



Notebook Master  
Sp. z o.o.



## Python / Etap I / Podstawy programowania w języku Python

Numer usługi 2024/11/19/158529/2419590

📍 zdalna w czasie rzeczywistym

🏠 Usługa szkoleniowa

🕒 40 h

📅 02.12.2024 do 06.12.2024

3 567,00 PLN brutto

2 900,00 PLN netto

89,18 PLN brutto/h

72,50 PLN netto/h

## Informacje podstawowe

<b>Kategoria</b>	Informatyka i telekomunikacja / Programowanie
<b>Sposób dofinansowania</b>	wsparcie dla osób indywidualnych wsparcie dla pracodawców i ich pracowników
<b>Grupa docelowa usługi</b>	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.  Usługa również adresowana dla Uczestników Projektu "Małopolski pociąg do kariery - sezon 1".
<b>Minimalna liczba uczestników</b>	1
<b>Maksymalna liczba uczestników</b>	8
<b>Forma prowadzenia usługi</b>	zdalna w czasie rzeczywistym
<b>Liczba godzin usługi</b>	40
<b>Podstawa uzyskania wpisu do BUR</b>	Znak Jakości Małopolskich Standardów Usług Edukacyjno-Szkoleniowych (MSUES) - wersja 2.0

## Cel

### Cel edukacyjny

Usługa "Python / Etap I / Podstawy programowania w języku Python.", prowadzi do nabycia specjalistycznych kompetencji w obszarze tematycznym szkolenia (w tym do rozwoju umiejętności w obszarze TIK (ITC) oraz kompetencji

cyfrowych) i przygotowuje do samodzielnego i prawidłowego wykonywania obowiązków w zakresie podstaw programowania z wykorzystaniem języka Python.

## Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Pobiera i instaluje Python.	Instaluje Python zgodnie z zaleceniami.	Test teoretyczny
	Potwierdza poprawną instalację poprzez sprawdzenie wersji Pythona.	Test teoretyczny
Sprawdza wersję Pythona.	Korzysta z odpowiedniej komendy w terminalu lub interpreterze Pythona do sprawdzenia zainstalowanej wersji Pythona.	Test teoretyczny
Instaluje edytor kodu, np. Visual Studio Code.	Instaluje edytor kodu Visual Studio Code na swoim systemie.	Test teoretyczny
	Potwierdza poprawną instalację i uruchomienie edytora kodu.	Test teoretyczny
Rozróżnia typy danych, operatory arytmetyczne, porównania, przypisania (w tym skrótowych) i logiczne.	Wyjaśnia różne typy danych oraz ich zastosowanie w Pythonie.	Test teoretyczny
	Przedstawia działanie operatorów arytmetycznych, porównań, przypisań (w tym skrótowych) i logicznych w kontekście Pythona.	Test teoretyczny
Korzysta z terminala do uruchamiania skryptów Pythona i komunikacji z terminalem i zarządzania zmiennymi środowiskowymi.  Pracuje z operacjami na plikach, takimi jak otwieranie, zapisywanie i odczytywanie, oraz obsługą błędów związanymi z operacjami na plikach.	Wykorzystuje terminal do uruchamiania skryptów Pythona.	Test teoretyczny
	Przedstawia działanie komunikacji z terminalem oraz zarządzania zmiennymi środowiskowymi.	Test teoretyczny
	Wykonuje operacje na plikach w Pythonie, w tym otwieranie, zapisywanie i odczytywanie danych.	Test teoretyczny
	Radzi sobie z obsługą błędów związanych z operacjami na plikach.	Test teoretyczny
Tworzy interfejs graficzny, w tym okna, widżety oraz układ i rozmieszczenie elementów interfejsu.	Konstruuje interfejs graficzny z wykorzystaniem narzędzi dostępnych we frameworku tkinter.	Test teoretyczny
	Tworzy okna, widżety i układy, a także odpowiednio rozmieszcza elementy interfejsu.	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Korzysta z Jupyter Notebook do tworzenia komórek kodu.	Korzysta z Jupyter Notebook do tworzenia, edycji i wykonywania komórek kodu Pythona.	Test teoretyczny
	Demonstruje umiejętność zapisywania i udostępniania notatników Jupyter.	Test teoretyczny
Wprowadza pojęcie enkapsulacji, praca z lokalnymi modułami.	Omawia pojęcie enkapsulacji w Pythonie.	Test teoretyczny
	Przedstawia sposób pracy z lokalnymi modułami w kontekście enkapsulacji.	Test teoretyczny
Instaluje dodatkowe moduły i importuje je.  Tworzy i aktywuje środowisko wirtualne (venv).	Instaluje dodatkowe moduły za pomocą menedżera pakietów pip.	Test teoretyczny
	Poprawnie importuje dodatkowe moduły do swoich projektów Pythona.	Test teoretyczny
	Tworzy środowisko wirtualne za pomocą narzędzia venv.	Test teoretyczny
Instaluje pakiety w środowisku wirtualnym.	Aktywuje środowisko wirtualne i wykorzystuje je do pracy z projektem.	Test teoretyczny
	Opisuje proces instalacji dodatkowe pakiety za pomocą menedżera pakietów w aktywowanym środowisku wirtualnym.	Test teoretyczny
	Potwierdza poprawność instalacji i dostępność pakietów w środowisku wirtualnym.	Test teoretyczny
	Tworzy plik requirements.txt zawierający niezbędne pakiety projektu.	Test teoretyczny
Porównuje menedżery pakietów: Pip, Conda, Poetry.	Omawia różnice między menedżerami pakietów Pip, Conda i Poetry.	Test teoretyczny
	Przedstawia zalety i wady każdego z menedżerów pakietów.	Test teoretyczny
Omawia typy danych liczbowych, takie jak int, float, complex oraz funkcje matematyczne z modułu 'math'.	Charakteryzuje różne typy danych liczbowych oraz ich zastosowanie.	Test teoretyczny
	Przedstawia funkcje matematyczne dostępne w module 'math' i ich działanie.	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Pracuje z napisami (string), w tym formatowaniem, indeksowaniem i wycinaniem fragmentów tekstu, oraz korzysta z metod dla napisów.	Wykorzystuje różne operacje na napisach, takie jak formatowanie, indeksowanie i wycinanie fragmentów tekstu.	Test teoretyczny
	Korzysta z różnych metod dostępnych dla napisów w Pythonie.	Test teoretyczny
Charakteryzuje kolekcje, takie jak listy, krotki, zbiory i słowniki, oraz wykonuje operacje na nich.	Omawia cechy i zastosowanie różnych kolekcji w Pythonie.	Test teoretyczny
	Wykonuje operacje na kolekcjach, takie jak dodawanie, usuwanie, indeksowanie itp.	Test teoretyczny
Stosuje konstrukcje logiczne, takie jak operacje warunkowe (if, elif, else), pętle (for, while), list comprehension, set comprehension i dict comprehension.	Wykorzystuje różne konstrukcje logiczne do manipulacji danymi w Pythonie.	Test teoretyczny
	Korzysta z pętli oraz składni comprehension.	Test teoretyczny
Definiuje funkcje za pomocą 'def' i korzysta z funkcji anonimowych (lambda).	Definiuje funkcje za pomocą instrukcji 'def'.	Test teoretyczny
	Wykorzystuje funkcje anonimowe (lambda) w odpowiednich kontekstach.	Test teoretyczny
Zwraca dane z funkcji za pomocą 'return' oraz obsługuje wartość 'None'.	Poprawnie stosuje instrukcję 'return' w funkcjach do zwracania wartości.	Test teoretyczny
	Obsługuje przypadki, w których funkcja może zwrócić wartość 'None'.	Test teoretyczny
Pracuje z parametrami funkcji, argumentami pozycyjnymi, domyślnymi oraz korzysta z *args i **kwargs w funkcjach.	Wykorzystuje różne rodzaje parametrów funkcji, takie jak argumenty pozycyjne, domyślne, *args i **kwargs.	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, dokument potwierdza, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji.

**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 w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji.

**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

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. Instalacja.

1. Pobieranie i instalacja interpretera Python.
2. Sprawdzanie wersji Pythona.
3. Semantyka, a prezentacja. Instalacja edytora kodu (na przykładzie Visual Studio Code).

