



JavaScript od podstaw do technik zaawansowanych

Numer usługi 2026/02/12/202247/3330813

3 690,00 PLN brutto

3 000,00 PLN netto

105,43 PLN brutto/h

85,71 PLN netto/h

157,50 PLN cena rynkowa ⓘ

JSYSTEMS SPÓŁKA
Z OGRANICZONĄ
ODPOWIEDZIALNOŚ
CIĄ

Brak ocen dla tego dostawcy

- 📄 Usługa szkoleniowa
- 📺 zdalna w czasie rzeczywistym
- 🕒 35:00 h
- 📅 23.11.2026 do 27.11.2026

Informacje podstawowe

| | |
|--|--|
| Kategoria | Informatyka i telekomunikacja / Programowanie |
| Grupa docelowa usługi | Szkolenie jest odpowiednie zarówno dla osób, które chcą rozpocząć naukę programowania w JavaScript od podstaw, jak i dla tych, którzy mają już pierwsze doświadczenia z tym językiem i chcą poszerzyć swoje umiejętności oraz poznać bardziej zaawansowane techniki. |
| Minimalna liczba uczestników | 6 |
| Maksymalna liczba uczestników | 15 |
| Data zakończenia rekrutacji | 19-11-2026 |
| Forma prowadzenia usługi | zdalna w czasie rzeczywistym |
| Liczba godzin usługi | 35 |
| Podstawa uzyskania wpisu do BUR | Certyfikat systemu zarządzania jakością wg. ISO 9001:2015 (PN-EN ISO 9001:2015) - w zakresie usług szkoleniowych |

Cel

Cel edukacyjny

Nabycie przez uczestników kompleksowych umiejętności programowania w języku JavaScript, od podstaw składni po zaawansowane techniki programowania asynchronicznego, umożliwiającym samodzielne tworzenie interaktywnych aplikacji webowych i skryptów JavaScript w nowoczesnym standardzie ES6+.

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

| Efekty uczenia się | Kryteria weryfikacji | Metoda walidacji |
|--|--|---|
| Stosuje podstawowe konstrukcje języka JavaScript (zmienne, typy, funkcje, tablice, obiekty). | Uczestnik implementuje funkcję przetwarzającą tablicę obiektów (filtrowanie, mapowanie, sortowanie) z użyciem metod tablicowych ES6 i zwracającą poprawny wynik. | Test teoretyczny z wynikiem generowanym automatycznie |
| Manipuluje DOM i obsługuje zdarzenia przeglądarki. | Uczestnik tworzy interaktywny komponent UI (np. formularz z walidacją, lista z filtrowaniem) wyłącznie przy użyciu JavaScript bez zewnętrznych bibliotek | Test teoretyczny z wynikiem generowanym automatycznie |
| Stosuje programowanie asynchroniczne (Promises, async/await, fetch API). | Uczestnik pobiera dane z publicznego API przy użyciu fetch i async/await, obsługuje błędy i wyświetla dane w DOM bez błędów w konsoli | Test teoretyczny z wynikiem generowanym automatycznie |
| Stosuje zaawansowane koncepcje JavaScript (closures, prototypy, moduły ES6, klasy). | Uczestnik implementuje moduł JS z klasą korzystającą z dziedziczenia i eksportuje go przy użyciu ES6 modules, demonstrując działanie w przeglądarce | Test teoretyczny z wynikiem generowanym automatycznie |

Kwalifikacje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem zawierają opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji i zgodnie z zaplanowanymi metodami walidacji?

TAK

Pytanie 3. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

Program

Wprowadzenie

Krótką historia JavaScript

Środowisko pracy

Edytory kodu

Przeglądarki i inne środowiska uruchomieniowe

Narzędzia debugowania

Składnia JavaScript

Czym są zmienne i jak je deklarować? Kiedy let a kiedy const? Czy używamy jeszcze var?

Komentarze

Typy danych - proste i złożone

Operatory

Instrukcje warunkowe i pętle

Dobre praktyki związane ze składnią i formatowaniem kodu

Funkcje

Czym są funkcje?

Różne rodzaje tworzenia funkcji (definicja funkcji, wyrażenie funkcyjne, funkcja strzałkowa)

Parametry i zwracane wartości

Zakres zmiennych w funkcjach

Dobre praktyki

Obiekty i tablice

Tworzenie obiektów i tablic

Dodawanie i usuwanie elementów

Iterowanie przez obiekty i tablice

Kopiowanie przez wartość i przez referencje

Destrukturyzacja obiektów i tablic

Rest / spread syntax - czyli pobieranie reszty i rozpraszanie tablic i obiektów

Dobre praktyki

DOM - czyli Document Object Model

Czym jest DOM?

Wyszukiwanie elementów DOM

Manipulacja elementami DOM i ich stylami

Dodawanie i usuwanie elementów DOM

Obsługa zdarzeń - czyli interakcji użytkownika takich jak klikanie, najeżdżanie itp.

