

SZKOLENIE PODSTAWOWE

Python podstawy

PYTHON/F

Czas trwania: 5 dni (40h)

Cele szkolenia

- Wprowadzenie do języka Python
- Poznanie zmiennych, typów danych, sekwencji, pętli, instrukcji warunkowych, plików, funkcji, klas i obiektów
- Wprowadzenie do pracy ze zintegrowanym środowiskiem programistycznym (IDE): debugging, refactoring, kontrola wersji, REPL, testy itp.
- Wprowadzenie do dobrych praktyk i technik inżynierii oprogramowania

Zalety

- Szkolenie pozwoli osobom, które nie miały styczności z programowaniem rozpocząć pisanie kodu w Python
- Omówione zostaną wszystkie podstawowe zagadnienia, które konieczne są do samodzielnej pracy
- Dostaniesz wysokiej jakości materiały i odnośniki do stron, filmików i webinarów, które pomogą kontynuować naukę programowania
- Kurs ma zbalansowany stosunek teorii do praktyki
- Praktyczny warsztat z ponad tysiącem linii do samodzielnego napisania przez każdego uczestnika
- Przykłady kodu na podstawie realnych przypadków użycia i studiów przypadku, które demonstrują praktyczne zastosowanie
- Wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how

Dla kogo?

- Szkolenie Python od podstaw adresowane jest dla osób chcących nauczyć się programować w języku Python, zrozumieć jego składnię oraz instrukcje sterowania przepływem

Wymagania

- Biegła umiejętność obsługi komputera
- Mile widziana znajomość innych języków programowania
- Mile widziana umiejętność korzystania ze środowisk programistycznych



- Mile widziane wcześniejsze zetknięcie z językiem

Program

1. Przygotowanie środowiska do pracy i wstęp do programowania
 - a. Wprowadzenie do języka Python i cykli wydań
 - b. Wprowadzenie do zintegrowanego środowiska deweloperskiego (IDE)
 - c. Elementy składni języka programowania: zmienne, komentarze, operatory
 - d. Wprowadzenie do konsoli Python (REPL), składni, wcięć i standardów
2. Składnia
 - a. Zmienne
 - b. Komentarze
 - c. Print i podstawianie wartości
 - d. Podstawowe operatory
 - e. Funkcje wbudowane
 - f. Dobre praktyki
3. Typy i struktury danych
 - a. Numeryczne typy danych (int, float)
 - b. Logiczne typy danych (bool, None)
 - c. Łańcuch znaków (str)
 - d. Sekwencje i zbiory (tuple, list, set)
 - e. Mapy (dict)
 - f. Zagnieżdżone struktury danych (list[tuple], list[list], list[dict])
 - g. Wybieranie, wycinanie elementów, rozpakowywanie
4. Instrukcje sterujące
 - a. Operatory warunkowe, logiczne, przynależności
 - b. Instrukcje warunkowe (if, else, elif)
 - c. Wyrażenia warunkowe, algebra Boole'a (and, or, not)
 - d. Wyjątki: rodzaje, analiza stosu wywołań, podnoszenie wyjątków
 - e. Pętle (while, for)
 - f. Pętle po zagnieżdżonych strukturach danych
5. Pliki
 - a. Ścieżki względne i bezwzględne
 - b. Tryby dostępu i kodowanie znaków
 - c. Operacje na pliku: odczyt, zapis, dopisywanie
6. Funkcje
 - a. Definiowanie funkcji i zwracanie wartości
 - b. Parametry: wymagane, opcjonalne
 - c. Argumenty: pozycyjne, nazwane (keyword)
 - d. Zakres przestrzeni nazw i dobre praktyki
7. Wyjątki
 - a. Podnoszenie i przechwytywanie
 - b. Najczęściej występujące i hierarchia
 - c. Asercje
 - d. Analiza stosu wywołań



e. Wyjątki: analiza stosu wywołań, podnoszenie wyjątków, definiowanie

8. Programowanie obiektowe

- a. Klasy i instancje
- b. Atrybuty i metoda inicjalizacyjna
- c. Metody
- d. Dziedziczenie

9. Moduły

- a. Krótkie wprowadzenie do biblioteki standardowej
- b. Krótkie wprowadzenie do najczęściej wykorzystywanych modułów dodatkowych
- c. Wyszukiwanie i instalacja nowych modułów

