



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 2025/01/13/11051/2499153

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

📅 22.03.2025 do 16.11.2025

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

20-03-2025

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
obsługuje 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
obsługuje 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
obsługuje 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
obsługuje programowanie obiektowe (OOP) w Pythonie obsługuje 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
obsługuje framework Django obsługuje 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
obsługuje wzorce projektowe oraz refaktoring i debugowanie kodu obsługuje 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
obsługuje API: HTTP, REST, WebAPI obsługuje 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. charakteryzuje 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
obsługuje Deploy Aplikacji z wykorzystaniem AWS i Dockera obsługuje 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
obsługuje 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
obsługuje 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 osiągnięcia wspólnych celów.	Test teoretyczny z wynikiem generowanym automatycznie
uzasadnia 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	Konrad Uciechowski	22-03-2025	15:00	18:00	03:00
2 z 69 narzędzia programisty, uwzględniono 15 minut przerwy	Jarosław Majka	22-03-2025	18:00	21:00	03:00
3 z 69 podstawy programowania 1/4	Jarosław Majka	23-03-2025	15:00	18:00	03:00
4 z 69 GIT 1/3, uwzględniono 15 minut przerwy	Tomasz Jurkiewicz	23-03-2025	18:00	21:00	03:00
5 z 69 podstawy programowania 2/4, uwzględniono 15 minut przerwy	Jarosław Majka	05-04-2025	15:00	21:00	06:00
6 z 69 podstawy programowania 3/4	Jarosław Majka	06-04-2025	15:00	18:00	03:00
7 z 69 GIT 2/3, uwzględniono 15 minut przerwy	Tomasz Jurkiewicz	06-04-2025	18:00	21:00	03:00
8 z 69 podstawy programowania 4/4, uwzględniono 15 minut przerwy	Jarosław Majka	12-04-2025	15:00	21:00	06:00
9 z 69 Python - wyjątki / debugger	Jarosław Majka	13-04-2025	15:00	18:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
10 z 69 GIT 3/3, uwzględniono 15 minut przerwy	Tomasz Jurkiewicz	13-04-2025	18:00	21:00	03:00
11 z 69 Python - praca z plikami	Jarosław Majka	26-04-2025	15:00	18:00	03:00
12 z 69 OOP w Pythonie 1/4, uwzględniono 15 minut przerwy	Wiktor Piela	26-04-2025	18:00	21:00	03:00
13 z 69 Scrum, uwzględniono 15 minut przerwy	Piotr Chudzik	27-04-2025	15:00	18:00	03:00
14 z 69 OOP w Pythonie 2/4	Wiktor Piela	27-04-2025	18:00	21:00	03:00
15 z 69 OOP w Pythonie 3/4, uwzględniono 15 minut przerwy	Wiktor Piela	10-05-2025	15:00	21:00	06:00
16 z 69 OOP w Pythonie 4/4	Wiktor Piela	11-05-2025	15:00	18:00	03:00
17 z 69 Projekt Planowanie S1, uwzględniono 15 minut przerwy	Michał Adach	11-05-2025	18:00	21:00	03:00
18 z 69 pip / wirtualne środowiska	Wiktor Piela	17-05-2025	15:00	18:00	03:00
19 z 69 Linux, uwzględniono 15 minut przerwy	Mateusz Podolak	17-05-2025	18:00	21:00	03:00
20 z 69 Python VM - REPL - Python interpreter	Piotr Chudzik	18-05-2025	15:00	18:00	03:00
21 z 69 Projekt, uwzględniono 15 minut przerwy	Jarosław Majka	18-05-2025	18:00	21:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
22 z 69 Django (wstęp do Django) 1/3, uwzględniono 15 minut przerwy	Piotr Chudzik	31-05-2025	15:00	21:00	06:00
23 z 69 Python - clean code, code review	Piotr Chudzik	01-06-2025	15:00	18:00	03:00
24 z 69 Projekt, uwzględniono 15 minut przerwy	Piotr Chudzik	01-06-2025	18:00	21:00	03:00
25 z 69 SQL - bazy danych - definicja modelu danych, uwzględniono 15 minut przerwy	Anna Cielas	14-06-2025	15:00	21:00	06:00
26 z 69 wzorce projektowe	Konrad Uciechowski	15-06-2025	15:00	18:00	03:00
27 z 69 Projekt, uwzględniono 15 minut przerwy	Michał Adach	15-06-2025	18:00	21:00	03:00
