



ALX Academy Sp. z  
o.o.

★★★★★ 4,7 / 5

5 ocen

## Python Developer XL - analiza danych i AI

Numer usługi 2025/12/08/47449/3198455

📍 Warszawa / mieszana (stacjonarna połączona z usługą  
zdalną w czasie rzeczywistym)

🏠 Usługa szkoleniowa

🕒 176 h

📅 20.12.2025 do 14.06.2026

12 839,97 PLN brutto

10 439,00 PLN netto

72,95 PLN brutto/h

59,31 PLN netto/h

## Informacje podstawowe

### Kategoria

Informatyka i telekomunikacja / Programowanie

### Grupa docelowa usługi

Kurs prowadzony od podstaw języka Python skierowany do tych uczestników, którzy wcześniej nie mieli styczności z programowaniem. Odnajdą się na nim osoby, które posiadają analityczne podejście do problemów, systematyczność oraz nie boją się rozwiązywania logicznych problemów. Kursantami mogą być osoby chcące zmienić branżę, poznać nowy język programowania, szukające nowego zawodu, pracownicy zdobywający nowe kompetencje i kwalifikacje czy freelancerzy chcący świadczyć kolejne usługi na rynku.

### Minimalna liczba uczestników

6

### Maksymalna liczba uczestników

16

### Data zakończenia rekrutacji

19-12-2025

### Forma prowadzenia usługi

mieszana (stacjonarna połączona z usługą zdalną w czasie rzeczywistym)

### Liczba godzin usługi

176

### Podstawa uzyskania wpisu do BUR

Certyfikat systemu zarządzania jakością wg. ISO 9001:2015 (PN-EN ISO 9001:2015) - w zakresie usług szkoleniowych

## Cel

### Cel edukacyjny

Kurs prowadzony od podstaw języka Python skierowany do tych uczestników, którzy wcześniej nie mieli styczności z programowaniem. Odnajdą się na nim osoby, które posiadają analityczne podejście do problemów, systematyczność oraz nie boją się rozwiązywania logicznych problemów. Kursantami mogą być osoby chcące zmienić branżę, poznać nowy język programowania, szukające nowego zawodu, pracownicy zdobywający nowe kompetencje i kwalifikacje czy freelancerzy chcący świadczyć kolejne usługi na rynku.

## Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

| Efekty uczenia się                                                                                                | Kryteria weryfikacji                                                                                                                                                                                                                                                                                                                                                                                                                                  | Metoda walidacji |
|-------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| W zakresie wiedzy uczestnik wykazuje się znajomością podstawowych elementów języka Python.                        | <ul style="list-style-type: none"><li>- opanował wiedzę teoretyczną z zakresu podstaw programowania,</li><li>- monitoruje i kontroluje środowisko Python,</li><li>- definiuje najważniejsze pojęcia.</li></ul>                                                                                                                                                                                                                                        | Test teoretyczny |
| W zakresie umiejętności uczestnik opanował podstawy i operuje uniwersalnymi aspektami programowania komputerowego | <ul style="list-style-type: none"><li>- charakteryzuje uniwersalne koncepcje programowania komputerowego,</li><li>- rozróżnia typy danych, kontenery, funkcje, warunki i pętle,</li><li>- kontroluje składnię i semantykę języka programowania Python</li><li>- obsługuje środowisko wykonawcze.</li><li>- pisze funkcje,</li><li>- obsługuje/zgłasza wyjątki,</li><li>- pracuje z narzędziami ułatwiającymi pisanie programów, debugowanie</li></ul> | Test teoretyczny |
| W zakresie umiejętności społecznych uczestnik używa zdobytej wiedzy w celu usprawnienia swojej pracy.             | <ul style="list-style-type: none"><li>- wykazuje umiejętność pracy w zespole,</li><li>- organizuje i realizuje zadania przy pracy nad projektem grupowym,</li><li>- charakteryzuje się odpowiedzialnością za realizację przydzielonych zadań,</li></ul>                                                                                                                                                                                               | Test teoretyczny |

## Kwalifikacje

### Kompetencje

Usługa prowadzi do nabycia kompetencji.

#### Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji zawiera opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument potwierdza, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji?

TAK

Pytanie 3. Czy dokument potwierdza zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

# Program

| Lp. | Temat zajęć edukacyjnych                                                                                                                                                                                                                                                                           | Liczba godzin zajęć teoretycznych | Liczba godzin zajęć praktycznych |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|----------------------------------|
| 1   | Wprowadzenie do programowania i środowiska – Podstawy działania komputera – Systemy operacyjne – Przegląd języków – Wstęp do Python (Geneza, Wersje 2.x vs 3.x) – Instalacja i konfiguracja środowiska – Interpreter – Wirtualne środowisko (venv) – Konfiguracja IDE (PyCharm) – Pierwsze skrypty | 4                                 | 4                                |
| 2   | Podstawy składni języka Python – Interakcja z użytkownikiem (input/output) – Zmienne i typy danych – Operatory – Struktury danych: Listy i ich metody – Krotki – Słowniki i Zbiory – Mutowalność vs niemutowalność – Konwersja typów                                                               | 4                                 | 4                                |
| 3   | Sterowanie przepływem programu – Instrukcje warunkowe (if, else, elif) – Pętle (for, while) – Instrukcje sterujące (break, continue, pass) – Wyrażenia typu "comprehension" (list, dict comprehension) – Zagnieżdżone struktury danych                                                             | 4                                 | 4                                |
| 4   | Programowanie proceduralne – Definiowanie funkcji – Argumenty pozycyjne i nazwane – Wartości domyślne – *args i **kwargs – Zasięg zmiennych (local vs global) – Funkcje lambda – Dokumentacja (docstrings) i adnotacje typów                                                                       | 4                                 | 4                                |
| 5   | Programowanie obiektowe (OOP) – Część 1 – Paradygmat obiektowy – Klasy i obiekty – Atrybuty i metody instancji – Konstruktor __init__ – Parametr self – Enkapsulacja i ukrywanie danych                                                                                                            | 4                                 | 4                                |
| 6   | Programowanie obiektowe (OOP) – Część 2 – Dziedziczenie – Polimorfizm – Metody specjalne (magiczne) – Metody statyczne i klasowe – Dekoratory w Pythonie – Abstrakcja                                                                                                                              | 4                                 | 4                                |
| 7   | Organizacja kodu i obsługa błędów – Moduły i pakiety – Importowanie – Struktura projektu Python – __main__ – Obsługa wyjątków (try, except, else, finally) – Rzucanie wyjątków (raise) – Tworzenie własnych wyjątków                                                                               | 4                                 | 4                                |
| 8   | Biblioteka standardowa i operacje na plikach – Moduły: os, sys, pathlib – Obsługa daty i czasu (datetime, time) – Kolekcje (collections) – Operacje na plikach tekstowych (open, read, write) – Serializacja danych (JSON, Pickle) – Kontekst menedżerowie (with)                                  | 4                                 | 4                                |
| 9   | Narzędzia profesjonalisty i jakość kodu – Instalacja bibliotek zewnętrznych (PyPI, pip) – Zarządzanie zależnościami (requirements.txt) – Styl kodowania PEP8 – Lintery – Debugowanie kodu (pdb, debugger w IDE) – Podstawy testowania jednostkowego (unittest, pytest)                             | 4                                 | 4                                |
| 10  | Praktyczne zastosowania Pythona – Wstęp do tworzenia aplikacji – Wyrażenia regularne (re) – Obsługa argumentów linii poleceń (argparse) – Podstawy komunikacji sieciowej (HTTP requests) – Wstęp do aplikacji webowych (koncepty frameworków np. Django/Flask)                                     | 4                                 | 4                                |

