



Akademia
Techniczno-
Artystyczna Nauk
Stosowanych w
Warszawie

★★★★★ 4,6 / 5
25 ocen

Python & AI Developer - studia podyplomowe

Numer usługi 2025/07/16/14367/2882010

📍 zdalna w czasie rzeczywistym

🎓 Studia podyplomowe

🕒 160 h

📅 15.11.2025 do 30.06.2026

7 500,00 PLN brutto

7 500,00 PLN netto

46,88 PLN brutto/h

46,88 PLN netto/h

Informacje podstawowe

Kategoria

Informatyka i telekomunikacja / Programowanie

Grupa docelowa usługi

Studia podyplomowe **Python & AI Developer** zostały zaprojektowane z myślą o osobach, które chcą zdobyć praktyczne umiejętności programowania w języku Python oraz rozpocząć karierę w rozwijającej się dziedzinie sztucznej inteligencji. Program łączy naukę od podstaw z zaawansowanymi zagadnieniami związanymi z analizą danych, uczeniem maszynowym (Machine Learning) oraz tworzeniem aplikacji webowych i usług opartych o AI. Programowanie w python wsparte AI pozwala na analizę i wizualizację na przykład danych środowiskowych pozwalających na przewidywanie pogody czy emisji zużycia energii, a także tworzenie aplikacji wspierających zrównoważony rozwój. W tym kontekście absolwent może starać się o prace w nowym zielonym sektorze gospodarki tworzącym zielone miejsca pracy.

Minimalna liczba uczestników

15

Maksymalna liczba uczestników

24

Data zakończenia rekrutacji

07-11-2025

Forma prowadzenia usługi

zdalna w czasie rzeczywistym

Liczba godzin usługi

160

Podstawa uzyskania wpisu do BUR

art. 163 ust. 1 ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (t.j. Dz. U. z 2023 r. poz. 742, z późn. zm.)

Cel

Cel edukacyjny

Przygotowanie do roli Python Developera i analityka danych – od podstaw programowania po budowanie modeli AI. Umiejętność tworzenia aplikacji webowych i usług opartych na danych – m.in. predykcyjnych REST API. Znajomość narzędzi i bibliotek stosowanych w przemyśle – Pandas, NumPy, scikit-learn, Django, Flask. Gotowość do pracy w sektorach takich jak IT, analiza danych, web development, automatyzacja czy fintech.

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
--------------------	----------------------	------------------

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Wyjaśnia istotę programowania w języku Python	Wymienia przykłady zastosowań AI w Pythonie.	Test teoretyczny
	Definiuje pojęcie programowania w Python.	Test teoretyczny
	Instaluje Pythona i konfiguruje środowisko pracy	Obserwacja w warunkach rzeczywistych
	Pisze pierwszy skrypt w Pythonie: Hello World	Obserwacja w warunkach symulowanych
	Wymienia podstawy składni Pythona: zmienne, typy danych, operatory.	Test teoretyczny
	Identyfikuje zjawisko sterowania przepływem: instrukcje warunkowe if-elif-else.	Wywiad swobodny
	Definiuje pętle : for i while.	Test teoretyczny
	Wykonuje listy i operacje na listach.	Obserwacja w warunkach rzeczywistych
	Wskazuje na słowniki, krotki i zbiory charakterystyczne dla języka Python.	Wywiad swobodny
	Wymienia zaawansowane koncepcje Pythona.	Test teoretyczny
	Wymienia i identyfikuje elementy języka Python obejmujące charakterystyczną składnię, struktury danych, funkcje, a także bardziej zaawansowane koncepcje takie jak programowanie obiektowe i programowanie funkcyjne.	Test teoretyczny
	Projektuje, implementuje i utrzymuje aplikacje webowe przy użyciu popularnych frameworków Pythona, takich jak Django lub Flask.	Obserwacja w warunkach symulowanych
	Tworzy proste, jak i zaawansowane aplikacje webowe.	Obserwacja w warunkach symulowanych

