



INFOSHARE
ACADEMY SPÓŁKA
Z OGRANICZONĄ
ODPOWIEDZIALNOŚ
CIĄ
★★★★★ 4,6 / 5
190 ocen

Kurs Python + Django + AI – weekendowy - pakiet IT Nawigator | forma zdalna w czasie rzeczywistym

Numer usługi 2025/03/19/11051/2633969

9 900,00 PLN brutto
8 048,78 PLN netto
41,25 PLN brutto/h
33,54 PLN netto/h

📍 zdalna w czasie rzeczywistym

🏠 Usługa szkoleniowa

🕒 240 h

📅 23.08.2025 do 12.04.2026

Informacje podstawowe

Kategoria

Informatyka i telekomunikacja / Programowanie

Grupa docelowa usługi

Kurs Python + Django + AI – weekendowy jest dla każdego entuzjasty technologicznego, który chce zostać programistą.

W czasie całego cyklu nauczania staramy się dać adeptom nie tylko wiedzę, ale przede wszystkim możliwość zrozumienia, jak wygląda codzienna praca dobrego programisty. Wszystkie zajęcia mają więc formę warsztatową i prowadzone są przez doświadczonych praktyków, specjalistów w wybranych dziedzinach.

Dla kogo jest kurs Python + Django + AI:

1. osób, które chcą tworzyć aplikacje i skrypty,
2. osób zainteresowanych automatyzacją zadań i procesów,
3. osób, które chcą tworzyć strony internetowe i backend,
4. osób, które chcą rozwijać się w programowaniu i rozwoju oprogramowania

Minimalna liczba uczestników

12

Maksymalna liczba uczestników

18

Data zakończenia rekrutacji

21-08-2025

Forma prowadzenia usługi

zdalna w czasie rzeczywistym

Liczba godzin usługi

240

Podstawa uzyskania wpisu do BUR

Znak Jakości TGLS Quality Alliance

Cel

Cel edukacyjny

Celem szkolenia jest przygotowanie osób do podjęcia pracy jako Junior Python Developer. Ma ono wyposażyć uczestników w znajomość technologii, języków i narzędzi, niezbędnych do pracy, ale także wyrobić dobre nawyki i solidny warsztat programistyczny.

- Uczestnik po kursie potrafi:
- *tworzyć nowoczesne aplikacje webowe
 - *efektywnie posługiwać się narzędziami developerskimi
 - *pracować w zespole programistycznym w metodyce Scrum
 - *programować w Pythonie
 - *korzystać z frameworków Django, Flask i FAST API

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Charakteryzuje narzędzia Linux i Terminal	Stosuje zaawansowane polecenia i płynnie nawiguje po systemie plików. Automatyzacja zadań: potrafi tworzyć skrypty do automatyzacji powtarzalnych zadań. Konfiguracja środowiska: dokonuje personalizacji i konfiguracji swojego środowiska pracy zgodnie z indywidualnymi preferencjami lub wymaganiami projektu – tak jak w prawdziwej pracy.	Test teoretyczny z wynikiem generowanym automatycznie
Opisuje system Kontroli Wersji Git	Obsługuje rozgałęzienie i scalenie: potrafi dokonywać zaawansowanych operacji na gałęziach projektu. Zarządzanie konfliktami: obsługuje narzędzia oraz techniki potrzebne do rozwiązywania konfliktów wynikających z równoczesnej edycji kodu przez różnych programistów. Praca z GitHub: stosuje zasady publikowania i współpracy nad projektem na platformie GitHub.	Test teoretyczny z wynikiem generowanym automatycznie
Charakteryzuje podstawy Programowania Python	Obsługuje struktury danych: definiuje zastosowanie podstawowych struktur danych, takich jak listy, krotki, słowniki oraz dowie się, jak z nimi efektywnie pracować. Obsługa plików: charakteryzuje metody manipulacji plikami, wczytywania i zapisywania danych. Podstawy algorytmów. Na zajęciach omawia sposoby implementacji prostych algorytmów do rozwiązywania programistycznych problemów.	Test teoretyczny z wynikiem generowanym automatycznie

