



Program kursu

PythON: Academy

self-learning

603

lekcji wideo

dożywotni dostęp do kursu

Postaw na
praktykę

PythON: Academy

Od składni Pythona do realnych projektów.

Kurs wideo Pythona — od składni po realne projekty (603 lekcje). Uczysz się w swoim tempie i przygotowujesz do certyfikacji PCEP.

Co będziesz potrafić po kursie?

- ✓ Poznasz składnię Pythona w praktycznych projektach
- ✓ Rozwiązujesz zadania programistyczne krok po kroku
- ✓ Przygotowujesz się do certyfikacji PCEP
- ✓ Uczysz się w swoim tempie z dożywotnim dostępem

PythON: Academy

14 modułów

1

MODUŁ 1

Fundamenty programowania i Python

Wprowadzenie do programowania i języka Python: przygotowanie środowiska pracy, pierwszy program oraz praca z interaktywną konsolą.

ZAKRES MODUŁU

- ✓ Czym jest Python i dlaczego warto rozumieć programowanie
- ✓ Pierwszy program "Hello World" w Pythonie
- ✓ Przygotowanie środowiska pracy (Windows, Linux, MacOS)
- ✓ Interaktywna konsola i zmienne
- ✓ Funkcja print, napisy i funkcja input
- ✓ Komentarze w kodzie

2

MODUŁ 2

Typy danych, instrukcje warunkowe i pętle

Podstawowe typy zmiennych oraz sterowanie przepływem programu: instrukcje warunkowe, operatory logiczne i pętle.

ZAKRES MODUŁU

- ✓ Typy int, float, str, list, dict, NoneType
- ✓ Operatory logiczne and / or / not oraz in / is
- ✓ Porównania i typ bool
- ✓ Pętle while i for (w tym for in range)
- ✓ Instrukcje if / else / elif
- ✓ Komendy break i continue

3

MODUŁ 3

Funkcje

Definiowanie i wykorzystanie funkcji w Pythonie wraz z argumentami, zakresem zmiennych oraz konstrukcjami zaawansowanymi.

ZAKRES MODUŁU

- ✓ Definiowanie funkcji i ich wywoływanie
- ✓ Argumenty pozycyjne, nazwane i tylko nazwane
- ✓ Parametry domyślne i zakres zmiennych
- ✓ Inner functions i rekurencja
- ✓ Funkcja jako obiekt oraz lambda (funkcja anonimowa)
- ✓ Unpacking operators

4

MODUŁ 4

Programowanie obiektowe (OOP)

Paradygmat obiektowy w Pythonie: klasy, obiekty, metody magiczne, enkapsulacja, dziedziczenie i polimorfizm wraz z dobrymi praktykami.

ZAKRES MODUŁU

- ✓ Klasy i obiekty, konstruktor i pola obiektu
- ✓ Metody oraz metody magiczne (`__str__`, `__repr__`, `__eq__`, `__add__` i inne)
- ✓ Enkapsulacja, metody i pola klasy, metody statyczne
- ✓ Dziedziczenie, metoda `super`, wielodziedziczenie i MRO
- ✓ Abstrakcja i polimorfizm
- ✓ Property (getter/setter), refaktoryzacja
- ✓ Dobre praktyki: reguły SOLID i DRY

5

MODUŁ 5

Wbudowane struktury danych

Przegląd wbudowanych i zaawansowanych struktur danych Pythona oraz różnice między obiektami mutowalnymi a niemutowalnymi.

ZAKRES MODUŁU

- ✓ Listy i list comprehensions, słowniki i dict comprehensions
- ✓ Set i frozenset, mutable vs immutable
- ✓ Tuple oraz named tuple
- ✓ Dataclass
- ✓ Enum
- ✓ Moduł collections: defaultdict, OrderedDict, deque, Counter, ChainMap

6

MODUŁ 6

Środowisko, biblioteki i zarządzanie zależnościami

Praca ze środowiskami wirtualnymi, instalacja i zarządzanie zależnościami oraz przegląd przydatnych bibliotek zewnętrznych.

ZAKRES MODUŁU

- ✓ Wirtualne środowiska: virtualenv, virtualenvwrapper, pipenv
- ✓ Instalacja bibliotek (pip) i zarządzanie zależnościami
- ✓ Zależności developerskie i produkcyjne
- ✓ Skąd biorą się biblioteki i jak je znaleźć
- ✓ Przydatne biblioteki: Black, GeoPy, Folium, openpyxl, pydantic, requests

7

MODUŁ 7

Obsługa błędów, praca z plikami i scraping

Obsługa wyjątków, praca z plikami w formatach CSV i JSON oraz wprowadzenie do web scrapingu.

ZAKRES MODUŁU

- ✓ Wyjątki w Pythonie i tworzenie własnych wyjątków
- ✓ Blok try... except..., finally oraz else
- ✓ Praca z plikami (konstrukcja with), formaty CSV i JSON
- ✓ Web scraping: HTML, protokół HTTP, biblioteka requests
- ✓ Przetwarzanie treści i biblioteka BeautifulSoup

8

MODUŁ 8

Git i kontrola wersji

Podstawy i praca zespołowa z systemem kontroli wersji Git oraz przyjęte workflow programistyczne.

