



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

Numer usługi 2024/09/17/11051/2312965

8 900,00 PLN brutto

7 235,77 PLN netto

37,08 PLN brutto/h

30,15 PLN netto/h

INFOSHARE

ACADEMY SPÓŁKA

Z OGRANICZONĄ

ODPOWIEDZIALNOŚĆ

CIA



📍 zdalna w czasie rzeczywistym

👤 Usługa szkoleniowa

🕒 240 h

📅 30.11.2024 do 20.07.2025

Informacje podstawowe

| | |
|--|--|
| Kategoria | Informatyka i telekomunikacja / Programowanie |
| Sposób dofinansowania | wsparcie dla osób indywidualnych wsparcie dla pracodawców i ich pracowników |
| Grupa docelowa usługi | <p>Kurs Python + Django + AI – weekendowy jest dla każdego entuzjasty technologicznego, który chce zostać programistą.</p> <p>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.</p> |
| Minimalna liczba uczestników | 1 |
| Maksymalna liczba uczestników | 18 |
| Data zakończenia rekrutacji | 27-11-2024 |
| 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 |
|--|---|--------------------------------------|
| pozna narzędzia Linux i Terminal | Zaawansowana nawigacja: nauczy się używać zaawansowanych poleceń i płynnie nawigować po systemie plików. Automatyzacja zadań: zaznajomi się z tworzeniem skryptów do automatyzacji powtarzalnych zadań. Konfiguracja środowiska dokona personalizacji i konfiguracji swojego środowiska pracy zgodnie z indywidualnymi preferencjami lub wymaganiami projektu – tak jak w prawdziwej pracy! | Obserwacja w warunkach rzeczywistych |
| pozna system Kontroli Wersji Git | Rozgałęzienie i scalenie: nauczy się zaawansowanych operacji na gałęziach projektu. Zarządzanie konfliktami: pozna narzędzia oraz techniki potrzebne do rozwiązywania konfliktów wynikających z równoczesnej edycji kodu przez różnych programistów. Praca z GitHub: wdroży się w zasady publikowania i współpracy nad projektem na platformie GitHub. | Obserwacja w warunkach rzeczywistych |
| nauczy się podstawy Programowania Python | Struktury danych: pozna zastosowanie podstawowych struktur danych, takich jak listy, krotki, słowniki oraz dowie się, jak z nimi efektywnie pracować. Obsługa plików: zaznajomi się z metodami manipulacji plikami, wczytywania i zapisywania danych. Podstawy algorytmów. Na zajęciach pozna sposoby implementacji prostych algorytmów do rozwiązywania programistycznych problemów. | Test teoretyczny |

| Efekty uczenia się | Kryteria weryfikacji | Metoda walidacji |
|---|--|---|
| <p>nauczy się programowanie obiektowe (OOP) w Pythonie</p> <p>nauczy się pracy z wirtualnymi środowiskami</p> <p>pozna framework Django</p> | <p>tworzenie klas i obiektów: pozna podstawy projektowania i implementacji klas oraz ich instancji. Dziedziczenie i polimorfizm: na zajęciach dowie się, jak wykorzystywać dziedziczenie do współdzielenia kodu oraz stosować polimorfizm. Enkapsulacja: ukrywanie implementacji i ograniczanie dostępu do danych zapewnia bezpieczeństwo i integralność kodu.</p> <p>Instalacja pakietów: pozna techniki skutecznego korzystania z narzędzia pip do instalacji pakietów Python. Tworzenie wirtualnych środowisk: nauczy się, jak tworzyć i zarządzać wirtualnymi środowiskami za pomocą narzędzi takich jak virtualenv. Zarządzanie zależnościami projektu: wymaganymi wersjami pakietów dla projektu jest kluczowe w przypadku pracy z wieloma bibliotekami Pythona.</p> <p>Tworzenie aplikacji w Django: zapozna się z zasadami projektowania i rozwijania prostych aplikacji webowych przy użyciu Django, Obsługa bazy danych: korzystanie z ORM do komunikacji z bazą danych. Administracja aplikacją: konfiguracja i korzystanie z wbudowanego Admin Panelu w celu zarządzania danymi pozwoli patrzeć globalnie na cały projekt.</p> | <p>Test teoretyczny</p> <p>Obserwacja w warunkach rzeczywistych</p> <p>Test teoretyczny</p> |
| <p>pozna bazy danych i język SQL</p> | <p>Podstawy zapytań w języku SQL: pozna zasady tworzenia prostych zapytań do baz danych opartych o: SELECT, INSERT, UPDATE, DELETE. Operacje na tabelach: zapozna się z dobrymi praktykami efektywnego projektowania, modyfikowania i usuwanie tabel, a także definiowanie kluczy głównych i obcych. Zaawansowane zapytania: dowie się, jakie są zaawansowane funkcje języka SQL, takie jak JOIN, GROUP BY, HAVING.</p> | <p>Test teoretyczny</p> |

