biblioteka JavaScript służąca do tworzenia interfejsów użytkownika

React

tutorial version 1.11

React

frontend JavaScript framework

tool for building UI components

Instead of manipulating the browser's DOM directly, React creates a virtual DOM in memory, where it does all the necessary manipulating, before making the changes in the browser DOM

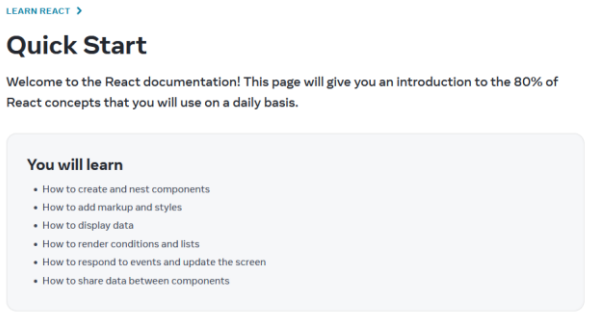
https://react.dev/

JavaScript library created by Facebook

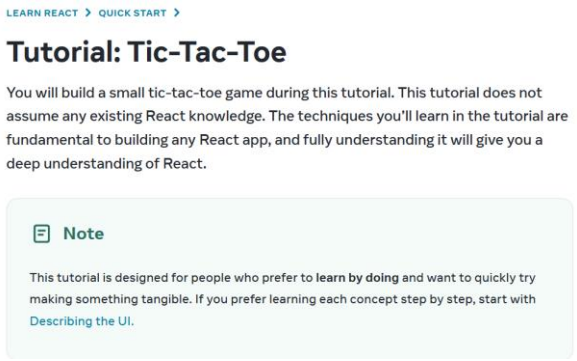
creates a VIRTUAL DOM in memory

https://www.w3schools.com/REACT/react_intro.asp

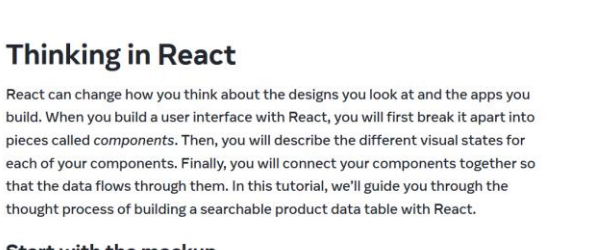
React tutorials



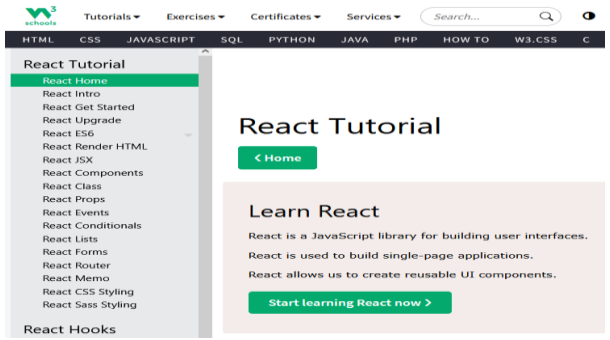
https://react.dev/learn



https://react.dev/learn/tutorial-tic-tac-toe



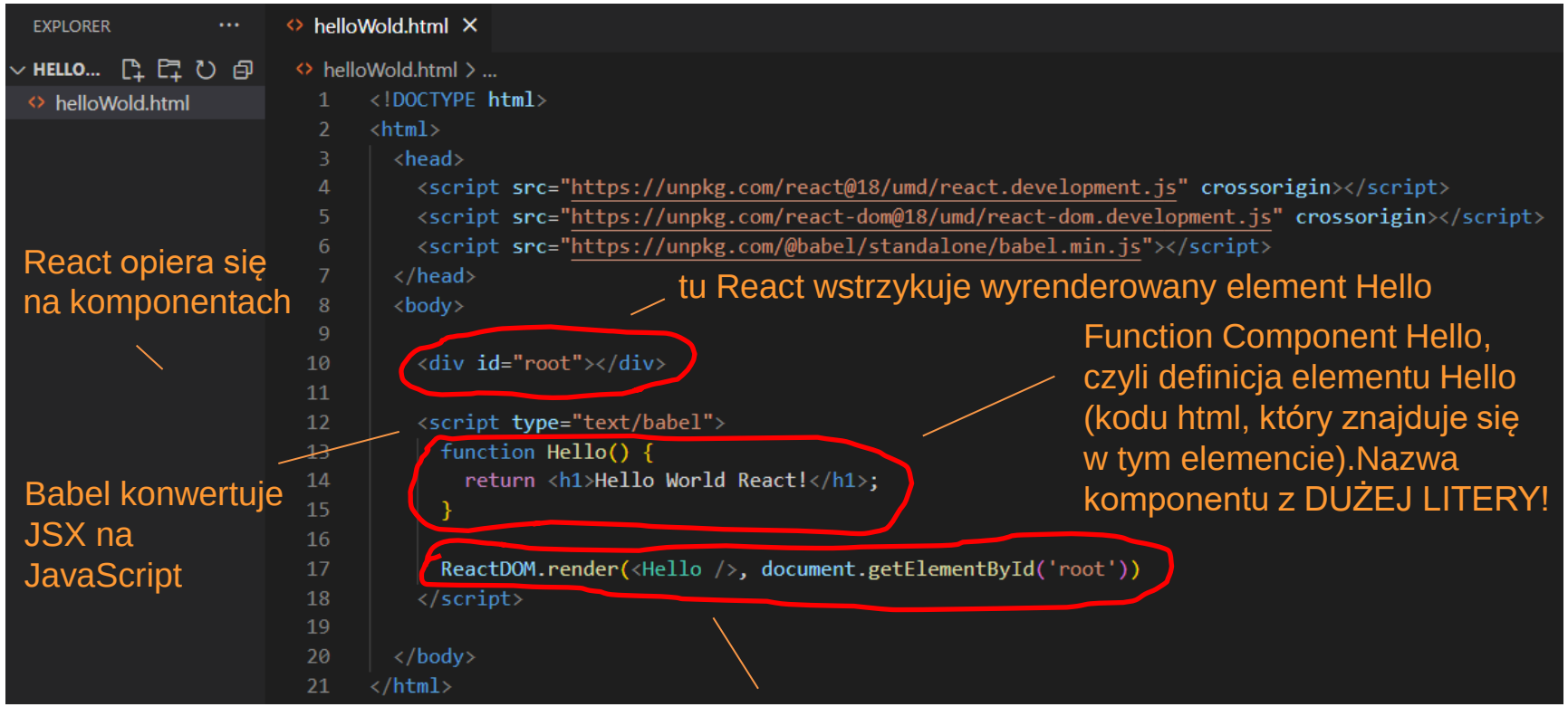
https://react.dev/learn/thinking-in-react



