# Aplikacje LLM w Pythonie: RAG i agenci AI (kod: PYTHON-AI-LLM-API)

**Zapytaj o szczegóły**
tel. 22 63 64 164
akademia@alx.pl

## Opis i cel szkolenia

Coraz więcej firm chce mieć model językowy we własnym systemie, a nie tylko w oknie czatu: asystenta, który odpowiada na podstawie firmowych dokumentów, usługę, która wyciąga dane z umów, albo agenta, który sam sięga do bazy i wykonuje zadania. Na tym szkoleniu budujemy takie rozwiązania w Pythonie, krok po kroku – od pierwszego wywołania API, przez RAG, po agenta z narzędziami.

Pracujemy na API OpenAI i kilku sprawdzonych bibliotekach Pythona, m.in. LangChain. Zaczynamy od tego, na czym stoi każda aplikacja LLM: promptów utrzymywanych w kodzie, odpowiedzi w ustalonej strukturze i wywoływania narzędzi. Drugi dzień poświęcamy **RAG**, czyli odpowiadaniu na podstawie własnych dokumentów: przygotowaniu danych, bazie wektorowej, poprawianiu trafności i mierzeniu jakości. Trzeciego dnia dochodzą **agenci AI**, protokół MCP, bezpieczeństwo oraz to, co trzeba wiedzieć przed wdrożeniem: koszty, logowanie, testy. Kod piszemy tak, jak pisze się go dziś, czyli z pomocą asystenta AI.

**Po szkoleniu:**

- napiszesz usługę w Pythonie, która korzysta z modelu przez API i zwraca dane w ustalonej strukturze,
- podłączysz do modelu własne funkcje, bazę danych i zewnętrzne API jako narzędzia,
- zbudujesz asystenta, który odpowiada na podstawie dokumentów firmy i wskazuje źródła,
- sprawdzisz na liczbach, czy po zmianie promptu albo modelu odpowiedzi są lepsze, czy gorsze,
- zbudujesz agenta, który sam dobiera narzędzia, i podłączysz go do danych przez MCP,
- będziesz wiedzieć, jak ograniczyć koszty, halucynacje i ryzyko prompt injection, zanim aplikacja trafi do użytkowników.

Szkolenie kierujemy do wszystkich, którzy mają wprowadzić modele językowe do firmowego oprogramowania: programistów Pythona i backendu, inżynierów danych i analityków piszących w Pythonie, DevOpsów, architektów i liderów technicznych, a także osób, które takie rozwiązania zamawiają i odbierają od wykonawców.

Koncentrujemy się na modelach językowych i pracy z tekstem oraz dokumentami. Jeśli chcesz przede wszystkim sprawniej pisać kod z pomocą AI, zobacz szkolenie AI Assisted Development. Agentów i automatyzacje bez programowania obejmuje szkolenie Agenci AI (low-code/no-code). Klasyczne uczenie maszynowe i sieci neuronowe od podstaw to zakres kursu Programowanie AI i ML w Pythonie.

## Czas trwania

3 dni

## Program

1. Model językowy przez API
   - API OpenAI w Pythonie: zapytanie, instrukcje, historia rozmowy, odpowiedź strumieniowa
   - Tokeny, okno kontekstu, limity i koszty
   - Dobór modelu do zadania: jakość, szybkość, cena

   - Rzut okiem na API innych dostawców i modele uruchamiane lokalnie
2. Prompty i odpowiedzi strukturalne
   - Prompty w kodzie: instrukcje, przykłady, szablony, wersjonowanie
   - Odpowiedź w formacie JSON zgodnym ze schematem, walidacja danych
   - Wyciąganie danych z tekstu i dokumentów, klasyfikacja, streszczenia
   - Obsługa błędów, ponowienia, zachowanie przy odpowiedzi niezgodnej z oczekiwaniem
3. Wywoływanie narzędzi (function calling)
   - Własne funkcje jako narzędzia modelu: opis, parametry, wynik
   - Pętla model – narzędzie – model
   - Baza danych i zewnętrzne API jako narzędzia
   - Narzędzia wbudowane: wyszukiwanie w sieci, wykonywanie kodu
4. Pierwsza aplikacja
   - Prosty interfejs czatu w Streamlit
   - Usługa REST w FastAPI
   - Przechowywanie historii rozmów
   - Klucze API i dane poufne w konfiguracji
5. RAG: odpowiedzi na podstawie własnych dokumentów
   - Embeddingi i wyszukiwanie semantyczne
   - Baza wektorowa (Chroma lub analogiczna)
   - Wczytywanie dokumentów (PDF, DOCX), podział na fragmenty, metadane
   - Odpowiedź ze wskazaniem źródeł; kiedy model ma odpowiedzieć „nie wiem”
   - Poprawa trafności: wyszukiwanie hybrydowe, filtrowanie po metadanych, reranking
   - RAG w bibliotece LangChain: ten sam potok krócej; kiedy biblioteka pomaga, a kiedy lepiej bez niej
6. Ewaluacja, czyli jak sprawdzić, że działa
   - Zestaw pytań testowych i oczekiwanych odpowiedzi
   - Miary jakości wyszukiwania i wierności odpowiedzi względem źródeł
   - Ocena odpowiedzi przez drugi model
   - Testy regresji po zmianie promptu, modelu lub danych
7. Agenci AI w kodzie
   - Agent a zwykły przepływ: kiedy które rozwiązanie
   - Agent z narzędziami w OpenAI Agents SDK
   - RAG jako narzędzie agenta, pamięć i stan
   - Rzut okiem na układy wieloagentowe
8. Model Context Protocol (MCP)
   - Podłączanie gotowych serwerów MCP do aplikacji i agenta
   - Własny prosty serwer MCP udostępniający dane firmowe
   - Uprawnienia: co agent może czytać, a co zmieniać
9. Bezpieczeństwo i utrzymanie
   - Prompt injection, także z dokumentów i wyników narzędzi
   - Zatwierdzanie działań przez człowieka, ograniczanie narzędzi
   - Dane osobowe i poufne w zapytaniach do modelu
   - Logowanie i śledzenie wywołań, koszty, czas odpowiedzi, pamięć podręczna
10. Warsztat końcowy
   - Asystent do dokumentów z narzędziami lub własny scenariusz uczestnika
   - Omówienie rozwiązań i typowych błędów