Dobre praktyki

AJAX - czyli Asynchroniczny JavaScript i XML

Czym jest asynchroniczność ? - oraz kilka słów o tym jak działa JavaScript

Wyjaśnienie, dlaczego AJAX jest używany w tworzeniu interaktywnych aplikacji internetowych

Asynchroniczne żądania do serwera

Pobieranie i wysyłanie danych

Obsługa odpowiedzi serwera i aktualizacja strony

Obsługa błędów AJAX

Dobre praktyki

Obsługa wyjątków

Przegląd różnych typów błędów w JavaScript, takich jak ReferenceError, TypeError, SyntaxError itp.

Bloki try-catch

Rzucanie wyjątków - za pomocą instrukcji - throw

Debugowanie kodu

Unikanie zagrożeń bezpieczeństwa

Dobre praktyki w obsłudze wyjątków

Moduły

Zrozumienie czym są moduły w JavaScript i dlaczego są używane?

Tworzenie, eksportowanie i importowanie modułów

Omówienie zakresu zmiennych w modułach

Sposoby zarządzania zależnościami między modułami.

Generatory

Czym są generatory?

Omówienie podstawowych pojęć

Wyjaśnienie, do czego generatory są używane i jakie problemy rozwiązują.

Składnia generatorów

Zapoznanie się ze składnią deklaracji generatorów za pomocą funkcji function* oraz instrukcji yield.

Praktyczne przykłady definiowania i wywoływania generatorów.

Iteracja przez generatory

Wyjaśnienie, jak iterować przez wartości zwracane przez generator za pomocą pętli for...of.

Praktyczne zastosowanie iteracji przez generatory do przetwarzania danych

Wartości zwracane przez generatory

Omówienie sposobów zwracania wartości za pomocą instrukcji yield.

Wyjaśnienie, jak przekazywać wartości do generatora i je odbierać.

Kontrola przepływu w generatorach

Generatory jako iteratory

Praktyczne przykłady

Map i Weak Map

Czym są mapy? - Omówienie koncepcji map w JavaScript i porównanie ich z tradycyjnymi obiektami.

Tworzenie map - jak tworzyć mapy za pomocą konstruktora Map() oraz dodawanie i usuwanie elementów z mapy.

Operacje na mapach - Zapoznanie się z różnymi operacjami, takimi jak pobieranie wartości, sprawdzanie obecności klucza, iterowanie po elementach itp.

Czym są weak mapy? - Wyjaśnienie koncepcji weak map w JavaScript i różnic między weak mapami a standardowymi mapami.

Tworzenie weak map - jak tworzyć weak mapy za pomocą konstruktora WeakMap() oraz dodawanie i usuwanie z nich elementów.

Praktyczne przykłady zastosowania, takie jak np.: przechowywanie dodatkowych danych bez ryzyka wycieku pamięci.

Zarządzanie danymi w przeglądarce

Czym są Local Storage i Session Storage?

Wyjaśnienie różnic

Omówienie sposobów, w jaki dane są przechowywane w przeglądarce za pomocą tych mechanizmów.

Dostęp do Local Storage i Session Storage

Zapisywanie danych - Praktyczne przykłady zapisywania danych wykorzystując metody setItem()

Pobieranie danych - jak pobierać dane korzystając z metody getItem().

Usuwanie danych - Omówienie sposobów usuwania danych przy użyciu metody removeItem().

Praca z różnymi typami danych

Jak obsługiwać błędy, takie jak przekroczenie limitu pamięci lub brak dostępu.

Omówienie kwestii bezpieczeństwa oraz praktyk, które należy stosować w celu ochrony danych użytkowników.

Praktyczne przykłady.

Dziedziczenie w JavaScript

Wprowadzenie do dziedziczenia

Obiekty i prototypy

Jak tworzyć obiekty za pomocą literalów obiektów oraz konstruktorów obiektów?

Omówienie koncepcji prototypów

Jak działa łańcuch prototypów (prototype chain) w JavaScript?

Dziedziczenie prototypowe

Wyjaśnienie mechanizmu dziedziczenia prototypowego

Jak dziedziczyć właściwości i metody z innego obiektu za pomocą prototypów?

Metoda Object.create()

Funkcje konstruktora i dziedziczenie

Klasy w JavaScript (ES6+)

Wyjaśnienie koncepcji klas w JavaScript

Porównanie klas z tradycyjnym mechanizmem prototypowym.

