

# Tytuł szkolenia: Podstawy programowania w Pythonie

## Kod szkolenia: PYTHON01

### Wprowadzenie

Już od wielu lat język Python wpisuje się w kanon podstawowych narzędzi, których znajomość jest wymagana w działach informatycznych zarówno dużych, jak i mniejszych firm. Python jest wysokopoziomowym językiem skryptowym, silnie dynamicznym, o bardzo dużym wsparciu dla paradygmatów zarówno programowania obiektowego, funkcyjnego czy strukturalnego. Ogromny nacisk w języku kładzie się na jego gramatykę, której priorytetem jest język naturalny, a zarazem hasło *There's Only One Way To Do It*. Dzięki takiemu podejściu, rozwiązanie idealnie wpasowuje się w ekonomiczne aspekty każdego przedsięwzięcia czy projektu. Stworzony kod jest czytelny co minimalizuje koszty jego utrzymania czy debugowania. Celem jest skupienie się nad rozwiązywanym problemem, a nie technicznymi aspektami samego języka, co często jest bolączką innych zaawansowanych języków programowania.

Pomimo tych zalet, język pozostaje wszechstronnym narzędziem, zarówno dla programisty, a także dla administratora systemów lub sieci, oraz testera czy nawet badacza naukowego. Python wykorzystywany jest do tworzenia sporego odsetku istniejących już web aplikacji (m.in. dzięki frameworkowi Django), jak i serwisów backendowych (frameworki Twisted, Celery i inne). Świetnie sprawuje się jako bardziej uporządkowany i o większych możliwości następcą skryptów systemowych zazwyczaj pisanych w bashu czy Perlu, a także przy automatyzacji testów czy procesów systemowych. O wszechstronności języka może świadczyć także fakt, iż światowej sławy instytucje jak NASA czy CERN korzystają z języka Python do obliczeń numerycznych, gdzie wyparł on, wydawałoby się, niezwyciężonego Mathlaba.

Język Python jest narzędziem darmowym, stworzonym i utrzymanym przez środowisko Open Source. Dzięki temu, istnieje niezliczona ilość bibliotek i frameworków pokrywająca zastosowanie w nawet najbardziej wymagających zadaniach.

Kod źródłowy języka Python cechuje się wyjątkową czystością i czytelnością. Język ten jest często wybierany także ze względu na czas potrzebny do zakończenia zadania, która zazwyczaj jest wielokrotnie krótszy, aniżeli korzystając z tradycyjnych języków. Społeczność przekonuje *Life is too short, use Python*.

### Adresaci szkolenia

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

Szkolenie z podstaw programowania w języku Python nie posiada minimalnych wymagań od uczestników, aczkolwiek zalecana jest znajomość ogólnie popularnych pojęć programistycznych jak funkcja, klasa czy iteracja.

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

- Programowanie w dowolnym innym języku
- Programowanie w języku obiektowym (Java, C#, C++ itp.)
- Programowanie w języku skryptowym (Perl, bash itp.)

### Cel szkolenia

Szkolenie koncentruje się na poznaniu możliwości języka Python zarówno od strony teoretycznej jak i praktycznej. Program podstawowy obejmuje wszystkie główne aspekty języka, które wykorzystywane są praktycznie w każdej aplikacji napisanej w Pythonie, począwszy od krótkich skryptów automatyzacyjnych zadania systemowe, jak i dużych systemów informatycznych.

Po zakończeniu szkolenia aktywny uczestnik potrafi:

- Czytać, rozumieć oraz nanieść stosowne poprawki na kod źródłowy napisany w języku Python
- Stworzyć programy oparte na zarządzaniu plikami na poziomie systemu
- Stworzyć dowolną aplikację przy jednoczesnym poznaniu bibliotek lub frameworka wspierającego
- Automatyzować testy

Szkolenie jest skoncentrowane na podstawach języka. Dzięki ugruntowaniu wiedzy podstawowej istnieje możliwość dowolnego rozwoju w Pythonie w zależności od zainteresowań i potrzeb.

## Czas i forma szkolenia

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

## Plan szkolenia

1. Wprowadzenie do języka Python
  - a. Idea języka
  - b. Konwencja
  - c. Dokumentacja
  - d. Pierwszy program
2. Podstawowe typy wbudowane
  - a. Liczby
  - b. Łańcuchy tekstowe
  - c. Kontenery
  - d. Operacje na typach
3. Podstawowa funkcjonalność
  - a. Funkcje wbudowane
  - b. Wyrażenia i instrukcje
  - c. Operacje przypisania
  - d. Instrukcje sterujące
  - e. Instrukcje iteracyjne
4. Definiowanie funkcji
  - a. Deklaracja funkcji
  - b. Obsługa dynamiczności argumentów
  - c. Namespacing i zasięg
  - d. Funkcje lambda
  - e. Wstęp do programowania funkcyjnego
5. Programowanie OOP
  - a. Wprowadzenie do OOP
  - b. Python, a OOP
  - c. Definiowanie klas
  - d. Dziedziczenie i polimorfizm
  - e. Enkapsulacja
  - f. Getters & Setters
  - g. Różnice i nowości między starym, a nowym typem obiektów
6. Wyjątki
  - a. Rodzaje wyjątków
  - b. Obsługa i podnoszenie wyjątków
  - c. Tworzenie nowych typów
  - d. Instrukcja finally
  - e. Instrukcje warunkowe
7. Moduły i pakiety
  - a. Wprowadzenie
  - b. Definiowanie modułów i pakietów
  - c. Importowanie modułów
  - d. Hermetyzacja
8. Operacje I/O
  - a. Formatowanie wyjścia
  - b. Odczyt i zapis plików
  - c. Serializacja obiektów
9. Standardowa Biblioteka
  - a. Wprowadzenie
  - b. Obsługa wiersza poleceń
  - c. Obsługa kompresji