Dla grup na zamówienie możemy oprzeć warsztaty o API innego dostawcy (np. Gemini lub Claude) albo o modele uruchamiane lokalnie, na własnej infrastrukturze. Szkolenie można też zrealizować w wersji 2-dniowej, bez części o agentach i MCP, lub rozszerzyć o tematy takie jak LangGraph, układy wieloagentowe, wdrożenie w chmurze czy dostrajanie modeli.

**Zapytaj o szczegóły**
tel. 22 63 64 164
akademia@alx.pl

## Przeznaczenie i wymagania

Szkolenie jest przeznaczone przede wszystkim dla:

— programistów Pythona i backendu, którzy mają dodać model językowy do aplikacji lub usługi,
— inżynierów danych i analityków piszących w Pythonie, którzy budują asystenta do firmowych dokumentów,
— DevOpsów i administratorów, którzy będą takie rozwiązania uruchamiać i utrzymywać,
— architektów i liderów technicznych, którzy decydują, co budować samodzielnie, i odbierają pracę od wykonawców.

Szkolenie będzie za proste dla osób, które mają już na produkcji własny system RAG albo agentów.

Świadomie nie uczymy:

— Pythona od podstaw,
— trenowania ani dostrajania modeli (fine-tuning) i klasycznego uczenia maszynowego,
— budowy agentów w narzędziach no-code,
— tworzenia rozbudowanego interfejsu użytkownika.

Od uczestników oczekujemy:

— podstawowej znajomości Pythona: funkcje, moduły, instalowanie i używanie bibliotek,
— ogólnej orientacji w tym, czym są REST API i JSON,
— choćby okazjonalnego korzystania z czatu AI.

Nie jest wymagane doświadczenie z API modeli językowych ani z uczeniem maszynowym. Osobom, które nie programowały w Pythonie, polecamy wcześniej szkolenie Skrypty w Pythonie.

Do ćwiczeń potrzebne jest aktywne konto na platformie OpenAI z dostępem do API i niewielkim doładowaniem środków (to osobne konto niż abonament ChatGPT). Więcej informacji podamy przed zajęciami. Dla grup na zamówienie możemy zapewnić konta i dostępy uczestnikom.

**Zapytaj o szczegóły**
tel. 22 63 64 164
akademia@alx.pl

## Certyfikaty

Uczestnicy szkolenia otrzymują imienne certyfikaty sygnowane przez ALX.

## Lokalizacje

— Warszawa – ul. Jasna 14/16A
— Zdalnie – zajęcia realizowane poprzez platformę Zoom
— Kraków – ul. św. Filipa 23
— Warsaw (English) – Jasna 14/16A
— Online (English) – your home, office or wherever you want
— na życzenie dowolne miejsce w Polsce, lub UE (zajęcia prowadzone w języku angielskim)

## Cena szkolenia

2690 PLN netto (VAT 23%)

W cenę szkoleń organizowanych w naszej siedzibie wliczone są:

— autorskie materiały szkoleniowe,
— indywidualne stanowisko komputerowe do pracy podczas zajęć,

— certyfikaty ukończenia szkolenia,
— drobny poczęstunek oraz ciepłe i zimne napoje,
— możliwość jednorazowego kontaktu z instruktorem (instruktorami) po szkoleniu i zadawania pytań dotyczących materiału szkolenia.

Cena szkolenia nie zawiera obiadów. Można je dokupić w cenie 35 zł netto za obiad.

**Zapytaj o szczegóły**
tel. 22 63 64 164
akademia@alx.pl