



Notebook Master  
Sp. z o.o.



## Python / Etap II / Python dla średnio zaawansowanych

Numer usługi 2024/05/16/158529/2152410

📍 zdalna w czasie rzeczywistym

🏠 Usługa szkoleniowa

🕒 32 h

📅 02.12.2024 do 05.12.2024

3 936,00 PLN brutto

3 200,00 PLN netto

123,00 PLN brutto/h

100,00 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 na poziomie średnio zaawansowanym i wykorzystać ją w ramach prowadzonej działalności gospodarczej i etatu.
<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>	32
<b>Podstawa uzyskania wpisu do BUR</b>	Znak Jakości Małopolskich Standardów Usług Edukacyjno-Szkoleniowych (MSUES) - wersja 2.0

## Cel

### Cel edukacyjny

Usługa "Python / Etap II / Python dla średnio zaawansowanych", 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.	Opisuje różne typy zmiennych i ich zastosowania w kontekście programowania.	Test teoretyczny
	Omawia funkcje, ich definicje oraz różne sposoby przekazywania argumentów funkcji.	Test teoretyczny
Wykorzystuje biblioteki 'typing' i 'mypy'.	Stosuje adnotacje typów z biblioteki 'typing' w kodzie Pythona.	Test teoretyczny
	Wykorzystuje narzędzie 'mypy' do statycznej analizy typów i sprawdzania zgodności typów.	Test teoretyczny
Tworzy czytelne i dokładne docstringi opisujące cel funkcji, klasy i modułów. Kryteria weryfikacji:	Tworzy czytelne docstringi opisujące funkcje, klasy oraz moduły.	Test teoretyczny
Pracuje z docstringami typu reStructuredText i potrafi generować dokumentację na podstawie docstringu za pomocą narzędzi takich jak Sphinx.	Stosuje docstringi w konwencji reStructuredText.	Test teoretyczny
	Generuje dokumentację na podstawie docstringów przy użyciu narzędzi takich jak Sphinx.	Test teoretyczny
Przestrzega podstawowych zasad stylu zgodnie z PEP-8, zarządzając nazewnictwem, długością linii, wcięciami, pustymi liniami i odstępami.	Stosuje zasady PEP-8 dotyczące nazewnictwa zmiennych, funkcji, klas oraz zarządza formatowaniem kodu, wcięciami i długością linii.	Test teoretyczny
	Zachowuje czytelność kodu poprzez odpowiednie użycie pustych linii i odstępów.	Test teoretyczny
Korzysta z narzędzi do autoformatowania, takich jak Black, Blue, autopep8, yapf, i integruje je z edytorami kodu.	Stosuje narzędzia do autoformatowania kodu, takie jak Black, Blue, autopep8, yapf.	Test teoretyczny
	Integruje narzędzia autoformatujące z edytorem kodu.	Test teoretyczny
Wykorzystuje narzędzia lintingu, takie jak pylint, flake8, pyright, do analizy statycznej kodu, wykrywania potencjalnych błędów i niedoskonałości.	Stosuje narzędzia lintingu do analizy statycznej kodu w celu wykrywania potencjalnych błędów i niedoskonałości.	Test teoretyczny
	Efektywnie interpretuje wyniki narzędzi lintingu i podejmuje odpowiednie działania naprawcze.	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
Tworzy własne wyjątki.	Definiuje własne wyjątki zgodnie z konwencjami Pythona.	Test teoretyczny
	Wykorzystuje własne wyjątki w kodzie w odpowiednich sytuacjach.	Test teoretyczny
Pracuje z testami jednostkowymi, wykorzystując moduły doctest i pytest tworzy testy, testuje przypadki graniczne i izolację.	Tworzy testy jednostkowe zgodnie z konwencjami danego modułu testowego (np. pytest, unittest).	Test teoretyczny
	Testuje różne przypadki, w tym przypadki graniczne i izolację w testach.	Test teoretyczny
Dokonuje logowania, używając funkcji print, logowania do plików oraz modułów logging i nezu, z uwzględnieniem różnych poziomów logowania.	Stosuje funkcję print do prostego logowania komunikatów.	Test teoretyczny
	Wykorzystuje moduł logging do logowania z uwzględnieniem różnych poziomów logowania i formatowania logów.	Test teoretyczny
Stosuje koncepcje programowania obiektowego, tworzy klasy i instancje, inicjalizuje obiekty, zarządza atrybutami klasowymi i instancyjnymi oraz korzysta z getterów i setterów.	Wykorzystuje moduł nezu do logowania z uwzględnieniem różnych poziomów logowania i formatowania logów.	Test teoretyczny
	Zapisuje logi do plików zgodnie z wymaganiami projektu.	Test teoretyczny
	Tworzy klasy zgodnie z koncepcjami programowania obiektowego.	Test teoretyczny
Pracuje z metodami instancji, statycznymi i klasowymi, stosuje dekoratory, oraz tworzy konstruktory i destruktory, zarówno z domyślnymi jak i z parametrami.	Poprawnie inicjalizuje obiekty, zarządza atrybutami klasowymi i instancyjnymi, oraz stosuje gettery i settery w zależności od potrzeb.	Test teoretyczny
	Definiuje metody instancji, statyczne i klasowe zgodnie z potrzebami projektu.	Test teoretyczny
	Wykorzystuje dekoratory do stosowania różnych funkcjonalności metod.	Test teoretyczny
	Tworzy konstruktory i destruktory zgodnie z konwencjami języka Python.	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
<p>Korzysta z metod specjalnych, włączając str, repr, len, getitem, setitem, i interakcję z obiektami.</p> <p>Przeładowuje operatory, w tym arytmetyczne, porównania, logiczne, oraz rozumie ich zastosowania i ograniczenia.</p>	<p>Implementuje metody specjalne, takie jak str, repr, len, getitem, setitem zgodnie z potrzebami projektu.</p>	<p>Test teoretyczny</p>
	<p>Efektywnie korzysta z metod specjalnych do interakcji z obiektami i manipulacji nimi.</p>	<p>Test teoretyczny</p>
	<p>Przeładowuje operatory arytmetyczne, porównania, logiczne zgodnie z potrzebami projektu.</p>	<p>Test teoretyczny</p>
	<p>Rozumie zastosowania i ograniczenia związane z przeładowywaniem operatorów.</p>	<p>Test teoretyczny</p>
<p>Dokonuje konwersji między typami oraz rzutowania typów.</p> <p>Wykorzystuje metody dla stringów, list, zbiorów i słowników.</p>	<p>Wykonuje konwersje między różnymi typami danych w sposób bezpieczny i efektywny.</p>	<p>Test teoretyczny</p>
	<p>Stosuje rzutowanie typów w przypadkach, gdy jest to konieczne lub uzasadnione.</p>	<p>Test teoretyczny</p>
	<p>Korzysta z wbudowanych metod dla stringów, list, zbiorów i słowników w celu manipulacji nimi.</p>	<p>Test teoretyczny</p>
	<p>Rozumie działanie poszczególnych metod i potrafi je zastosować w praktyce.</p>	<p>Test teoretyczny</p>
<p>Pracuje z metodami krotek oraz korzysta z operacji na zbiorach i słownikach.</p>	<p>Wykorzystuje metody krotek oraz operacje na zbiorach i słownikach zgodnie z wymaganiami projektu.</p>	<p>Test teoretyczny</p>
	<p>Manipuluje danymi przechowywanymi w krotkach, zbiorach i słownikach.</p>	<p>Test teoretyczny</p>
<p>Dodaje, usuwa i aktualizuje elementy w słownikach, a także stosuje metodę get.</p>	<p>Poprawnie dodaje, usuwa i aktualizuje elementy w słownikach zgodnie z potrzebami projektu.</p>	<p>Test teoretyczny</p>
	<p>Stosuje metodę get do bezpiecznego dostępu do wartości w słownikach.</p>	<p>Test teoretyczny</p>