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Opisuje programowanie obiektowe (OOP) w Pythonie Definiuje prace z wirtualnymi środowiskami	Obsługuje tworzenie klas i obiektów: opisuje podstawy projektowania i implementacji klas oraz ich instancji. Dziedziczenie i polimorfizm: na zajęciach omawia, jak wykorzystywać dziedziczenie do współdzielenia kodu oraz stosować polimorfizm. Enkapsulacja: potrafi ukrywać implementację i ograniczać dostęp do danych zapewniając bezpieczeństwo i integralność kodu. Obsługuje Instalację pakietów: stosuje techniki skutecznego korzystania z narzędzia pip do instalacji pakietów Python. Tworzenie wirtualnych środowisk: opisuje, jak tworzyć i zarządzać wirtualnymi środowiskami za pomocą narzędzi takich jak virtualenv. Zarządzanie zależnościami projektu: zarządza wymaganymi wersjami pakietów dla projektu jest kluczowe w przypadku pracy z wieloma bibliotekami Pythona.	Test teoretyczny z wynikiem generowanym automatycznie Test teoretyczny z wynikiem generowanym automatycznie
Opisuje framework Django Charakteryzuje bazy danych i język SQL	Obsługuje Tworzenie aplikacji w Django: charakteryzuje zasady projektowania i rozwijania prostych aplikacji webowych przy użyciu Django, Obsługa bazy danych: korzysta z ORM do komunikacji z bazą danych. Administracja aplikacją: potrafi skonfigurować i korzystać z wbudowanego Admin Panelu w celu zarządzania danymi, co pozwala patrzeć globalnie na cały projekt. Definiuje podstawy zapytań w języku SQL: omawia zasady tworzenia prostych zapytań do baz danych opartych o: SELECT, INSERT, UPDATE, DELETE. Operacje na tabelach: opisuje dobre praktyki efektywnego projektowania, modyfikowania i usuwania tabel, a także definiowanie kluczy głównych i obcych. Zaawansowane zapytania: omawia, jakie są zaawansowane funkcje języka SQL, takie jak JOIN, GROUP BY, HAVING.	Test teoretyczny z wynikiem generowanym automatycznie Test teoretyczny z wynikiem generowanym automatycznie

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Definiuje wzorce projektowe oraz refaktoring i debugowanie kodu Charakteryzuje proces testowania	Obsługuje wzorce projektowe: opisuje i wdraża w projekcie zespołowym, zastosowanie popularnych wzorców projektowych, takich jak Singleton, Factory, Observer. Clean Code: potrafi stworzyć czytelny, zorganizowany i efektywny kod w języku Python zgodnie z zasadami Clean Code. Refaktoring: potrafi poprawić istniejący kod bez zmiany jego funkcjonalności. Omawia zasady tworzenia testów przy użyciu frameworka PyTest oraz korzystania z asercji do sprawdzania poprawności ich wyników. Unittest.mock: opisuje izolowanie kodu testowanego poprzez mockowanie zależności. Testy funkcjonalne: potrafi tworzyć testy, które sprawdzają całe funkcjonalności systemu, co jest ostatnim ogniwem na drodze do kompleksowej weryfikacji działania aplikacji.	Test teoretyczny z wynikiem generowanym automatycznie Test teoretyczny z wynikiem generowanym automatycznie
Opisuje API: HTTP, REST, WebAPI Charakteryzuje Frontend: HTML, CSS, Bootstrap	Używa protokołu HTTP: omawia zasady działania protokołu HTTP w tym metod żądań oraz nagłówków. Architektura REST: omawia projektowanie i implementacja interfejsów RESTful. Obsługa autoryzacji: opisuje zabezpieczanie dostępu do swojego API za pomocą różnych mechanizmów autoryzacji. Opisuje obsługę HTML: omawia tworzenie struktury strony internetowej, korzystanie z różnych tagów i atrybutów. CSS: opisuje jak ostylewać stronę, nadawać wygląd poszczególnym jej elementom, a także pracować z selektorami. Bootstrap: charakteryzuje szybkie projektowanie responsywnych interfejsów przy użyciu gotowych komponentów.	Test teoretyczny z wynikiem generowanym automatycznie Test teoretyczny z wynikiem generowanym automatycznie

