

SZKOLENIE ŚREDNIO ZAAWANSOWANE

Python średnio zaawansowany

PYTHON/MID

Czas trwania: 5 dni (40h)

Uczestnicy szkolenia zapoznają się z biblioteką standardową Python, serializacją danych do różnych formatów i obsługą relacji. Szczególny nacisk kładziemy na praktyczne aspekty programowania oraz osadzenie technik wytwarzania oprogramowania w języku Python we współczesnych realiach. Szkolenie odpowiada potrzebom średnio zaawansowanych i zaawansowanych programistów i nie jest przeznaczone dla osób początkujących lub nieprogramujących!

Cele szkolenia

- Poszerzenie wiedzy dotyczącej programowania w języku Python
- Zapoznanie się z alternatywnymi wersjami interpretera
- Nauka refaktoringu i dobrych praktyk związanych z tworzeniem czystego kodu

Zalety

- Uczestnicy nauczą się korzystać z biblioteki standardowej Python
- Uczestnicy zdobędą umiejętność refaktoryzacji i pracy z debuggerem
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretnie umiejętności - w ramach każdego szkolenia rozwijamy praktyczne umiejętności związane z daną technologią i tematyką
- Nauka z praktykami - wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how

Dla kogo?

- Programiści języka Python
- Analitycy Danych, Data Scientist lub osoby zajmujące się Machine Learning

Wymagania

- Podstawowa umiejętność programowania i debugowania w języku Python
- Dobra znajomość wybranego środowiska IDE (preferowany PyCharm)
- Znajomość koncepcji programowania obiektowego

Program

1. Wprowadzenie



- a. Zadania określające poziom grupy
- b. Zmiany w najnowszych wydaniach Python
2. Składnia
 - a. Rozwinięcia list, set, dict i wyrażenia generatorowe
 - b. Generowanie (zip, enumerate, range, map, filter)
 - c. Unpacking Assignment
3. Pattern Matching
 - a. Wprowadzenie do składni
 - b. Wzorce: literal, or, wildcard, capture, constant
 - c. Wzorce: sequence, mapping, class, walrus
 - d. Guard i subpattern
4. Print i PPrint
 - a. Zaawansowane opcje formatowania - print
 - b. Formatowanie wyświetlania obiektów - pprint
 - c. Stringifikacja obiektów (str, repr, format)
5. Daty i strefy czasowe
 - a. Tworzenie obiektów dat, parsowanie i formatowanie - datetime
 - b. Przesunięcia czasu i dat - timedelta
 - c. Praca z timestamp - time, datetime
 - d. Obsługa stref czasowych - zoneinfo
 - e. Obsługa kalendarza - calendar
6. Format CSV
 - a. Specyfika formatu i najczęstsze problemy
 - b. Zapis i odczyt danych płaskich, moduł csv
 - c. Zapis i odczyt danych relacyjnych
 - d. Moduł csv oraz pandas
7. Format JSON
 - a. Specyfika formatu i najczęstsze problemy
 - b. Zapis i odczyt prostych danych
 - c. Enkoder i dekodery dla dat i obiektów złożonych
8. Format TOML
 - a. Specyfika formatu i najczęstsze problemy
 - b. Odczyt danych
 - c. Co to jest plik pyproject.toml
9. Format Pickle
 - a. Serializacja i deserializacja obiektów, dat, relacji
 - b. Zapis do pliku, odczyt z pliku
 - c. Wersje protokołów i kompatybilność
10. Wyrażenia regularne
 - a. Składnia (identifier, qualifiers, quantifiers, grouping, backreference)
 - b. Wyrażenia chciwe i leniwe
 - c. Wyszukiwanie wieloliniowe
 - d. Grupy nazwane, pozycyjne, niechwytające i komentarze
 - e. Flagi



f. Korzystanie z wyrażeń regularnych w Python

11. Bazy danych

- a. DB API i bazy relacyjne na przykładzie SQLite3
- b. Tworzenie i zmienianie schema (CREATE TABLE, CREATE INDEX, ALTER, DROP)
- c. Tworzenie i zmienianie danych (INSERT, UPDATE, DELETE)
- d. Wybór danych (SELECT, podzapytania, kursory, relacje, CTE)
- e. Problematyka SQL injection

12. Projekt praktyczny

- a. Zastosowanie technologii ze szkolenia
- b. Dobre praktyki
- c. Wykorzystanie debuggera w IDE