# 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 opis efektów 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 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. Jakość kodu.

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

## II. Testowanie.

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

## III. Programowanie obiektowe.

1. Klasa i obiekt.
  - a: Tworzenie klas i instancji.
  - b: Inicjalizacja obiektu.
2. Atrybuty.
  - a: Atrybuty klasowe i instancji.
  - b: Gettery i settery.
  - c: Atrybuty prywatne i chronione.
3. Metody.
  - a: Metody instancji, statyczne, klasowe.
  - b: Dekoratory.
4. Konstruktor i destruktork.
  - a: init i del.
  - b: Tworzenie obiektów z parametrami.

5. Metody dunder.
  - a: str, repr, len, getitem, setitem.
  - b: Interakcja z obiektami.
6. Przeladowanie operatorów.
  - a: Operatory arytmetyczne, porównania, logiczne.
  - b: Zastosowania i ograniczenia.
7. Konwersje.
  - a: Konwersje między typami.
  - b: Rzutowanie typów.

#### IV. Kontenery i stringi.

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

Szkolenie trwa 32 godziny dydaktyczne 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).

## Harmonogram

Liczba przedmiotów/zajęć: 29

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>1 z 29</b> Jakość kodu. Wskazówki typów. Typy zmiennej, funkcji, argumentów funkcji. Biblioteki typing i mypy. Zaawansowane wskazówki typów. (Wykłady, dyskusja, ćwiczenia, testy.)	Karol Kołodziejczyk	02-12-2024	08:45	10:15	01:30
<b>2 z 29</b> Przerwa.	Karol Kołodziejczyk	02-12-2024	10:15	10:30	00:15

