# Course : CrewAI, AI agent programming

*Practical course - 3d - 21h00 - Ref. PYY*
*Price : 2030 € E.T.*

**NEW**

CrewAI is a modern, lightweight framework designed specifically for the creation, orchestration and deployment of multi-agent artificial intelligence (AI) systems. It enables the creation of "teams" (or crews) of collaborative AI agents, each with a specialized role, clear objectives and specific tools. The aim of this course is to establish the theoretical foundations of agent-based programming using the CrewAI environment and the Python language.

## Teaching objectives

**At the end of the training, the participant will be able to:**

- Understanding the CrewAI environment and its various concepts
- Schedule and synchronize AI prompt tasks
- Learn the syntax of the Python language to build CrewAI teams
- Master collaboration mechanisms and integrate external tools to empower agents

## Intended audience
project managers wanting to acquire a precise vision of the techniques for using AI, traditional developers wanting to evolve towards AI.

## Prerequisites
Fundamental knowledge of programming and the use of artificial intelligence.

## Practical details
**Hands-on work**
The course alternates lectures and practical work illustrating the concepts described.

## Course schedule

## 1 Introduction and key AI concepts

- Fundamentals and introduction to AI.
- AI agent: definition, components (language model, memory, tools, planning).
- Multi-agent systems (MAS): advantages (specialization, robustness, complexity).
- Overview of orchestration frameworks (CrewAI, LangChain, AutoGen).

## 2 CrewAI: the AI agent team orchestrator

- Agent AI fundamentals and first steps with CrewAI.
- The basics of agent-based programming with CrewAI.

### Hands-on work
Set up the CrewAI environment to create a first agent.

## 3 CrewAI: focus on collaboration

- The fundamental concepts of CrewAI.
- Agent: definition of roles, objectives and personality.
- Task: definition of the objective to be achieved and the tools required.
- Crew: collaboration orchestrator, process definition (sequential versus hierarchical).
- The role of the manager or orchestrator.

### Hands-on work
Installation et configuration de CrewAI, configuration de l'environnement Python (pip, venv). Installation de la librairie CrewAI. Configuration des clés API (OpenAI ou autres modèles locaux/ouverts).

## 4 Creating agents and simple tasks

- Creation of a first agent.
- Step-by-step code for instantiating an agent.
- Attribute implementation: role, goal, backstory, verbose.
- Best prompt engineering practices to define an effective role.
- Define and execute static tasks.
- Definition of a simple task with a single agent.
- Execute a simple crew with a single task and a single agent.
- Analysis of the execution process (the agent thinks, performs the task, gives the result).

### Hands-on work
Objectif : créer un agent capable de générer un contenu court sur un sujet donné. Mise en application des concepts.

## 5 Interagents collaboration

- Setting up a core team.
- Create two agents with distinct roles (e.g. researcher and analyst).
- Sequential task definition.
- Instantiate crew with a sequential process (Process.sequential).

## 6  Mastering collaboration language

- How do agents communicate?
- Optimization of the setpoint (task) to ensure good information transfer.
- Debug collaboration errors (infinite loop, bad output).

### Hands-on work
Build a team in which one agent gathers information and the other synthesizes it.

## 7  Tool integration and autonomy

- The concept of tools.
- What is a tool in CrewAI: external action capabilities.
- Use of predefined tools: Google Search Tool with crewai-tools.
- Assign tools to agents.
- Creation of a customized tool.
- The basics of creating a BaseTool class.
- Define tool description and function.
- Integration of the customized tool into an agent.

### Hands-on work
Create a team, for example a journalist and an editor, in which the journalist uses a research tool to gather facts before writing the article, and the editor corrects the style.

## 8  Advanced processes and best practices

- Advanced structuring and use cases.
- Process.hierarchical.
- Understand the manager's role in a hierarchical process.
- When to use the hierarchical approach?
- Example of implementation.
- Output and deliverables management.
- Structured extraction of results (Pydantic models or JSON formats).
- History management and intermediate steps.
- Added validation and software conditions.

### Hands-on work
Design a hierarchical system in which a manager delegates an analysis to several agents, then consolidates their reports to make a decision.

## 9  Optimization, ethics and future steps

- Optimization and improvement.
- Cost and performance management: choice of Language Model (LLM) per agent.
- Use a lightweight model for simple tasks, a powerful model for analysis.
- Techniques for reducing tokens and API calls.
- Introduction to memories: adding knowledge persistence.
- Recommended design patterns for CrewAI teams.
- Ethical considerations: biases, responsibilities, and transparency of autonomous agent actions.
- Security and management of sensitive data when using tools.

### Hands-on work
Launch of a realistic project (e.g. creation of a complete marketing plan).

## Dates and locations

**REMOTE CLASS**
2026 : 17 June, 7 Sep., 2 Dec.

**PARIS LA DÉFENSE**
2026 : 17 June, 7 Sep., 2 Dec.