28 z 69 wstęp do testowania (wstęp do PyTest)	Jan Ziobro	28-06-2025	15:00	18:00	03:00
29 z 69 HTTP, uwzględniono 15 minut przerwy	Jakub Dąbrowski	28-06-2025	18:00	21:00	03:00
30 z 69 API - WebAPI	Łukasz Pieńkowski	29-06-2025	15:00	18:00	03:00
31 z 69 Projekt, uwzględniono 15 minut przerwy	Konrad Uciechowski	29-06-2025	18:00	21:00	03:00
32 z 69 Django 2/3, uwzględniono 15 minut przerwy	Witold Bazela	12-07-2025	15:00	21:00	06:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
33 z 69 Django 3/3, uwzględniono 15 minut przerwy	Witold Bazela	13-07-2025	15:00	18:00	03:00
34 z 69 Projekt Przegląd S1	Michał Adach	13-07-2025	18:00	19:00	01:00
35 z 69 Projekt Retrospektywy	Michał Adach	13-07-2025	19:00	21:00	02:00
36 z 69 Python - refaktoring kodu	Przemysław Przywarty	26-07-2025	15:00	18:00	03:00
37 z 69 PyTest - asercje, fixtures i pluginy, uwzględniono 15 minut przerwy	Przemysław Przywarty	26-07-2025	18:00	21:00	03:00
38 z 69 unittest.mock - TDD	Przemysław Przywarty	27-07-2025	15:00	18:00	03:00
39 z 69 Projekt Planowanie S2, uwzględniono 15 minut przerwy	Michał Adach	27-07-2025	18:00	21:00	03:00
40 z 69 (FRONTEND) HTML - CSS - Bootstrap - JS 1/2, uwzględniono 15 minut przerwy	Tomasz Mariański	09-08-2025	15:00	21:00	06:00
41 z 69 (FRONTEND) HTML - CSS - Bootstrap - JS 2/2	Tomasz Mariański	10-08-2025	15:00	18:00	03:00
42 z 69 Projekt, uwzględniono 15 minut przerwy	Konrad Uciechowski	10-08-2025	18:00	21:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
43 z 69 Django's ORM - zaawansowane techniki, uwzględniono 15 minut przerwy	Wiktor Piela	23-08-2025	15:00	21:00	06:00
44 z 69 Django Admin Panel	Wiktor Piela	24-08-2025	15:00	18:00	03:00
45 z 69 Projekt, uwzględniono 15 minut przerwy	Michał Adach	24-08-2025	18:00	21:00	03:00
46 z 69 debugowanie	Jarosław Majka	06-09-2025	15:00	18:00	03:00
47 z 69 loggery, uwzględniono 15 minut przerwy	Jan Ziobro	06-09-2025	18:00	21:00	03:00
48 z 69 architektura aplikacji	Jarosław Majka	07-09-2025	15:00	18:00	03:00
49 z 69 Projekt, uwzględniono 15 minut przerwy	Michał Adach	07-09-2025	18:00	21:00	03:00
50 z 69 docker, uwzględniono 15 minut przerwy	Tomasz Mariański	13-09-2025	15:00	21:00	06:00
51 z 69 Jenkins CI, uwzględniono 15 minut przerwy	Konrad Uciechowski	14-09-2025	15:00	18:00	03:00
52 z 69 Projekt Przegląd S2	Michał Adach	14-09-2025	18:00	19:00	01:00
53 z 69 Projekt Retrospektywy	Michał Adach	14-09-2025	19:00	21:00	02:00
54 z 69 REST API, uwzględniono 15 minut przerwy	Piotr Chudzik	27-09-2025	15:00	21:00	06:00
55 z 69 Migracje	Piotr Chudzik	28-09-2025	15:00	18:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
56 z 69 Projekt Planowanie S3, uwzględniono 15 minut przerwy	Piotr Chudzik	28-09-2025	18:00	21:00	03:00
57 z 69 FAST API, uwzględniono 15 minut przerwy	Piotr Chudzik	11-10-2025	15:00	21:00	06:00
58 z 69 web scrapping	Piotr Chudzik	12-10-2025	15:00	18:00	03:00
59 z 69 Projekt, uwzględniono 15 minut przerwy	Piotr Chudzik	12-10-2025	18:00	21:00	03:00
60 z 69 asynchroniczność, uwzględniono 15 minut przerwy	Piotr Chudzik	18-10-2025	15:00	21:00	06:00
61 z 69 UX	Piotr Chudzik	19-10-2025	15:00	18:00	03:00
62 z 69 Projekt, uwzględniono 15 minut przerwy	Piotr Chudzik	19-10-2025	18:00	21:00	03:00
63 z 69 AWS, uwzględniono 15 minut przerwy	Piotr Chudzik	08-11-2025	15:00	21:00	06:00
64 z 69 UML, uwzględniono 15 minut przerwy	Piotr Chudzik	09-11-2025	15:00	18:00	03:00
65 z 69 Walidacja za pomocą testu z wynikiem generowanym automatycznie	Piotr Chudzik	09-11-2025	18:00	19:00	01:00
66 z 69 Projekt	Piotr Chudzik	09-11-2025	19:00	21:00	02:00
67 z 69 Flask, uwzględniono 15 minut przerwy	Piotr Chudzik	15-11-2025	15:00	21:00	06:00
68 z 69 zadania rekrutacyjne, uwzględniono 15 minut przerwy	Piotr Chudzik	16-11-2025	15:00	18:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
69 z 69 Przegląd końcowy Retrospektywy końcowe Zakończenie kursu	Piotr Chudzik	16-11-2025	18:00	21:00	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: 15



