



Programowanie w języku Python - poziom średnio zaawansowany

Numer usługi 2026/02/13/202247/3333655

2 952,00 PLN brutto

2 400,00 PLN netto

140,57 PLN brutto/h

114,29 PLN netto/h

157,50 PLN cena rynkowa ⓘ

JSYSTEMS SPÓŁKA
Z OGRANICZONĄ
ODPOWIEDZIALNOŚ
CIĄ

★★★★★ 4,6 / 5

5 ocen

- 🏠 Usługa szkoleniowa
- 📄 zdalna w czasie rzeczywistym
- 👥 Zajęcia grupowe
- 🕒 21:00 h
- 📅 07.12.2026 do 09.12.2026

Informacje podstawowe

Kategoria	Informatyka i telekomunikacja / Programowanie
Grupa docelowa usługi	Szkolenie jest odpowiednie zarówno dla osób, które mają już podstawy Pythona i chcą je pogłębić, jak i dla tych, którzy chcą uporządkować oraz usystematyzować swoją wiedzę przed przejściem do bardziej zaawansowanych tematów i praktycznych zastosowań.
Minimalna liczba uczestników	6
Maksymalna liczba uczestników	10
Data zakończenia rekrutacji	03-12-2026
Forma prowadzenia usługi	zdalna w czasie rzeczywistym
Liczba godzin usługi	21
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

Po ukończeniu tego szkolenia uczestnik będzie potrafił:

- pobierać wartości z list,
- modyfikować zawartości list,
- tworzyć słowniki,

- optymalizować funkcje przez cache,
- uruchamiać testy z konsoli.

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Po ukończeniu tego szkolenia uczestnik będzie potrafił: • pobierać wartości z list, • modyfikować zawartości list, • tworzyć słowniki, • optymalizować funkcje przez cache, • uruchamiać testy z konsoli.	Każde omawiane zagadnienie jest utrwalane rzędem ćwiczeń o wzrastającym poziomie trudności. Podczas szkolenia uczestnicy wykonują ponad 50 warsztatów. W szkoleniu duży nacisk jest kładziony na dobre zrozumienie zasad działania wykorzystywanych technik, a nie ich odtwórcze stosowanie.	Obserwacja w warunkach rzeczywistych

Kwalifikacje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem zawierają opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji i zgodnie z zaplanowanymi metodami walidacji?

TAK

Pytanie 3. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

Program

1. Listy

- o Tworzenie list
- o Pobieranie wartości z list
- o Iterowanie po listach
- o Sprawdzanie czy element znajduje się na liście
- o Modyfikowanie zawartości listy

- o Funkcje wbudowane w listy
- o Sortowanie i odwracanie list
- o Inne ciekawe funkcje i możliwości

2. Słowniki

- o Tworzenie słowników
- o Pobieranie wartości ze słowników
- o Modyfikacja zawartości słowników

3. Powrót do funkcji

- o *args
- o **kwargs
- o Funkcja wewnątrz funkcji
- o Funkcja jako argument funkcji
- o Funkcja zwracająca funkcje
- o Rekurencja
- o Optymalizacja funkcji przez cache

4. Generatory

- o yield
- o Generatory skończone
- o Generatory nieskończone

5. Obiektość

- o Klasy a instancje
- o Atrybuty
 - ▮ Atrybuty domyślne
 - ▮ Obiekty jako atrybuty obiektu ?
- o Metody
 - ▮ Dlaczego lepsze niż zewnętrzne funkcje ?
 - ▮ Widoczność
- o Konstruktor __init__
 - ▮ self
 - ▮ Parametryzacja konstruktora
- o Dziedziczenie
 - ▮ Dziedziczenie po jednej klasie
 - ▮ Dziedziczenie po wielu klasach
 - ▮ Kolejność dziedziczenia
 - ▮ Metody i klasy abstrakcyjne
- o Metody magiczne

▣ __str__

▣ __getitem__

▣ __setitem__

o Hermetyzacja

▣ Ukrywanie atrybutów

▣ Próby nadpisania z zewnątrz

▣ Kontrolowany dostęp

o Dokumentowanie klas

6. Iteratory

o __iter__

o __next__

7. Dekoratory

o Zasada działania

o Dekorator jako wzorzec projektowy

o Dekoratory funkcji

8. Pakiety

o Import pakietów

o Kolejne zagnieżdżenia

o __init__.py

o Pułapki w które można wpaść

9. Wprowadzenie do asynchroniczności w Pythonie

o Biblioteka threading

o Wątki jako demony

10. Konfiguracja Pythona

o Parametry wiersza poleceń - moduł click

▣ Stosowanie parametrów

▣ Stosowanie wielu parametrów

▣ Automatyczne generowanie pomocy

o Ręczne tworzenie wirtualnego środowiska

11. Testy jednostkowe

o pytest

o Uruchamianie testów z konsoli

o Makiety

o Sprawdzanie pokrycia kodem

Harmonogram

Liczba pozycji harmonogramu: 0

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
Brak wyników.					

Cennik

Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	2 952,00 PLN
Koszt przypadający na 1 uczestnika netto	2 400,00 PLN
Koszt osobogodziny brutto	140,57 PLN
Koszt osobogodziny netto	114,29 PLN

Prowadzący

Liczba prowadzących: 1



1 z 1

Przemysław Smaga

Trener programowania Python z doświadczeniem systemowym i komercyjnym, który łączy praktyczną wiedzę developerską z realnym doświadczeniem projektowym zdobytym w środowisku biznesowym. Posiada doświadczenie w tworzeniu, rozwijaniu i utrzymywaniu aplikacji opartych o Python – od skryptów automatyzujących, przez API, aż po systemy backendowe i integracje między systemami.

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Uczestnicy otrzymają komplet materiałów PDF

Warunki uczestnictwa

podstawowa znajomość języka Python, umiejętność korzystania z komputera.

Informacje dodatkowe

Szkolenie kończy się wydaniem certyfikatu imiennego

Warunki techniczne

Dostęp do internetu, laptop/komputer, wideoaplikacja ZOOM lub dostęp do wersji w przeglądarce, możliwość podłączenia się do hostów przygotowanych na pulpicie zdalnym.

Kontakt



Biuro Obsługi Klienta

E-mail biuro@jsystems.pl

Telefon (+48) 534 506 503