JavaScript i React (Frontend Developer XL) (kod: K-JS-XL)

Opis i cel kursu

JavaScript to obecnie podstawowy język programistyczny w internecie. Działa w każdej przeglądarce, a poza nią także na serwerach i w aplikacjach mobilnych. Jest też często wybierany jako pierwszy język do nauki programowania.

Nasz kurs programowania w JavaScript prezentuje podstawy budowy stron i świata aplikacji internetowych, składnię samego języka JS od zera, komunikację z API, pracę z narzędziami AI i dodawanie funkcji AI do własnej aplikacji – oraz finalnie bibliotekę React, jedną z najpopularniejszych bibliotek JavaScriptu. Node.js i npm poznajemy jako środowisko pracy programisty.

W wariancie XL kontynuujemy w drugiej części kursu: TypeScript, React z TypeScriptem, architektura większej aplikacji SPA, Redux Toolkit, testy i Storybook, wdrożenie aplikacji, Next.js oraz AI w większej aplikacji.

Zapytaj o szczegóły

tel. 22 63 64 164
akademia@alx.pl

Najbliższe terminy

2026-12-10 (Zdalnie)
2026-12-10 (Warszawa)
2026-12-19 (Zdalnie)
2026-12-19 (Warszawa)

Program

1. **Wprowadzenie do programowania internetowego**
   - — Podstawowe pojęcia programowania internetowego: internet, frontend, backend, aplikacja webowa
   - — Czym jest strona internetowa, a czym aplikacja webowa?
   - — Rola HTML, CSS i JavaScript w budowie stron
   - — Podstawowe pojęcia związane z pracą programisty
2. **Narzędzia i środowisko pracy**
   - — Edytory i środowiska deweloperskie, np. VS Code, Cursor
   - — Panel deweloperski w przeglądarkach internetowych: Chrome, Firefox
   - — Inspekcja i debugowanie stron oraz skryptów
   - — Podstawy pracy z plikami projektu
   - — Uruchamianie prostej strony lokalnie w przeglądarce
   - — Ćwiczenie I: przygotowanie środowiska pracy i uruchomienie pierwszej strony.
3. **Podstawy budowy stron - HTML, CSS i model DOM**
   - — Co to jest znacznik/tag?
   - — Organizacja elementów strony
   - — Semantyka HTML5
   - — Czym jest DOM?
   - — Selektory: element, class, id
   - — Podstawowe właściwości CSS
   - — Model pudełkowy: margin, border, padding, content
   - — Podstawy responsywności stron
   - — Podstawy accessibility i WCAG: semantyka, teksty alternatywne, etykiety formularzy
   - — Podstawy SEO: struktura nagłówków, tytuł strony, opis, znaczniki semantyczne
   - — Podstawowe tips and tricks HTML5/CSS
   - — Ćwiczenie II: stworzenie prostej, responsywnej strony HTML/CSS.
4. **Narzędzia AI w pracy programisty**
   - — Wsparcie AI w pracy z kodem: autouzupełnianie, refaktoryzacja, generowanie fragmentów HTML/CSS
   - — Praca z istniejącym kodem w Cursorze, GitHub Copilot lub podobnym narzędziu

- Wykorzystanie modeli językowych do rozwiązywania zadań i wyjaśniania kodu, np. ChatGPT, Claude lub Gemini
- Jak pisać dobre prompty techniczne
- Jak prosić AI o poprawę struktury HTML, uporządkowanie CSS lub znalezienie błędu
- Jak weryfikować kod wygenerowany przez AI
- Granice wykorzystania AI: bezpieczeństwo, prywatność, prawa autorskie
- Ćwiczenie III: ulepszanie prostej strony HTML/CSS z pomocą AI.