## II. Operatory i zmienne.

1. Typy danych - wprowadzenie.
2. Operatory arytmetyczne.
3. Operatory porównania.
4. Operatory przypisania.
5. Operatory logiczne.

## III. Komunikacja z programem.

1. Terminal.
  - a: Uruchamianie skryptów Pythona.
  - b: Komunikacja z terminalem.
  - c: Zmienne środowiskowe.
2. Operacje na plikach.
  - a: Otwieranie, zapisywanie, odczytywanie.
  - b: Obsługa błędów.
3. Interfejs graficzny.
  - a: Tworzenie okien.
  - b: Tworzenie widgetów.
  - c: Układ i rozmieszczenie.
4. Jupyter Notebook.
  - a: Instalacja i uruchomienie.
  - b: Tworzenie komórek kodu i Markdown.

## IV. Środowisko.

1. Enkapsulacja.
  - a: Wprowadzenie do koncepcji.
  - b: Lokalne moduły.
2. Instalacja dodatkowych modułów.
  - a: Importowanie modułów.
  - b: Użycie pip do instalacji pakietów.
3. Środowiska wirtualne.
  - a: Tworzenie i aktywacja środowiska wirtualnego (venv).
  - b: Instalacja pakietów w środowisku wirtualnym.
4. Pip/Anaconda/Poetry.
  - a: Różnice między menedżerami pakietów.
  - b: Instalacja i podstawowe komendy.

## V. Typy danych - Szczegółowo.

1. Typy liczbowe.
  - a: Int, float, complex.
  - b: Funkcje matematyczne (math).
2. Tekst.
  - a: String.
  - b: Formatowanie, indeksowanie, wycinanie.
  - c: Metody stringów.

- 3. Kontenery.
  - a: Listy, krotki, zbiory, słowniki.
  - b: Operacje na kontenerach.

## VI. Konstrukcje logiczne.

- 1. Operacje warunkowe i operator trójskładnikowy.
  - a: if, elif, else.
  - b: Warunki złożone.
  - c: Operator trójskładnikowy.
- 2. Pętle i wyrażenia kontenerów.
  - a: Pętla for, while.
  - b: Pętla zagnieżdżone.
  - c: List comprehension.
  - d: Set comprehension.
  - e: Dict comprehension.

## VII. Funkcje.

- 1. Definicje def i lambda.
  - a: Tworzenie funkcji za pomocą def.
  - b: Funkcje anonimowe (lambda).
- 2. Zwracanie danych.
  - a: Instrukcja return.
  - b: Wartość None.
- 3. Parametry, args, kwargs.
  - a: Parametry pozycyjne, domyślne, nazwane.
  - b: \*args i \*\*kwargs.
- 4. Zmienne lokalne.
  - a: Zasięg zmiennych.
  - b: Zmienne globalne w funkcji.
  - c: Shadowing.

Szkolenie trwa 40 godzin dydaktycznych i realizowane jest w kameralnych grupach, maksymalnie 8-osobowych. Do uczestników realizujących usługę rozwojową w formie zdalnej w czasie rzeczywistym, dostarczamy kurierem wyposażenie stanowiska szkoleniowego (po ukończeniu szkolenia sprzęt zostaje odebrany przez kuriera).

Na czas trwania usługi składają się 13 godzin zajęć teoretycznych i 27 godzin zajęć praktycznych.

Przerwy nie są wliczane do czasu trwania usługi .

# Harmonogram

Liczba przedmiotów/zajęć: 36

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>1 z 36</b> Pobieranie i instalacja interpretera Python. Sprawdzanie wersji Pythona. (Wykłady, dyskusja, ćwiczenia, testy.)	Karol Kołodziejczyk	02-12-2024	08:45	10:15	01:30

