PROGRAM SZKOLENIA JĘZYKI

Szkolenie JavaScript od podstaw

Nauczysz się JavaScript od zera po async/await i fetch i zbudujesz aplikację, która pobiera dane z API.

CZAS TRWANIA	POZIOM	FORMA
4 dni	**od podstaw**	**warsztat dla zespołu, zdalnie lub stacjonarnie**

Zaczynasz przygodę z JavaScript, a kursy wideo zostawiają Cię z kodem przepisanym z ekranu, którego nie potrafisz samodzielnie zmienić? Na szkoleniu poznasz fundamenty języka w takiej kolejności, żeby każdy kolejny temat opierał się na poprzednim, i od pierwszej godziny piszesz kod sam.

Uczysz się nowoczesnego JavaScript, takiego, jaki spotkasz dziś w projektach: `let` i `const`, metody tablic, optional chaining, klasy, moduły ES oraz `async` i `await`. Poznasz też mechanizmy, na których najczęściej potykają się początkujący: `this`, domknięcia i kolejność wykonywania kodu asynchronicznego. Na koniec łączysz wszystko w aplikację, która pobiera dane z API i działa w przeglądarce.

PROGRAM

4 dni

- Pierwsze kroki z JavaScript
  - Czym jest JavaScript i ECMAScript, gdzie działa kod
  - Konsola przeglądarki i narzędzia deweloperskie
  - Terminal: poruszanie się po katalogach i uruchamianie komend, `node` i `npm` z wiersza poleceń
  - Uruchamianie skryptu w przeglądarce i w Node.js
  - npm: instalacja paczek i serwer deweloperski do projektu
- Zmienne, typy i operatory
  - Tworzenie zmiennych: `let`, `const` i starszy `var`
  - Zakres blokowy i funkcyjny, hoisting i strefa martwa (TDZ)
  - Typy proste i obiekty, operator `typeof`
  - Mutowalne i niemutowalne wartości
  - Operatory arytmetyczne, logiczne i porównania: `==` a `===`
  - Konwersja typów i wartości fałszywe (falsy values)
- Instrukcje warunkowe i pętle
  - Instrukcje `if` i `switch` oraz operator `?:`
  - Pętle `for`, `while`, `do...while`
  - Pętle `for...of` i `for...in`: czym się różnią
  - Instrukcje `break` i `continue`

- Funkcje
  - Deklaracja, wyrażenie funkcyjne i funkcja strzałkowa
  - Parametry domyślne i parametr `...rest`
  - Wartość zwracana i czyste funkcje
  - Wywołania zwrotne (callback) i funkcje wyższego rzędu
- Tablice
  - Dodawanie, usuwanie i wyszukiwanie: `push`, `includes`, `indexOf`
  - Metody `map`, `filter`, `reduce`, `find`, `some`, `every`
  - Sortowanie bez mutacji tablicy: `toSorted`
  - Grupowanie danych: `Object.groupBy`
- Obiekty
  - Literał obiektu, właściwości i metody
  - Destrukturyzacja oraz składnia rest i spread (`...`)
  - Optional chaining `?.` i operator `??`
  - `Object.keys`, `Object.values`, `Object.entries`
  - Kopia płytka i głęboka: spread a `structuredClone` z API przeglądarki
- Funkcje od środka
  - Domknięcia (closures) i prywatny stan funkcji
  - Słowo kluczowe `this` w metodach obiektu
  - `this` w funkcji strzałkowej
- Tekst, liczby, daty i JSON
  - Najczęstsze metody tekstu: `slice`, `split`, `trim`, `replaceAll`
  - Szablony literałowe (template literals) i łączenie tekstu
  - `Math` i liczby: zaokrąglanie, losowanie, `Number.isNaN`
  - `JSON.parse` i `JSON.stringify`
  - Daty: obiekt `Date` i jego pułapki
- Obsługa błędów
  - `try`, `catch`, `finally` i `throw`
  - Typy błędów: `Error`, `TypeError`, `RangeError`, `SyntaxError`
  - Czytanie komunikatów błędów i debugger w przeglądarce
- Klasy
  - Konstruktor, metody, gettery i settery
  - Dziedziczenie: `extends` i `super`
  - Kiedy klasa, a kiedy zwykły obiekt i funkcje
- Moduły ES
  - `import` i `export`: nazwane i domyślne
  - Moduły w przeglądarce: `<script type="module">`
  - Tryb ścisły (strict mode) domyślnie w modułach
- JavaScript w przeglądarce
  - DOM: pobieranie, tworzenie i usuwanie elementów
  - Obsługa zdarzeń: `addEventListener`, propagacja i delegacja
  - Formularze: `FormData` i walidacja wbudowana w przeglądarkę
  - Obiekty przeglądarki: `location`, `navigator`, `URL`

