

# Warsztat - programowanie w języku Python na poziomie średniozaawansowanym

## Samodzielne tworzenie prostych aplikacji

---

### Cele szkolenia

Celem szkolenia jest usystematyzowanie i praktyczne wykorzystanie podstawowej wiedzy dotyczącej programowania w języku Python. W trakcie zajęć uczestnicy pracują warsztatowo pisząc proste aplikacje z użyciem języka Python oraz zapoznają się z różnymi zastosowaniami struktur językowych na konkretnych przykładach.

Warsztat ma na celu przygotowanie uczestników do podjęcia nauki języka Python na poziomie zaawansowanym.

### Umiejętności

Dzięki szkoleniu uczestnik będzie potrafił:

- Instalować i konfigurować środowisko programistyczne języka Python
- Pisać programy w wybranym IDE dla języka Python
- Projektować i pisać proste aplikacje stosując najważniejsze zasady programowania w języku Python
- Programować struktury danych w postaci kolekcji oraz danych zewnętrznych
- Programować funkcyjnie i obiektowo
- Tworzyć obsługę błędów
- Optymalizować programy
- Analizować cykl życia aplikacji języka Python

### Profil uczestników

Szkolenie przeznaczone jest dla osób, które ugruntować wiedzę na poziomie podstawowym i przygotować się do nauki na etapie zaawansowanym.

Szkolenie dla programistów języka Python

# Przygotowanie uczestników

Uczestnik powinien posiadać podstawową wiedzę i doświadczenie w programowaniu w języku Python (na poziomie kursu podstawowego).

## Szczegółowy program szkolenia

1. *Analiza Środowiska programistycznego dla Pythona*
  - 1.1. Konfiguracja środowiska
  - 1.2. Podstawowe zasady korzystania ze środowiska
  - 1.3. Proces debugingu
  - 1.4. Analiza błędów
2. *Zasady programowania w języku Python*
  - 2.1. Styl PEP8
  - 2.2. Interpolacja ciągów string
  - 2.3. Wdrożenie zasad programowania i konstruowania programów
  - 2.4. Używanie funkcji i obiektów
  - 2.5. Zasady użycia instrukcji
3. *Programowanie kolekcji*
  - 3.1. Kolekcje: listy, krotki, zbiory oraz słowniki
  - 3.2. Operacje na danych
  - 3.3. Stosowanie dedykowanych metod dla kolekcji
  - 3.4. Asocjacje key – value
  - 3.5. Zastosowania metody `__missing__()`
4. *Podstawy programowania funkcyjnego*
  - 4.1. Pisanie prostych funkcji
  - 4.2. Paradygmaty programowania funkcyjnego
  - 4.3. Funkcje rekurencyjne
  - 4.4. Iteratory i generatory
  - 4.5. Użycie funkcji do przetwarzania danych: format XML
  - 4.6. Wprowadzenie elementów dynamicznych do programowania funkcyjnego
  - 4.7. Funkcje wyższego rzędu: `lambda`, `sorted()`, `filter()`, `iter()`, `map()` i inne
  - 4.8. Funkcje zwracające n-wyników
  - 4.9. Definiowanie zachowań opcjonalnych
  - 4.10. Metoda `yield()` – użycie wielu generatorów
5. **Własny projekt uczestnika szkolenia – programowanie funkcyjne**
  - 5.1. Wykonanie indywidualnego projektu mini-aplikacji opartej na modelu funkcyjnym programowania
  - 5.2. Napisanie kodu źródłowego prostej aplikacji opartej na programowaniu funkcyjnym
  - 5.3. Analiza błędów i testy własnej aplikacji
6. *Programowanie obiektowe*
  - 6.1. Pisanie obiektów
  - 6.2. Diagram klas, jak go czytać?
  - 6.3. Programowanie klas
  - 6.4. Paradygmaty programowania obiektowego: Polimorfizm, Enkapsulacja, Dziedziczenie, Abstrakcja
  - 6.5. Porównanie rozwiązań obiektowych i funkcyjnych
  - 6.6. Klasy abstrakcyjne i interfejsy
  - 6.7. Praktyczne aspekty programowania obiektowego
7. **Własny projekt uczestnika szkolenia – programowanie obiektowe**
  - 7.1. Wykonanie indywidualnego projektu mini-aplikacji opartej na modelu obiektowym programowania

- 7.2. *Napisanie kodu źródłowego prostej aplikacji opartej na programowaniu obiektowym*
- 7.3. *Analiza błędów i testy własnej aplikacji*
- 8. *Programowanie i modelowanie struktur danych*
  - 8.1. *Pliki tekstowe*
  - 8.2. *Pliki XML*
  - 8.3. *Pliki JSON*
  - 8.4. *Proces parsowania danych*
  - 8.5. *Sprawdzanie poprawności danych*
- 9. *Obsługa wyjątków i wydajność*
  - 9.1. *Procesy testowania i debugowania*
  - 9.2. *Konstrukcja try/except/else/finally*
  - 9.3. *Analiza wyników testów*
  - 9.4. *Optymalizacja programu*
  - 9.5. *Analiza wykorzystania pamięci, wycieki pamięci*
- 10. **Własny projekt uczestnika szkolenia – programowanie parsera źródła danych**
  - 10.1. *Wykonanie indywidualnego projektu mini-aplikacji parsującej wybrany model źródła danych*
  - 10.2. *Napisanie kodu źródłowego zaprojektowanej aplikacji*
  - 10.3. *Analiza błędów i testy własnej aplikacji*
- 11. *Cykl życia aplikacji języka Python*
- 12. *Analiza zastosowań języka Python*

## Metoda realizacji szkolenia

Szkolenie prowadzone jest jako warsztat – laboratorium. W trakcie warsztatu uczestnik pracuje indywidualnie i grupowo pod okiem doświadczonego trenera.

## Ilość dni, ilość godzin szkoleniowych

4 dni, 32 godziny szkoleniowe

## Ścieżka rozwoju

Programowanie w języku Python – poziom zaawansowany