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
2 z 36 Przerwa.	Karol Kołodziejczyk	02-12-2024	10:15	10:30	00:15
3 z 36 Semantyka, a prezentacja. Instalacja edytora kodu (na przykładzie Visual Studio Code). (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	02-12-2024	10:30	12:00	01:30
4 z 36 Przerwa.	Karol Kołodziejczyk	02-12-2024	12:00	12:45	00:45
5 z 36 Operatory i zmienne. Typy danych - wprowadzenie. Operatory arytmetyczne. Operatory przypisania. Operatory logiczne. (Wykłady, dyskusja, ćwiczenia.)	Karol Kołodziejczyk	02-12-2024	12:45	14:15	01:30
6 z 36 Przerwa.	Karol Kołodziejczyk	02-12-2024	14:15	14:30	00:15
7 z 36 Komunikacja z programem. Terminal. Uruchamianie skryptów Pythona. Komunikacja z terminalem. Zmienne środowiskowe. (Wykłady, dyskusja, ćwiczenia.)	Karol Kołodziejczyk	02-12-2024	14:30	16:00	01:30



Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>8 z 36</b> Operacje na plikach. Otwieranie, zapisywanie, odczytywanie. Obsługa błędów. (Wykłady, dyskusja, ćwiczenia.)	Karol Kołodziejczyk	03-12-2024	08:45	10:15	01:30
<b>9 z 36</b> Przerwa.	Karol Kołodziejczyk	03-12-2024	10:15	10:30	00:15
<b>10 z 36</b> Interfejs graficzny. Tworzenie okien. Tworzenie widgetów. Układ i rozmieszczenie. (Wykłady, dyskusja, ćwiczenia.)	Karol Kołodziejczyk	03-12-2024	10:30	12:00	01:30
<b>11 z 36</b> Przerwa.	Karol Kołodziejczyk	03-12-2024	12:00	12:45	00:45
<b>12 z 36</b> Jupyter Notebook. Instalacja i uruchomienie. Tworzenie komórek kodu i Markdown. (Wykłady, dyskusja, ćwiczenia.)	Karol Kołodziejczyk	03-12-2024	12:45	14:15	01:30
<b>13 z 36</b> Przerwa.	Karol Kołodziejczyk	03-12-2024	14:15	14:30	00:15
<b>14 z 36</b> Środowisko. Enkapsulacja. Wprowadzenie do koncepcji. Lokalne moduły. (Wykłady, dyskusja, ćwiczenia.)	Karol Kołodziejczyk	03-12-2024	14:30	16:00	01:30

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>15 z 36</b> Instalacja dodatkowych modułów. Importowanie modułów. Użycie pip do instalacji pakietów. (Wykłady, dyskusja, ćwiczenia.)	Karol Kołodziejczyk	04-12-2024	08:45	10:15	01:30
<b>16 z 36</b> Przerwa.	Karol Kołodziejczyk	04-12-2024	10:15	10:30	00:15
<b>17 z 36</b> Środowiska wirtualne. Tworzenie i aktywacja środowiska wirtualnego (venv). Instalacja pakietów w środowisku wirtualnym. (Wykłady, dyskusja, ćwiczenia.)	Karol Kołodziejczyk	04-12-2024	10:30	12:00	01:30
<b>18 z 36</b> Przerwa.	Karol Kołodziejczyk	04-12-2024	12:00	12:45	00:45
<b>19 z 36</b> Pip/Anaconda/Pytropy. Różnice między menedżerami pakietów. Instalacja i podstawowe komendy. (Wykłady, dyskusja, ćwiczenia.)	Karol Kołodziejczyk	04-12-2024	12:45	14:15	01:30
<b>20 z 36</b> Przerwa.	Karol Kołodziejczyk	04-12-2024	14:15	14:30	00:15