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Definiuje Deploy Aplikacji z wykorzystaniem AWS i Dockera Opisuje Loggery, Asynchroniczność, Web Scraping	Obsługuje Docker: omawia tworzenie i zarządzanie kontenerami. Opisuje niezbędne elementy związane z tym narzędziem. AWS: opisuje wdrażanie aplikacji na platformie AWS – potrafi określić główne rozwiązania chmury Amazona. Jenkins CI: opisuje konfigurację procesu Continuous Integration i Continuous Deployment, który wpływa na automatyzację procesów. Obsługuje Loggery w Pythonie: i opisuje korzystanie z modułu logging w celu efektywnego logowania zdarzeń w aplikacji. Programowanie asynchroniczne: omawia wykorzystanie mechanizmów asynchronicznych w Pythonie. Web Scraping z BeautifulSoup i Requests: opisuje pozyskiwanie danych z witryn internetowych, przetwarzanie i analiza zebranych informacji są kluczowe w procesie analizy konkurencji.	Test teoretyczny z wynikiem generowanym automatycznie Test teoretyczny z wynikiem generowanym automatycznie
Definiuje Flask	Obsługuje Podstawy frameworka Flask: omawia podstawy projektowania tras, obsługi żądań i odpowiedzi HTTP. Szablony Jinja2: opisuje korzystanie z szablonów do dynamicznego generowania treści HTML, co jest niezbędne przy efektywnej pracy z Flaskiem. Rozwijanie aplikacji Flask: omawia dodawanie funkcjonalności, obsługę formularzy czy korzystanie z rozszerzeń, jako niektóre z tematów, które wchodzą w skład podstaw Flaska.	Test teoretyczny z wynikiem generowanym automatycznie
Charakteryzuje UX oraz Zadania Rekrutacyjne	Obsługuje Projektowanie interfejsów: opisuje jak budować interfejsy tak, aby w najlepszy sposób wykorzystać zasady designu i ergonomii Testy użyteczności omawia: przeprowadzanie testów użyteczności, które sprawdzają, jak dobrze użytkownicy radzą sobie z interfejsem, co jest kluczem do kompleksowej weryfikacji działania aplikacji. Opisuje efektywną prezentację.	Test teoretyczny z wynikiem generowanym automatycznie
Definiuje czym jest empatia	Charakteryzuje umiejętności zrozumienia i odczuwania emocji innych osób w projekcie zespołowym	Test teoretyczny z wynikiem generowanym automatycznie

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Definiuje metodykę Scrum	Charakteryzuje umiejętności współpracy pracy w grupie, dzielenia się zadaniami i osiągania wspólnych celów.	Test teoretyczny z wynikiem generowanym automatycznie
Charakteryzuje czym jest otwartość na feedback	Uczestnik nauczy się gotowości do przyjmowania konstruktywnej krytyki i wykorzystywania jej do własnego rozwoju.	Test teoretyczny z wynikiem generowanym automatycznie

Kwalifikacje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji zawiera opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument potwierdza, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji?

TAK

Pytanie 3. Czy dokument potwierdza zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

Program

Celem szkolenia jest przygotowanie osób do podjęcia pracy jako Junior Python Developer. Ma ono wyposażyć uczestników w znajomość technologii, języków i narzędzi, niezbędnych do pracy, ale także wyrobić dobre nawyki i solidny warsztat programistyczny.

W czasie całego cyklu nauczania staramy się dać adeptom nie tylko wiedzę, ale przede wszystkim możliwość zrozumienia, jak wygląda codzienna praca dobrego programisty. Wszystkie zajęcia mają więc formę warsztatową i prowadzone są przez doświadczonych praktyków, specjalistów w wybranych dziedzinach.

Zajęcia trwać będą przez 20 weekendów. Każdy weekend w Akademii będzie miał podobną formę. W sobotę odbywać się będą zajęcia warsztatowe, na których uczestnicy nabywać będą nowe umiejętności. W niedzielę dzień podzielony będzie na 3 godziny warsztatowe na temat narzędzi bądź dobrych praktyk oraz 3 godziny projektowe. W ramach projektów uczestnicy będą mieli możliwość wykorzystania zdobytych kompetencji w pracy zespołowej.