ZAKRES MODUŁU

- ✓ Czym jest system kontroli wersji i Git, Git vs GitHub
- ✓ Instalacja i konfiguracja Gita
- ✓ Repozytorium, pierwszy commit i historia zmian
- ✓ Branche, mergowanie i rozwiązywanie konfliktów
- ✓ Tagi, pull requests i code review
- ✓ Workflow: Gitflow i GitHub flow

9

MODUŁ 9

Data, czas i strefy czasowe

Obsługa daty i czasu w Pythonie: typy date, time, datetime, formatowanie, interwały czasowe oraz strefy czasowe.

ZAKRES MODUŁU

- ✓ Typy date, time oraz datetime
- ✓ Data i czas w formacie ISO oraz UNIX Timestamp
- ✓ Formatowanie i parsowanie daty i czasu z różnych formatów
- ✓ Porównywanie dat i czasu, timedelta
- ✓ Strefy czasowe i obiekty aware
- ✓ Bezpieczne metody pracy z czasem

10

MODUŁ 10

Wyrażenia regularne

Praca z wyrażeniami regularnymi w Pythonie: budowa wzorców, dopasowywanie i wyszukiwanie tekstu.

ZAKRES MODUŁU

- ✓ Metaznaki: klasy znaków i specjalne symbole
- ✓ Raw strings i określanie liczby wystąpień
- ✓ Grupowanie
- ✓ Metody match, fullmatch i search
- ✓ Metody findall i finditer
- ✓ Flagi wyrażen regularnych, kompilacja a wydajność

11

MODUŁ 11

Type hint i statyczna kontrola typów

Typowanie statyczne w Pythonie: adnotacje typów, kontrola zgodności narzędziem mypy oraz typy złożone i generyczne.

ZAKRES MODUŁU

- ✓ Typowanie dynamiczne i statyczne
- ✓ Type hints w funkcjach i zmiennych
- ✓ Kontrola zgodności typów narzędziem mypy
- ✓ Deklarowanie złożonych typów i generyki
- ✓ Rzutowanie typu oraz przypadki szczególne (Any, ignore)

12

MODUŁ 12

Zaawansowany Python: iteratory, dekoratory, metaklasy, wzorce

Zaawansowane konstrukcje języka: abstrakcja i Protocol, iteratory i generatory, context manager, dekoratory, metaklasy oraz wybrane wzorce projektowe.

ZAKRES MODUŁU

- ✓ Abstrakcja, interfejs i Protocol (structural subtyping), stub files
- ✓ Protokół iteratora i generatory (send, throw, close)
- ✓ Context manager (własny i jako funkcja generator)
- ✓ Dekoratory: z argumentami, funkcji i klas, functools.wrap, lru_cache, total_ordering
- ✓ Metaklasy: dynamiczne tworzenie i modyfikacja klas
- ✓ Wzorce projektowe: Factory, Builder, Adapter, Fasada, Obserwator, Iterator

13

MODUŁ 13

Testowanie kodu

Pisanie testów automatycznych w Pythonie z użyciem pytest i unittest oraz podejście Test Driven Development.

ZAKRES MODUŁU

- ✓ Biblioteka pytest i pisanie prostych testów
- ✓ Testowanie wyjątków i fixtury
- ✓ Stubby i mocki
- ✓ Moduł unittest, pytest a unittest
- ✓ Test Driven Development (TDD)

14

MODUŁ 14

Współbieżność, równoległość i CI

Programowanie współbieżne i równoległe w Pythonie (wątki, procesy, asynchroniczność) oraz praktyki ciągłej integracji.

ZAKRES MODUŁU

- ✓ Wątki, wyścigi, sekcja krytyczna i synchronizacja, moduł queue
- ✓ Global Interpreter Lock, CPU bound vs IO bound
- ✓ Procesy, komunikacja międzyprocesowa, subprocess, skalowalność
- ✓ Wykonanie asynchroniczne i model kooperatywny
- ✓ Proces CI: testy automatyczne, kontrola typów, linter i formatowanie

Kurs w formule self-learning

603 lekcji wideo w Twoim tempie, z dożywotnim dostępem do materiałów.



Wideo na żądanie

Uczysz się we własnym tempie, wracasz do lekcji kiedy chcesz.



Certyfikat

Kurs przygotowuje do certyfikatu PCEP.



Technologie

Python



Praktyka

Materiał oparty na realnych przykładach i nowoczesnym warsztacie pracy.

Sfinansuj kurs tak, jak Ci wygodnie

W infoShare Academy oferujemy różne formy sfinansowania kursu — bez względu na to, który pakiet wybierzesz. Sprawdź, która forma jest dla Ciebie najlepsza!

01 Raty PayU 0%

Najszybszy sposób na rozłożenie płatności na niskie, comiesięczne raty. Do wyboru 5, 10 lub 15 rat 0% albo 24, 36, a nawet 50 rat nisko oprocentowanych.

02 Płatność jednorazowa

Płacisz i zaczynasz naukę. Jednorazowa płatność często wiąże się z dużym rabatem — zapytaj o niego nasze konsultantki.

Stawiamy na jakość

2015

uczymy IT
od tego roku

9970+

przeszkolonych
osób

330+

trenerów
w naszej kadrze

Łączymy kompetencje techniczne z praktyką biznesową, a każdy kurs prowadzą trenerzy pracujący na co dzień w branży. Dzięki temu uczysz się tego, co realnie przydaje się w pracy.



MASZ PYTANIA?

Sylwia Liedtke

Doradczyni ds. kursów

+48 730 822 825 · sylwia.liedtke@infohareacademy.com