

SZKOLENIE ŚREDNIO ZAAWANSOWANE

Python: Poziom średnio zaawansowany

PYTHON/MID

Czas trwania: 5 dni

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
- Konkretność 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 debuggowania w języku Python



- Dobra znajomość wybranego środowiska IDE (preferowany PyCharm)
- Znajomość koncepcji programowania obiektowego



Program

1. Rozpakowywanie obiektów
 - a. Unpacking Assignment
 - b. Rozpakowywanie parametrów (*args, kwargs)
 - c. Rozpakowywanie argumentów (*args, kwargs)
 - d. Assignment Expression
 - e. Składnia parametrów do definiowania API
2. Type Annotations
 - a. Typy proste
 - b. Sekwencje
 - c. Mapy
 - d. Funkcje
 - e. Obiekty i metody
3. Daty i strefy czasowe
 - a. Tworzenie obiektów dat, parsowanie i formatowanie - datetime
 - b. Przesunięcia czasu i dat - timedelta
 - c. Praca z timestamp
 - d. Obsługa stref czasowych - zoneinfo, pytz
 - e. Obsługa kalendarza - calendar
4. Serializacja obiektów, dat, relacji oraz zapis do pliku i bazy danych
 - a. Format CSV (dane proste i relacyjne)
 - b. Format JSON (serializacja prostych danych, dat oraz obiektów)
 - c. Format Pickle
5. 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. Korzystanie z wyrażeń regularnych w Python
6. Bazy danych
 - a. DB API
 - b. Bazy relacyjne na przykładzie SQLite3
 - c. Tworzenie i zmienianie schema (CREATE, ALTER, DROP)
 - d. Tworzenie i zmienianie danych (INSERT, UPDATE, DELETE)
 - e. Wybór danych (SELECT, podzapytania, kursory, relacje, indeksy)
 - f. SQL injection
7. Programowanie obiektowe
 - a. Przeciążanie operatorów
 - b. Enumeracja (enum)
 - c. Dataclasses (dataclasses)
8. Kontrola jakości, logowanie i debugging
 - a. Testy jednostkowe i Test Driven Development (TDD) - doctest, unittest



- b. Logowanie zdarzeń i informacje ostrzegawcze - logging, warnings
 - c. Praca z debuggerem w IDE
 - d. Wydajność wbudowanych struktury i typów danych - timeit, cProfile
 - e. Statyczna analiza kodu źródłowego - pylint, pylama, sonarlint, pyflakes
 - f. Standardy programowania PEP8, PEP20 i dobre praktyki - pycodestyle, pydocstyle
 - g. Zaawansowane opcje formatowania tekstu - print
 - h. Formatowanie wyświetlania obiektów - pprint
9. Moduły, paczkowanie i alternatywne interpretery i preprocesory
- a. Środowiska wirtualne - venv
 - b. Tworzenie własnych paczek - setup.py, setup.cfg, pyproject.toml
 - c. Zarządzanie zależnościami - requirements.txt
 - d. Upload do pypi.org - twine, format wheel
 - e. Generowanie pakietów - zipapp
 - f. Generowanie plików wykonywalnych dla systemu operacyjnego - pyinstaller
 - g. Alternatywny kompilator Python - pypy
10. Matematyka w bibliotece standardowej
- a. Podstawowe zagadnienia matematyczne - math
 - b. Podstawowa statystyka - statistics
 - c. Liczby pseudolosowe - random
 - d. Ułamki zwykłe - fractions
 - e. Zagadnienia precyzji operacji numerycznych i typu float - decimal
 - f. Liczby zespolone - complex
11. Lokalizacja i internacjonalizacja
- a. Locale
 - b. i18n
 - c. l10n
12. System Operacyjny
- a. Praca ze ścieżkami - pathlib
 - b. Parsowanie plików konfiguracyjnych - configparser
 - c. Parsowanie argumentów linii poleceń - argparse
 - d. Uruchamianie poleceń w systemie - subprocess
 - e. Pliki tymczasowe - tempfile
 - f. Praca z systemem operacyjnym, plikami i katalogami - os