Zajęcia odbywają się w godzinach zegarowych.

W harmonogramie ujęto przerwy, przerwy wliczają się w czas trwania szkolenia.

Przerwy w czasie trwania kursu są dostosowane do tempa pracy uczestników. Są indywidualne ustalane z trenerem prowadzącym zajęcia z uczestnikami kursu.

Wymagania sprzętowe: system operacyjny Linux lub macOS.

Optymalna konfiguracja sprzętowa: procesor i5+ lub podobny, 8GB+ pamięci RAM, zalecany dysk SSD.

Poniżej znajduje się lista modułów szkoleniowych, które zostaną zrealizowane podczas kursu „Python + Django + AI” w infoShare Academy.

W przypadku każdej z grup szkoleniowych program będzie dostosowywany do poziomu zaawansowania grupy, także w trakcie trwania szkolenia.

W ramach projektów uczestnicy będą mieli możliwość wykorzystania zdobytych kompetencji w pracy zespołowej.

1. Narzędzia programistyczne

Poznanie podstawowych narzędzi programistycznych, używanych w codziennej pracy programisty, ze szczególnym uwzględnieniem IDE (PyCharm), podstaw używania repozytorium kodu git (GitHub) i konteneryzacji (Docker).

2. Scrum

Reguły zwinnego wytwarzania oprogramowania, ze szczególnym uwzględnieniem metodyki Scrum.

3. Podstawy programowania

Wprowadzenie do programowania, na przykładzie języka Python. Podstawowe struktury danych oraz ich sprawne przetwarzanie.

4. Django

Wprowadzenie do tworzenia aplikacji webowych w oparciu o framework Django. Zapoznanie z architekturą aplikacji webowych oraz przedstawienie idei frameworka webowego.

5. Testowanie automatyczne

Automatycznie testowanie aplikacji na poziomie jednostkowym i integracyjnym. Idea i praktyczne zastosowanie Test Driven Development, w tym mockowanie obiektów. Nabycie umiejętności debugowania i profilowania kodu aplikacji.

6. Loggery

Praktyczne zastosowanie najpopularniejszych loggerów. Umiejętności wartościowego logowania istotnych informacji.

7. Refaktoryzacja

Poznanie idei, celu i zrozumienie potrzeby refaktoryzacji kodu oraz nabycie umiejętności jego realizacji przy aktywnym wsparciu przez IDE.

8. System kontroli wersji git

Warsztaty ze skutecznego używania systemu kontroli wersji git, w tym rozwiązywanie konfliktów, git flow oraz narzędzia wspomagające.

9. Przeglądy kodu

Zrozumienie idei, wartości i sztuki przeglądów kodu. Zrozumienie znaczenia czystości kodu oraz dbania o jego utrzymywalność.

10. Środowisko ciągłej integracji

Poznanie wartości środowiska ciągłej integracji oraz umiejętności efektywnego jego wykorzystania, na przykładzie systemu Jenkins.

11. UML i wzorce projektowe

Poznanie i zrozumienie znaczenia podstawowych diagramów UML w celu skuteczniejszej komunikacji z innymi developerami. Poznanie i zrozumienie podstawowych wzorców projektowych.

12. Budowanie i używanie REST API

Poznanie REST oraz nabycie umiejętności prawidłowego definiowania API oraz jego implementowania w języku Python.

13. Bazy danych

Wykorzystanie baz danych SQL do przechowywania danych.

14. Flask

Podstawy frameworku do tworzenia prostych aplikacji webowych.

