



Notebook Master
Sp. z o.o.

★★★★★ 4,8 / 5

188 ocen

Python / Etap II / Python dla średnio zaawansowanych - szkolenie

Numer usługi 2025/08/20/158529/2949684

📍 zdalna w czasie rzeczywistym

🏠 Usługa szkoleniowa

🕒 32 h

📅 01.12.2025 do 04.12.2025

3 936,00 PLN brutto

3 200,00 PLN netto

123,00 PLN brutto/h

100,00 PLN netto/h

Informacje podstawowe

Kategoria

Informatyka i telekomunikacja / Programowanie

Identyfikatory projektów

Małopolski Pociąg do kariery, Zachodniopomorskie Bony Szkoleniowe, Regionalny Fundusz Szkoleniowy II, Kierunek - Rozwój, FELB.06.08-IZ.00-0083/24

Grupa docelowa usługi

Szkolenie skierowane jest zarówno osób fizycznych, jak i do przedsiębiorców i ich pracowników pracujących w branży IT, chcących nabyć umiejętności programowania w języku Python na poziomie średnio zaawansowanym i wykorzystać ją w ramach prowadzonej działalności gospodarczej i etatu.

Usługa rozwojowa adresowa również dla Uczestników projektów, m.in.:

- Małopolski pociąg do kariery
- Zachodniopomorskie Bony Szkoleniowe
- Kierunek – Rozwój
- Regionalny Fundusz Szkoleniowy II
- Lubuskie Bony Rozwojowe
- Usługi rozwojowe dla mieszkańców województwa lubuskiego

Minimalna liczba uczestników

1

Maksymalna liczba uczestników

8

Data zakończenia rekrutacji

30-11-2025

Forma prowadzenia usługi

zdalna w czasie rzeczywistym

Liczba godzin usługi

32

Cel

Cel edukacyjny

Usługa "Python / Etap II / Python dla średnio zaawansowanych", prowadzi do nabycia specjalistycznych kompetencji w obszarze tematycznym szkolenia i przygotowuje do samodzielnego i prawidłowego wykonywania obowiązków w zakresie programowania na poziomie średnio zaawansowanym z wykorzystaniem języka Python.