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Obsługuje funkcje i moduły w Python	Wymienia i definiuje funkcje i moduły w Pythonie	Test teoretyczny
	Wyjaśnia istotę funkcji i tworzy te funkcje w Python	Test teoretyczny
	Stosuje argumenty funkcji i zwracanie wartości	Obserwacja w warunkach symulowanych
	Definiuje pojęcie zasięg zmiennych, wyjaśnia czym są moduły oraz wskazuje jak z nich korzystać.	Wywiad swobodny
	Instaluje i importuje pakiety	Obserwacja w warunkach rzeczywistych
	Obsługuje błędy i wyjątki	Obserwacja w warunkach symulowanych
	Definiuje co to są błędy i wyjątki	Test teoretyczny
	Obsługuje błędy: try, except, finally	Obserwacja w warunkach symulowanych
	Tworzy reguły dla własnych wyjątków	Obserwacja w warunkach symulowanych
	Wymienia i identyfikuje zaawansowane Koncepcje Pythona związane z programowaniem obiektowym i funkcyjnym.	Wywiad swobodny
	Instaluje i importuje biblioteki ML/AI (NumPy, Pandas, scikit-learn)	Obserwacja w warunkach rzeczywistych

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Programuje w języku Python z wykorzystaniem AI	Wyjaśnia istotę programowania obiektowego.	Test teoretyczny
	Wymienia definiuje i rozróżnia klasy i obiekty.	Test teoretyczny
	Identyfikuje metody i atrybuty.	Test teoretyczny
	Definiuje pojęcie dziedziczenie Wyjaśnia pojęcie Klasa „Model” – enkapsulacja danych i metod treningu ML	Test teoretyczny
	Identyfikuje metody statyczne, klasowe oraz detektory	Test teoretyczny
	Wyjaśnia zaawansowane koncepcje Pythona, takie jak: dekoratory, generatory, list comprehensions, wielowątkowość.	Wywiad swobodny
	Testuje i debuguje aplikacje. Stosuje różne techniki testowania i debugowania aplikacji w Pythonie.	Obserwacja w warunkach symulowanych
	Tworzy testy jednostkowe, integrowane oraz debuguje kod.	Obserwacja w warunkach symulowanych

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Pracuje z plikami i bazami danych tworząc zaawansowane struktury danych wykorzystując elementy AI.	Definiuje i tworzy zaawansowane struktury danych.	Obserwacja w warunkach symulowanych
	Różnicuje stosy i kolejki.	Wywiad swobodny
	Buduje drzewa i grafy.	Obserwacja w warunkach symulowanych
	Odczytuje i zapisuje pliki.	Obserwacja w warunkach rzeczywistych
	Różnicuje formaty plików: TXT, CSV, JSON, XML.	Wywiad swobodny
	Tworzy proste zapytania w SQL.	Obserwacja w warunkach rzeczywistych
	Integruje kodowanie w Python z tworzeniem zapytań w SQL: sqlite3.	Obserwacja w warunkach symulowanych
	Zarządza bazami danych – SQLAlchemy.	Obserwacja w warunkach symulowanych
	Przygotowuje dane treningowe, wyjaśnia na czym polega czyszczenie i ekstrakcja cech.	Obserwacja w warunkach rzeczywistych
	Tworzy drzewa decyzyjne z wykorzystaniem AI	Obserwacja w warunkach rzeczywistych

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Wyjaśnia istotę WEB developmentu	Wykorzystuje protokoły komunikacyjne w sieci Internet (np. HTTP/HTTPS) do wymiany danych między klientem a serwerem.	Obserwacja w warunkach symulowanych
	Wykorzystuje Django: routing, szablony, formularze	Obserwacja w warunkach symulowanych
	Wyjaśnia na czym polega integracja Django z bazą danych SQLite	Wywiad swobodny
	Zabezpiecza aplikacje webowe.	Obserwacja w warunkach symulowanych
	Buduje framework na bazie FLASK	Obserwacja w warunkach rzeczywistych
	Wykorzystuje prosty REST API do serwisu predykcji ML (Flask lub Django REST Framework)	Obserwacja w warunkach rzeczywistych
	Tworzy testy jednostkowe, integrowane oraz debuguje kod.	Obserwacja w warunkach symulowanych

