

## Python Developer XL - backend/web (kod: K-PYTHON-XL)

### Opis i cel kursu

Python jest jednym z najbardziej przystępnych oraz uniwersalnych języków programowania. Właśnie ta wszechstronność powoduje, że jest on często wybierany nawet przez osoby nie będące programistami do uproszczenia ich codziennej pracy.

Nasz kurs przeznaczony jest dla osób nigdy nie mających styczności z programowaniem, a Pythona z założenia może nauczyć się każdy.

Bootcamp **Python XL Backend** jest kompletnym kursem języka, prowadzonym od podstaw, wraz z nauką programowania - stopniowo do coraz bardziej złożonych konstrukcji, narzędzi i różnych ciekawych trików Pythona. Poznasz tu klasyczne i najbardziej popularne sposoby wykorzystania języka Python do automatyzacji czynności systemowych, operacji na plikach, tworzenia aplikacji różnego typu. W drugiej części kursu poświęcimy więcej czasu na tworzenie rozbudowanych aplikacji webowych przy pomocy najpopularniejszego frameworku **Django**, z wykorzystaniem baz danych. Przy okazji, poznając podstawowe, typowe schematy (wzorce) projektowania nieco większych aplikacji i myślenia o nich. Dowiesz się także, jak działa HTTP, co to jest REST, jak w projektach wykorzystywać GIT, TDD i inne techniki pracy.

Wspomnimy też o metodach deploymentu aplikacji.

### Program

- Wprowadzenie do programowania**
  - podstawy działania komputera
  - systemy operacyjne, programy, frameworki i biblioteki
  - przegląd języków programowania
- Wstęp do języka Python**
  - geneza i historia Pythona
  - zastosowania i możliwości
  - Python 2.x vs. Python 3.x
- Instalacja i konfiguracja środowiska**
  - interpreter języka Python
  - wirtualne środowisko (venv)
  - zintegrowane środowisko programistyczne (IDE) (edytor `PyCharm`)
- Podstawy składni języka Python**
  - interakcja z użytkownikiem
  - zmienne i podstawowe typy danych
  - struktury danych
  - instrukcja warunkowa
  - pętle
  - wyrażenia "comprehension"
- Programowanie proceduralne**
  - podstawy definiowania funkcji
  - przekazywania argumentów
  - dokumentacja i adnotacje
- Programowanie obiektowe**
  - podstawy definiowania klas
  - metody specjalne
  - metody statyczne i klasowe

### Zapytaj o szczegóły

tel. 22 63 64 164

akademia@alx.pl

### Najbliższe terminy

2024-12-21 (Zdalnie)

2024-12-21 (Warszawa)

2025-02-20 (Zdalnie)

2025-02-20 (Warszawa)

2025-03-22 (Zdalnie)

2025-03-22 (Warszawa)

2025-04-03 (Zdalnie)

2025-04-03 (Warszawa)

- dziedziczenie
- 7. **Obsługa wyjątków**
  - rzucanie i przechwytywanie wyjątków
  - definiowanie własnych wyjątków
- 8. **Organizacja kodu**
  - moduły i pakiety
  - struktura projektu
- 9. **Biblioteka standardowa języka Python**
  - wyrażenia regularne (moduł ``re``)
  - obsługa parametrów linii poleceń (moduł ``argparse``)
  - obsługa daty i czasu (moduł ``datetime``)
  - zaawansowane kolekcje (moduł ``collections``)
  - graficzny interfejs użytkownika (GUI) (moduł ``tkinter``)
- 10. **Instalacja zewnętrznych bibliotek**
  - repozytorium pakietów (PyPI)
  - instalator pakietów (narzędzie ``pip``)
  - zarządzanie zależnościami w projekcie
- 11. **Operacje wejścia/wyjścia**
  - operacje na plikach
  - zapytania HTTP
  - serializacja (moduł ``json`` i ``pickle``)
- 12. **Testowanie i debugowanie oprogramowania**
  - testowanie z wykorzystaniem biblioteki standardowej (moduł ``unittest``)
  - zewnętrzne narzędzia wspierające testowanie (projekt ``pytest``)
  - debugowanie kodu (moduł ``pdb`` oraz debugger wbudowany w ``PyCharm``)
- 13. **Zastosowania języka Python**
  - aplikacje webowe (framework ``Django``)
  - analiza i wizualizacja danych (biblioteka ``pandas`` i ``matplotlib``)
  - obliczenia naukowe (biblioteka ``SciPy`` i ``NumPy``)
- 14. **Dobre praktyki programowania w Pythonie**
  - styl kodowania i konwencje (PEP8)
  - idiomatyczne rozwiązania
- 15. **Python oczami programisty backend**
  - Wykorzystywane struktury danych
  - Wykorzystywane koncepcje programistyczne
  - Powtórzenie koniecznych podstaw
- 16. **Struktura projektu pythonowego, dodatkowe narzędzia**
  - Projekt jako moduł; zależności między projektami, struktura katalogu
  - GIT i kontrola wersji kodu
    - Zapisywanie i odtwarzanie historii zmian
    - Rozgałęzienia i ich późniejsze łączenie (branch/merge)
    - Repozytoria lokalne i zdalne; różne scenariusze pracy zespołowej
- 17. **Testy i kontrola jakości kodu**
  - TDD w praktyce
  - Zapewnianie poprawności oprogramowania
  - Testy jednostkowe
    - Istota testu jednostkowego
    - Typowa postać testu, proste asercje, uruchamianie
    - Dodatkowe moduły przydatne w pisaniu testów: Mock, patch
- 18. **Warstwa prezentacji**
  - Model DOM
  - Projektowanie witryn od strony programisty – HTML i CSS
  - Elementy formularza, metody POST i GET, żądania HTTP