w3schools tutorial

React

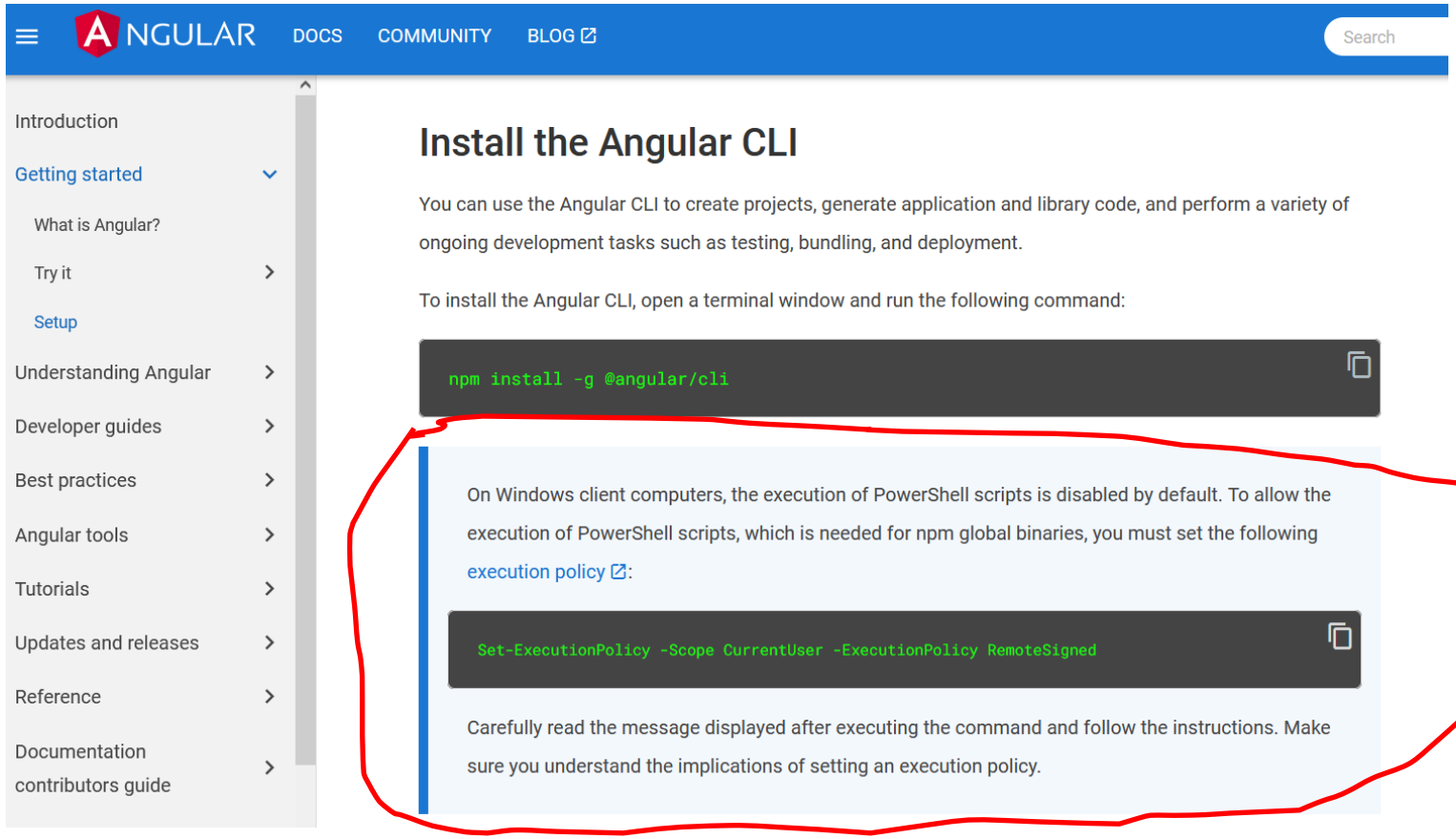
React bezpośrednio w pliku HTML (do celów testowych, w produkcji stosuje się specjalne środowisko)



React renderuje element Hello w znaczniku

Hello World React!

