



Programowanie w języku Python

Numer usługi 2025/04/02/168224/2666135

4 200,00 PLN brutto

4 200,00 PLN netto

105,00 PLN brutto/h

105,00 PLN netto/h

NTF.PL SPÓŁKA Z
OGRANICZONĄ
ODPOWIEDZIALNOŚ
CIĄ

📍 Lublin / stacjonarna

🏠 Usługa szkoleniowa

🕒 40 h

📅 22.04.2025 do 13.06.2025

Informacje podstawowe

Kategoria	Informatyka i telekomunikacja / Programowanie
Sposób dofinansowania	wsparcie dla osób indywidualnych wsparcie dla pracodawców i ich pracowników
Grupa docelowa usługi	Kurs przeznaczony jest dla osób, które: <ul style="list-style-type: none">• chcą w własnej inicjatywy nauczyć się programowania w języku Python• znają podstawy programowania w innym języku, i chcą rozszerzyć posiadane umiejętności• zmienić zawód• zdobyć nowe kompetencje i kwalifikacje
Minimalna liczba uczestników	1
Maksymalna liczba uczestników	3
Data zakończenia rekrutacji	15-04-2025
Forma prowadzenia usługi	stacjonarna
Liczba godzin usługi	40
Podstawa uzyskania wpisu do BUR	Standard Usługi Szkoleniowo-Rozwojowej PIFS SUS 2.0

Cel

Cel edukacyjny

Kurs programowania w języku Python przygotowuje uczestnika do pozyskania umiejętności samodzielnego tworzenia prostych aplikacji webowych w języku Python, pracy z narzędziami ułatwiającymi pisanie programów, debugowania.

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
<p>Rozumie podstawowe cechy języka Python</p> <p>Znajomość zastosowań Pythona w różnych dziedzinach.</p>	<p>Wyjaśnia czym jest Python i podać przykłady jego zastosowań</p> <p>Rozumie różnicę między kompilacją a interpretacją kodu</p>	<p>Obserwacja w warunkach symulowanych</p>
<p>Instaluje Pythona.</p> <p>Używa edytorów i IDE.</p> <p>Uruchamia skrypty w różnych środowiskach.</p> <p>Używa różnych typów danych (int, float, str).</p> <p>Wykonuje operacje na stringach i liczbach.</p> <p>Znajomość konwersji typów.</p>	<p>Poprawnie instaluje Pythona i wybrane IDE (np. VS Code, PyCharm).</p> <p>Tworzy i uruchamia prosty skrypt w wybranym środowisku.</p> <p>Poprawnie deklaruje zmienne różnych typów.</p> <p>Wykonuje operacje na stringach (np. split(), join()) i liczbach.</p> <p>Konwertuje typy danych (int(), str(), float()).</p>	<p>Obserwacja w warunkach symulowanych</p> <p>Obserwacja w warunkach symulowanych</p>
<p>Stosuje podstawowe struktury danych: listy, krotki, słowniki, zbiory.</p> <p>Potrafi manipulować danymi w strukturach.</p> <p>Pobiera dane od użytkownika (input()).</p> <p>Znajomość różnych metod formatowania wyjścia (f-string, .format()).</p> <p>Stosuje instrukcje if, elif, else.</p> <p>Stosuje operatory logiczne</p>	<p>Tworzy i modyfikuje listy, krotki, słowniki i zbiory.</p> <p>Stosuje podstawowe metody (append(), remove(), sort(), keys(), values()).</p> <p>Pisze program, który pobiera dane od użytkownika i wyświetla sformatowany wynik.</p> <p>Tworzy program wykorzystujący warunki logiczne.</p> <p>Uczestnik poprawnie stosuje operatory and, or, not.</p>	<p>Obserwacja w warunkach symulowanych</p> <p>Obserwacja w warunkach symulowanych</p> <p>Obserwacja w warunkach symulowanych</p>
<p>Stosuje pętle for i while.</p> <p>Stosuje instrukcje sterujące (break, continue, pass).</p>	<p>Pisze program wykorzystujący pętle do iteracji po danych.</p> <p>Stosuje break i continue w praktycznych przykładach.</p>	<p>Obserwacja w warunkach symulowanych</p>
<p>Odczytuje i zapisuje do plików.</p> <p>Stosuje kontekstowego menedżera with.</p>	<p>Tworzy skrypt odczytujący i zapisujący dane do pliku.</p>	<p>Obserwacja w warunkach symulowanych</p>

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Definiuje funkcji (def). Stosuje argumenty (*args, **kwargs, wartości domyślne). Rozumienie zasięg zmiennych (global, local).	Pisze funkcje z różnymi typami argumentów. Poprawnie stosuje funkcje lambda.	Obserwacja w warunkach symulowanych
Zna podstawy OOP (klasy, obiekty, dziedziczenie, polimorfizm). Definiuje metody i atrybuty.	Tworzy klasę z metodami i dziedziczy po innej klasie.	Obserwacja w warunkach symulowanych
Obsługuje wyjątki (try-except-finally). Zgłasza wyjątki (raise).	Pisze kod z obsługą błędów.	Obserwacja w warunkach symulowanych
Rozumienie działania generatorów (yield). Zna dekoratory (@staticmethod, @classmethod).	Tworzy prosty generator i dekorator.	Obserwacja w warunkach symulowanych
Debuguje kod (pdb, assert). Posiada znajomość logowania (logging).	Debuguje kod za pomocą pdb i stosuje assert.	Obserwacja w warunkach symulowanych

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?