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>3 z 29</b> Docstring. Opisywanie celu funkcji, klas i modułów. Styl docstring (np. Google, NumPy, reStructuredText). Dokumentacja na podstawie docstringu (np. Sphinx). (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	02-12-2024	10:30	12:00	01:30
<p><b>4 z 29</b> Przerwa.</p>	Karol Kołodziejczyk	02-12-2024	12:00	12:45	00:45
<p><b>5 z 29</b> PEP-8. Podstawowe zasady stylu. Nazewnictwo, długość linii, wcięcia. Puste linie i spacje. (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	02-12-2024	12:45	14:15	01:30
<p><b>6 z 29</b> Przerwa.</p>	Karol Kołodziejczyk	02-12-2024	14:15	14:30	00:15
<p><b>7 z 29</b> Autoformatowanie. Narzędzia: Black, autopep8, yapf. (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	02-12-2024	14:30	16:00	01:30
<p><b>8 z 29</b> Linting. Narzędzia: pylint, flake8, pyright. Analiza statyczna kodu. Detekcja potencjalnych błędów i niedoskonałości. (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	03-12-2024	08:45	10:15	01:30
<p><b>9 z 29</b> Przerwa.</p>	Karol Kołodziejczyk	03-12-2024	10:15	10:30	00:15



Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>10 z 29</b> Testowanie. Wyjątki. Obsługa błędów: try, except, finally, raise. Własne wyjątki. (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	03-12-2024	10:30	12:00	01:30
<p><b>11 z 29</b> Przerwa.</p>	Karol Kołodziejczyk	03-12-2024	12:00	12:45	00:45
<p><b>12 z 29</b> Testy jednostkowe. Moduły (pytest, robot, unittest). Tworzenie testów. Testowanie przypadków granicznych. Mockowanie i izolacja. (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	03-12-2024	12:45	14:15	01:30
<p><b>13 z 29</b> Przerwa.</p>	Karol Kołodziejczyk	03-12-2024	14:15	14:30	00:15
<p><b>14 z 29</b> Logi. Funkcja print. Logowanie do plików. Moduł logging. Poziomy logowania: DEBUG, INFO, WARNING, ERROR, CRITICAL. Formatowanie logów. (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	03-12-2024	14:30	16:00	01:30

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>15 z 29</b>  Programowanie obiektowe. Klasa i obiekt.  Tworzenie klas i instancji.  Inicjalizacja obiektu.  (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	04-12-2024	08:45	10:15	01:30
<p><b>16 z 29</b> Przerwa.</p>	Karol Kołodziejczyk	04-12-2024	10:15	10:30	00:15
<p><b>17 z 29</b> Atrybuty.  Atrybuty klasowe i instancji.  Gettery i settery.  Atrybuty prywatne i chronione.  (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	04-12-2024	10:30	12:00	01:30
<p><b>18 z 29</b> Przerwa.</p>	Karol Kołodziejczyk	04-12-2024	12:00	12:45	00:45
<p><b>19 z 29</b> Metody.  Metody instancji, statyczne, klasowe.  Dekoratory.  Konstruktor i destruktork. init i del. Tworzenie obiektów z parametrami.  (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	04-12-2024	12:45	14:15	01:30
<p><b>20 z 29</b> Przerwa.</p>	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 29</b> Metody dunder. str, repr, len, getitem, setitem. Interakcja z obiektami. Przeladowanie operatorów. Operatory arytmetyczne, porównania, logiczne. (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	04-12-2024	14:30	16:00	01:30
<p><b>22 z 29</b> Konwersje. Konwersje między typami. Rzutowanie typów. Kontenery i stringi. (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	05-12-2024	08:45	10:15	01:30
<p><b>23 z 29</b> Przerwa.</p>	Karol Kołodziejczyk	05-12-2024	10:15	10:30	00:15
<p><b>24 z 29</b> Metody stringów. Użycie metod: split, join, replace, find, count, etc. Funkcje związane z wielkością liter: upper, lower, capitalize, title, etc. (Wykłady, dyskusja, ćwiczeni)</p>	Karol Kołodziejczyk	05-12-2024	10:30	12:00	01:30
<p><b>25 z 29</b> Przerwa.</p>	Karol Kołodziejczyk	05-12-2024	12:00	12:45	00:45

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>26 z 29</b> Metody krotek. Metody count, index. Metody list. Dodawanie i usuwanie elementów. Sortowanie i reorganizacja. (Wykłady, dyskusja, ćwiczeni)	Karol Kołodziejczyk	05-12-2024	12:45	14:15	01:30
<b>27 z 29</b> Przerwa.	Karol Kołodziejczyk	05-12-2024	14:15	14:30	00:15
<b>28 z 29</b> Metody i operatory zbiorów. Dodawanie i usuwanie elementów. Operacje: union, intersection, difference, issubset, issuperset, etc. Metody słowników. (Wykłady, dyskusja, ćwiczenia, testy.)	Karol Kołodziejczyk	05-12-2024	14:30	15:30	01:00
<b>29 z 29</b> Walidacja.	-	05-12-2024	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 jest bardzo szczegółowe, ponieważ zależy nam na przekazaniu jak największej ilości informacji. Trwa przez 4 dni, co stanowi łącznie 32 godziny dydaktyczne.

Harmonogram uwzględnia łączną liczbę godzin szkolenia, jako 29 godzin zegarowych, ponieważ uwzględnia również przerwy pomiędzy poszczególnymi blokami zajęć (I przerwa - 15 min, II przerwa - 45 min, III przerwa 15 min / 1 dzień).

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](mailto:szkolenia@notebookmaster.pl)

**Telefon** (+48) 573 956 049