5. **JavaScript - fundamenty**
   - Czym jest ECMAScript i wersje JavaScript - krótkie omówienie
   - Zmienne, obowiązkowe i rekomendowane praktyki nazywania zmiennych: let, const
   - Komentowanie i dokumentowanie kodu
   - Prymitywne typy danych, konwersja typów
   - Operatory: arytmetyczne, przypisania, inkrementacji, dekrementacji, relacyjne, logiczne
   - Interakcja z użytkownikiem
   - Instrukcje warunkowe, instrukcja switch
   - Pętle
   - Funkcje, w tym funkcje strzałkowe
   - Obiekty, metody, klasy i podstawy OOP
   - Tablice
   - Mapy i zbiory - wprowadzenie
   - Moduły: import i export
   - Czym różni się JavaScript od TypeScript - krótkie omówienie
   - Podsumowanie: zadania i pytania sprawdzające
   - Ćwiczenie IV: zadania praktyczne z podstaw JavaScript.
6. **JavaScript praktycznie - VanillaJS**
   - Komunikacja pomiędzy JavaScript, HTML i CSS
   - Obsługa zdarzeń: click, submit, input, addEventListener
   - Animacje i manipulacja stroną DOM
   - Tworzenie modali, czyli okien dialogowych
   - Przechowywanie danych na stronie: localStorage, cookies
   - Walidacja formularza
   - Prosta aplikacja SPA - HTML + CSS + JS
   - Ćwiczenie V: stworzenie prostej aplikacji SPA w HTML, CSS i JavaScript.
7. **Jak frontend komunikuje się z backendem**
   - Podstawy protokołu HTTP - jak działają strony?
   - Asynchroniczność: Promise i async/await
   - Technologia AJAX i Fetch API
   - JSON jako format wymiany danych
   - Obsługa błędów podczas komunikacji z API
   - Przygotowanie środowiska do pracy - Node.js
   - Jak instalować moduły? Przydatne polecenia npm
   - Ćwiczenie VI: pobieranie i wyświetlanie danych z API.
8. **Projekt praktyczny - frontend do prostego bloga**
   - REST - komunikacja z backendem
   - Dynamiczne ładowanie treści
   - Dodawanie nowych postów
   - Wyświetlanie postów
   - Edycja postów
   - Dodawanie i usuwanie komentarzy
   - Ćwiczenie VII: stworzenie frontendu do prostego bloga.
9. **AI w aplikacjach JavaScript - integracja z modelami językowymi**

Zapytaj o szczegóły
tel. 22 63 64 164
akademia@alx.pl

Najbliższe terminy
2026-12-10 (Zdalnie)
2026-12-10 (Warszawa)
2026-12-19 (Zdalnie)
2026-12-19 (Warszawa)

- Jak działają LLM API: klucze, endpointy, format żądań
- Bezpieczne korzystanie z kluczy API
- Łączenie z API wybranego dostawcy modeli, np. OpenAI
- Wysyłanie wiadomości i odbieranie odpowiedzi
- Streaming odpowiedzi: ReadableStream, efekt pisania w UI
- Przechowywanie historii konwersacji w JS - tablica messages
- Dodanie funkcji AI do aplikacji blogowej
- Ćwiczenie VIII: dodanie funkcji AI do bloga, np. generowanie tytułu, podsumowania lub sugestii treści posta.

