

SZKOLENIE PODSTAWOWE

Python - Wprowadzenie do składni języka i środowiska dla programistów

PYTHON/F-DEVEL

Czas trwania: 3 dni (24h)

Podczas tego szkolenia uczestnicy od podstaw poznają składnię i środowisko języka Python. To szkolenie pomoże Ci w rozwoju kariery programistycznej.

Cele szkolenia

- Wprowadzenie do języka Python dla osób programujących

Zalety

- Popularność: Python jest jednym z najpopularniejszych języków programowania na świecie. Jest wykorzystywany w wielu różnych branżach, takich jak nauka danych, sztuczna inteligencja, robotyka, aplikacje webowe i wiele innych.
- Łatwość nauki: Python jest językiem programowania, który jest stosunkowo łatwy do nauki, szczególnie dla początkujących programistów. Ma prostą i czytelną składnię, co oznacza, że można go szybko opanować.
- Wszechstronność: Python jest wszechstronnym językiem programowania, który może być wykorzystany do tworzenia różnych rodzajów aplikacji. Można go używać do tworzenia skryptów, aplikacji webowych, gier, narzędzi automatyzacji i wiele więcej.
- Wsparcie społeczności: Python ma bardzo silną społeczność programistów, którzy tworzą różne biblioteki i narzędzia, które ułatwiają pracę z tym językiem. Społeczność ta jest również bardzo pomocna i chętnie dzieli się swoją wiedzą i doświadczeniem z innymi programistami.
- Przydatność w pracy: Wielu pracodawców poszukuje programistów, którzy mają doświadczenie w Pythonie. Znajomość tego języka programowania może zwiększyć szanse na znalezienie pracy lub awans w obecnej pracy.

Dla kogo?

- Programistów innych języków programowania
- Inżynierów oprogramowania
- Inżynierów jakości
- Devopsów
- Architektów

Wymagania

- Znajomość zagadnień programistycznych tj. zmienne, pętle, instrukcje warunkowe, funkcje, klasy
- znajomość dowolnego IDE (mile widziane IDE z rodziny JetBrains, np. IntelliJ albo WebStorm)
- Umiejętność tworzenia oprogramowania w dowolnym języku
- Dla osób chcących od podstaw poznać język Python kierowane jest szkolenie: Python Podstawy (Python/F)

Program

1. Przygotowanie środowiska do pracy i wstęp do programowania
 - a. Wprowadzenie do Python
 - b. PyCharm jako IDE



- c. Python wersje i cykle wdrożeniowe
- 2. Struktury danych
 - a. Elementy składni języka
 - b. Numeryczne typy danych (int, float)
 - c. Logiczne typy danych (bool, None)
 - d. Łańcuchy znaków (str)
 - e. Sekwencje (tuple, list, set, frozenset)
 - f. Zagnieżdżone sekwencje
 - g. Wybieranie, wycinanie elementów, rozpakowywanie
 - h. Mapy (dict)
 - i. Generowanie Map (zip, enumerate)
 - j. Zagnieżdżone mapy
 - k. Type Annotation
- 3. Instrukcje sterujące
 - a. Operatory warunkowe, logiczne, przynależności
 - b. Instrukcje warunkowe (if, else, elif)
 - c. Wyrażenia warunkowe, algebra Boola (and, or)
 - d. Wyjątki: interpretacja, podnoszenie, definiowanie
 - e. Pętle (while, for)
 - f. Pętle po zagnieżdżonych strukturach danych, rozpakowywanie
 - g. Rozwinięcia i wyrażenia generatorowe
- 4. Pliki
 - a. Ścieżki
 - b. Tryby dostępu
 - c. Operacje na pliku: odczyt, zapis, dopisywanie
 - d. Kodowanie znaków
 - e. Tryb binarny
- 5. Funkcje
 - a. Definiowanie
 - b. Zwracanie wartości
 - c. Parametry: wymagane, opcjonalne
 - d. Argumenty: pozycyjne, nazwane
 - e. Zakres przestrzeni nazewniczej
- 6. Programowanie obiektowe
 - a. Klasy i instancje
 - b. Metody
 - c. Atrybuty
 - d. Metoda inicjalizacyjna
 - e. Dziedziczenie
 - f. Stringifikacja (str, repr)
 - g. Architektura OOP