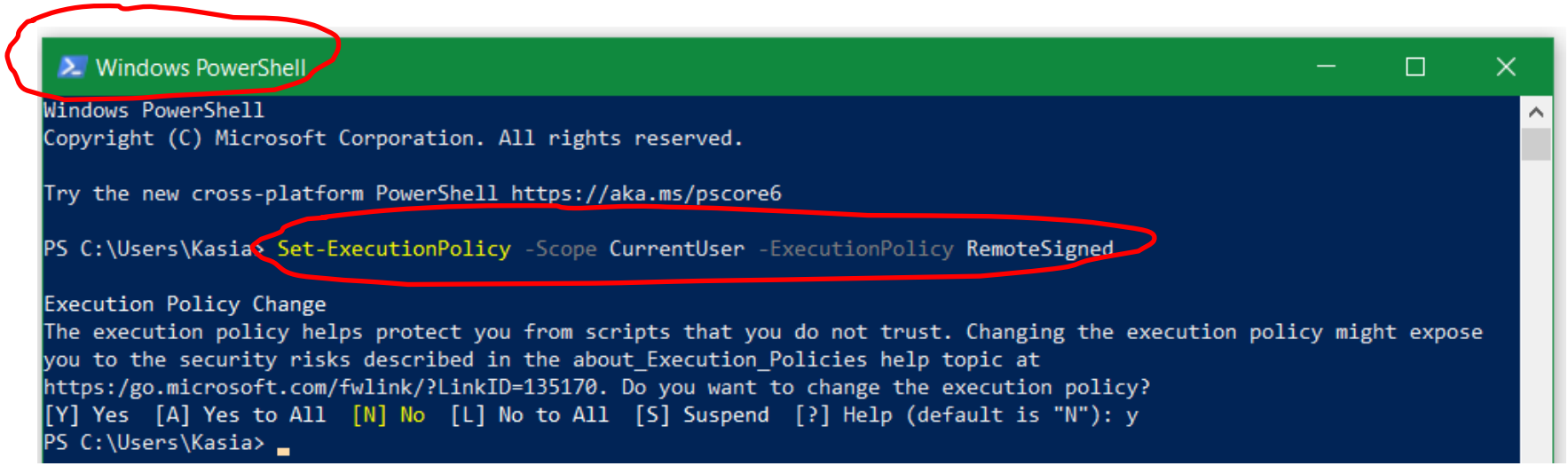
React instalacja



https://angular.io/guide/setup-local

w systemie Windows należy włączyć możliwość wykonywania skryptów w PowerShellu

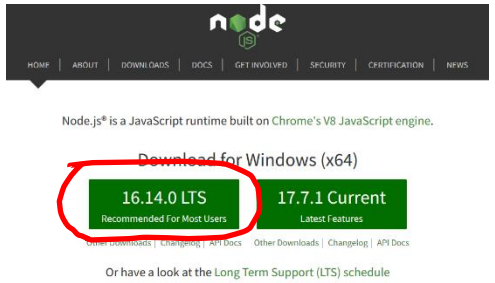
React instalacja



komendę wykonujemy w PowerShellu, a nie w cmd

React
instalacja środowiska w Visual Studio Code

instalacja środowiska Node.js



https://nodejs.org/en/

instalacja środowiska React w terminalu

```
npm install -g create-react-app
```

utworzenie projektu helloworld w terminalu

```
npx create-react-app helloworld
```

```
Success! Created helloworld at D:\React\programy\HelloWorld2\helloworld
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd helloworld
  npm start

Happy hacking!
```

błędy przy tworzeniu nowego projektu

```
Initialized a git repository.

Installing template dependencies using npm...
npm error code ERESOLVE
npm error ERESOLVE unable to resolve dependency tree
npm error
npm error While resolving: mikolaj@0.1.0
npm error Found: react@19.0.0
npm error node_modules/react
npm error   react@"^19.0.0" from the root project
npm error
npm error Could not resolve dependency:
npm error peer react@"^18.0.0" from @testing-library/react@13.4.0
npm error node_modules/@testing-library/react
npm error peer react@"^18.0.0" from @testing-library/react@13.4.0
```