Harmonogram

Liczba przedmiotów/zajęć: 69

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
1 z 69 Rozpoczęcie kursu Organizacja	Piotr Chudzik	23-08-2025	08:00	11:00	03:00
2 z 69 Narzędzia programisty	Piotr Chudzik	23-08-2025	11:00	14:00	03:00
3 z 69 podstawy programowania 1/4	Jarosław Majka	24-08-2025	08:00	11:00	03:00
4 z 69 GIT 1/3	Jarosław Majka	24-08-2025	11:00	14:00	03:00
5 z 69 podstawy programowania 2/4	Jarosław Majka	06-09-2025	08:00	14:00	06:00
6 z 69 podstawy programowania 3/4	Jarosław Majka	07-09-2025	08:00	11:00	03:00
7 z 69 GIT 2/3	Jarosław Majka	07-09-2025	11:00	14:00	03:00
8 z 69 podstawy programowania 4/4	Jarosław Majka	13-09-2025	08:00	14:00	06:00
9 z 69 Python - wyjątki / debugger	Jarosław Majka	14-09-2025	08:00	11:00	03:00
10 z 69 GIT 3/3	Jarosław Majka	14-09-2025	11:00	14:00	03:00
11 z 69 Python - praca z plikami	Jarosław Majka	27-09-2025	08:00	11:00	03:00
12 z 69 OOP w Pythonie 1/4	Wiktoria Piela	27-09-2025	11:00	14:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
13 z 69 OOP w Pythonie 2/4	Wiktor Piela	28-09-2025	08:00	11:00	03:00
14 z 69 Scrum	Wiktor Piela	28-09-2025	11:00	14:00	03:00
15 z 69 OOP w Pythonie 3/4	Wiktor Piela	04-10-2025	08:00	14:00	06:00
16 z 69 OOP w Pythonie 4/4	Wiktor Piela	05-10-2025	08:00	11:00	03:00
17 z 69 Projekt Planowanie S1	Konrad Uciechowski	05-10-2025	11:00	14:00	03:00
18 z 69 pip / wirtualne środowiska	Wiktor Piela	18-10-2025	08:00	11:00	03:00
19 z 69 Linux	Piotr Chudzik	18-10-2025	11:00	14:00	03:00
20 z 69 Python VM - REPL - Python interpreter	Piotr Chudzik	19-10-2025	08:00	11:00	03:00
21 z 69 Projekt	Konrad Uciechowski	19-10-2025	11:00	14:00	03:00
22 z 69 Django (wstęp do Django) 1/3	Piotr Chudzik	25-10-2025	08:00	14:00	06:00
23 z 69 Python - clean code, code review	Piotr Chudzik	26-10-2025	08:00	11:00	03:00
24 z 69 Projekt	Konrad Uciechowski	26-10-2025	11:00	14:00	03:00
25 z 69 SQL - bazy danych - definicja modelu danych	Konrad Uciechowski	08-11-2025	08:00	14:00	06:00
26 z 69 wzorce projektowe	Konrad Uciechowski	09-11-2025	08:00	11:00	03:00
27 z 69 Projekt	Konrad Uciechowski	09-11-2025	11:00	14:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
28 z 69 wstęp do testowania (wstęp do PyTest)	Konrad Uciechowski	15-11-2025	08:00	11:00	03:00
29 z 69 HTTP	Konrad Uciechowski	15-11-2025	11:00	14:00	03:00
30 z 69 API - WebAPI	Konrad Uciechowski	16-11-2025	08:00	11:00	03:00
31 z 69 Projekt	Konrad Uciechowski	16-11-2025	11:00	14:00	03:00
32 z 69 Django 2/3	Konrad Uciechowski	29-11-2025	08:00	14:00	06:00
33 z 69 Django 3/3	Konrad Uciechowski	30-11-2025	08:00	11:00	03:00
34 z 69 Projekt Przegląd S1	Konrad Uciechowski	30-11-2025	11:00	12:00	01:00
