

# Python od podstaw



w CODE:ME

## program kursu

### 1. Środowisko programistyczne:

- konfiguracja środowiska,
- instalacja i konfiguracja Pythona oraz PyCharma.

### 2. Podstawowe typy danych:

- wprowadzenie do podstawowych typów danych w Pythonie,
- prezentacja operacji arytmetycznych,
- używanie Pythona jako kalkulatora.

### 3. Instrukcje sterujące:

- przedstawienie instrukcji warunkowych oraz pętli.

### 4. Podstawowe kolekcje:

- używanie listy oraz słownika,
- praca z krotkami,
- pakowanie i rozpakowywanie krotek.

### 5. Proste aplikacje konsolowe:

- drukowanie danych w konsoli,
- pobieranie i konwertowanie danych od użytkownika.

### 6. Funkcje:

- pisanie własnych funkcji,
- przekazywanie parametrów do funkcji,
- tworzenie parametrów z wartościami domyślnymi,
- zwracanie danych z funkcji.

### 7. Paczki i moduły:

- tworzenie kodu, którego można użyć ponownie, w postaci modułu,
- używanie w skryptach modułów napisanych przez innych oraz dostępnych w standardowej bibliotece Pythona.

## 8. Wyjątki i obsługa:

- wprowadzenie do wyjątków,
- prawidłowa obsługa wyjątków,
- samodzielne zgłaszanie wyjątków,
- samodzielne tworzenie własnych wyjątków.

## 9. Pliki:

- operacje na plikach,
- tworzenie, wczytywanie oraz zapisywanie plików.

## 10. Dokumentacja

- praca z dokumentacją Pythona

## 11. Klasy i obiekty:

- wprowadzenie do klas i obiektów,
- definicja metody i różnice między funkcją a metodą,
- tworzenie obiektów,
- przekazywanie danych do obiektów,
- podstawowe zagadnienia związane z dziedziczeniem oraz nadpisywaniem metod.

## 12. Paradygmaty i dobre praktyki programowania obiektowego:

- jak poprawnie projektować klasy,
- kiedy tworzyć klasy, a kiedy korzystać z funkcji.

## 13. Serializacja:

- definicja serializacji,
- praktyczne zastosowania,
- podstawowe formaty dostępne w standardowej bibliotece Pythona.

## 14. Moduł unittest:

- wprowadzenie do tworzenia testów jednostkowych dla napisanych na zajęciach skryptów.

## 15. Moduł datetime:

- praca z obiektami reprezentującymi datę oraz godzinę.

## 16. Środowisko uruchomieniowe:

- wprowadzenie do uruchamiania skryptów pythonowych,
- wirtualne środowiska i izolacja zależności,
- instalowanie bibliotek stworzonych przez społeczność Pythona.