https://github.com/facebook/react/issues/31699

npm install -g yarn
yarn create react-app my-app
cd my-app
yarn start

alternatywny sposób utworzenia projektu my-app oraz uruchomienia serwera

React
instalacja środowiska w Visual Studio Code

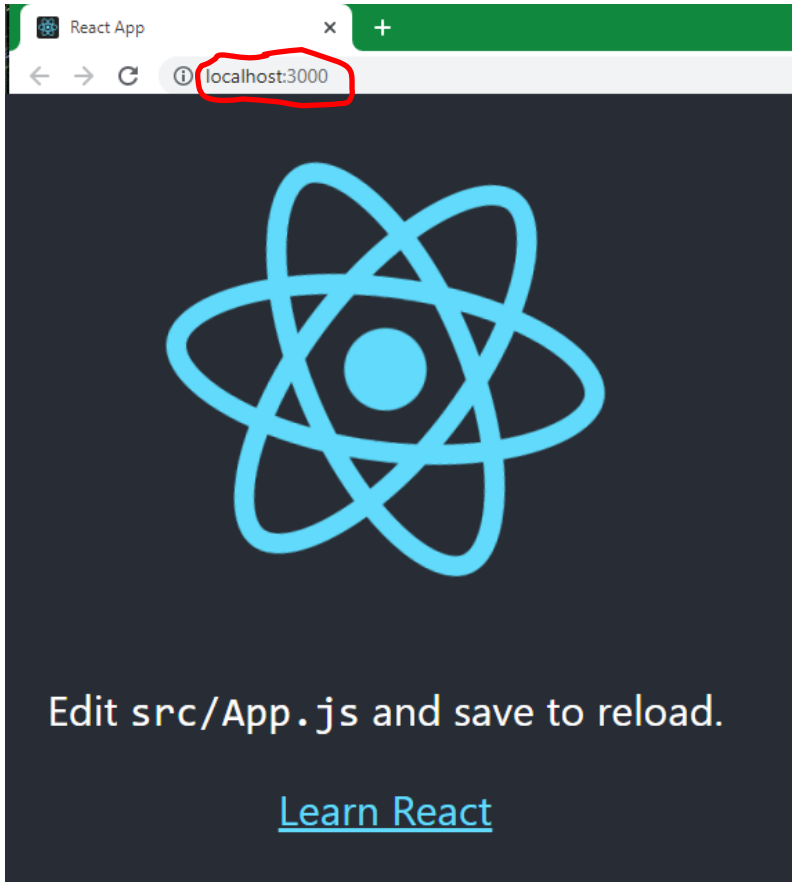
start serwera React

przejście do katalogu projektu

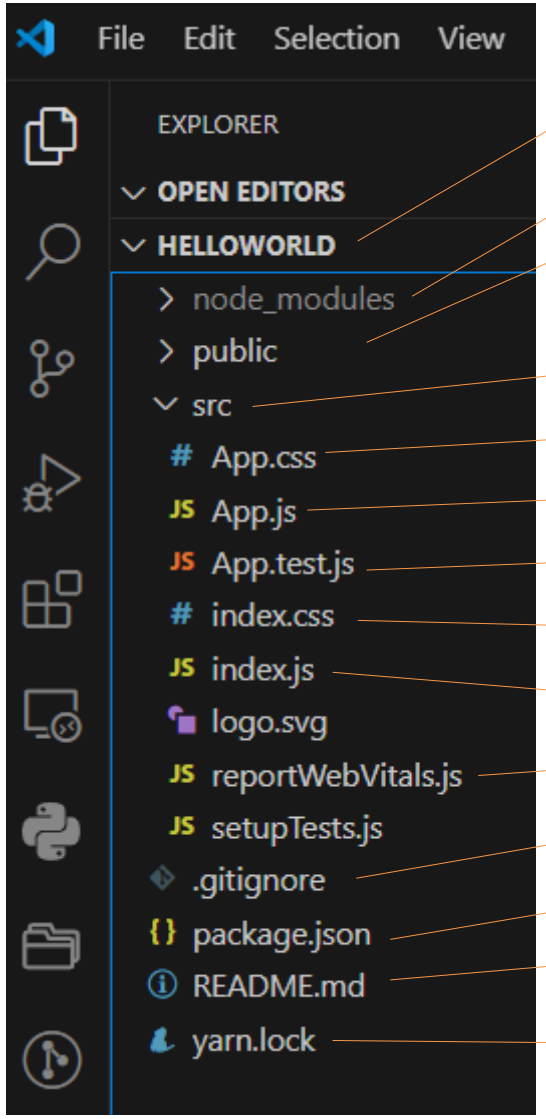
cd helloworld
npm start

serwer React nasłuchuje na porcie 3000

```
PS D:\React\programy\HelloWorld2> cd .\helloworld\
PS D:\React\programy\HelloWorld2\helloworld> npm start