10. **Wprowadzenie do Reacta**
    - Jak zacząć?
    - Tworzenie nowego projektu
    - Struktura projektu
    - Jak działają komponenty?
    - Co składa się na komponent?
    - Czym jest JSX?
    - Props i stan komponentu (hook useState)
    - Obsługa zdarzeń w komponentach
    - Routing - wprowadzenie
    - Tworzenie prostej aplikacji w React
    - Ćwiczenie IX: stworzenie prostej aplikacji w React.
11. **Rynek, aktualne oczekiwania i rola AI we frontendzie**
    - Obecne oczekiwania wobec front-end developera
    - Co jest dziś standardem: JavaScript, TypeScript, React, Git, praca z API, testy, podstawy deployu
    - Co się zdeaktualizowało lub zeszło na dalszy plan
    - Co jest aktualnie na topie: TypeScript, React, Next.js, Vite, komponentowe UI, testowanie, Storybook, AI-assisted development
    - Rola AI w pracy developera
    - Ograniczenia AI: halucynacje, bezpieczeństwo, błędy w kodzie, konieczność review
    - Omówienie projektu końcowego kursu
    - Ćwiczenie I: analiza przykładowej oferty pracy frontendowej i mapowanie wymagań na tematy kursu.
12. **Node.js, npm, edytor z AI i workflow projektowy**
    - Node.js jako środowisko narzędziowe dla frontendu
    - Menedżer npm, package.json, skrypty projektowe
    - Instalacja zależności
    - Vite jako środowisko projektowe
    - Zmienne środowiskowe i pliki .env
    - Konfiguracja edytora z asystentem AI (np. Cursor) do pracy z projektem
    - Podstawowa struktura projektu frontendowego
    - Konfiguracja ESLint i Prettier
    - Ćwiczenie II: utworzenie projektu React/Vite, konfiguracja edytora z AI i podstawowego workflow developerskiego.
13. **Git, GitHub i praca projektowa**
    - Instalacja i konfiguracja Git
    - Repozytorium lokalne i zdalne
    - Polecenia commit, branch, merge, pull, push
    - Praca na branchach
    - Pull request - idea i uproszczony workflow
    - Rozwiązywanie prostych konfliktów przy łączeniu gałęzi
    - README projektu
    - Dobre praktyki commitowania

Zapytaj o szczegóły
tel. 22 63 64 164
akademia@alx.pl

Najbliższe terminy
2026-12-10 (Zdalnie)
2026-12-10 (Warszawa)
2026-12-19 (Zdalnie)
2026-12-19 (Warszawa)

— Ćwiczenie III: założenie repozytorium, pierwszy branch, prosty pull request i rozwiązanie konfliktu.

14. **Zaawansowany JavaScript**
    - — Obsługa błędów: try/catch, throw, własne błędy
    - — Callbacki i callback hell
    - — Promises
    - — Składnia async/await
    - — Promise.all, Promise.allSettled
    - — Event loop - koncepcyjnie i praktycznie
    - — Mikro- i makro-zadania
    - — Moduły ES: import/export
    - — Organizacja kodu w modułach
    - — Praktyczne wzorce pracy z kodem asynchronicznym
    - — Ćwiczenie IV: refaktoryzacja kodu z callbacków do Promise i async/await.
15. **TypeScript w aplikacjach frontendowych**
    - — Po co TypeScript w większych aplikacjach frontendowych
    - — Różnice między JavaScript i TypeScript
    - — Typy podstawowe
    - — Typowanie zmiennych
    - — Typowanie funkcji: parametry i wartości zwracane
    - — Typowanie tablic i obiektów
    - — Typy: type vs interface
    - — Union types
    - — Literal types
    - — Optional properties
    - — Type narrowing - zawężanie typów
    - — Type guards
    - — Generics - podstawy
    - — Utility types: Pick, Omit, Partial, Record
    - — Typowanie danych z API
    - — Typowanie propsów komponentów React
    - — Dobre praktyki pracy z TypeScriptem w projekcie frontendowym
    - — Ćwiczenie V: przygotowanie typów dla modeli danych aplikacji oraz typowanie funkcji i komponentów.
16. **Komunikacja client-server i bezpieczeństwo**
    - — REST API w praktyce
    - — Metody HTTP, statusy, nagłówki
    - — JSON i struktury odpowiedzi API
    - — Obsługa błędów API
    - — Uwierzytelnianie i autoryzacja - koncepcyjnie
    - — Tokeny, sesje, JWT - podstawy
    - — Przechowywanie tokenów - ryzyka i dobre praktyki
    - — CORS - koncepcyjnie
    - — Mock backend / fake API
    - — Krótkie omówienie GraphQL i WebSockets: kiedy się ich używa, bez głębokiego wdrożenia
    - — Ćwiczenie VI: połączenie aplikacji z mock API i przygotowanie prostego przepływu logowania/autoryzacji.
17. **React z TypeScriptem - komponenty, state i hooks**
    - — Myślenie komponentowe
    - — JSX i renderowanie UI
    - — Props i typowanie propsów
    - — Renderowanie warunkowe
    - — Renderowanie list

