



Comarch SA



Comarch Bootcamp - Junior Python Developer

Numer usługi 2025/01/17/7733/2508566

📍 zdalna w czasie rzeczywistym

🏠 Usługa szkoleniowa

🕒 208 h

📅 10.05.2025 do 28.09.2025

12 054,00 PLN brutto

9 800,00 PLN netto

57,95 PLN brutto/h

47,12 PLN netto/h

Informacje podstawowe

Kategoria	Informatyka i telekomunikacja / Administracja IT i systemy komputerowe
Identyfikator projektu	Małopolski Pociąg do kariery
Sposób dofinansowania	wsparcie dla osób indywidualnych wsparcie dla pracodawców i ich pracowników
Grupa docelowa usługi	Szkolenie prowadzone jest w formie intensywnego kursu i przeznaczone jest dla osób które chcą się przygotować do programowania w języku Python na poziomie zaawansowanym. Szkolenie dla programistów języka Python w zakresie Junior Python Developer <i>„Usługa również adresowana dla uczestników projektu Kierunek Kariera Zawodowa”</i>
Minimalna liczba uczestników	5
Maksymalna liczba uczestników	14
Data zakończenia rekrutacji	04-04-2025
Forma prowadzenia usługi	zdalna w czasie rzeczywistym
Liczba godzin usługi	208
Podstawa uzyskania wpisu do BUR	Znak Jakości Małopolskich Standardów Usług Edukacyjno-Szkoleniowych (MSUES) - wersja 2.0

Cel

Cel edukacyjny

Przygotowanie uczestnika do zawodu programisty języka Python w zakresie wymagań na stanowisko Junior Python Developer.

Wprowadzenie uczestników w zaawansowane zagadnienia związane z programowaniem w języku Python, w tym:

- przygotowanie uczestników do pisania programów opartych na złożonych strukturach języka Python.
- przygotowanie uczestników do pisania programów służących do analizy danych (big data i data science).

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Typuje zmienne i rozumie kolekcje	Uczestnik poprawnie deklaruje zmienne w Pythonie, przypisując odpowiednie typy (np. int, str, list, dict). Używa kolekcji takich jak listy, słowniki, zbiory i krotki, rozumie ich właściwości oraz zastosowanie w różnych kontekstach programistycznych.	Test teoretyczny
Programuje funkcyjnie i obiektowo w zaawansowany sposób	Uczestnik efektywnie stosuje funkcje wyższego rzędu, dekoratory oraz zamknięcia (closures) w paradygmacie funkcyjnym. Umiejętnie korzysta z programowania obiektowego, implementując klasy, dziedziczenie, polimorfizm, a także wzorce projektowe takie jak Singleton, Factory, czy Strategy.	Test teoretyczny
W złożony sposób pracuje z modułami	Uczestnik tworzy, importuje i zarządza własnymi modułami oraz bibliotekami, a także integruje zewnętrzne pakiety, zarządzając ich zależnościami za pomocą narzędzi takich jak pip. Potrafi organizować kod w moduły, co poprawia czytelność i utrzymanie aplikacji.	Test teoretyczny
Stosuje meta programowanie do realizacji modeli obiektowych	Uczestnik wykorzystuje techniki metaprogramowania, takie jak dynamiczne tworzenie klas, modyfikowanie atrybutów obiektów oraz używanie dekoratorów do dostosowywania zachowań obiektów w czasie wykonania.	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Wdraża programy współbieżne, równoległe i asynchroniczne	<p>Uczestnik implementuje aplikacje współbieżne i równoległe, wykorzystując wątki, procesy oraz mechanizmy takie jak asyncio do przetwarzania danych w tle. Potrafi zrealizować asynchroniczne operacje I/O, efektywnie wykorzystując mechanizmy asynchroniczne do optymalizacji wydajności.</p>	Test teoretyczny
<p>Programuje dostęp do baz danych, SQL i noSQL</p> <p>Stosuje język Python do obliczeń i analizy danych</p>	<p>Uczestnik tworzy aplikacje z dostępem do baz danych SQL (np. MySQL, PostgreSQL) i noSQL (np. MongoDB), wykorzystując odpowiednie biblioteki i ORM (np. SQLAlchemy). Potrafi tworzyć zapytania SQL, zarządzać połączeniami oraz operować na danych z różnych typów baz danych.</p> <p>Uczestnik efektywnie korzysta z bibliotek takich jak NumPy, Pandas, SciPy do przeprowadzania obliczeń matematycznych, analizy danych oraz operacji na dużych zbiorach danych. Stosuje odpowiednie metody analizy i transformacji danych, takie jak sortowanie, grupowanie czy agregowanie.</p>	<p>Test teoretyczny</p> <p>Test teoretyczny</p>
Tworzy aplikacje webowe z użyciem platformy Flask i Django	<p>Uczestnik projektuje i implementuje aplikacje webowe z wykorzystaniem frameworków Flask i Django, tworząc zarówno backend, jak i interfejsy API. Potrafi konfigurować routing, middleware, zarządzanie sesjami, uwierzytelnianie oraz autoryzację użytkowników.</p>	Test teoretyczny
Stosuje różne metody analizy danych	<p>Uczestnik stosuje techniki eksploracji danych (data exploration), wizualizacji (np. Matplotlib, Seaborn) oraz analizy statystycznej do odkrywania wzorców i trendów w zbiorach danych. Potrafi dobierać odpowiednie metody analizy w zależności od rodzaju danych i problemu.</p>	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Analizuje dane za pomocą algorytmów AI	<p>Uczestnik stosuje algorytmy sztucznej inteligencji, takie jak klasyfikacja, regresja, grupowanie czy algorytmy oparte na głębokim uczeniu (deep learning) do analizy danych.</p> <p>Wykorzystuje biblioteki AI, takie jak Scikit-learn, TensorFlow czy Keras, do tworzenia modeli predykcyjnych i rozwiązywania problemów analitycznych.</p>	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?