- Asynchroniczność i komunikacja z serwerem
  - Pętla zdarzeń (event loop): dlaczego kod nie wykonuje się po kolei
  - Zegary `setTimeout` i `setInterval`
  - Od callbacków do `Promise`, potem `async` i `await`
  - Fetch API: żądania, nagłówki, JSON i obsługa błędów HTTP i sieci
  - Kilka żądań naraz: `Promise.all` i `Promise.allSettled`
- Projekt: lista zadań
  - Dodawanie, usuwanie i filtrowanie zadań
  - Pobieranie i zapisywanie danych przez API
  - Stan ładowania i błędu w interfejsie
  - Zapis danych w `localStorage`
  - Podział kodu na moduły i małe funkcje

DLA KOGO?

- Dla osób, które napisały pierwsze proste programy i zaczynają naukę JavaScript.
- Dla programistów innych języków, którzy przechodzą na frontend.
- Dla testerów i osób technicznych, które chcą czytać i poprawiać kod JavaScript.

Po tym szkoleniu możesz przyjść na kolejny poziom: JavaScript zaawansowany.

CO MUSISZ UMIEĆ?

- Obsługiwać komputer, przeglądarkę i edytor kodu.
- Mieć za sobą pierwsze proste programy w dowolnym języku: zmienna, warunek, pętla.
- Znać podstawy HTML: znaczniki, atrybuty, formularze.

CZEGO SIĘ NAUCZYSZ?

- Poruszysz się po katalogach w terminalu i uruchomisz z niego `node` i `npm`.
- Uruchomisz kod JavaScript w przeglądarce i w Node.js, a paczki zainstalujesz przez npm.
- Rozróżnisz typy danych i unikniesz pułapek konwersji typów.
- Użyjesz `let` i `const` i zrozumiesz, gdzie widać zmienną.
- Napiszesz warunki, pętle i funkcje, w tym funkcje strzałkowe.
- Przewidzisz wartość `this` i świadomie użyjesz domknięć.
- Przetworzysz dane metodami `map`, `filter`, `reduce` i `find`.
- Bezpiecznie odczytasz zagnieżdżone dane przez `?.` i `??`.
- Obsłużysz daty, liczby i JSON wbudowanymi obiektami języka.
- Obsłużysz błędy przez `try` i `catch` i znajdziesz je debuggerem.
- Zaprojektujesz prostą klasę i podzielisz kod na moduły ES.
- Obsłużysz zdarzenia i formularze w DOM, także przez delegację.
- Zrozumiesz pętlę zdarzeń i napiszesz kod asynchroniczny z `async` i `await`.

- Pobierzesz dane z API przez `fetch` i obsłużysz błędy sieci.
- Zbudujesz aplikację, która reaguje na działania użytkownika i pobiera dane z serwera.

OPINIE UCZESTNIKÓW

> *„Gruntownie, krok po kroku stworzyliśmy aplikację sklepu internetowego, po drodze pisząc testy jednostkowe, e2e, web komponenty, autorski backend oparty na Node.js + Express.js, jak również wykorzystaliśmy narzędzia CI."*
>
> Łukasz, JavaScript Developer (WarsawJS MasterClass)

> *„The whole team was able to learn from the mini-workshops and presentations he regularly shared with us, covering advanced Git tricks, React, JavaScript and more."*
>
> Paul Ngei, współpracownik z zespołu w Box (rekomendacja na LinkedIn)

TRENER

Piotr Kowalski

W branży IT od 2008 roku. Od 2014 jest mentorem dla wielu developerów. Jako trener spędził na sali szkoleniowej ponad 1000 godzin, a na LinkedIn ma 52 rekomendacje. Każde jego szkolenie to dawka solidnej wiedzy oraz humoru.

Więcej: piecioshka.pl/piotr-kowalski

ZORGANIZUJMY TO SZKOLENIE W TWOIM ZESPOLE

Program dopasuję do Waszego projektu i poziomu grupy. Napisz, z czym pracujecie, a odpowiem z propozycją terminu i zakresu.

KONTAKT **piecioshka.pl/kontakt**
E-MAIL **piecioshka+blog [małpka] gmail.com**
SZKOLENIE **piecioshka.pl/szkolenia/javascript/**