Zapytaj o szczegóły
tel. 22 63 64 164
akademia@alx.pl

Najbliższe terminy
2026-12-10 (Zdalnie)
2026-12-10 (Warszawa)
2026-12-19 (Zdalnie)
2026-12-19 (Warszawa)

- Obsługa zdarzeń
- State komponentu
- Aktualizacja obiektów i tablic w stanie
- Formularze kontrolowane
- Lifting state up
- Kompozycja komponentów
- Context API - kiedy wystarczy, a kiedy lepszy jest Redux Toolkit
- Hook useEffect i synchronizacja z zewnętrznymi systemami
- Hook useRef
- Custom hooks
- Organizacja komponentów w większej aplikacji
- Ćwiczenie VII: zbudowanie bazowych komponentów aplikacji w React + TypeScript.

18. **Analiza wymagań, Atomic Design i dekompozycja UI**
    - Definiowanie wymagań aplikacji
    - User stories
    - Analiza widoków i przepływów użytkownika
    - Podstawy pracy z makietą
    - Przejście od makiety do komponentów
    - Wprowadzenie do Atomic Design
    - Atomy, molekuły, organizmy, template'y/widoki
    - Dekompozycja komponentowa na podstawie mockupów
    - Uwzględnianie accessibility i WCAG na etapie projektowania UI
    - Przygotowanie backlogu / listy zadań na bazie user stories
    - Ćwiczenie VIII: analiza wymagań, praca z makietą i dekompozycja UI zgodnie z Atomic Design.
19. **Architektura aplikacji SPA**
    - Projektowanie struktury folderów
    - Podział aplikacji na warstwy: UI, logika, API, routing, stan
    - Routing aplikacji
    - Organizacja komponentów, widoków i modułów
    - Separacja logiki biznesowej od warstwy prezentacyjnej
    - Organizacja komunikacji z API
    - Obsługa stanów aplikacji: loading, error, success
    - Przygotowanie aplikacji pod dalszą rozbudowę
    - Planowanie implementacji większej funkcjonalności
    - Ćwiczenie IX: zaprojektowanie architektury aplikacji SPA i przygotowanie struktury projektu.
20. **Stan aplikacji, Redux Toolkit i integracja z API**
    - Local state vs global state
    - Kiedy globalny store ma sens
    - Redux Toolkit - idea i podstawy
    - Store
    - Slices
    - Actions i reducers
    - Typowanie Redux Toolkit w TypeScript
    - Asynchroniczna komunikacja z API
    - Loading/error/success state
    - Autoryzowane zapytania do API
    - Organizacja logiki aplikacji
    - Ćwiczenie X: dodanie globalnego stanu i integracji z API do aplikacji SPA.
21. **AI w zaawansowanej aplikacji frontendowej**
    - Jak działają LLM API: klucze, endpointy, format żądań
    - Bezpieczne korzystanie z kluczy API

Zapytaj o szczegóły
tel. 22 63 64 164
akademia@alx.pl

Najbliższe terminy
2026-12-10 (Zdalnie)
2026-12-10 (Warszawa)
2026-12-19 (Zdalnie)
2026-12-19 (Warszawa)

- Backend/proxy/serverless jako pośrednik między frontendem a LLM API
- Łączenie z API wybranego dostawcy modeli, np. OpenAI
- Wysyłanie wiadomości i odbieranie odpowiedzi
- Przekazywanie kontekstu aplikacji do modelu
- Streaming odpowiedzi: ReadableStream, efekt pisania w UI
- Przechowywanie historii konwersacji w JS - tablica messages
- Obsługa stanów: loading, error, success
- Dodanie funkcji AI do budowanej aplikacji SPA
- Ćwiczenie XI: dodanie funkcji AI do aplikacji, np. generatora opisu, podsumowania, sugestii lub asystenta kontekstowego.