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>21 z 36</b> Typy danych - Szczegółowo. Typy liczbowe. Int, float, complex. Funkcje matematyczne (math). (Wykłady, dyskusja, ćwiczenia.)</p>	Karol Kołodziejczyk	04-12-2024	14:30	16:00	01:30
<p><b>22 z 36</b> Tekst. String. Formatowanie, indeksowanie, wycinanie. Metody stringów. (Wykłady, dyskusja, ćwiczenia.)</p>	Karol Kołodziejczyk	05-12-2024	08:45	10:15	01:30
<p><b>23 z 36</b> Przerwa.</p>	Karol Kołodziejczyk	05-12-2024	10:15	10:30	00:15
<p><b>24 z 36</b> Kontenery. Listy, krotki, zbiory, słowniki. Operacje na kontenerach. (Wykłady, dyskusja, ćwiczenia.)</p>	Karol Kołodziejczyk	05-12-2024	10:30	12:00	01:30
<p><b>25 z 36</b> Przerwa.</p>	Karol Kołodziejczyk	05-12-2024	12:00	12:45	00:45
<p><b>26 z 36</b> Konstrukcje logiczne. Operacje warunkowe i operator trójskładnikowy. (Wykłady, dyskusja, ćwiczenia.)</p>	Karol Kołodziejczyk	05-12-2024	12:45	14:15	01:30
<p><b>27 z 36</b> Przerwa.</p>	Karol Kołodziejczyk	05-12-2024	14:15	14:30	00:15

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>28 z 36</b> Pętle i wyrażenia kontenerów. Pętle for, while. Pętle zagnieżdżone. List/set/dict comprehension. (Wykłady, dyskusja, ćwiczenia.)</p>	Karol Kołodziejczyk	05-12-2024	14:30	16:00	01:30
<p><b>29 z 36</b> Funkcje. Definicje def i lambda. Tworzenie funkcji za pomocą def. Funkcje anonimowe (lambda). (Wykłady, dyskusja, ćwiczenia.)</p>	Karol Kołodziejczyk	06-12-2024	08:45	10:15	01:30
<p><b>30 z 36</b> Przerwa.</p>	Karol Kołodziejczyk	06-12-2024	10:15	10:30	00:15
<p><b>31 z 36</b> Zwracanie danych. Instrukcja return. Wartość None. (Wykłady, dyskusja, ćwiczenia.)</p>	Karol Kołodziejczyk	06-12-2024	10:30	12:00	01:30
<p><b>32 z 36</b> Przerwa.</p>	Karol Kołodziejczyk	06-12-2024	12:00	12:45	00:45
<p><b>33 z 36</b> Parametry, args, kwargs. Parametry pozycyjne, domyślne, nazwane. *args i **kwargs. (Wykłady, dyskusja, ćwiczenia.)</p>	Karol Kołodziejczyk	06-12-2024	12:45	14:15	01:30
<p><b>34 z 36</b> Przerwa.</p>	Karol Kołodziejczyk	06-12-2024	14:15	14:30	00:15

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<span>35 z 36</span> Zmienne lokalne. Zasięg zmiennych. Zmienne globalne w funkcji. Shadowing. (Wykłady, dyskusja, ćwiczenia, testy.)	Karol Kołodziejczyk	06-12-2024	14:30	15:30	01:00
<span>36 z 36</span> Walidacja.	-	06-12-2024	15:30	16:00	00:30

## Cennik

### Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	3 567,00 PLN
Koszt przypadający na 1 uczestnika netto	2 900,00 PLN
Koszt osobogodziny brutto	89,18 PLN
Koszt osobogodziny netto	72,50 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.

Sprzęt i materiały przesłane do uczestnika "zdalnego w czasie rzeczywistym" na czas trwania szkolenia (w celu realizacji części praktycznej szkolenia):

- 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 5 dni co stanowi łącznie 40 godzin dydaktycznych.

Harmonogram uwzględnia łączną liczbę godzin szkolenia, jako 36,25 godzin zegarowych, ponieważ uwzględnia również przerwy pomiędzy poszczególnymi blokami zajęć (I przerwa - 15 min, II przerwa - 45 min, III przerwa 15 min / 1 dzień).

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.

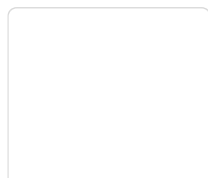
Szkolenie rozpoczyna się pre-testem weryfikującym początkową wiedzę uczestnika usługi rozwojowej i zakończone jest wewnętrznym egzaminem (post-test) weryfikującym i potwierdzającym pozyskaną wiedzę, pozytywne jego zaliczenie honorowane jest certyfikatem potwierdzającym jego ukończenie i uzyskane efekty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



**Artur Kowalewski**

**E-mail** szkolenia@notebookmaster.pl



**Telefon** (+48) 573 436 635