35 z 69 Projekt Retrospektywy	Konrad Uciechowski	30-11-2025	12:00	14:00	02:00
36 z 69 Python - refaktoring kodu	Konrad Uciechowski	13-12-2025	08:00	11:00	03:00
37 z 69 PyTest - asercje, fixtures i pluginy	Konrad Uciechowski	13-12-2025	11:00	14:00	03:00
38 z 69 unittest.mock - TDD	Konrad Uciechowski	14-12-2025	08:00	11:00	03:00
39 z 69 Projekt Planowanie S2	Konrad Uciechowski	14-12-2025	11:00	14:00	03:00
40 z 69 (FRONTEND) HTML - CSS - Bootstrap - JS 1/2	Konrad Uciechowski	10-01-2026	08:00	14:00	06:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
41 z 69 (FRONTEND) HTML - CSS - Bootstrap - JS 2/2	Konrad Uciechowski	11-01-2026	08:00	11:00	03:00
42 z 69 Projekt	Konrad Uciechowski	11-01-2026	11:00	14:00	03:00
43 z 69 Django's ORM - zaawansowane techniki	Konrad Uciechowski	17-01-2026	08:00	14:00	06:00
44 z 69 Django Admin Panel	Konrad Uciechowski	18-01-2026	08:00	11:00	03:00
45 z 69 Projekt	Konrad Uciechowski	18-01-2026	11:00	14:00	03:00
46 z 69 debugowanie	Konrad Uciechowski	31-01-2026	08:00	11:00	03:00
47 z 69 loggery	Konrad Uciechowski	31-01-2026	11:00	14:00	03:00
48 z 69 architektura aplikacji	Konrad Uciechowski	01-02-2026	08:00	11:00	03:00
49 z 69 Projekt	Konrad Uciechowski	01-02-2026	11:00	14:00	03:00
50 z 69 docker	Konrad Uciechowski	07-02-2026	08:00	14:00	06:00
51 z 69 Jenkins CI	Konrad Uciechowski	08-02-2026	11:00	14:00	03:00
52 z 69 Projekt Przegląd S2	Konrad Uciechowski	08-02-2026	11:00	12:00	01:00
53 z 69 Projekt Retrospektywy	Konrad Uciechowski	08-02-2026	12:00	14:00	02:00
54 z 69 REST API	Konrad Uciechowski	21-02-2026	08:00	14:00	06:00
55 z 69 Migracje	Konrad Uciechowski	22-02-2026	08:00	11:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
56 z 69 Projekt Planowanie S3	Konrad Uciechowski	22-02-2026	11:00	14:00	03:00
57 z 69 FAST API	Konrad Uciechowski	28-02-2026	08:00	14:00	06:00
58 z 69 web scrapping	Konrad Uciechowski	01-03-2026	08:00	11:00	03:00
59 z 69 Projekt	Konrad Uciechowski	01-03-2026	11:00	14:00	03:00
60 z 69 asynchroniczność	Konrad Uciechowski	14-03-2026	08:00	14:00	06:00
61 z 69 UX	Konrad Uciechowski	15-03-2026	11:00	14:00	03:00
62 z 69 Projekt	Konrad Uciechowski	15-03-2026	11:00	14:00	03:00
63 z 69 AWS	Konrad Uciechowski	21-03-2026	08:00	14:00	06:00
64 z 69 UML	Konrad Uciechowski	22-03-2026	08:00	11:00	03:00
65 z 69 Walidacja za pomocą testu z wynikiem generowanym automatycznie	Konrad Uciechowski	22-03-2026	11:00	12:00	01:00
66 z 69 Projekt	Konrad Uciechowski	22-03-2026	12:00	14:00	02:00
67 z 69 Flask	Konrad Uciechowski	11-04-2026	08:00	14:00	06:00
68 z 69 zadania rekrutacyjne	Konrad Uciechowski	12-04-2026	08:00	11:00	03:00
69 z 69 Przegląd końcowy Retrospektywy końcowe Zakończenie kursu	Konrad Uciechowski	12-04-2026	11:00	14:00	03:00