22. **Storybook i dokumentowanie UI**
    - Po co dokumentować komponenty wizualnie
    - Wprowadzenie do Storybooka
    - Stories dla komponentów
    - Warianty komponentów
    - Dokumentowanie propsów i stanów komponentów
    - Komponenty w izolacji
    - Dokumentowanie komponentów powstałych dla funkcji AI
    - Storybook jako narzędzie komunikacji w zespole
    - Ćwiczenie XII: przygotowanie Storybooka dla wybranych komponentów aplikacji.
23. **Testowanie aplikacji frontendowej**
    - Po co testować frontend
    - Testy jednostkowe logiki JS/TS
    - Testy komponentów React
    - React Testing Library - podstawy
    - Mockowanie API
    - Testowanie loading/error/success state
    - Testowanie wybranej logiki funkcji AI
    - Testy E2E - koncepcyjnie
    - Playwright/Cypress - krótkie omówienie
    - Co testować, a czego nie testować
    - Ćwiczenie XIII: napisanie podstawowych testów dla logiki, komponentu i wybranego przepływu użytkownika.
24. **Wdrożenie aplikacji na Netlify**
    - Build aplikacji
    - Zmienne środowiskowe na produkcji
    - Deploy aplikacji SPA na Netlify
    - Konfiguracja projektu po stronie Netlify
    - Obsługa routingu SPA po wdrożeniu
    - Typowe problemy po deployu
    - Sprawdzenie działania aplikacji produkcyjnej
    - Ćwiczenie XIV: wdrożenie aplikacji na Netlify.
25. **Next.js - SSR, hydratacja i prosty projekt**
    - Czym Next.js różni się od klasycznej aplikacji React SPA
    - Routing w Next.js
    - Client-side rendering vs server-side rendering
    - SSR
    - SSG - krótkie omówienie
    - Hydratacja - o co chodzi i dlaczego ma znaczenie
    - Server Components - koncepcyjnie
    - Client Components
    - Fetchowanie danych w Next.js
    - Layouty

Zapytaj o szczegóły
tel. 22 63 64 164
akademia@alx.pl

Najbliższe terminy
2026-12-10 (Zdalnie)
2026-12-10 (Warszawa)
2026-12-19 (Zdalnie)
2026-12-19 (Warszawa)

- SEO w aplikacjach renderowanych po stronie serwera
- Metadane, struktura treści i wpływ renderowania na indeksowanie
- Jamstack i headless CMS - krótkie omówienie
- Prosty projekt w Next.js
- Kiedy wybrać klasyczne SPA, a kiedy Next.js
- Ćwiczenie XV: stworzenie prostego projektu Next.js z routingiem i pobieraniem danych.

Przeznaczenie i wymagania

Kurs jest przeznaczony dla osób początkujących, które do tej pory nie programowały w żadnym języku, ale zdecydowanych na dłuższy wysiłek i przejście całej ścieżki kursowej od podstaw do praktycznych zastosowań.

Mile widziana może być - ale nie jest wymagana ponieważ tematy te zostają omówione na początku kursu - znajomość HTML i CSS na podstawowym poziomie.

Część kursantów przychodzi na niniejszy kurs również po ukończeniu kursu Web Designer - który ma przeznaczenie bardziej "kreatywne" (tworzenie treści, grafika...) niż programistyczne. Natomiast wcześniejsze ukończenie kursu Web Designer nie jest w żaden sposób wymagane. Są to różne kursy, tworzone z myślą o różnych zastosowaniach. Web Designer - to kierunek bardziej "kreatywny", Frontend Developer - kierunek "programistyczny".

W razie wszelkich wątpliwości, pytań odnośnie tego który kurs lepiej wybrać - zapraszamy serdecznie do kontaktu z naszym biurem.

Certyfikaty

Uczestnicy szkolenia otrzymują imienne certyfikaty sygnowane przez ALX.

Zapytaj o szczegóły

tel. 22 63 64 164
akademia@alx.pl

Najbliższe terminy

2026-12-10 (Zdalnie)
2026-12-10 (Warszawa)
2026-12-19 (Zdalnie)
2026-12-19 (Warszawa)