| Efekty uczenia się | Kryteria weryfikacji | Metoda walidacji |
|---|---|---|
| <p>pozna wzorce projektowe oraz refaktoring i debugowanie kodu</p> <p>pozna proces testowania</p> | <p>Wzorce projektowe: pozna i wdroży w projekcie zespołowym, zastosowanie popularnych wzorców projektowych, takich jak Singleton, Factory, Observer. Clean Code: nauczy się jak tworzyć czytelny, zorganizowany i efektywny kod w języku Python zgodnie z zasadami Clean Code.</p> <p>Refaktoring: nabędzie umiejętności poprawy istniejącego kodu bez zmiany jego funkcjonalności.</p> <p>PyTest i asercje: pozna zasady tworzenia testów przy użyciu frameworka PyTest oraz korzystania z asercji do sprawdzania poprawności ich wyników.</p> <p>Unittest.mock: izolowanie kodu testowanego poprzez mockowanie zależności.</p> <p>Testy funkcjonalne: tworzenie testów, które sprawdzają całe funkcjonalności systemu, jest ostatnim ogniwem na drodze do kompleksowej weryfikacji działania aplikacji.</p> | <p>Obserwacja w warunkach symulowanych</p> <p>Obserwacja w warunkach symulowanych</p> |
| <p>pozna API: HTTP, REST, WebAPI</p> <p>pozna Frontend: HTML, CSS, Bootstrap</p> | <p>Protokół HTTP: zasady działania protokołu HTTP w tym metod żądań oraz nagłówków. Dzięki zapoznaniu się z tymi podstawami rozpocznie przygodę z tematyką RESTową.</p> <p>Architektura REST: projektowanie i implementacja interfejsów RESTful.</p> <p>Obsługa autoryzacji: zabezpieczanie dostępu do swojego API za pomocą różnych mechanizmów autoryzacji.</p> <p>HTML: jego znajomość pozwala na tworzenie struktury strony internetowej, korzystanie z różnych tagów i atrybutów.</p> <p>CSS: dzięki biegłości w posługiwaniu się CSS może ostylewać stronę, nadawać wygląd poszczególnym jej elementom, a także pracować z selektorami.</p> <p>Bootstrap: szybkie projektowanie responsywnych interfejsów przy użyciu gotowych komponentów.</p> | <p>Obserwacja w warunkach symulowanych</p> <p>Test teoretyczny</p> |

| Efekty uczenia się | Kryteria weryfikacji | Metoda walidacji |
|--|--|--|
| <p>pozna Deploy Aplikacji z wykorzystaniem AWS i Dockera</p> | <p>Docker: tworzenie i zarządzanie kontenerami. Pozna niezbędne elementy związane z tym narzędziem. AWS: wdrażanie aplikacji na platformie AWS – na zajęciach zapozna się z głównymi rozwiązaniami chmury Amazona. Jenkins CI: konfiguracja procesu Continuous Integration i Continuous Deployment wpływa na automatyzację procesów.</p> | <p>Test teoretyczny</p> |
| <p>pozna Loggery, Asynchroniczność, Web Scraping</p> | <p>Loggery w Pythonie: i korzystanie z modułu logging w celu efektywnego logowania zdarzeń w aplikacji. Programowanie asynchroniczne: wykorzystanie mechanizmów asynchronicznych w Pythonie. Web Scraping z BeautifulSoup i Requests: pozyskiwanie danych z witryn internetowych, przetwarzanie i analiza zebranych informacji są kluczowe w procesie analizy konkurencji.</p> | <p>Test teoretyczny</p> |
| <p>pozna Flask</p> | <p>Podstawy frameworka Flask: zapozna się z podstawami projektowania tras, obsługi żądań i odpowiedzi HTTP. Szablony Jinja2: korzystanie z szablonów do dynamicznego generowania treści HTML jest niezbędne przy efektywnej pracy z Flaskiem. Rozwijanie aplikacji Flask: dodawanie funkcjonalności, obsługa formularzy czy korzystanie z rozszerzeń, to niektóre z tematów, które wchodzą w skład podstaw Flaska.</p> | <p>Test teoretyczny</p> |
| <p>pozna UX oraz Zadania Rekrutacyjne</p> | <p>Projektowanie interfejsów: nauczy się budować interfejsy tak, aby w najlepszy sposób wykorzystać zasady designu i ergonomii Testy użyteczności: przeprowadzanie testów użyteczności, które sprawdzają, jak dobrze użytkownicy radzą sobie z interfejsem, jest kluczem do kompleksowej weryfikacji działania aplikacji. Efektywna prezentacja.</p> | <p>Obserwacja w warunkach symulowanych</p> |
| <p>nauczy się aktywnego słuchania</p> | <p>zdolność uważnego słuchania, aby zrozumieć perspektywę i potrzeby innych uczestników kursu</p> | <p>Obserwacja w warunkach symulowanych</p> |