## Zapytaj o szczegóły

tel. 22 63 64 164

akademia@alx.pl

## Najbliższe terminy

2024-12-21 (Zdalnie)

2024-12-21 (Warszawa)

2025-02-20 (Zdalnie)

2025-02-20 (Warszawa)

2025-03-22 (Zdalnie)

2025-03-22 (Warszawa)

2025-04-03 (Zdalnie)

2025-04-03 (Warszawa)

- Renderowanie widoku po stronie serwera - technologie szablonowe
- AJAX i REST od strony frontendu
- 19. **Działanie strony serwerowej aplikacji webowej**
  - Protokół HTTP
  - Aplikacje uruchamiane na serwerze; serwery aplikacji
- 20. **Framework Django**
  - Poznanie Django jako platformy
  - Wstępna instalacja i konfiguracja Django
  - Konfiguracja i praca ze środowiskiem developerskim
  - Omówienie komponentów typowej aplikacji Django
  - Projekt,
  - Aplikacja,
  - Model,
  - Widok,
  - Obiektowy dostęp do bazy danych (ORM),
  - Tworzenie panelu administracyjnego,
  - i18n.
  - Tworzenie komponentów aplikacji, na przykładach.
    - Widoki
    - Szablony
    - Wykorzystanie szablonów w widokach
    - Różne sposoby tworzenia formularzy – ręczne i korzystające z gotowych mechanizmów.
    - Korzystanie z widoków generycznych.
    - Zasady dobrych praktyk programistycznych.
  - Stworzenie przykładowej aplikacji webowej
    - Wspólne stworzenie przykładowego projektu na platformie Django
- 21. **Model REST, REST API i Django**
  - Stworzenie przykładowego REST API przy wykorzystaniu Django
    - Omówienie modelu REST
    - Założenia REST w kontekście Django
    - Przydatne biblioteki
    - Implementacja aplikacji
- 22. **Dodatkowe narzędzia programisty backendowego**
  - Celery, RabbitMQ i inne
- 23. **Wdrożenie aplikacji**
  - Deployment aplikacji
    - Środowisko produkcyjne (omówienie serwerów webowych i serwerów aplikacji)
    - Różne sposoby wdrożenia aplikacji na produkcji

## Zapytaj o szczegóły

tel. 22 63 64 164

akademia@alx.pl

## Najbliższe terminy

2024-12-21 (Zdalnie)

2024-12-21 (Warszawa)

2025-02-20 (Zdalnie)

2025-02-20 (Warszawa)

2025-03-22 (Zdalnie)

2025-03-22 (Warszawa)

2025-04-03 (Zdalnie)

2025-04-03 (Warszawa)

## Przeznaczenie i wymagania

Kurs przeznaczony jest dla osób, które chcą poznać Pythona z myślą o tworzeniu aplikacji internetowych.

Od uczestników wymagana jest znajomość ogólnej obsługi komputera (kopiowanie plików, korzystanie z edytora tekstu) i umiejętność sprawnego korzystania z klawiatury.

Programowanie, szczególnie na początku, jest trudne, a jego nauka jest porównywalna z nauką obcego języka lub gry na instrumencie muzycznym. Dlatego warunkiem skutecznego uczestnictwa w kursie jest zarezerwowanie odpowiedniej ilości czasu w domu na ćwiczenia.

## Certyfikaty

Uczestnicy szkolenia otrzymują imienne certyfikaty sygnowane przez ALX.

### Zapytaj o szczegóły

tel. 22 63 64 164

akademia@alx.pl

### Najbliższe terminy

2024-12-21 (Zdalnie)

2024-12-21 (Warszawa)

2025-02-20 (Zdalnie)

2025-02-20 (Warszawa)

2025-03-22 (Zdalnie)

2025-03-22 (Warszawa)

2025-04-03 (Zdalnie)

2025-04-03 (Warszawa)