Tak

Program

Moduł A: Język Python, jego interpreter i środowiska IDE (4 godz.)

Programowanie w języku Python wymaga na początek zaznajomienia się z ideą języka, jego interpreterem, dokumentacją a następnie z różnymi środowiskami programistycznymi.

Język Python (2 godziny)

Geneza

Zastosowania

Mocne i słabe strony języka

Środowisko kompilacyjne, wersjonowanie

IDE (Zintegrowane Środowisko Programistyczne) dla Pythona (2 godz)

Porównanie różnych IDE dla języka Python

PyCharm a Microsoft Visual Studio 2019

Instalacja i konfiguracja środowisk

Strona www.python.org i jej zawartość

Moduł B: 2. Wstęp do programowania (4 godz.)

Celem modułu jest zapoznanie Uczestników z ideą programowania z podstawowymi pojęciami dotyczącymi ogólnie programowania: definicją, algorytmem, instrukcjami. Omówienie zagadnień związanych z typowaniem oraz zmiennymi.

Co to są języki programowania?

Algorytm – definicja oraz użycie

Instrukcje programistyczne

Przykłady programowania wybranych algorytmów. Jak algorytmy prezentują się w kodzie źródłowym?

Co to są zmienne i typy danych?

Analiza wybranych struktur języków programowania

Moduł C: Podstawy programowania w języku Python (36 godz.)

Celem modułu jest wprowadzenie uczestnika w świat programowania w języku Python. Zapoznanie ze składnią języka oraz podstawowymi strukturami. Wprowadzenie do programowania funkcyjnego, jak również obiektowego.

Typy danych w języku Python

Rodzaje typów wbudowanych (wewnętrznych) – operacje na danych oraz metody specyficzne dla typu

Liczby

Łańcuchy znaków (tekst)

Logiczne typy danych (prawda, fałsz)

Struktury danych – ich charakterystyka i metody

Listy

Krotki

Zbiory

Słowniki

Instrukcje Sterujące (operacje I/O, przypisania, warunki i pętle)

Podstawowe zasady tworzenia kodu w języku Python – znaczenie wcięć. Jak pisać w Pythonie?