| Efekty uczenia się | Kryteria weryfikacji | Metoda walidacji |
|-----------------------------------|--|-------------------------------------|
| pozna czym jest empatia | nabędzie umiejętności zrozumienia i odczuwania emocji innych osób w projekcie zespołowym | Obserwacja w warunkach symulowanych |
| pozna metodykę Scrum | nabędzie umiejętności współpracy pracy w grupie, dzielenia się zadaniami i osiągnięcia wspólnych celów. | Obserwacja w warunkach symulowanych |
| nauczy się otwartości na feedback | uczestnik nauczy się gotowości do przyjmowania konstruktywnej krytyki i wykorzystywania jej do własnego rozwoju. | Obserwacja w warunkach symulowanych |

Kwalifikacje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

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

Tak, dokument zawiera efekty uczenia się, do których uzyskania uczestnik przygotowywał się w procesie uczenia się.

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

Tak, dokument potwierdza, że walidacja została przeprowadzona zgodnie z ustalonymi kryteriami weryfikacji zawartymi w efektach uczenia się.

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

Tak, dokument potwierdza zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji.

Program

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.

15. AWS

Podstawy pracy ze środowiskiem chmurowych Amazonu.

Harmonogram

Liczba przedmiotów/zajęć: 69

| Przedmiot / temat zajęć | Prowadzący | Data realizacji zajęć | Godzina rozpoczęcia | Godzina zakończenia | Liczba godzin |
|--|------------|-----------------------|---------------------|---------------------|---------------|
| 1 z 69 Rozpoczęcie kursu Organizacja | - | 30-11-2024 | 08:00 | 11:00 | 03:00 |
| 2 z 69 narzędzia programisty | - | 30-11-2024 | 11:00 | 14:00 | 03:00 |
| 3 z 69 podstawy programowania 1/4 | - | 01-12-2024 | 08:00 | 11:00 | 03:00 |
| 4 z 69 GIT 1/3 | - | 01-12-2024 | 11:00 | 14:00 | 03:00 |
| 5 z 69 podstawy programowania 2/4 | - | 14-12-2024 | 08:00 | 14:00 | 06:00 |
| 6 z 69 podstawy programowania 3/4 | - | 15-12-2024 | 08:00 | 11:00 | 03:00 |
| 7 z 69 GIT 2/3 | - | 15-12-2024 | 11:00 | 14:00 | 03:00 |
| 8 z 69 podstawy programowania 4/4 | - | 11-01-2025 | 08:00 | 14:00 | 06:00 |
| 9 z 69 Python - wyjątki / debugger | - | 12-01-2025 | 08:00 | 11:00 | 03:00 |
| 10 z 69 GIT 3/3 | - | 12-01-2025 | 11:00 | 14:00 | 03:00 |
| 11 z 69 Python - praca z plikami | - | 18-01-2025 | 08:00 | 11:00 | 03:00 |
| 12 z 69 OOP w Pythonie 1/4 | - | 18-01-2025 | 11:00 | 14:00 | 03:00 |
| 13 z 69 OOP w Pythonie 2/4 | - | 19-01-2025 | 08:00 | 11:00 | 03:00 |
| 14 z 69 Scrum | - | 19-01-2025 | 11:00 | 14:00 | 03:00 |
| 15 z 69 OOP w Pythonie 3/4 | - | 01-02-2025 | 08:00 | 14:00 | 06:00 |