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Analizuje dane	Wykorzystuje język Python do analizy danych.	Obserwacja w warunkach symulowanych
	Wyjaśnia budowę i funkcjonalnie analizy danych	Test teoretyczny
	Manipuluje i analizuje dane z wykorzystaniem bibliotek NumPy i Pandas	Obserwacja w warunkach symulowanych
	Czyści i przetwarza dane	Obserwacja w warunkach rzeczywistych
	Wizualizuje dane z Matplotlib i Seaborn	Obserwacja w warunkach symulowanych
	Analizuje dane z wykorzystaniem Machine Learning	Obserwacja w warunkach symulowanych
	Wykorzystuje Supervised vs. Unsupervised Learning	Obserwacja w warunkach symulowanych
	Korzysta z scikit-learn: regresja i klasyfikacja	Obserwacja w warunkach symulowanych
	Waliduje model krzyżowo i tworzy metryki jakości modelu	Obserwacja w warunkach symulowanych
	Dobiera odpowiednią bibliotekę i narzędzie do przetwarzania, wizualizacji i interpretacji danych.	Obserwacja w warunkach rzeczywistych
	Przeprowadza analizę statystyczną.	Obserwacja w warunkach rzeczywistych
	Identyfikuje wzorce, prezentuje wyniki w sposób przejrzysty.	Wywiad swobodny

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

Studia podyplomowe Python & AI Developer trwają 2 semestry i mają charakter praktyczny.

Podczas 160h dydaktycznych uczestnicy studiów zdobędą 30 punktów ECTS.

Studia organizowane są w wersji online na platformie Teams.

Dni zajęć to sobota i niedziele średnio raz lub 2 razy w miesiącu za wyjątkiem okresu świąt, ferii

i wakacji. Zajęcia prowadzone są w formie ćwiczeń, warsztatów, case study oraz multimedialnych

wykładów. Wykładowcami są osoby będącymi ekspertami w zakresie języka Python.

Uczestnicy studiów po ukończeniu programu i zdaniu egzaminu przed komisją otrzymają świadectwo

ukończenia studiów podyplomowych. Przerwy po między zajęciami nie wliczają się czas trwania usługi.

Czas realizacji zajęć to 160h dydaktycznych tj 120h zegarowych.

Egzamin końcowy prowadzony jest przez dedykowaną komisję w skład, której wchodzi Przewodniczący

Komisji - reprezentant uczelni oraz opiekun merytoryczny jako członek komisji co zapewnia to

rozdzielność procesu dydaktycznego od walidacji.

Egzamin dyplomowy odbędzie się nie później niż do 30.06.2026

Treści programowe:

1. **Wprowadzenie do Pythona**
2. Wprowadzenie do programowania i przykłady zastosowań AI w Pythonie
3. Instalacja Pythona i konfiguracja środowiska pracy
4. Pierwszy skrypt w Pythonie: Hello World
5. Podstawy składni Pythona: zmienne, typy danych, operatory
6. Sterowanie przepływem: instrukcje warunkowe if-elif-else
7. Pętle: for i while
8. Listy i operacje na listach
9. Słowniki, krotki i zbiory
10. **Funkcje i moduły w Pythonie**
11. Co to są funkcje i jak je tworzyć
12. Argumenty funkcji i zwracanie wartości
13. Zasięg zmiennych
14. Moduły: co to są i jak z nich korzystać
15. Instalacja i import bibliotek ML/AI (NumPy, Pandas, scikit-learn)
16. **Obsługa błędów i wyjątków**
17. Co to są błędy i wyjątki
18. Jak obsługiwać błędy: try, except, finally
19. Własne wyjątki
20. **Programowanie obiektowe**
21. Wprowadzenie do programowania obiektowego
22. Klasy i obiekty
23. Klasa „Model” – enkapsulacja danych i metod treningu ML
24. Metody i atrybuty
25. Dziedziczenie
26. **Zaawansowane struktury danych**
27. Stosy i kolejki
28. Drzewa i grafy
29. Drzewa decyzyjne jako przykład zastosowania grafów w AI
30. **Praca z plikami i bazami danych**
31. Odczyt i zapis plików
32. Formaty plików: TXT, CSV, JSON, XML
33. Przygotowanie danych treningowych: czyszczenie i ekstrakcja cech
34. Podstawy SQL
35. Python i SQL: sqlite3
36. **Wprowadzenie do web developmentu**
37. Wprowadzenie do sieci i protokołu HTTP
38. Django: routing, szablony, formularze
39. Integracja Django z bazą danych SQLite
40. Prosty REST API do serwisu predykcji ML (Flask lub Django REST Framework)
41. Bezpieczeństwo aplikacji webowych
42. **Analiza danych i wstęp do Machine Learning**
43. Supervised vs. Unsupervised Learning
44. Praktyka z scikit-learn: regresja i klasyfikacja
45. Walidacja krzyżowa i metryki jakości modelu
46. Wprowadzenie do analizy danych
47. Biblioteki NumPy i Pandas
48. Czyszczenie i przetwarzanie danych
49. Wizualizacja danych z Matplotlib i Seaborn

