

Comarch Bootcamp

Junior Python Developer

Cele szkolenia

Celem szkolenia jest:

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

Wprowadzenie uczestników w zaawansowane zagadnienia związane z programowaniem 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).*

Umiejętności

Dzięki szkoleniu uczestnik będzie:

- Typować zmienne, rozumieć kolekcje
- Programować funkcyjnie i obiektowo w zaawansowany sposób
- W złożony sposób pracować z modułami
- Stosować metaprogramowanie do realizacji modeli obiektowych
- Wdrażać program współbieżne, równoległe i asynchroniczne
- Programować dostęp do baz danych, SQL i noSQL
- Stosować język Python do obliczeń i analizy danych
- Stworzyć aplikację www z użyciem platformy Flask i Django
- Stosować różne metody analizy danych
- Analizować dane za pomocą algorytmów AI

Profil uczestników

Szkolenie 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.*

Przygotowanie uczestników

Uczestnik powinien posiadać przynajmniej podstawowe doświadczenie w programowaniu w dowolnych językach programowania oraz podstawową wiedzę na temat algorytmów.

Szczegółowy program szkolenia

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: 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 tworzenie kodu w języku Python – znaczenie wcięć. Jak pisać w Pythonie?
- Dobre praktyki w pisaniu kodu – PEP8 – dokument definiujące 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 zagnieżdzone
- 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 (48 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 wiązane 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 (36 godz.)

Biblioteka pandas

- Wstęp do pandas
- Tworzenie struktur: Series, DataFrame
- Podstawowe Funkcjonalności
- Operacje I/O
- Indeksowanie i selekcjonowanie danych
- Zaawansowane operacje na danch
- 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 dwuwumiarowe
- 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 (40 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 (36 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, wieloetykietykowa, 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
 1. Przygotowanie do rozmowy
 2. Zasady prowadzenia rozmowy
 3. Pytania od i do rekrutera
 4. Stres – geneza i antidotum

Pytania i odpowiedzi

Metoda realizacji szkolenia

Szkolenie jest realizowane różnymi metodami takimi jak: mikrowykład, ćwiczenia wspólne, ćwiczenia indywidualne, praca indywidualna, z głębokim naciskiem położonym na aspekt praktyczny programowania.

W trakcie szkolenia uczestnicy będą realizowali trzy projekty praktyczne:

Projekt realizowany w trakcie trwania modułu C.

- Celem projektu jest przygotowanie prostej aplikacji w języku Python z użyciem poznanych narzędzi i metod poznanych w trakcie realizacji modułu C

Projekt realizowany w trakcie trwania modułu D.

- Celem projektu jest przygotowanie aplikacji w języku Python z użyciem zaawansowanych narzędzi i metod poznanych w trakcie realizacji modułu D

Projekt realizowany w trakcie trwania modułu EFG.

- Celem projektu jest przygotowanie aplikacji w języku Python z użyciem zaawansowanych narzędzi analitycznych z oprawą www a także z użyciem algorytmów AI

Projekt końcowy

- Jest to projekt podsumowujący wiedzę nabytą w trakcie kursu. Projekt będzie obejmował różnorodne techniki programowania realizowane w trakcie trwania całego kursu. Wyjściem ma być albo konsola z

użyciem złożonych elementów obliczeniowych, albo prosty interfejs www. Dodatkowo każdy z modelu ma się odwoływać do zdefiniowanego wcześniej źródła danych

Liczba dni, liczba godzin szkoleniowych

26 dni, 208 godzin szkoleniowych

Ścieżka rozwoju po szkoleniu

- *Szkolenia z zakresu Machine Learning oraz programowania aplikacji webowych w języku Python*