|                         |                                                                                                                                                                                                                                                                |           |           |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-----------|
| 11                      | Środowisko pracy Data Science — Anaconda i Conda — Jupyter Notebook i Jupyter Lab — Markdown — Notacja LaTeX — Wstęp do NumPy — Tablice (arrays) — Typy danych w NumPy — Wydajność vs listy Pythona                                                            | 4         | 4         |
| 12                      | Obliczenia numeryczne z NumPy — Operacje na macierzach i wektorach — Indeksowanie i wycinanie (slicing) — Wektoryzacja operacji — Broadcasting — Algebra liniowa w NumPy — Generowanie liczb losowych — Rozwiązywanie układów równań                           | 4         | 4         |
| 13                      | Przetwarzanie danych z Pandas — Część 1 — Struktury: Series i DataFrame — Wczytywanie danych (CSV, Excel, SQL, JSON) — Eksploracja danych (head, describe, info) — Indeksowanie i selekcja danych (loc, iloc) — Filtrowanie danych                             | 4         | 4         |
| 14                      | Przetwarzanie danych z Pandas — Część 2 — Czyszczenie danych (Data Cleaning) — Obsługa brakujących danych (Missing Data) — Usuwanie duplikatów — Zmiana typów danych — Operacje na napisach w Pandas — Funkcje apply i map                                     | 4         | 4         |
| 15                      | Zaawansowane operacje na danych — Grupowanie i agregacja (groupby) — Tabele przestawne (pivot tables) — Łączenie zbiorów danych (merge, join, concat) — Przekształcanie wymiarów (reshaping, stack/unstack) — Szeregi czasowe w Pandas                         | 4         | 4         |
| 16                      | Wizualizacja i Analiza Statystyczna — Matplotlib (tworzenie wykresów, formatowanie) — Seaborn (wizualizacje statystyczne) — Wykresy interaktywne (wstęp) — Podstawy statystyki opisowej — Korelacja — Rozkłady zmiennych — Testowanie hipotez                  | 4         | 4         |
| 17                      | Machine Learning — Fundamenty — Podział metod (nadzorowane, nienadzorowane) — Proces ML — Przygotowanie zbiorów (treningowy, testowy, walidacyjny) — Feature Engineering — Skalowanie i normalizacja danych — Problem przeuczenia (Overfitting) i niedouczenia | 4         | 4         |
| 18                      | Algorytmy uczenia nadzorowanego — Regresja liniowa i wielomianowa — Regresja logistyczna — Metryki oceny regresji i klasyfikacji — Drzewa decyzyjne — Lasy losowe — Maszyny wektorów nośnych (SVM) — K-4 najbliższych sąsiadów (KNN)                           | 4         | 4         |
| 19                      | Uczenie nienadzorowane i optymalizacja — Klasteryzacja (K-means, hierarchiczna) — Redukcja wymiarowości (PCA) — Dobór hiperparametrów (Grid Search) — Walidacja krzyżowa (Cross-validation) — Pipelines w Scikit-learn                                         | 4         | 4         |
| 20                      | Wstęp do Deep Learning i Sieci Neuronowych — Budowa neuronu — Perceptron — Funkcje aktywacji (ReLU, Sigmoid, Softmax) — Propagacja wsteczna (Backpropagation) — Funkcje straty — Biblioteki DL: TensorFlow vs PyTorch — Wstęp do Keras API                     | 4         | 4         |
| 21                      | Architektury sieci głębokich — Wielowarstwowe perceptrony (MLP) — Konwolucyjne sieci neuronowe (CNN) do analizy obrazu — Warstwy konwolucyjne i pooling — Rekurencyjne sieci neuronowe (RNN, LSTM) do analizy sekwencji — Transfer Learning                    | 4         | 4         |
| 22                      | Nowoczesne AI i Projekt Końcowy — Modele Transformers (Hugging Face) — Wstęp do JAX — Wdrażanie modeli (Deployment - wstęp) — Etyka w AI — Projekt zaliczeniowy: End-to-end ML project (od danych do predykcji) — WALIDACJA EFEKTÓW UCZENIA                    | 4         | 4         |
| <b>OG<br/>ÓŁE<br/>M</b> | <b>LICZBA GODZIN NAUCZANIA</b>                                                                                                                                                                                                                                 | <b>88</b> | <b>88</b> |