Dziedziczenie klas

Metoda extends i super

Harmonogram

Liczba pozycji harmonogramu: 16

| Przedmiot / temat | Prowadzący | Data realizacji zajęć | Godzina rozpoczęcia | Godzina zakończenia | Liczba godzin |
|--|-------------|-----------------------|---------------------|---------------------|---------------|
| 1 z 16 Wprowadzenie/Krótką historią JavaScript | Agata Malec | 23-11-2026 | 09:00 | 12:30 | 03:30 |
| 2 z 16 Przerwa Obiadowa | Agata Malec | 23-11-2026 | 12:30 | 13:00 | 00:30 |
| 3 z 16 Środowisko pracy/Składnia JavaScript | Agata Malec | 23-11-2026 | 13:00 | 16:00 | 03:00 |
| 4 z 16 Funkcje/Obiekty i tablice | Agata Malec | 24-11-2026 | 09:00 | 12:30 | 03:30 |
| 5 z 16 Przerwa Obiadowa | Agata Malec | 24-11-2026 | 12:30 | 13:00 | 00:30 |
| 6 z 16 DOM - czyli Document Object Model/AJAX - czyli Asynchroniczny JavaScript i XML | Agata Malec | 24-11-2026 | 13:00 | 16:00 | 03:00 |
| 7 z 16 Obsługa wyjątków | Agata Malec | 25-11-2026 | 09:00 | 12:30 | 03:30 |
| 8 z 16 Przerwa Obiadowa | Agata Malec | 25-11-2026 | 12:30 | 13:00 | 00:30 |
| 9 z 16 Moduły | Agata Malec | 25-11-2026 | 13:00 | 16:00 | 03:00 |
| 10 z 16 Generatory | Agata Malec | 26-11-2026 | 09:00 | 12:30 | 03:30 |
| 11 z 16 Przerwa Obiadowa | Agata Malec | 26-11-2026 | 12:30 | 13:00 | 00:30 |
| 12 z 16 Map i Weak Map | Agata Malec | 26-11-2026 | 13:00 | 16:00 | 03:00 |

| Przedmiot / temat | Prowadzący | Data realizacji zajęć | Godzina rozpoczęcia | Godzina zakończenia | Liczba godzin |
|---|-------------|-----------------------|---------------------|---------------------|---------------|
| 13 z 16 Zarządzanie danymi w przeglądarce | Agata Malec | 27-11-2026 | 09:00 | 12:30 | 03:30 |
| 14 z 16 Przerwa Obiadowa | Agata Malec | 27-11-2026 | 12:30 | 13:00 | 00:30 |
| 15 z 16 Dziedziczenie w JavaScript | Agata Malec | 27-11-2026 | 13:00 | 15:30 | 02:30 |
| 16 z 16 Walidacja | Agata Malec | 27-11-2026 | 15:30 | 16:00 | 00:30 |

Cennik

Cennik

| Rodzaj ceny | Cena |
|---|--------------|
| Koszt przypadający na 1 uczestnika brutto | 3 690,00 PLN |
| Koszt przypadający na 1 uczestnika netto | 3 000,00 PLN |
| Koszt osobogodziny brutto | 105,43 PLN |
| Koszt osobogodziny netto | 85,71 PLN |

Prowadzący

Liczba prowadzących: 1



1 z 1

Agata Malec

Trener posiada wieloletnie doświadczenie w tworzeniu aplikacji webowych w języku JavaScript, przy czym kluczowe kwalifikacje w zakresie programowania JavaScript (ES6+), frameworków frontendowych oraz nowoczesnych technik tworzenia aplikacji webowych zostały zdobyte i są czynnie wykorzystywane w okresie ostatnich 5 lat (od 2021 roku do chwili obecnej). Potwierdzają to zrealizowane projekty developerskie oraz przeprowadzone szkolenia z programowania JavaScript w latach 2022–2026.

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Informacje o materiałach dla uczestników usługi - Uczestnicy otrzymają komplet materiałów PDF. Każdy uczestnik otrzymuje kod dostępu i

dane logowania do platformy ZOOM na 7 dni przed datą rozpoczęcia szkolenia. Dane przesyłane są na adres e-mail podany podczas rejestracji.

Warunki uczestnictwa

Podstawowe zrozumienie podstaw programowania oraz podstawowa znajomość HTML i CSS, umiejętność obsługi komputera.

Informacje dodatkowe

Warunkiem ukończenia szkolenia i otrzymania zaświadczenia jest uzyskanie minimalnej frekwencji na poziomie 80% całkowitego czasu trwania usługi. Obecność uczestnika będzie potwierdzana na podstawie codziennych list obecności lub logów z platformy online.

Warunki techniczne

Uczestnik musi dysponować sprzętem i łączem o parametrach:

- Procesor: min. 4-rdzeniowy (np. Intel i5/i7 lub odpowiednik AMD/M1/M2)
- Pamięć RAM: min. 16 GB
- Dysk: min. 20 GB wolnej przestrzeni
- System operacyjny: Windows 10/11 Pro, Linux lub macOS
- Multimedia: Sprawna kamera internetowa oraz mikrofon (wymagane do komunikacji i weryfikacji obecności)
- Łącze internetowe: Stabilne połączenie o minimalnej prędkości 10 Mbps (download) / 5 Mbps (upload)
- Oprogramowanie: Uprawnienia administratora pozwalające na instalację narzędzi

Kontakt



Biuro Obsługi Klienta

E-mail biuro@jssystem.com

Telefon (+48) 534 506 503