



REACT - ALE O CO CHODZI?

KURS REACT OD PODSTAW

DLA CIEBIE?

Czy warto uczyć się React?

Tak, jeśli wiążesz swoją przyszłość z programowaniem, np.

- [Front-end developer](#) w firmie (opanować dobrze jeden z trzech frameworków - Angular, Vue, React)
- [Freelancer](#) - praca zdalna przy różnych projektach (SPA)
- [Własny startup](#) - React nie pozwoli Ci zbudować pełnego projektu (nie zajmujemy się tu backendem), ale jest jedną z kluczowych technologii i pozwoli Ci zbudować interfejs aplikacji (także aplikacji mobilnej, jeśli skorzystasz z React Native)

Czy warto uczyć się React?

Tak, jeśli wiążesz swoją przyszłość z programowaniem, np.

- [Front-end developer](#) w firmie (opanować dobrze jeden z trzech frameworków - Angular, Vue, React)
- [Freelancer](#) - praca zdalna przy różnych projektach (SPA)
- [Własny startup](#) - React nie pozwoli Ci zbudować pełnego projektu (nie zajmujemy się tu backendem), ale jest jedną z kluczowych technologii i pozwoli Ci zbudować interfejs aplikacji (także aplikacji mobilnej, jeśli skorzystasz z React Native)

React otwiera nowe możliwości zarówno jeśli chodzi o pracę jak i własne projekty.

Co musisz umieć by rozpocząć naukę

HTML - Choć nie piszemy prawie w ogóle w HTML, to tak naprawdę nasza składnia jest niemal identyczna z HTML.

CSS - Nie będziemy poświęcać mu wiele czasu, kiedy jednak będziemy go używać, nie będę mówił, czym jest właściwość w CSS czy jak deklaruje się klasę w CSS.

JavaScript - React jest biblioteką JS, wszystko, co piszemy piszemy w JS, choć nawet dla osób znających JS na początku może być trudno zrozumieć nie tyle składnię (choć początkowo też), co sposób działania.

Twój poziom JavaScript ?

m.in.:

arrow function,

class i tworzenie instancji klasy

mechanizm this

metody do pracy z tablicami (map, forEach, filter)

Jeśli nie wiesz, o co tutaj chodzi, to przed nauką React musisz jeszcze poświęcić trochę czasu na naukę JavaScript.

- skorzystać z mojego kursu JS na Udemy lub
- wybrać inne kursy/materiały, by nauczyć się JS.
- Będzie szybka powtórka!

Jaki sposób nauki lubisz?

Kurs od podstaw, tłumaczone dokładnie różne zagadnienia, to nie jest kurs dla osób posiadających już praktykę w pracy w React, to jest kurs dla kompletnie początkujących w React.

Czasami powtarzam tematy - coś w kodzie, potem w prezentacji, znowu w kodzie.

Czasami zagłębiam się bardziej.

Czasami robię dygresję.

Czasami powiem niby śmieszną rzecz (ale tak naprawdę nieśmieszną).

PROJEKT "Od podstaw do pierwszej pracy..."

Kurs "React od podstaw" jest częścią większego projektu edukacyjnego. Celem jest dać Ci argumenty przy szukaniu pracy jako junior front-end developer.

Jest to 4. z 6 etapów PROJEKTU. Więcej o nim przeczytasz na <https://websamuraj.pl>

Każdy kurs jest jednak osobną całością. Nie musisz przejść poprzednich czy robić kolejnych!

REACT - CO W KURSIE

Czego będziesz się uczyć

CO PODCZAS TEGO KURSU

REACT - TEORIA

REACT - PRAKTYKA

CREATE-REACT-APP (Webpack, Babel, moduły)

REACT ROUTER

GIT

Pierwszy krok

Potraktuj ten kurs jako pierwszy etap do poznania React.

Praktyka, cierpliwość, praktyka, cierpliwość.