| Przedmiot / temat zajęć | Prowadzący | Data realizacji zajęć | Godzina rozpoczęcia | Godzina zakończenia | Liczba godzin |
|---|------------|-----------------------|---------------------|---------------------|---------------|
| 16 z 69 OOP w Pythonie 4/4 | - | 02-02-2025 | 08:00 | 11:00 | 03:00 |
| 17 z 69 Projekt Planowanie S1 | - | 02-02-2025 | 11:00 | 14:00 | 03:00 |
| 18 z 69 pip / wirtualne środowiska | - | 08-02-2025 | 08:00 | 11:00 | 03:00 |
| 19 z 69 Linux | - | 08-02-2025 | 11:00 | 14:00 | 03:00 |
| 20 z 69 Python VM - REPL - Python interpreter | - | 09-02-2025 | 08:00 | 11:00 | 03:00 |
| 21 z 69 Projekt | - | 09-02-2025 | 11:00 | 14:00 | 03:00 |
| 22 z 69 Django (wstęp do Django) 1/3 | - | 22-02-2025 | 08:00 | 14:00 | 06:00 |
| 23 z 69 Python - clean code, code review | - | 23-02-2025 | 08:00 | 11:00 | 03:00 |
| 24 z 69 Projekt | - | 23-02-2025 | 11:00 | 14:00 | 03:00 |
| 25 z 69 SQL - bazy danych - definicja modelu danych | - | 01-03-2025 | 08:00 | 14:00 | 06:00 |
| 26 z 69 wzorce projektowe | - | 02-03-2025 | 08:00 | 11:00 | 03:00 |
| 27 z 69 Projekt | - | 02-03-2025 | 11:00 | 14:00 | 03:00 |
| 28 z 69 wstęp do testowania (wstęp do PyTest) | - | 15-03-2025 | 08:00 | 11:00 | 03:00 |
| 29 z 69 HTTP | - | 15-03-2025 | 11:00 | 14:00 | 03:00 |
| 30 z 69 API - WebAPI | - | 16-03-2025 | 08:00 | 11:00 | 03:00 |

| Przedmiot / temat zajęć | Prowadzący | Data realizacji zajęć | Godzina rozpoczęcia | Godzina zakończenia | Liczba godzin |
|--|------------|-----------------------|---------------------|---------------------|---------------|
| 31 z 69 Projekt | - | 16-03-2025 | 11:00 | 14:00 | 03:00 |
| 32 z 69 Django 2/3 | - | 22-03-2025 | 08:00 | 14:00 | 06:00 |
| 33 z 69 Django 3/3 | - | 23-03-2025 | 08:00 | 11:00 | 03:00 |
| 34 z 69 Projekt Przegląd S1 | - | 23-03-2025 | 11:00 | 12:00 | 01:00 |
| 35 z 69 Projekt Retrospektywy | - | 23-03-2025 | 12:00 | 14:00 | 02:00 |
| 36 z 69 Python - refaktoring kodu | - | 05-04-2025 | 08:00 | 11:00 | 03:00 |
| 37 z 69 PyTest - asercje, fixtures i pluginy | - | 05-04-2025 | 11:00 | 14:00 | 03:00 |
| 38 z 69 unittest.mock - TDD | - | 06-04-2025 | 08:00 | 11:00 | 03:00 |
| 39 z 69 Projekt Planowanie S2 | - | 06-04-2025 | 11:00 | 14:00 | 03:00 |
| 40 z 69 (FRONTEND) HTML - CSS - Bootstrap - JS 1/2 | - | 12-04-2025 | 08:00 | 14:00 | 06:00 |
| 41 z 69 (FRONTEND) HTML - CSS - Bootstrap - JS 2/2 | - | 13-04-2025 | 08:00 | 11:00 | 03:00 |
| 42 z 69 Projekt | - | 13-04-2025 | 11:00 | 14:00 | 03:00 |
| 43 z 69 Django's ORM - zaawansowane techniki | - | 26-04-2025 | 08:00 | 14:00 | 06:00 |
| 44 z 69 Django Admin Panel | - | 27-04-2025 | 08:00 | 11:00 | 03:00 |