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Charakteryzuje typy zmiennych, funkcji i argumentów funkcji.	Rozpoznaje typ zmiennej w podanym fragmencie kodu.	Test teoretyczny
	Wskazuje przeznaczenie konstruktorów z modułu typing.	Test teoretyczny
Wykorzystuje biblioteki 'typing' i 'mypy'.	Stosuje adnotacje typów z biblioteki 'typing' w kodzie Pythona.	Test teoretyczny
	Rozpoznaje narzędzie mypy jako program do statycznej analizy typów.	Test teoretyczny
Tworzy czytelne i dokładne docstringi opisujące cel funkcji, klasy i modułów. Kryteria weryfikacji:	Wskazuje poprawne zasady tworzenia docstringów.	Test teoretyczny
Pracuje z docstringami typu reStructuredText i potrafi generować dokumentację na podstawie docstringu za pomocą narzędzi takich jak Sphinx.	Wskazuje możliwość formatowania docstringów w konwencji reStructuredText.	Test teoretyczny
	Dobiera narzędzia do generowania dokumentacji na podstawie docstringów.	Test teoretyczny
Przestrzega podstawowych zasad stylu zgodnie z PEP-8, zarządzając nazewnictwem, długością linii, wcięciami, pustymi liniami i odstępami.	Określa standardy języka Python dotyczący formatowania kodu.	Test teoretyczny
Korzysta z narzędzi do autoformatowania, takich jak Black, Blue, autopep8, yapf, i integruje je z edytorami kodu.	Wskazuje i integruje narzędzia automatycznej analizy kodu wśród różnych narzędzi programistycznych.	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Obsługuje wyjątki, stosując konstrukcje try, except, finally, oraz raise. Kryteria weryfikacji:	Stosuje konstrukcje try, except, finally do obsługi wyjątków w kodzie.	Test teoretyczny
Pracuje z testami jednostkowymi, wykorzystując moduły doctest i pytest tworzy testy, testuje przypadki graniczne i izolację.	Wskazuje przeznaczenie testów jednostkowych w procesie programowania.	Test teoretyczny
Stosuje koncepcje programowania obiektowego, tworzy klasy i instancje, inicjalizuje obiekty, zarządza atrybutami klasowymi i instancyjnymi oraz korzysta z getterów i setterów.	Rozpoznaje metodę konstruktora klasy w Pythonie.	Test teoretyczny
Pracuje z metodami instancji, statycznymi i klasowymi, stosuje dekoratory, oraz tworzy konstruktory i destruktory, zarówno z domyślnymi jak i z parametrami.	Rozpoznaje cechy metod statycznych.	Test teoretyczny
	Omawia proces tworzenia konstruktorów i destruktorków.	Test teoretyczny
Korzysta z metod specjalnych, włączając str, repr, len, getitem, setitem, i interakcję z obiektami.	Rozpoznaje metody specjalne wpływające na prezentację obiektu.	Test teoretyczny
Przeładowuje operatory, w tym arytmetyczne, porównania, logiczne, oraz rozumie ich zastosowania i ograniczenia.	Wskazuje sposób przeładowania operatorów w Pythonie.	Test teoretyczny
	Rozumie zastosowania i ograniczenia związane z przeładowywaniem operatorów.	Test teoretyczny
Dokonuje konwersji między typami oraz rzutowania typów. Wykorzystuje metody dla stringów, list, zbiorów i słowników.	Wykonuje konwersje między różnymi typami danych w sposób bezpieczny i efektywny.	Test teoretyczny
	Stosuje rzutowanie typów w przypadkach, gdy jest to konieczne lub uzasadnione.	Test teoretyczny
	Wykonuje konwersje między różnymi typami danych w sposób bezpieczny i efektywny.	Test teoretyczny
	Stosuje rzutowanie typów w przypadkach, gdy jest to konieczne lub uzasadnione.	Test teoretyczny
Pracuje z metodami krotek oraz korzysta z operacji na zbiorach i słownikach.	Stosuje operacje na zbiorach i słownikach.	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Dodaje, usuwa i aktualizuje elementy w słownikach, a także stosuje metodę get.	Określa działanie metody słownika get().	Test teoretyczny

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

Szkolenie skierowane jest do przedsiębiorców i ich pracowników pracujących w branży IT oraz osób, chcących nabyć umiejętności programowania w języku Python i wykorzystać ją w ramach prowadzonej działalności gospodarczej i etatu.

Ramowy plan kształcenia:

I. Jakość kodu. (teoria+praktyka)

1. Wskazówki typów.
2. a: Typy zmiennej, funkcji, argumentów funkcji.
3. b: Biblioteki typing i mypy.
4. c: Zaawansowane wskazówki typów
5. Docstring.
6. a: Opisywanie celu funkcji, klas i modułów.
7. b: Styl docstring (np. Google, NumPy, reStructuredText).
8. c: Dokumentacja na podstawie docstringu (np. Sphinx).
9. PEP-8.
10. a: Podstawowe zasady stylu.
11. b: Nazewnictwo, długość linii, wcięcia.
12. c: Puste linie i spacje.
13. Autoformatowanie.
14. a: Narzędzia: Black, autopep8, yapf.
15. b: Integracja z edytorami kodu.
16. Linting.
17. a: Narzędzia: pylint, flake8, pyright.
18. b: Analiza statyczna kodu.
19. c: Detekcja potencjalnych błędów i niedoskonałości.

II. Testowanie. (teoria+praktyka)

1. Wyjątki.
2. a: Obsługa błędów: try, except, finally, raise.
3. b: Własne wyjątki.
4. Testy jednostkowe.
5. a: Moduły (pytest, robot, unittest).
6. b: Tworzenie testów.
7. c: Testowanie przypadków granicznych.
8. d: Mockowanie i izolacja.
9. Logi.
10. a: Funkcja print.
11. b: Logowanie do plików.
12. c: Moduł logging.
13. d: Poziomy logowania: DEBUG, INFO, WARNING, ERROR, CRITICAL.
14. e: Formatowanie logów.

