

Zaawansowane techniki programowania w języku Python



Przeznaczenie szkolenia

Szkolenie przeznaczone jest dla osób chcących podnieść oraz utrwalić, a także wzbogacić swoją dotychczasową wiedzę o nowe funkcjonalności, oraz bardziej zaawansowane mechanizmy z zakresu programowania w języku Python.



Korzyści wynikające z ukończenia szkolenia

Szkolenie utrwała oraz znacząco poszerza wiedzę z zakresu języka Python o zaawansowane techniki, które będą omawiane w trakcie części warsztatowej. Po ukończeniu szkolenia słuchacze będą potrafili poradzić sobie z większością problemów, które mogą napotkać w trakcie pracy z językiem. Poruszone zostaną ważne aspekty takie jak nowoczesne praktyki oraz omówione zostaną popularne dziś moduły, asynchroniczność czy optymalizacja procesów dzięki wykorzystaniu dostępnych funkcjonalności języka.



Oczekiwane przygotowanie słuchaczy

Udział w szkoleniu PYTH01 – Podstawy programowania w języku Python lub równoważna wiedza.
Podstawowa znajomość środowiska Linux/Unix/Windows.
Mile widziana znajomość podstawowych zagadnień bazodanowych



Język szkolenia

- Szkolenie: polski
- Materiały: polski



Szkolenie obejmuje

- wykłady
- warsztaty



Czas trwania

5 dni / 35 godzin

Agenda szkolenia

1. PROGRAMOWANIE FUNKCYJNE

- *args & **kwargs
- Rozpakowywanie Argumentów oraz Rozpakowywanie Kolekcji
- Funkcja jako parametr (First Class Citizen)
- Funkcje lambda (Anonimowe)
- Wyrażenia Listowe, Słownikowe, etc... - zaawansowane
- Zagnieżdżone Wyrażenia Listowe, Słownikowe, etc...
- Moduł ITERTOOLS - omówienie
- Moduł FUNCTOOLS - omówienie
- Generatory i Iteratory (różne sposoby definiowania)
- Wzorzec Dekoratora - Tworzenie własnych Dekoratorów (dekorator prosty)

2. PROGRAMOWANIE OOP - POZIOM ZAAWANSOWANY

- Dokumentowanie Kodu
- Atrybuty Klas
- Wielodziedziczenie oraz Method Resolution Order
- Metoda Super
- Dostęp do Atrybutów, Atrybuty Prywatne
- Deskryptory i definiowanie właściwości
(`__get__`, `__set__`, `__delete__`, `__set_name__`)
- `__getitem__`, `__setitem__`, `__delitem__`
- Implementacja Iterowalności dla Klas
- Przeciążanie operatorów

- Klasy Abstrakcyjne – podstawowe zagadnienia
- Dekoratory Klasowe z Argumentami
- MetaClasses – podstawowe zagadnienia

3. POMOCNE NARZĘDZIA

- Adnotacje Typów (Hints) – informacje i przykład zastosowania
- Moduł TIMEIT – na przykładach (porównanie czasu wykonywania się algorytmów)
- Moduł LOGGING – logowanie zdarzeń (konfiguracja i sposoby zastosowania)
- Moduł OS – powtórzenie oraz dodatkowe informacje
- Moduł SYS – powtórzenie oraz dodatkowe informacje
- Interakcja z systemem operacyjnym i systemem plików – moduły SYS i OS

4. KOLEKCJE – rozszerzenie wbudowanych typów złożonych

• **Moduł COLLECTIONS:**

- a) NamedTuple
- b) DataClass
- c) DefaultDict
- d) Deque
- e) Counter

5. WYRAŻENIA REGULARNE

• **Moduł RE:**

- a) Składnia – symbole, budowa wyrażeń regularnych
- b) funkcje match & search
- c) funkcje findall & finditer
- d) obiekt Pattern
- e) obiekt Match
- f) funkcje sub & split
- g) flagi DOTALL & MULTILINE
- h) Narzędzia online do tworzenia WR (np. regex101 i pythex)

6. PRZETWARZANIE DANYCH

- Moduł REQUESTS – HTTP Protocol (podstawowe informacje oraz ządania z poziomu Pythona)
- Moduł BEAUTIFUL SOUP – WebScrapping (XML & HTML) (przykład zastosowania w przeszukiwaniu dokumentów hipertekstowych)
- Moduł PARAMIKO – połączenia SSH (przykład zastosowania)
- JSON, YAML, PICKLE – praktyczne użycie
- Wstęp do Pandas (odczyt i zapis XLS, CSV, etc...)
- Przegląd innych bibliotek

7. BAZY DANYCH

- Przegląd popularnych „Connectorów” dla relacyjnych baz danych – na podstawie połączeń z MYSQL / PostgreSQL / ORACLE lub innego wybranego silnika
- Obsługa zapytań z poziomu Pythona – podstawowe zapytania
- Połączenie z bazami nierelacyjnymi – na przykładzie PYMONGO (MongoDB)

- ORM na przykładzie SQLAlchemy

8. WĄTKI I PROCESY

- **Moduł THREADING - podstawowe koncepcje:**

- a) Uruchamianie Wątków
- b) Synchronizacja Wątków
- c) Rlocks
- d) Semaphores
- e) GIL (Global Interpreter Lock)

- **Moduł MULTIPROCESSING - podstawowe koncepcje:**

- a) Processes, Queues, Locks
- b) Pools
- c) Daemons
- d) Wymiana Danych pomiędzy Procesami

9. ASYNCHRONICZNY PYTHON - Wstęp i Podstawy

- **Moduł ASYNCIO - omówienie podstawowych zagadnień:**

- a) Coroutines, Tasks
- b) Streams
- c) Subprocesses
- d) Queues
- e) Exceptions
- f) Event Loop
- g) Futures
- h) Asynchronous (Non-Blocking) HTTP Calls

10. WSTĘP DO TESTÓW

- Testy jednostkowe
- Wstęp do TDD
- Biblioteka Unittest
- Przegląd innych bibliotek

12. DODATKOWO:

- Inne biblioteki wybrane wspólnie z uczestnikami kursu