+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 02/10/2025

Sign up

Codex Open AI Training

3 days (21 hours)

Presentation

Codex Open AI training introduces you to an artificial intelligence technology designed to understand and generate code. Part of the GPT family of models, it transforms the way developers create, test and document their applications.

Capable of working with a wide range of languages (Python, JavaScript, Java, SQL...), Codex integrates directly into your development environments, notably Visual Studio Code, either locally via a client or via the API.

Our Codex Open AI training course will enable you to take full advantage of this technology to accelerate your development cycles and improve the quality of your deliverables.

You'll learn how to install and configure the Codex client, integrate the VS Code extension, and use good prompt engineering practices to obtain reliable, reproducible results.

You'll discover how to integrate it into your CI/CD pipelines, automatically generate unit tests, automate documentation and secure your workflows with appropriate practices.

At the end of the course, you'll be able to use Codex Open AI as a real development assistant.

Like all our training courses, this one is based on the latest version of the OpenAI tools, and takes a practical, operational approach.

Objectives

- Discover how Codex Open AI works and how to use it.

- Generate, complete, test and document multi-language code.
- Integrate Codex into Visual Studio Code and CI/CD pipelines.
- Optimize prompts to improve quality of results.
- Implement appropriate security and ethical practices.
- Deploy a complete project with Codex.

Target audience

- Software developers and engineers
- Tech leads and DevOps teams
- Data scientists

Prerequisites

- Good command of a programming language (e.g. Python, JavaScript, Java)
- Basic knowledge of REST APIs and Git
- Current use of an IDE

Codex Open AI training program

[Day 1 - Morning]

Discover Codex and lay the foundations

- Understanding generative AI applied to code and the role of Codex
- Overview of use cases: completion, refactoring, documentation, testing
- Tools ecosystem: CLI client, VS Code extension, APIs
- Key parameters: model, temperature, output length, stops
- Basic security: API key and secret management
- Practical workshop: Installation, first prompt, function generation.

[Day 1 - Afternoon]

Local client and model settings

- CLI flow: repository navigation, editing, test execution
- Structuring context and work files
- Prompt engineering: role, constraints, examples, counter-examples
- Evaluation and recovery strategies (retry/backoff)
- Traceability: local logs and generation logs
- Practical workshop: CLI scenarios.

Initial integration into VS Code

- Installation and configuration of the VS Code extension
- Autocompletion, snippets and multi-file block generation
- Assisted refactoring and context-sensitive suggestions
- Customization: shortcuts, editor settings, team policies
- Local quality: lint, format, Git pre-hooks
- Practical workshop: Equipping an existing project with Codex + VS Code.

[Day 2 - Morning]

Software quality and testing with Codex

- Unit and integration test generation (pytest, Jest, etc.)
- Orchestration of data sets and test doubles
- Code smell detection and secure refactoring
- Prompt reliability strategies (constraints, assertions)
- Measurement: coverage, mutations, quality gates
- Practical workshop: Increasing the coverage of a critical module.

[Day 2 - Afternoon]

Documentation, refactoring and technical debt

- Generating docstrings, READMEs, ADRs and user guides
- Create reusable snippets and team templates
- Mapping and reducing technical debt with AI
- Style rules and project conventions
- Guided review: comments and assisted PR
- Practical workshop: documentation and refactoring on a legacy service.

CI/CD, governance and collaboration

- Integrating Codex into CI/CD (GitHub Actions, GitLab CI)
- Automating lint, format, patches and suggestions
- Good security & ethics practices (data, licenses, secrets)
- AI-assisted peer programming and acceptance requirements
- Dashboards: quality and productivity metrics
- Practical workshop: CI/CD Pipeline generating tests and automated Pull Request.

[Day 3 - Morning]

Codex-driven back-end and APIs

- Creation of REST APIs (Python/Node): routes, validation, errors
- Client and contract generation (OpenAPI)

- Error management: timeouts, retries, circuit breakers
- API and contract testing
- Observability: logs, traces, miniprofiling
- Practical workshop: Generating an API + battery of tests.

[Day 3 - Afternoon]

Data, productivity and gas pedals

- Generating optimized SQL queries and verifying plans
- Gas pedals: templates for prompts, macros and checklists
- Migration between languages or frameworks (AI guidance)
- Performance management and micro-benchmarks
- Multi-repo and single-repo organization
- Practical workshop: Pack of prompts/snippets gas pedals for the team.

Final project and production launch

- Mini-product design (back + front or back only)
- Deployment strategies (containers, cloud), secrets and variables
- Quality criteria, SLO/SLI, runbook
- Transition plan and internal communication
- Tooling roadmap (VS Code, CLI, integrations)
- Practical workshop: Full project and controlled release.

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced IT technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the forthcoming training course, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical training: 60% hands-on, 40% theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Certification

A certificate will be awarded to each trainee who has completed the entire course.