Cennik

Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	9 900,00 PLN
Koszt przypadający na 1 uczestnika netto	8 048,78 PLN
Koszt osobogodziny brutto	41,25 PLN
Koszt osobogodziny netto	33,54 PLN

Prowadzący

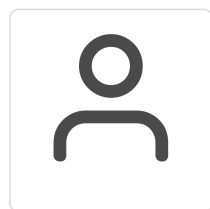
Liczba prowadzących: 8



1 z 8

Filip Szyler

Ukończył Politechnikę Warszawską na kierunku Big Data oraz Politechnikę Gdańską na kierunku Inżynieria Danych. Przez 2 lata był wykładowcą na Wyższej Szkole Bankowej, prowadził zajęcia z Marketingu internetowego i e-commerce oraz Media społecznościowe i zintegrowany marketing internetowy. Posiada wieloletnie doświadczenie jako SEO Manager, a jako Data Scientist pracuje od ponad czterech lat. Z InfoShare Academy związany jest od 2019 roku.



2 z 8

Damian Filipkowski

Ukończył Uniwersytet Technologiczny w Irlandii oraz Uniwersytet Morski w Gdyni na Wydziale Nawigacyjnym. W 2017 roku przebranżowił się - ukończył dodatkowe kursy z programowania i tworzenia stron internetowych. Posiada wieloletnie doświadczenie zawodowe jako Python Developer. Z InfoShare Academy współpracuje od prawie 6-ciu lat.



3 z 8

Tomasz Słupik

Ukończył Politechnikę Warszawską na kierunku Zarządzanie i Inżynieria produkcji oraz Akademię Ekonomiczno - Humanistyczną w Warszawie. Dodatkowo zrobił studia podyplomowe na Akademii Leona Koźmińskiego na kierunkach Frontend Development z React oraz Python Development. Pracuje na stanowisku Senior Python Developer ale jest również Fullstack Developerem. Z InfoShare Academy związany jest od roku. Jego doświadczenie zawodowe wynosi 5 lat.



4 z 8

Piotr Chudzik

Ukończył Politechnikę Łódzką na kierunku Informatyka oraz studia podyplomowe na Uniwersytecie Łódzkim na kierunku Przechowywanie danych i administracja bazami danych Microsoft. Z IT jest

związany już prawie 10 lat. Pracował w różnych sektorach związanych z danymi: od administratora baz danych, poprzez DataOps, a skończywszy na Data Engineer. Zajmuje się również planowaniem i zarządzaniem zespołami (aktualnie pracuje jako Technical Lead). Autor wideo kursów i książek o tematyce technicznej. Z InfoShare Academy współpracuje od prawie 3 lat.



5 z 8

Konrad Uciechowski

W 2017, będąc jeszcze studentem Akademii Morskiej w Gdyni, zaczął pracę jako software developer dla startupu zajmującego się rozwijaniem aplikacji dla rynku nieruchomości, od 2023 pracuje w Airbus Helicopters Polska na stanowisku software developer. Z infoShare współpracuje jako trener Python od 2024 roku, natomiast na przełomie 2019-2020 był również kursantem na Data Science od podstaw.



6 z 8

Jarosław Majka

Trener Jarosław Majka ukończył Ekonomię w Wyższej Szkole Bankowej w Gdańsku. Z branżą IT związany od 2022 roku, kiedy to ukończył bootcamp z Pythona w infoShare Academy. Zdobywał doświadczenie w budowie bibliotek open source oraz pracy z technologiami front-endowymi. Obecnie specjalizuje się w rozwiązaniach serverless i projektowaniu zaawansowanych systemów API w chmurze za pomocą Pythona. Zajmuje się także tworzeniem infrastruktury oraz jej zarządzaniem.



7 z 8

Wiktoria Piel

Licencjat- kierunek ekonomia - Uniwersytet Ekonomiczny we Wrocławiu, specjalizacja: analityka gospodarcza 2016-2019 magister - kierunek audyt finansowy - Uniwersytet Ekonomiczny w Krakowie, 2019 - 2022. W 2021 rozpoczął pracę jako młodszy analityk danych w branży FMCG, zajął się tam modelowaniem statystycznym, tworzeniem analiz rynku na potrzeby handlu oraz marketingu. Głównym jego zadaniem było wspomaganie decyzji biznesowych przy pomocy danych, optymalizacja listingu produktów spółki w sieciach handlowych oraz pomiary efektywności działań handlu oraz marketingu na rzecz zwiększenia sprzedaży (promocje, merchandising, ekspozycje). Potem pracował jako data scientist w branżach: bankowość, finanse/ubezpieczenia i aktualnie - telekomunikacja. Zajmuje się tworzeniem, wdrażaniem i utrzymaniem modeli uczenia maszynowego, a także rozwiązań opartych o LLM. Jest ponadto odpowiedzialny za budowanie całej infrastruktury systemu predycyjnego - od eksperymentowania z danymi wejściowymi, przez wdrożenie modelu, aż po udostępnienie wyników predykcji interesariuszom lub innemu systemowi informatycznemu.