Harmonogram

Liczba przedmiotów/zajęć: 0

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
Brak wyników.					

Cennik

Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	7 500,00 PLN
Koszt przypadający na 1 uczestnika netto	7 500,00 PLN
Koszt osobogodziny brutto	46,88 PLN
Koszt osobogodziny netto	46,88 PLN

Prowadzący

Liczba prowadzących: 2



1 z 2

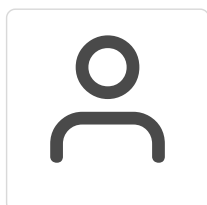
Grzegorz Małek

Doświadczony praktyk i trener.

Grzegorz jest doświadczonym trenerem i nauczycielem z 14 letnim stażem pracy dydaktycznej. Specjalizuje się w podnoszeniu kompetencji z zakresu programowania w Python, SQL oraz zarządzaniu systemami i sieciami informatycznymi ze szczególnym uwzględnieniem zagrożeń i zabezpieczeń. Ponadto pracuje jako wykładowca na wyższej uczelni, zarządza i administruje siecią.

Posiada ukończone studia informatyczne na Politechnice Lubelskiej. Dodatkowo ukończył studia podyplomowe z:

Informatyki śledczej,
Inżynierii oprogramowania
Administracji infrastruktury sieciowej.



2 z 2

Sebastian Bąk

Sebastian jest wieloletnim trenerem z obszaru IT, który przeprowadził ponad 2500 godzin kursów i szkoleń dla różnych grup odbiorców. Twórca kursów on-line oraz artykułów. Jego pasją jest dzielenie się wiedzą i doświadczeniem z innymi, a także ciągle poszerzanie swoich kompetencji w zakresie nowych technologii.

Specjalizuje się w szkoleniach związanych z bazami danych, analizą danych, wizualizacją danych, programowaniem w Pythonie, low-code/no-code i automatyzacją zadań. Prowadził szkolenia dla takich firm i instytucji jak LG, Alior, Pekao S.A., Mbank, Instytut Witolda Pileckiego, UM w Łodzi, UM we Wrocławiu i więcej. Lubi pracować z rozwiązaniami firm Microsoft, Oracle, Google i Adobe, a także pokazywać, jak można je wykorzystywać w praktyce.

Jako trener IT, ma na celu inspirować ludzi do odkrywania możliwości, jakie dają nowoczesne narzędzia informatyczne. Jego wizją jest tworzenie ciekawych i użytecznych projektów, które mogą pomóc w rozwiązywaniu problemów i podejmowaniu decyzji.

Prywatnie interesuje się książkami w gatunku fantasy, łucznictwem oraz strzelectwem.

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Wszelkie materiały zapewnią Test Army i Wrocławska Akademia Biznesu i Nauk Stosowanych

Podczas realizacji zadań na uczelni preferujemy posiadanie przez uczestników własnych komputerów przenośnych do pracy z pakietem Office. Uczelnia zapewnia oprogramowanie dla uczestników studiów – pakiet office jedynie w wersji on-line.