Dobre praktyki w pisaniu kodu – PEP8 – dokument definiujący reguły gramatyczne języka Python

Idea instrukcji sterujących

Przypisanie

Operacje wejścia/wyjścia

Instrukcje warunkowe (if)

Instrukcje iteracyjne – pętle (while i for)

Wielokrotne powtarzanie czynności – operacje na typach w strukturach danych.

Zasady programowania w języku Python

Styl PEP8

Interpolacja ciągów string

Wdrożenie zasad programowania i konstruowania programów

Używanie funkcji i obiektów

Zasady użycia instrukcji

Programowanie kolekcji

Kolekcje: listy, krotki, zbiory oraz słowniki

Operacje na danych

Stosowanie dedykowanych metod dla kolekcji

Asocjacje key – value

Zastosowania metody `__missing__()`

Zasady tworzenia dokumentacji w języku Python

Komentarze

Notatki dokumentacyjne (`__doc__`)

PyDoc

Pliki

Pliki binarne i tekstowe

Ścieżki i dostęp do plików

Operacje na plikach

Pliki CSV

Funkcje

Podstawy funkcji

Zasady tworzenia i nazewnictwa

Wywoływanie funkcji

Polimorfizm w funkcjach

Parametry w funkcjach

Wybrane aspekty użycia funkcji

Funkcje zagnieżdżone

Zasięgi

Argumenty funkcji

Zaawansowane zagadnienia dotyczące funkcji

Funkcje rekurencyjne

Funkcje Lambda

Listy składane i generatory

Wstęp do tworzenia modułów

Tworzenie modułów

Używanie modułów

Pakiety modułów

Przestrzenie nazw

Wstęp do programowania obiektowego w języku Python

Co to jest programowanie obiektowe?

Definicja obiektu

Paradygmaty programowania obiektowego

Obiekty

Czym jest klasa?

Tworzenie klas – instancja, konstruktor, metody

Dziedziczenie – wielodziedziczenie

Przeciążanie operatorów – tworzenie różnych wariantów operatorów

Klasy mieszane

Klasy abstrakcyjne, interfejsy

Wyjątki, obsługa błędów

Podstawy obsługi wyjątków – hierarchia

Instrukcja try/except/else/finally

Instrukcja raise

Instrukcja assert

Klasy wyjątków

Moduł D: Zaawansowane aspekty programowania w języku Python (40 godz.)

Zaawansowane aspekty typowania

Typowanie dynamiczne

Łańcuchy znaków

Generalne kategorie typów

Typy złożone

Programowanie funkcyjne

Pisanie funkcji prostych i złożonych

Paradygmaty programowania funkcyjnego

Funkcje iteracyjne, funkcje: zip(), reverse(), enumerate()

Funkcje wyższego rzędu, funkcje generatorowe

Złożone funkcje rekurencyjne

Redukcje

Moduły: itertools, functools, biblioteka PyMonad

Dekoratory

Złożone funkcje Lambda

Funkcyjne programowanie wielowątkowe

Programowanie obiektowe

Klasy, Interfejsy i metody

Paradygmaty programowania obiektowego

Klasy abstrakcyjne

Kontenery niestandardowe

Przeciążanie operatorów

Programowanie zorientowane obiektowo

Tworzenie klas mieszanych

Rozszerzanie typów wbudowanych

Dziedziczenie diamentowe – zmiany w wersji 3.x Pythona

Klasy w nowym stylu

Metody statyczne i metody klasy

Pułapki związane z klasami

Moduły

Przestrzenie nazw modułów

Pakiety modułów

Ukrywanie danych w modułach

Przechodnie przeładowywanie modułów

Pułapki związane z modułami

Metaprogramowanie

Metaklasy – model metaklasy

Tworzenie i dekorowanie metaklas

Algorytm dziedziczenie w Pythonie

Atrybuty, Atrybuty getter i setter

Refaktoryzacja atrybutów a @property

Podklasy i klasy nadrzędne

Zastoowanie metaklas

Współbieżność i równoległość

Tworzenie wątków

Programowanie równoległe

Klasy współbieżne: Lock, Queue, ThreadPoolExecutor

Asynchroniczność

Pułapki wiazane z programowaniem współbieżnym i asynchronicznym

Wydajność i optymalizacja

Testowanie aplikacji w języku Python

Analiza błędów

Profilowanie

Zarządzanie pamięcią

Testowanie wydajności

Zaawansowane aspekty programowania w języku Python

Wzorce projektowe

Systemy rozproszone

Przetwarzanie języka naturalnego (NLP)

Bazy danych

Połączenie z bazą danych SQL

Systemy ORM

Interpreter sqlite Pythona, baza SQLITE3

Peewee

Bazy noSQL

Moduł E: Analiza danych z użyciem języka Python (32 godz.)