III. Programowanie obiektowe. (teoria+praktyka)

1. Klasa i obiekt.
2. a: Tworzenie klas i instancji.
3. b: Inicjalizacja obiektu.
4. Atrybuty.
5. a: Atrybuty klasowe i instancji.
6. b: Gettery i settery.
7. c: Atrybuty prywatne i chronione.
8. Metody.
9. a: Metody instancji, statyczne, klasowe.
10. b: Dekoratory.
11. Konstruktor i destruktorktor.
12. a: init i del.
13. b: Tworzenie obiektów z parametrami.
14. Metody dunder.

15. a: str, repr, len, getitem, setitem.
16. b: Interakcja z obiektami.
17. Przeładowanie operatorów.
18. a: Operatory arytmetyczne, porównania, logiczne.
19. b: Zastosowania i ograniczenia.
20. Konwersje.
21. a: Konwersje między typami.
22. b: Rzutowanie typów.

IV. Kontenery i stringi. (teoria+praktyka)

1. Metody stringów.
2. a: Użycie metod: split, join, replace, find, count, etc..
3. b: Funkcje związane z wielkością liter: upper, lower, capitalize, title, etc.
4. Metody krotek.
5. a: Metody count, index.
6. Metody list.
7. a: Dodawanie i usuwanie elementów.
8. b: Sortowanie i reorganizacja.
9. Metody i operatory zbiorów.
10. a: Dodawanie i usuwanie elementów.
11. b: Operacje: union, intersection, difference, issubset, issuperset, etc.
12. Metody słowników.
13. a: Dodawanie, usuwanie, aktualizacja elementów.
14. b: Metoda get.

Szkolenie trwa 32 godziny dydaktyczne (przerwy nie wliczają się do czasu trwania usługi rozwojowej) i realizowane jest w kameralnych grupach, maksymalnie 8-osobowych. Każdy uczestnik realizujący szkolenie w formie zdalnej w czasie rzeczywistym ma możliwość otrzymania od nas (za pośrednictwem kuriera) wyposażenie stanowiska szkoleniowego (po ukończeniu szkolenia sprzęt zostaje odebrany przez kuriera).

Zawarto umowę z WUP w Szczecinie na świadczenie usług rozwojowych z wykorzystaniem elektronicznych bonów szkoleniowych.

Zawarto umowę z WUP w Toruniu w ramach Projektu Kierunek – Rozwój

Zaakceptowano Regulamin "Małopolskiego Pociągu do Kariery" dla instytucji szkoleniowych.

Wymagana jest frekwencja na poziomie 80%.

Obecność na zajęciach potwierdza raport z platformy Zoom.

Szkolenie prowadzone jest z wykorzystaniem metod nauczania aktywizujących uczestników: dyskusja w grupie, burza mózgów, ćwiczenia.

Harmonogram

Liczba przedmiotów/zajęć: 29

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
1 z 29 Jakość kodu. Wskazówki typów. Typy zmiennej, funkcji, argumentów funkcji. Biblioteki typing i mypy. Zaawansowane wskazówki typów. (Wykłady, dyskusja, ćwiczenia, testy.)	Karol Kołodziejczyk	01-12-2025	08:45	10:15	01:30
2 z 29 Przerwa.	Karol Kołodziejczyk	01-12-2025	10:15	10:30	00:15
3 z 29 Docstring. Opisywanie celu funkcji, klas i modułów. Styl docstring (np. Google, NumPy, reStructuredText) . Dokumentacja na podstawie docstringu (np. Sphinx). (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	01-12-2025	10:30	12:00	01:30
4 z 29 Przerwa.	Karol Kołodziejczyk	01-12-2025	12:00	12:45	00:45
5 z 29 PEP-8. Podstawowe zasady stylu. Nazewnictwo, długość linii, wcięcia. Puste linie i spacje. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	01-12-2025	12:45	14:15	01:30
6 z 29 Przerwa.	Karol Kołodziejczyk	01-12-2025	14:15	14:30	00:15

