

# Tytuł szkolenia: Programowanie wysokojakościowe w Pythonie

## Kod szkolenia: PYTHON03

### Wprowadzenie

Python to nowoczesny język, który od lat wpisuje się na stałe do kanonu najważniejszych języków programowania. Jest to język z rodziny wysokopoziomowych. Cechuje się dynamicznością z silnym wsparciem paradygmatów obiektowości i funkcyjności języka. Python jest językiem o szerokim spektrum zastosowań. Powstają w nim poważne systemy informatyczne, jak i proste skrypty systemowe.

Szkolenie z programowania wysokojakościowego skupia się na zrozumieniu wzorców projektowych, które występują w naturze tego języka. Jako nowoczesny język, większość wzorców projektowych została zaimplementowana bezpośrednio w samym Pythonie. Dzięki temu wiele popularnych technik znanych z innych języków nie ma bezpośredniej implementacji ponieważ leżą w naturze samego języka.

Szkolenie rozpoczyna się od usystematyzowania podstawowej wiedzy z języka Python z naciskiem na gramatykę języka, decyzje projektowe na poziomie front-endu interpretera itp. Uczestnik otrzymuje obraz szeregu decyzji, które wykształciły obecny charakter języka. Dzięki czemu rozwiązania bardziej skomplikowanych problemów stają się naturalne, a przez to oparte na logice i dedukcji.

W kolejnej części przedstawiony jest szereg zaawansowanych mechanizmów języka Python, które mają wpływ na rozwiązywanie typowych problemów. Omówienie wzorców związanych z programowaniem współbieżnym, ukazuje jak wiele istnieje rozwiązań typowego problemu, z czego decyzja wyboru większości podejmowana jest zazwyczaj niesłusznie.

Omówienie klasycznego podejścia do tematu wzorców projektowych, znanych z potocznych języków programowania, jest swoistym podsumowaniem omówionych do tej pory części szkolenia. Ukazuje naturalne rozwiązania, cechujące się wysoką jakością, ściśle związanych z naturą języka Python.

Szkolenie kończy wykład praktyczny w kwestii wytwarzania bardziej wydajnego oprogramowania w języku Python.

### Adresaci szkolenia

Adresatami szkolenia są programiści, administratorzy, testerzy, administratorzy baz danych, naukowcy i wszyscy chętnie zautomatyzować pewne procesy.

Minimalne wymagania wstępne:

- [Podstawy programowania w Pythonie \(kurs PYTHON01\)](#)

Dodatkowa wiedza i umiejętności, które pozwolą sprawniej pracować na szkoleniu i spojrzeć na zagadnienia szkolenia w szerszej perspektywie:

- [Zaawansowane programowanie w języku Python \(kurs PYTHON02\)](#)
- Programowanie w językach dynamicznych
- Programowanie w językach funkcyjnych
- Teoria języków i kompilatorów

### Cel szkolenia

Szkolenie koncentruje się na podniesieniu jakości wytwarzanego oprogramowania w języku Python. Opisuje podstawowe, jak i zaawansowane mechanizmy języka wpływające na decyzje o sposobie rozwiązania typowych problemów projektowych.

Po zakończeniu szkolenia aktywny uczestnik potrafi:

- Posiadać wiedzę o zaawansowanych mechanizmach dostępnych w języku, które przysłużą się do lepszej jakości kodu
- Posiadać wiedzę o wzorcach projektowych oraz ich wykorzystaniu w języku Python
- Posiadać aktywną wiedzę o front-endzie języka
- Zrozumieć możliwości i efektywnie wykorzystać współbieżność oferowaną przez język
- Optymalizować i tworzyć bardziej wydajny kod

## Czas i forma szkolenia

- 35 godzin (5 dni x 7 godzin), w tym wykłady i warsztaty praktyczne.

## Plan szkolenia

1. Wprowadzenie
  - a. Filozofia języka
  - b. Usystematyzowanie wiedzy z podstaw języka
  - c. Gramatyka języka
  - d. Paradygmaty
  - e. Częste pułapki i wyjątki
2. Protokół iteracyjny
  - a. Definicja protokołu
  - b. Definiowanie iteratorów
  - c. Generatory
3. Zaawansowane mechanizmy w Pythonie
  - a. Closures
  - b. Deskryptory
  - c. Dekoratory
  - d. Metaklasy
4. Interfejsy i przetwarzanie współbieżne
  - a. Programowanie wielowątkowe
  - b. Global Interpreter Lock
  - c. Filtry i potoki
  - d. Coroutines
  - e. Multimethods
5. Wzorce projektowe
  - a. Wstęp
  - b. Wzorce konstrukcyjne
  - c. Wzorce strukturalne
  - d. Wzorce behawioralne
  - e. Antywzorce
6. Optymalizacja
  - a. Optymalizacja kontenerów
  - b. Profilowanie i bottlenecks
  - c. Wprowadzenie do PyPy