Biblioteka pandas

Wstęp do pandas

Tworzenie struktur: Series, DataFrame

Podstawowe Funkcjonalności

Operacje I/O

Indeksowanie i selekcjonowanie danych

Zaawansowane operacje na danych

Przekształcenia, Tabele przestawne

Biblioteka Matplotlib

Elementy zaawansowane

Funkcje statystyczne

Grupowanie

Dane związane z osią czasową (timestamp)

Stylowanie

Skalowanie dużych zbiorów danych

Zastosowania biblioteki pandas w machine learning

Biblioteka NumPy

Podstawy pakietu

Tablice: tworzenie i podstawowe operacje

Tablice jedno i dwuwymiarowe

Macierze, operacje na macierzach

Generatory liczb pseudolosowych

Praca z zależnościami matematycznymi

Kreślenie wykresów za pomocą biblioteki Matplotlib

Typy danych, I/O, indeksowanie

Podklasa ndarrays

Połączenie z językiem C

Zastosowania biblioteki NumPy w machine learning

Moduł F: Programowanie aplikacji www z użyciem języka Python (36 godz)

Podstawy tworzenia aplikacji www

Front-end i Back-end

HTML5 i CSS3

JavaScript

Bootstrap

Architektura aplikacji www

Platforma Flask

Architektura platformy

Projekt – struktura aplikacji

Szablony

Obsługa formularzy www i baz danych

Platforma Django

Podstawy środowiska

Praca z modułami

Architektura MVC

Modele

Bezpieczeństwo

Moduł G: Wstęp do algorytmów sztucznej inteligencji – implementacja w języku Python (24 godz)

Pakiet Scikit - learn

Co zawiera pakiet? Strona główna scikit-learn

Instalacja

Klasyfikacja

Co to jest klasyfikacja?

Uczenie klasyfikatora binarnego.

Miary wydajności

Rodzaje klasyfikacji: wieloklasowa, wieloetykietowa, wielowyjściowa

Błędy

Uczenie modeli

Regresja, rodzaje regresji

Krzywe wielomianowe

Maszyny wektorów nośnych

Drzewa decyzyjne

Zbiory uczące się

Redukcja wymiarowości

Clustering, analiza skupień

Analiza obrazu

Deep Learning (uczenie głębokie) – zastosowania biblioteki scikit-learn

Inne pakiety i narzędzia języka Python: TensorFlow, Keras, DEAP

Moduł H: Konsultacje z doradcą HR (4 godz.)

Konsultacje z doradcą HR (4 godz.)

Statystyka rynku pracy w zawodzie programisty języka Python

Główne zasady rekrutacji pracowników

Zasady rekrutacji w Comarch SA

Klucz do dobrego CV

Przygotowanie profilu zawodowego na znanych portalach (LinkedIn, Goldenline)

Rozmowa rekrutacyjna

Przygotowanie do rozmowy

Zasady prowadzenia rozmowy

Pytania od i do rekrutera

Stres – geneza i antidotum

Pytania i odpowiedzi

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	12 054,00 PLN
Koszt przypadający na 1 uczestnika netto	9 800,00 PLN
Koszt osobogodziny brutto	57,95 PLN
Koszt osobogodziny netto	47,12 PLN

Prowadzący

Liczba prowadzących: 2



1 z 2

Marcin Albiński

Trener z obszaru programowania- Web, Mobile, Java, Python, C#, VB, JS, Groovy, Scala, Swift, HTML, inne, baz danych – analiza danych, algorytmy AI i grafiki komputerowej; Od 2013 roku współpracujący z Centrum Szkoleniowym Comarch; Wykształcenie: doktor informatyki w specjalności Algorytmy sztucznej inteligencji – Machine Learning, Deep Learning; Prawie 30-letnie doświadczenie trenerskie i akademickie; Autor programów szkoleniowych i ścieżek kompetencyjnych, licznych publikacji naukowych i odczytów; ponad 6 tysięcy godzin szkoleniowych, ponad 7500 godzin szkoleniowych.