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
7 z 29 Autoformatowani e. Narzędzia: Black, autopep8, yapf. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	01-12-2025	14:30	16:00	01:30
8 z 29 Linting. Narzędzia: pylint, flake8, pyright. Analiza statyczna kodu. Detekcja potencjalnych błędów i niedoskonałości. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	02-12-2025	08:45	10:15	01:30
9 z 29 Przerwa.	Karol Kołodziejczyk	02-12-2025	10:15	10:30	00:15
10 z 29 Testowanie. Wyjątki. Obsługa błędów: try, except, finally, raise. Własne wyjątki. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	02-12-2025	10:30	12:00	01:30
11 z 29 Przerwa.	Karol Kołodziejczyk	02-12-2025	12:00	12:45	00:45
12 z 29 Testy jednostkowe. Moduły (pytest, robot, unittest). Tworzenie testów. Testowanie przypadków granicznych. Mockowanie i izolacja. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	02-12-2025	12:45	14:15	01:30
13 z 29 Przerwa.	Karol Kołodziejczyk	02-12-2025	14:15	14:30	00:15

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
14 z 29 Logi. Funkcja print. Logowanie do plików. Moduł logging. Poziomy logowania: DEBUG, INFO, WARNING, ERROR, CRITICAL. Formatowanie logów. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	02-12-2025	14:30	16:00	01:30
15 z 29 Programowanie obiektowe. Klasa i obiekt. Tworzenie klas i instancji. Inicjalizacja obiektu. (Wyklady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	03-12-2025	08:45	10:15	01:30
16 z 29 Przerwa.	Karol Kołodziejczyk	03-12-2025	10:15	10:30	00:15
17 z 29 Atrybuty. Atrybuty klasowe i instancji. Gettery i settery. Atrybuty prywatne i chronione. (Wyklady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	03-12-2025	10:30	12:00	01:30
18 z 29 Przerwa.	Karol Kołodziejczyk	03-12-2025	12:00	12:45	00:45

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
19 z 29 Metody. Metody instancji, statyczne, klasowe. Dekoratory. Konstruktor i destrukt. init i del. Tworzenie obiektów z parametrami. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	03-12-2025	12:45	14:15	01:30
20 z 29 Przerwa.	Karol Kołodziejczyk	03-12-2025	14:15	14:30	00:15
21 z 29 Metody dunder. str, repr, len, getitem, setitem. Interakcja z obiektami. Przeładowanie operatorów. Operatory arytmetyczne, porównania, logiczne. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	03-12-2025	14:30	16:00	01:30
22 z 29 Konwersje. Konwersje między typami. Rzutowanie typów. Kontenery i stringi. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	04-12-2025	08:45	10:15	01:30
23 z 29 Przerwa.	Karol Kołodziejczyk	04-12-2025	10:15	10:30	00:15

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
24 z 29 Metody stringów. Użycie metod: split, join, replace, find, count, etc. Funkcje związane z wielkością liter: upper, lower, capitalize, title, etc. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	04-12-2025	10:30	12:00	01:30
25 z 29 Przerwa.	Karol Kołodziejczyk	04-12-2025	12:00	12:45	00:45
26 z 29 Metody krotek. Metody count, index. Metody list. Dodawanie i usuwanie elementów. Sortowanie i reorganizacja. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	04-12-2025	12:45	14:15	01:30
27 z 29 Przerwa.	Karol Kołodziejczyk	04-12-2025	14:15	14:30	00:15
28 z 29 Metody i operatory zbiorów. Dodawanie i usuwanie elementów. Operacje: union, intersection, difference, issubset, issuperset, etc. Metody słowników. (Wykłady, dyskusja, ćwiczenia, testy.)	Karol Kołodziejczyk	04-12-2025	14:30	15:30	01:00
29 z 29 Walidacja.	-	04-12-2025	15:30	16:00	00:30

Cennik

Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	3 936,00 PLN
Koszt przypadający na 1 uczestnika netto	3 200,00 PLN
Koszt osobogodziny brutto	123,00 PLN
Koszt osobogodziny netto	100,00 PLN

Prowadzący

Liczba prowadzących: 1



1 z 1

Karol Kołodziejczyk

Obszar specjalizacji: Programowanie w językach JavaScript, Python i gdScript, tworzenie aplikacji i gier.