CZYM JEST REACT

Biblioteka JavaScript

React to biblioteka JavaScript do budowania **interfejsów użytkownika**, czyli tej części aplikacji, którą użytkownik widzi i za pomocą której komunikuje się (wchodzi w interakcję) z aplikacją.

Interfejs użytkownika

Możesz potraktować interfejs szerzej, jako stronę internetową (wyświetlaną w przeglądarce, bez backendu) Do stworzenia takiego interfejsu możesz też więc wykorzystać HTML, CSS, czysty JS oraz JQ. Jednak **za pomocą React** możesz zrobić to inaczej.

Dynamiczny i interaktywny UI (User Interface)

React służy przede wszystkim do tworzenia dynamicznych i interaktywnych (reaktywnych) interfejsów (aplikacji/stron internetowych).

Jego głównym celem nie jest tworzenie landing page czy statycznych stron internetowych składających się z kilku podstron (choć nic nie stoi na przeszkodzie, by je zrobić). React to przede wszystkim **Single Page Application** (dynamiczne zmiany wywołane przez zmianę stanu, interakcję z użytkownikiem, bez przeładowywania strony (jeden plik HTML)).

Architektura React - komponenty

Interfejs zbudowany za pomocą React składa się z **komponentów**.

Komponenty tworzą **strukturę** i **logikę** interfejsu.

Architektura React - komponenty i stan

Aplikacja jako metafora mieszkania.

Są drzwi (komponent drzwi), są meble (różne **komponenty**, przede wszystkim struktura, bez zachowań), są pokoje (komponenty nadrzędne względem mebli i urządzeń), jest światło (jak włączmy światło, to w całym pokoju robi się jasno, **zmiana stanu** w komponencie, włącznik wpłynie na komponent pokój).

Mieszkanie składa się z wielu komponentów, podobnie jak aplikacja.

Architektura React - zmiana stanu

Jeden komponent może zmienić stan i strukturę w innym komponencie.

Do mieszkania można dodawać nowe rzeczy, np. klimatyzacja (po włączeniu **zmienia się stan** w pokoju, np. temperatura, zużycie prądu). Czyli akcja w jednym komponencie może wpływać na inne komponenty.

Architektura React - komponent główny i zagnieżdżenie

Mieszkania można potraktować jako **komponent główny**, w którym znajdują się inne komponenty, np. drzwi, pokoje. W pokoju znajdują się kolejne komponenty, np. szafa. W szafie kolejne, np. ubrania.

Komponent

Interfejs użytkownika (widok) tworzą w React komponenty.

Komponent powinien być mały, elastyczny (edytowalny).

Komponent powinien mieć swoje znaczenie (struktura, czasami także logika).

Przykłady komponentu: przycisk, input, header, tytuł w artykule, avatar, pozycja menu (i samo menu jako komponent nadrzędny).

React - połączenie logiki i struktury

Tradycyjne podejście

Struktura - HTML

Logika - JS i API DOM (getElementById, addEventListener, innerHTML itd.)

React łączy strukturę z logiką interfejsu: obsługą zdarzeń, zmianami stanu w czasie, sposobem przygotowania danych do wyświetlania. Zdaniem twórców React oddzielenie znaczników i logiki jest sztuczne. **React łączy elementy widoku z logiką za pomocą komponentów.**

React - projekt open source

Stworzony przez Facebooka w 2013 roku.

Możesz korzystać w swoich projektach.

Wielka społeczność i mnóstwo komercyjnych projektów.

Wersja 16.6 - cały czas rozwijana technologia.

Potrzebuję React?

A nie starczy mi jQuery czy
po prostu JavaScript bez
żadnej biblioteki?

A może jQuery albo po prostu JavaScript (DOM)

To nie jest tak, że łączenie logiki (zmiana stanów) ze strukturą nie jest możliwe w JS/JQ. Jest możliwe, ale nie jest to optymalne. Poza tym nawet coś, co działa, jest wypierane przez coś, co działa lepiej.

JQ/JS (DOM) przede wszystkim pozwalały manipulować poszczególnymi elementami DOM. Czym większa aplikacja, tym trudniejsze jest zarządzanie nią.

