- The theme I use is:
  - **AO Mirage**
- More specifically:
  - **AO Mirage Dark Border Theme**

- This theme:
    - Is the **overall theme**
    - Will be used **throughout the entire course**
- 

- **Docker and Container Tools**

- I have the following container-related extensions:
    - **Dev Containers**
    - **Docker**
    - **Docker DS**
  - These are used for:
    - Working with Docker
    - Container-based development workflows
- 

- **Icon Theme**

- The icon theme I use is:
    - **Material Icon Theme**
  - This controls:
    - Folder icons
    - File icons
    - Overall file explorer visuals
- 

- **Code Formatting Tool**

- **Prettier**
    - Prettier is a very useful extension
    - It is used to:
      - Format code
      - Keep code clean and consistent
      - Automatically manage spacing and structure
- 

- **Python Development Extensions**

- **Pylance**
  - This is a **main extension**
  - It is created by **Microsoft**
  - I use this extension **a lot**

- Highly recommended
- Helps with:
  - Python intelligence
  - Code analysis
  - Better development experience
- **Python Language Extension**
  - The official **Python extension for VS Code**
  - Also by **Microsoft**
  - Highly recommended to install
  - Essential for Python development
- **Python Debugger**
  - Python Debugger extension by **Microsoft**
  - I do not use this extension a lot
  - Still:
    - It is good to have
    - Useful for debugging Python applications

---

- **Summary of Required Extensions for Python Developers**

- These are the **major extensions** you need as a Python developer:
    - Python
    - Pylance
    - Python Debugger
    - Prettier
    - Material Icon Theme
  - No additional extensions are strictly required
- 

- **Final Confirmation**

- The theme used is:
    - **AO Mirage**
    - **AO Mirage Dark Border Theme**
  - This setup will remain consistent throughout the course
- 

- **Transition to Core Course Content**

- From the next section onwards:

- We will jump into the **real workings of LLMs**
  - We will **deep dive into Agentic AI**
  - This marks the start of:
    - Practical learning
    - Core AI concepts
    - Hands-on implementations
- 

## Notes Formatting Nested Lines

Create super depth notes in Markdown (.md) format with 100% information preserved, no loss. Use simple grammar and keep everything clear, direct, and well-structured. using headings, subheadings,paragraphs, statements and code blocks when needed. Include every detail, definition, example, and step exactly from the source. transform the given content into clean, readable .md format. and no #, just nested - lines plaintext, add bold wherever necessary

---

## End-of-File

The [KintsugiStack](#) repository, authored by Kintsugi-Programmer, is less a comprehensive resource and more an Artifact of Continuous Research and Deep Inquiry into Computer Science and Software Engineering. It serves as a transparent ledger of the author's relentless pursuit of mastery, from the foundational algorithms to modern full-stack implementation.

Made with  Kintsugi-Programmer