Doświadczenie zawodowe:

- 5 lat prowadzenia własnej działalności w branży informatycznej. Doświadczenie szkoleniowe:
- 3 lata pracy jako nauczyciel programowania w technikum i instruktor programowania na kursach zawodowych w TEB Edukacja.
- 3 lata doświadczenia szkoleniowego w zakresie HTML/CSS & JavaScript.
- 1 rok pracy jako nauczyciel programowania w Technikum Technologii Cyfrowych.

Wykształcenie: Techniczne - technik informatyk.

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Całość opracowanych materiałów składa się z: opisów, ćwiczeń, poleceń, komend, procedur i zdjęć. Po zakończeniu kształcenia wszyscy uczestnicy otrzymują materiały w formie skryptu dotyczące całości przekazywanej wiedzy.

Każdy uczestnik realizujący szkolenie w formie zdalnej w czasie rzeczywistym ma możliwość otrzymania od nas wyposażenia stanowiska szkoleniowego:

- jednostka sprzętowa z niezbędnym oprogramowaniem, peryferia

W/w wyposażenie stanowiska szkoleniowego jest przesłane i następnie odebrane od uczestnika "zdalnego w czasie rzeczywistym" po ukończeniu szkolenia za pośrednictwem kuriera.

Informacje dodatkowe

Faktura za usługę rozwojową podlega zwolnieniu z VAT dla osób korzystających z dofinansowania powyżej 70%.

Szkolenie trwa przez 4 dni, co stanowi łącznie 32 godziny dydaktyczne (przerwy nie są wliczane do czasu trwania usługi).

Pierwsza przerwa zaczyna się 10:15 i kończy 10:30.

Druga przerwa zaczyna się 12:00 i kończy 12:45.

Trzecia przerwa zaczyna się 14:15 i kończy 14:30.

Szkolenie rozpoczyna się pre-testem weryfikującym początkową wiedzę uczestnika usługi rozwojowej i zakończone jest wewnętrznym egzaminem (post-test) weryfikującym i potwierdzającym pozyskaną wiedzę, pozytywne jego zaliczenie honorowane jest certyfikatem potwierdzającym jego ukończenie i uzyskane efekty kształcenia.

Warunki techniczne

Warunki techniczne niezbędne do udziału w usłudze:

- Do połączenia zdalnego w czasie rzeczywistym pomiędzy uczestnikami, a trenerem służy program "Zoom Client for Meetings" (do pobrania ze strony <https://zoom.us/download>).
- Komputer/laptop z kamerką internetową z zainstalowanym klientem Zoom, minimum dwurdzeniowy CPU o taktowaniu 2 GHz.
- Mikrofon i słuchawki (ewentualnie głośniki).
- System operacyjny MacOS 10.7 lub nowszy, Windows 7, 8, 10, Linux: Mint, Fedora, Ubuntu, RedHat.
- Przeglądarkę internetową: Chrome 30 lub nowszy, Firefox 27 lub nowszy, Edge 12 lub nowszy, Safari 7 lub nowsze.
- Dostęp do internetu. Zalecane parametry przepustowości łącza: min. 5 Mbps - upload oraz min. 10 Mbps - download, zarezerwowane w danym momencie na pracę zdalną w czasie rzeczywistym. Umożliwi to komfortową komunikację pomiędzy uczestnikami, a trenerem. Link umożliwiający dostęp do szkolenia jest aktywny przez cały czas jego trwania, do końca zakończenia danego etapu szkolenia. Każdy uczestnik będzie mógł użyć go w dowolnym momencie trwania szkolenia.

Kontakt



Ewelina Klima

E-mail szkolenia@notebookmaster.pl

Telefon (+48) 573 956 049