1 z 15

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.



2 z 15

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. Trener infoShare Academy



3 z 15



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.



4 z 15

Michał Adach

trener infoShare Academy



5 z 15

Tomasz Jurkiewicz

trener infoShare Academy. Doświadczony programista JavaScript i React z ponad 6-letnim stażem, specjalizujący się w frontendzie i backendzie. Trener w infoShare Academy, prowadzący warsztaty programistyczne od 2 lat. Pasjonat dzielenia się wiedzą, optymalizacji kodu i nowoczesnych technologii webowych. Posiada doświadczenie w pracy dla globalnych firm, takich jak Airbnb, Shell i T-Mobile, oraz w prowadzeniu zespołów deweloperskich jako Frontend Lead.



6 z 15

Wiktor 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.



7 z 15

Mateusz Podolak

Programista Python z kilkuletnim doświadczeniem w tworzeniu aplikacji internetowych. Specjalizuje się w pracy z frameworkami Django i Flask, a także posiada solidne umiejętności w zakresie projektowania REST API, integracji z bazami danych (PostgreSQL, MySQL) oraz wdrażania aplikacji w środowiskach chmurowych (AWS).

Posiada wykształcenie informatyczne zdobyte na Uniwersytecie Warmińsko-Mazurskim w Olsztynie na kierunku Inżynieria Systemów Informatycznych oraz doświadczenie w pracy zarówno w startupach, jak i w większych zespołach deweloperskich. Ceni sobie czysty, czytelny kod i dobre praktyki w zakresie testowania kodu. Zorientowany na rozwój, nieustannie poszerza swoją wiedzę w obszarze programowania oraz wdrażania aplikacji.

8 z 15



Anna Cielas

to 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



9 z 15

Bartosz Sławecki

Trener z 5-letnim stażem w języku Python z doświadczeniem zdobytym w startupach w obszarze konstrukcji, e-commerce i neurosymbolicznej sztucznej inteligencji. Współorganizator konferencji PyCon PL, Rustmeet. Kontrybutor CPython i pasjonat wolnego oprogramowania.