2 z 2

Weronika Wasieczko

Specjalista z zakresu Human Resources, specjalizująca się w rekrutacji na stanowiska w branży IT; HR Business Partner w Comarch; wykształcenie wyższe; trenerka Centrum Szkoleniowego Comarch od 2018 roku, certyfikaty i szkolenia: Insightful Profiler (IP121) Professional Certificate, Extended DISC Certification Training; w ramach IT Camp prowadzi moduł HR – 4 godziny

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Uczestnicy otrzymują materiały szkoleniowe w wersji elektronicznej lub papierowej:

-Python. Wprowadzenie

<https://helion.pl/ksiazki/python-wprowadzenie-wydanie-v-mark-lutz-pyth5v.htm#format/d>

-Python w analizie danych. Przetwarzanie danych za pomocą pakietów Pandas i NumPy oraz środowiska IPython.

<https://helion.pl/ksiazki/python-w-analizie-danych-przetwarzanie-danych-za-pomoca-pakietow-pandas-i-numpy-oraz-srodowiska-ipy-wes-mckinney,pytand.htm#format/d>

Warunki uczestnictwa

Szkolenie prowadzone jest w formie intensywnego kursu i przeznaczone jest dla osób które chcą się przygotować do programowania w języku Python na poziomie zaawansowanym.

Szkolenie dla programistów języka Python w zakresie Junior Python Developer.

Warunkiem projektu KKZ jest egzamin: PCEP-30-02:PCEP Certified Entry-Level Python Programmer Certification

Egzamin wymagany - dla wszystkich uczestników zapisanych w BUR i jest on w cenie szkolenia. Do egzaminu należy podejść do 14 dni roboczych od daty zakończenia szkolenia. Na termin egzaminu uczestnicy zapisywani są indywidualnie z naszym koordynatorem do spraw egzaminów.

Informacje dodatkowe

Zawarto umowę z WUP Kraków na rozliczanie Usług z wykorzystaniem elektronicznych bonów szkoleniowych w ramach projektu „Kierunek Kariera Zawodowa”.

Egzamin wymagany - dla uczestników z Kierunek Kariera Zawodowa.

Warunki techniczne

Wymagania techniczne:

- Komputer / laptop ze stałym dostępem do Internetu (Szybkość pobierania/przesyłania: minimalna 2 Mb/s / 128 kb/s; zalecana 4 Mb/s / 512 kb/s)
- przeglądarka internetowa – zalecane: Google Chrome, Mozilla Firefox, Microsoft Edge
- słuchawki lub dobrej jakości głośniki
- mikrofon

Zalecane

- dodatkowy monitor
- kamera (w przypadku komputerów stacjonarnych)
- spokojne miejsce, odizolowane od zewnętrznych czynników rozpraszających
- podstawowa znajomość języka angielskiego (do sprawnego poruszania się po platformie zdalnej)

Szkolenie Zdalne prowadzone jest w czasie rzeczywistymi i transmitowane za pomocą kanału internetowego z wykorzystaniem systemu ZOOM lub Webex, który umożliwia komunikację głosową oraz wideo z Uczestnikami przebywających w dowolnym miejscu ze sprawnie działającym stałym łączem internetowym. Każdy z uczestników szkolenia otrzymuje przed szkoleniem link dostarczony w wiadomości mailowej z informacjami dotyczącymi szkolenia zdalnego.

Uczestnicy otrzymują link umożliwiający uczestnictwo w szkoleniu na 2 dni przed datą rozpoczęcia szkolenia.

Kontakt



Aneta Lewkowska

E-mail aneta.lewkowska@comarch.pl

Telefon (+48) 126 877 811