# Harmonogram

Liczba przedmiotów/zajęć: 0

| Przedmiot /<br>temat zajęć | Prowadzący | Data realizacji<br>zajęć | Godzina<br>rozpoczęcia | Godzina<br>zakończenia | Liczba godzin | Forma<br>stacjonarna |
|----------------------------|------------|--------------------------|------------------------|------------------------|---------------|----------------------|
| Brak wyników.              |            |                          |                        |                        |               |                      |

## Cennik

### Cennik

| Rodzaj ceny                               | Cena          |
|-------------------------------------------|---------------|
| Koszt przypadający na 1 uczestnika brutto | 12 839,97 PLN |
| Koszt przypadający na 1 uczestnika netto  | 10 439,00 PLN |
| Koszt osobogodziny brutto                 | 72,95 PLN     |
| Koszt osobogodziny netto                  | 59,31 PLN     |

## Prowadzący

Liczba prowadzących: 1



1 z 1

### Robert Wasik

Wykładowca języka JavaScript, Python, Java. Prowadzi również zajęcia z Tworzenia stron WWW oraz bootcamp Programista PHP. Zawodowy trener z dużym doświadczeniem praktycznym – poza prowadzeniem szkoleń posiada doświadczenie z pracy zarówno w dużym zespole programistycznym spółki telekomunikacyjnej, jak i przy niewielkich projektach, dzięki czemu rozumie realia pracy przy różnych projektach. W ALX, poza prowadzeniem szkoleń, zajmuje się pracowaniem programów szkoleń, materiałów szkoleniowych oraz ćwiczeń z technologii Front-endowych. Jako trener przeprowadził kilka tysięcy godzin szkoleniowych dotyczących głównie programowania w PHP oraz JavaScript. Jest jednym z najwyżej ocenianych trenerów w ALX, średnia ocen trenerskich z 50 ankiet spośród 10 ostatnich szkoleń wynosi 4,85 w 5-stopniowej skali ocen. Kursanci również bardzo wysoko oceniają jego prezentacje tworzone w ramach materiałów szkoleniowych.

## Informacje dodatkowe

### Informacje o materiałach dla uczestników usługi

Uczestnicy otrzymują autorskie materiały szkoleniowe ALX

## Informacje dodatkowe

Szkolenie może podlegać zwolnieniu z podatku VAT w w przypadku gdy udział w usłudze jest finansowany co najmniej w 70% ze środków publicznych.

## Warunki techniczne

### Czego potrzebujesz do kursu zdalnego?

- komputer z dostępem do Internetu (zalecana prędkość łącza: min 3 Mbit/s download/upload; całkowicie wystarczające są w szczególności połączenia przez sieć komórkową, oby jedynie były one stabilne - nie zrywające się)
- przeglądarka internetowa (Chrome, Firefox, Safari, Edge itp.),
- głośniki lub słuchawki, oraz mikrofon (aby słyszeć i rozmawiać z trenerem oraz innymi uczestnikami szkolenia).

### Opcjonalnie:

W miarę posiadania, można też wyposażyć stanowisko pracy w dodatkowy monitor. Jest wtedy możliwość jednoczesnego obserwowania udostępnionego obrazu (na jednym ekranie) i pracy w swoim edytorze/arkuszu/środowisku na drugim ekranie. Jeśli jednak nie posiadasz dodatkowego monitora, to również nie ma się czym martwić. Wystarczy przełączanie się między oknami w razie potrzeby, w ten sposób pracuje większość naszych kursantów.

Alternatywnie, istnieje jeszcze możliwość uruchomienia oprogramowania konferencyjnego (standardowo w ALX jest to Zoom) na tablecie lub ewentualnie telefonie Apple / Android (przy czym ekran telefonu jest jednak dość mały - kursanci raczej preferują tablety). Wtedy można uczestniczyć w sesji video poprzez urządzenie mobilne, a samodzielną pracę i ćwiczenia - wykonywać na komputerze.

## Adres

ul. Jasna 14/16A  
00-041 Warszawa  
woj. mazowieckie

### Udogodnienia w miejscu realizacji usługi

- Klimatyzacja
- Wi-fi
- Laboratorium komputerowe

## Kontakt



**Justyna Kamińska**

**E-mail** zakupy@alx.pl

**Telefon** (+48) 604 280 025