| Przedmiot / temat zajęć | Prowadzący | Data realizacji zajęć | Godzina rozpoczęcia | Godzina zakończenia | Liczba godzin |
|---------------------------------|------------|-----------------------|---------------------|---------------------|---------------|
| 45 z 69 Projekt | - | 27-04-2025 | 11:00 | 14:00 | 03:00 |
| 46 z 69 debugowanie | - | 10-05-2025 | 08:00 | 11:00 | 03:00 |
| 47 z 69 loggery | - | 10-05-2025 | 11:00 | 14:00 | 03:00 |
| 48 z 69 architektura aplikacji | - | 11-05-2025 | 08:00 | 11:00 | 03:00 |
| 49 z 69 Projekt | - | 11-05-2025 | 11:00 | 14:00 | 03:00 |
| 50 z 69 docker | - | 17-05-2025 | 08:00 | 14:00 | 06:00 |
| 51 z 69 Jenkins CI | - | 18-05-2025 | 08:00 | 11:00 | 03:00 |
| 52 z 69 Projekt Przegląd S2 | - | 18-05-2025 | 11:00 | 12:00 | 01:00 |
| 53 z 69 Projekt Retrospektywy | - | 18-05-2025 | 12:00 | 14:00 | 02:00 |
| 54 z 69 REST API | - | 31-05-2025 | 08:00 | 14:00 | 06:00 |
| 55 z 69 Migracje | - | 01-06-2025 | 08:00 | 11:00 | 03:00 |
| 56 z 69 Projekt Planowanie S3 | - | 01-06-2025 | 11:00 | 14:00 | 03:00 |
| 57 z 69 FAST API | - | 14-06-2025 | 08:00 | 14:00 | 06:00 |
| 58 z 69 web scrapping | - | 15-06-2025 | 08:00 | 11:00 | 03:00 |
| 59 z 69 Projekt | - | 15-06-2025 | 11:00 | 14:00 | 03:00 |
| 60 z 69 asynchroniczność | - | 28-06-2025 | 08:00 | 14:00 | 06:00 |
| 61 z 69 UX | - | 29-06-2025 | 08:00 | 11:00 | 03:00 |
| 62 z 69 Projekt | - | 29-06-2025 | 11:00 | 14:00 | 03:00 |
| 63 z 69 AWS | - | 05-07-2025 | 08:00 | 14:00 | 06:00 |

| Przedmiot / temat zajęć | Prowadzący | Data realizacji zajęć | Godzina rozpoczęcia | Godzina zakończenia | Liczba godzin |
|--|------------|-----------------------|---------------------|---------------------|---------------|
| 64 z 69 UML | - | 06-07-2025 | 08:00 | 11:00 | 03:00 |
| 65 z 69 Projekt Test końcowy | - | 06-07-2025 | 11:00 | 12:00 | 01:00 |
| 66 z 69 Projekt | - | 06-07-2025 | 12:00 | 14:00 | 02:00 |
| 67 z 69 Flask | - | 19-07-2025 | 08:00 | 14:00 | 06:00 |
| 68 z 69 zadania rekrutacyjne | - | 20-07-2025 | 08:00 | 11:00 | 03:00 |
| 69 z 69 Przegląd końcowy Retrospektywy końcowe Zakończenie kursu | - | 20-07-2025 | 11:00 | 14:00 | 03:00 |

Cennik

Cennik

| Rodzaj ceny | Cena |
|----------------------|--------------|
| Koszt usługi brutto | 8 900,00 PLN |
| Koszt usługi netto | 7 235,77 PLN |
| Koszt godziny brutto | 37,08 PLN |
| Koszt godziny netto | 30,15 PLN |

Prowadzący

Liczba prowadzących: 0

Brak wyników.

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.

Warunki uczestnictwa

- Wymagania sprzętowe: system operacyjny Linux lub macOS.

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.

Uczestnik powinien posiadać umiejętności analitycznego myślenia oraz znajomością języka angielskiego umożliwiającą czytanie oraz rozumienie dokumentacji.

- żeby mieć 80% obecności, aktywny udział w części projektowej, wykonywanie zadań domowych oraz zaliczenie egzaminu końcowego na ok 70%.

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.

szkolnie będzie realizowane zdalnie 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.

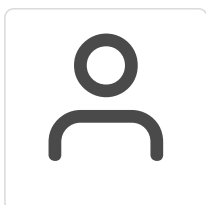
Warunki techniczne

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.

Kontakt



Agnieszka Frackiewicz

E-mail agnieszka.frackiewicz@infoshareacademy.com

Telefon (+48) 530 100 686