Tak

Program

Ramowy program szkolenia:

1. Wprowadzenie do Pythona
 - Co to jest Python?

- Zastosowania języka
 - Jak działa interpreter Pythona?
2. Przygotowanie środowiska
 - Instalacja Pythona
 - Edytory i IDE (PyCharm, VS Code, Jupyter)
 - Uruchamianie skryptów
 3. Podstawowe typy danych i operacje
 - Zmienne i ich deklaracja
 - Liczby (int, float) i operacje arytmetyczne
 - Stringi i operacje na nich (len(), upper(), lower(), split(), join())
 - Konwersja typów (int(), str(), float())
 4. Wprowadzenie do struktur danych
 - Listy (list): podstawowe operacje, indeksowanie, append(), remove(), sort()
 - Krotki (tuple) – różnice między listami
 - Słowniki (dict) – klucz-wartość, podstawowe metody
 - Zbiory (set)
 5. Operacje wejścia/wyjścia
 - input() – pobieranie danych od użytkownika
 - print() i formatowanie (f-strings, .format())
 6. Instrukcje warunkowe i operatory logiczne
 - if, elif, else
 - Operatory porównania (==, !=, >, <)
 - Operatory logiczne (and, or, not)
 7. Pętle w Pythonie
 - while i for
 - break, continue, pass
 - range()
 8. Praca z plikami
 - Otwieranie plików (open(), with)
 - Odczyt i zapis (read(), write())
 9. Funkcje w Pythonie
 - Tworzenie funkcji (def)
 - Argumenty (*args, **kwargs)
 - Wartości domyślne
 - Zasięg zmiennych (global, local, nonlocal)
 - Funkcje anonimowe (lambda)
 10. Programowanie obiektowe (OOP)
 - Klasy i obiekty
 - __init__ i atrybuty obiektów
 - Metody instancji i klasy
 - Dziedziczenie (super())
 - Polimorfizm
 11. Moduły i biblioteki
 - Importowanie modułów (import, from ... import ...)
 - Moduły standardowe (math, random, datetime)
 - Tworzenie własnych modułów
 12. Obsługa błędów i wyjątków
 - try-except – podstawowe użycie
 - finally, else w obsłudze wyjątków
 - raise – ręczne zgłaszanie błędów
 13. Generatory i dekoratory
 - Funkcja jako obiekt pierwszej klasy
 - Generatory (yield)
 - Dekoratory (@staticmethod, @classmethod, własne dekoratory)
 14. Debugowanie kodu

- print() vs logging
- assert – testowanie warunków
- pdb – debugger wbudowany

Program realizowany jest w godzinach dydaktycznych (45 min + przerwy). Każdy z Uczestników ma zapewnione samodzielne stanowisko pracy.

Harmonogram

Liczba przedmiotów/zajęć: 0

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

Cennik

Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	4 200,00 PLN
Koszt przypadający na 1 uczestnika netto	4 200,00 PLN
Koszt osobogodziny brutto	105,00 PLN
Koszt osobogodziny netto	105,00 PLN

Prowadzący

Liczba prowadzących: 1



1 z 1

Marcin Michnowicz

Marcin Michnowicz
Osoba prowadząca usługę

Z wykształcenia informatyk, wykładowca z bogatym doświadczeniem w branży informatycznej i programistycznej, specjalizującym się w e-commerce oraz obsłudze komputera, posiada ponad 20 lat doświadczenia w branży IT, pracując jako programista, analityk systemów oraz konsultant IT dla wielu firm i branży.

Jego specjalizacja w e-commerce obejmuje szeroki zakres tematów, od projektowania sklepów internetowych, przez optymalizację konwersji, aż po strategię marketingu cyfrowego i analizę

danych. W obszarze obsługi komputera, nie tylko uczy podstawowych i zaawansowanych umiejętności obsługi systemów i aplikacji, ale również wdraża wiedzę dotyczącą bezpieczeństwa cyfrowego, zarządzania danymi i efektywnej pracy z narzędziami IT.

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Uczestnicy otrzymują materiały szkoleniowe tj.:

- skrypt,
- zaświadczenie ukończenia kursu

Informacje dodatkowe

Warunkiem otrzymania dofinansowania na opisane szkolenie jest spełnienie warunków przedstawionych przez Operatora, który jest dysponentem funduszy publicznych. Informacje o Operatorach dostępne są na stronie Bazy Usług Rozwojowych <https://serwis-uslugirozwojowe.parp.gov.pl/component/site/site/dofinansowania-bur/#programy-krajowe>

Podana cena obejmuje szkolenie objęte dofinansowaniem w wysokości przynajmniej 70% ze środków publicznych (zgodnie z zapisami prawodawstwa szkolenia zawodowe finansowane w min. 70% ze środków publicznych podlegają zwolnieniu z VAT na podstawie §3 ust.1 pkt 14 Rozp.Ministra Finansów z dn. 20 grudnia 2013 r. w sprawie zwolnień od podatku od towarów i usług oraz warunków stosowania tych zwolnień (Dz. U. 2013 poz. 1722 § 3 ust. 1 pkt 14). W przypadku pozyskania mniejszego dofinansowania do podanej ceny należy doliczyć 23% VAT.

Adres

ul. Bolesława Prusa 8/12
20-064 Lublin
woj. lubelskie

Udogodnienia w miejscu realizacji usługi

- Klimatyzacja
- Wi-fi

Kontakt



Marcin Michnowicz

E-mail biuro@ntf.pl

Telefon (+48) 787 009 274