Warunki uczestnictwa

O przyjęciu na studia decyduje komplet dokumentów, które można wysłać tradycyjnie drogą pocztową, skanem lub złożyć osobiście w biurze Centrum Studiów MBA i Podyplomowych

Wymagane dokumenty:

- Podanie o przyjęcie na studia
- 1 zdjęcie w formie papierowej (35x45 mm)
- Kopia dyplomu ukończenia studiów pierwszego lub drugiego stopnia,
- Suplement lub wyciąg z indeksu - dla kończących studia przed 2005 r.
- Podpisana umowa o naukę wraz z załącznikiem - 2 egz.
- Dowód wniesienia opłaty wpisowej (250 PLN)
- Dowód osobisty do wglądu

Warunki techniczne

Własny komputer i internet do pracy w domu.

Uczelnia zapewnia oprogramowanie dla uczestników studiów – pakiet office jedynie w wersji on-line.

Wymagania sprzętowe aplikacji Teams na komputerze z systemem Windows

WYMAGANIA SPRZĘTOWE APLIKACJI TEAMS NA KOMPUTERZE Z SYSTEMEM WINDOWS Składnik	Wymaganie
Komputer i procesor	Minimum 1,1 GHz lub szybszy, 2 rdzenie Uwaga W przypadku procesorów Intel należy wziąć pod uwagę maksymalną prędkość osiągniętą przy użyciu technologii Intel Turbo Boost (Max Turbo Frequency)
Pamięć	4,0 GB PAMIĘCI RAM
Dysk twardy	3,0 GB dostępnego miejsca na dysku
Wyświetlać	rozdzielczość ekranu 1024 x 768
Sprzęt graficzny	System operacyjny Windows: Akceleracja sprzętowa grafiki wymaga DirectX 9 lub nowszego, z WDDM 2.0 lub nowszym dla Windows 10 (lub WDDM 1.3 lub nowszym dla Windows 10 Fall Creators Update)
System operacyjny	Windows 11, Windows 10 (z wyłączeniem Windows 10 LTSC), Windows 10 na ARM, Windows 8.1, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2. Uwaga: Zalecamy korzystanie z najnowszej wersji systemu Windows i dostępnych poprawek zabezpieczeń.
Wersja .NET	Wymaga środowiska .NET 4.5 CLR lub nowszego
Wideo	Kamera wideo USB 2.0
Urządzeń	Standardowa kamera, mikrofon i głośniki do laptopa
Rozmowy wideo i spotkania	- Wymaga 2-rdzeniowego procesora. W przypadku wyższej rozdzielczości wideo/współdzielenia ekranu i liczby klatek na sekundę zalecany jest procesor 4-rdzeniowy lub lepszy. - Efekty wideo w tle wymagają systemu Windows 10 lub procesora z zestawem instrukcji AVX2. - Zobacz Zalecenia dotyczące dekodera sprzętowego i sterownika kodera, aby uzyskać listę nieobsługiwanych dekodów i koderów. - Dołączanie do spotkania za pomocą wykrywania bliskości w pokoju Microsoft Teams wymaga technologii Bluetooth LE, która wymaga włączenia funkcji Bluetooth na urządzeniu klienckim, a w przypadku klientów z systemem Windows wymaga również 64-bitowego klienta usługi Teams. Ta funkcja nie jest dostępna na 32-bitowych klientach usługi Teams.
Wydarzenia na żywo dla zespołów	Jeśli tworzysz wydarzenie na żywo w usłudze Teams, zalecamy użycie komputera z procesorem Core i5 Kaby Lake, pamięcią RAM 4,0 GB (lub większą) i koderem sprzętowym. Zobacz Zalecenia dotyczące dekodera sprzętowego i sterownika kodera, aby uzyskać listę nieobsługiwanych dekodów i koderów.

Kontakt



Weronika Bereska

E-mail weronika.bereska@wab.edu.pl

Telefon (+48) 535 164 459