8 z 8

Anna Cielas

Data Scientistka zorientowana na nowe technologie, z ponad pięcioletnim doświadczeniem w branży. Łączy kompetencje z zakresu inżynierii oprogramowania oraz Data Science z zaawansowaną znajomością Pythona, SQL oraz technologii chmurowych. Posiada tytuł magistra informatyki z Uniwersytetu WSB Merito w Poznaniu ze specjalizacją Big Data oraz ukończyła inżynierię chemiczną i procesową na Politechnice Poznańskiej. W swojej karierze realizowała projekty z zakresu data science i inżynierii danych w firmach takich jak Allegro, Accenture, Kyndryl (dawniej IBM), Netguru oraz PSI. Jej doświadczenie, między innymi w branżach e-commerce oraz prawniczej, obejmuje klasyczne uczenie maszynowe, przetwarzanie języka naturalnego (NLP), generatywną AI oraz budowę potoków danych. Od 2024 roku dzieli się swoją wiedzą jako trenerka w infoShare Academy. Anna posiada liczne certyfikaty Google Cloud i Microsoft i z pasją wykorzystuje

zaawansowaną analitykę oraz uczenie maszynowe do rozwiązywania rzeczywistych problemów biznesowych

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Prework, czyli materiały przygotowujące do kursu. Uczestnik dostaje w trakcie kursu wszelkie materiały zawierające kod źródłowy, prezentacje i dostęp do repozytorium – ćwiczeń.

Kursanci otrzymują materiały po każdym bloku tematycznym. Trenerzy udostępniają autorskie materiały.

Dodatkowo w ramach tego pakietu kursanci otrzymują kurs online kurs **Prompt Engineering (6h online)** do przerobienia we własnym zakresie.

A także podstawowe wsparcie HR na slacku - między innymi kursant otrzyma wskazówki od Doradcy Kariery dotyczące przygotowania dobrego CV czy profilu na LinkedIn dostosowanego do tej branży (ok 4h).

Informacje dodatkowe

infoShare Academy uczy na bieżąco, poprzez obserwację pracy przy realizacji projektów, weryfikuje stopień przyswojenia wiedzy i motywuje do dalszej intensywnej pracy.

Uczestnikowi oferujemy:

- wiedzę na poziomie Junior Python Developera
- pomoc najlepszych trenerów
- wsparcie uczestnika do wejścia na rynek IT poprzez; doradztwo w przygotowaniu CV, profilu na LinkedIn, profilu na GitHub.

Całe szkolenie będzie realizowane zdalnie na żywo z trenerem i uczestnikami poprzez platformę Zoom

Po ukończeniu szkolenia uczestnik otrzymuje zaświadczenie ukończenia kursu.

Kurs prowadzony jest przez wielu trenerów, którzy mogą się powtarzać, w zależności od tematu zajęć.

Kurs również dedykowany jest dla osób chcących skorzystać z projektu "Małopolski pociąg do kariery"

Warunki techniczne

Wymagania sprzętowe: system operacyjny Windows 10/ lub /MacOS/ lub /Linux.

Optymalna konfiguracja sprzętowa: procesor i5+ lub podobny, 8GB+ pamięci RAM,

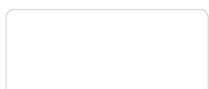
zalecany dysk SSD.

Konieczność posiadania wbudowanej kamery, słuchawek, Internetu 3Mb/s download i

3Mb/s upload.

Zajęcia będą się odbywać zdalnie na żywo z trenerem i uczestnikami na platformie zoom. Zarówno część warsztatowa, jak i projektowa.

Kontakt



Anna Mikulska



E-mail anna.mikulska@infohareacademy.com

Telefon (+48) 730 822 802