React wprowadza coś pomiędzy element a zdarzenie. Wprowadza stan (state). To stan zarządza aplikacją (zmianami w widoku). Jak się przekonasz, ta koncepcja naprawdę jest zdumiewająco dobra (dla kogoś, kto bezpośrednio manipulował DOM za pomocą JQ/JS).

React - nowe podejście

React to nowy sposób na tworzenie interfejsów. Jesteśmy przyzwyczajeni do tradycyjnego podejścia - JavaScript i jQuery do manipulacji DOM. React? Po prostu otwórz się na nowe, z czasem zobaczysz korzyści i to polubisz.

React oferuje Ci przede wszystkim nowe podejście, nowy paradygmat tworzenia interfejsów oparty na komponencie, który dba i o widok i logikę. Tworzysz szybciej, bardziej przejrzysty, czysty kod, kod bezpieczniejszy, łatwiejszy w rozwoju.

React - nowe podejście

Zapomnij (tak w przenośni oczywiście) w React od `document.querySelector`, `addEventListener`, `classList` itp. Nie potrzebujesz ani JS ani JQ do manipulowania DOM.

GŁÓWNE ZAGADNIENIA

Co musisz zrozumieć w
pierwszej kolejności

ISTOTNE ELEMENTY, BY ZROZUMIEĆ

JSX - składnia używana w React, przypomina bardzo HTML

komponenty - stanowe (komponent klasowy) i bezstanowe (komponent funkcyjny)

state - stan

props - właściwości

Renderowanie

Element React a element DOM. Virtual DOM a DOM.

ES6 i JSX nie są wymagane, ale...

ISTOTNE ELEMENTY, BY ZROZUMIEĆ

Najpierw te rzeczy zobaczymy w małych projektach, a potem dokładniej je omówimy w prezentacji i przećwiczymy w kolejnych projektach.

REACT - IDE

VSC (edytor kodu, terminal)

Node.js

VISUAL STUDIO CODE

Na początek...

Rozszerzenia:

- Live Server
- i wiele innych... (np. Prettier - Code formatter, snippets)

Zmiany w ustawieniach

- Format on Save (true)

VISUAL STUDIO CODE

Zintegrowany terminal (wierz polecen, CLI*) - za jego pomocą będziemy mogli instalować różne biblioteki, rozszerzenia, aplikacje. Będziemy też mogli uruchomić specjalne środowisko dla React.

Możemy też uruchomić taki interpreter poleceń bezpośrednio w naszym systemie (np. uruchamiając `cmd` w Windows), ale w VSC jest to zdecydowanie wygodniejsze.

*CLI - Command Line Interpreter

NODE.JS

Środowisko uruchomieniowe dla JS, które możemy zainstalować na naszym komputerze. Wraz z nim dostarczony jest menadżer pakietów (npm - node package manager).

Będziemy mogli za pomocą jednej komendy stworzyć bardzo przydatne środowisko do tworzenia aplikacji w React (zawierające już początkową minimalną aplikację jako stan początkowy).

```
npx create-react-app my-app
```


Zainstalujmy...

React - start

Czego potrzebujemy

Biblioteka React

Tworzenie elementów React, praca z komponentami.

Obiekt React w którym otrzymujemy różne metody.

Biblioteka React DOM

Obiekt ReactDOM, który pozwala nam renderować

Zazwyczaj użyjemy go raz w aplikacji

Kompilator Babel

Babel jest świetnym narzędziem, bez którego trudno się obyć, pracując z JavaScript. Babel kompiluje kod napisany w JS, ale w składni JSX i ES6 do ES5, czyli do poziomu zrozumiałego dla naszych przeglądarek.

Skorzystamy z CDN

W pierwszej fazie kursu pobierzemy trzy potrzebne pliki bezpośrednio w pliku HTML (cdn i import za pomocą `<script>`). W drugiej części kursu skorzystamy już z gotowego środowiska do developmentu aplikacji React (Create-React-App).