10 z 15

Jan Ziobro

Ukończył między innymi Politechnikę Warszawską, specjalność Data Science, Uniwersytet Jagielloński w Krakowie, a także bootcamp Data Science organizowany przez Kodołamacz (Sages). W ramach programu Erasmus studiował także na University of Exeter. Pracuje zawodowo z danymi jako Data Scientist od 2020 roku w NielsenIQ (wcześniej Nielsen). Prowadził liczne szkolenia wewnętrzne i zewnętrzne w obszarze analizy danych i programowania. Zanim rozpoczął pracę jako data scientist, pracował jako analityk i konsultant do spraw klienta biznesowego, biorąc udział w regularnych spotkaniach z klientem, gdzie prezentował przeprowadzone analizy, jak również bieżące tematy związane ze współpracą z Nielsenem. Z infoShare Academy współpracuje od 2 lat. Pasjonuje się programowaniem i rozwiązywaniem problemów. Lubi dzielić się zdobytą wiedzą.



11 z 15

Jakub Dąbrowski

Wiedzę i doświadczenie w IT zdobywał na mentoringach u doświadczonych programistów. Zawodowo zaczął zajmować się programowaniem w 2022 roku od startupu wspierającego działanie siłowni. Obecnie jako Data Engineer pracuje dla wiodącej firmy z branży Consumer Packaged Goods. Z infoShare Academy współpracuje od blisko półtora roku. Pasjonuje się technologiami inżynierii danych oraz sztuczną inteligencją.



12 z 15

Łukasz Pieńkowski

Trener Łukasz Pieńkowski ukończył Szkołę Główną Handlową w Warszawie w kierunku Finanse i Rachunkowość oraz Metody Ilościowe w Ekonomii i Systemy Informatyczne. Dodatkowo jest absolwentem Uniwersytetu Warszawskiego kierunku Data Science & Business Analytics. Pracuje zawodowo z danymi od 2021 roku. Najpierw jako analityk ryzyka, a następnie jako Data Scientist. Pracował w bankowości, ubezpieczeniach, R&D oraz konsultingu. Z infoShare Academy współpracuje od prawie 2 lat. Pracował w takich firmach jak Aviva, Samsung R&D Polska oraz Capgemini.



13 z 15

Witold Bazela

Zawodowo zaczynał jako Python developer, a obecnie jest inżynierem danych z ponad 6-letnim doświadczeniem. Ukończył Politechnikę Gdańską na kierunku Data Engineering. Jego podróż zaczęła się od programowania w języku C++ w wieku 17 lat, ewoluując w pasję do tworzenia potoków danych i skalowalnych architektur. Uważa, że praca w wielu branżach dała mu perspektywę wykorzystania danych jako strategicznego zasobu. Witold posiada dużą wiedzę i potrafi ją przekazywać kursantom, ponieważ ma prawie 3-letnie zaplecze szkoleniowe jako trener Pythona.



14 z 15

Przemysław Przywarty

Specjalista Python, trener infoShare Academy



15 z 15

Tomasz Mariański

Inżynier oprogramowania pracujący jako Full-stack developer. Na co dzień projektuje oraz wdraża rozwiązania w aplikacjach cloud-native z użyciem języków Python oraz TypeScript. Karierę zaczynał od analizy danych w badaniach aerodynamicznych przy użyciu technologii Machine Learning oraz AI wykorzystując do tego Pythona. Posiada również kilkuletnie doświadczenie jako wykładowca programowania na publicznym uniwersytecie oraz w prywatnych akademiach.

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).

informacyjnie 31.05.2025 Django (wstęp do Django) 1/3 i 3/3 poprowadził Wiktor Piel, a 1.06. 2025 Python - clean code, code review zajęcia poprowadziła trenerka Anna Cielas i część projektową Jarosław Majka i Michał Adach.

Informacje dodatkowe

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

Test końcowy odbędzie się 9.11.2025 r. jest on zaznaczony w harmonogramie zajęć. Będzie on przeprowadzony w formie testu teoretycznego. Walidację będzie przeprowadzać Grzegorz Labuda.

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@infoshareacademy.com

Telefon (+48) 730 822 802