Compiled successfully!

You can now view helloworld in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.25.80.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



React pliki projektu



nazwa projektu

folder z zainstalowanymi bibliotekami, modułami

folder z plikami statycznymi (ikony, logo itp.) oraz plikiem wejściowym aplikacji index,html

folder z kodem źródłowym aplikacji

css głównego komponentu aplikacji

główny komponent aplikacji (o nazwie App)

plik testowy głównego komponentu aplikacji

css strony aplikacji

plik, w którym w którym React renderuje komponent App

plik służący do zbierania i analizowania metryk wydajności aplikacji w celu jej optymalizacji

plik definiujący, które pliki lub foldery mają być ignorowane przez Git

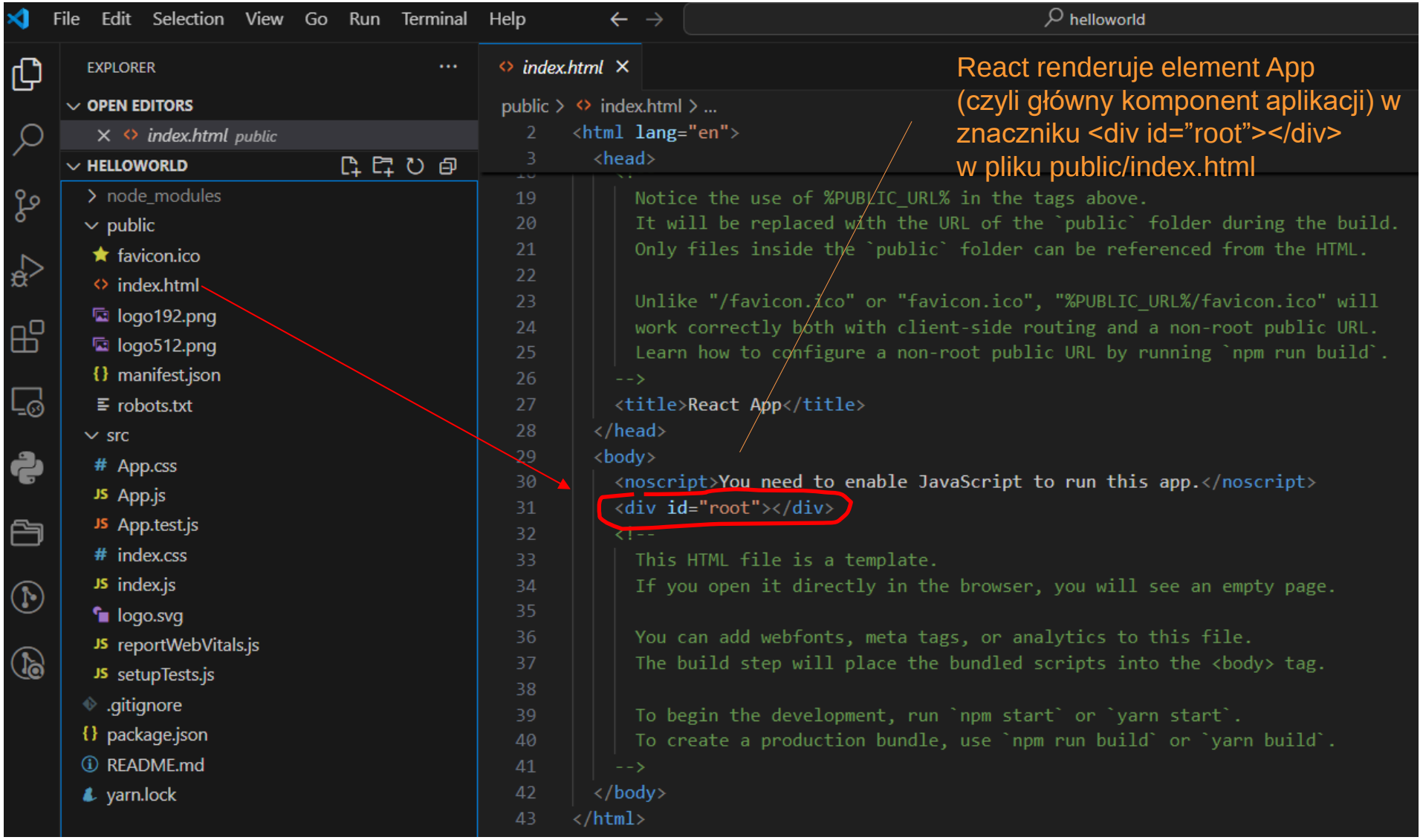
plik konfiguracyjny projektu

plik dokumentacji projektu

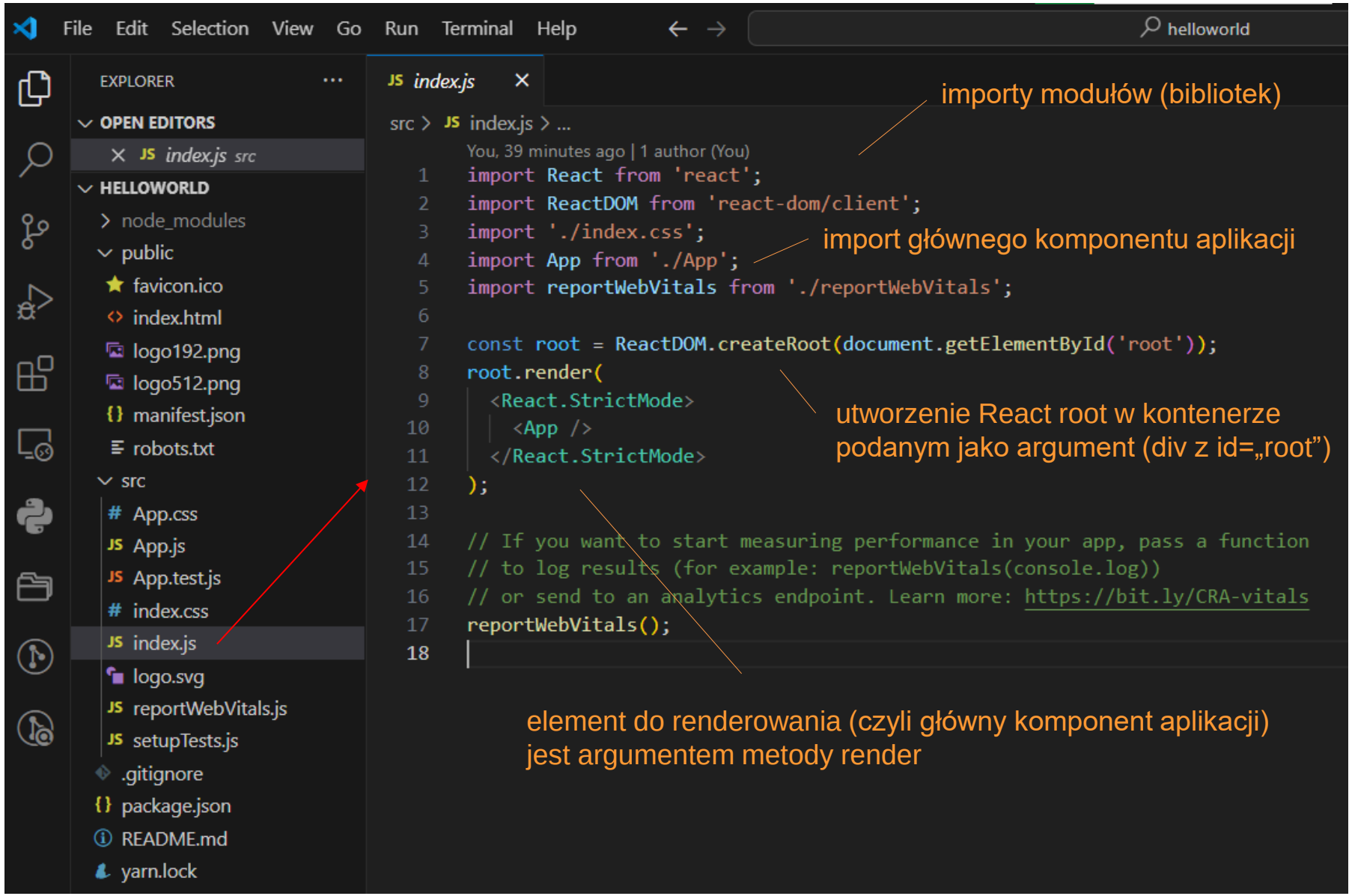
(lub package-lock.json) plik, który zapisuje dokładne wersje zależności,
aby zapewnić spójność instalacji

React renderowanie

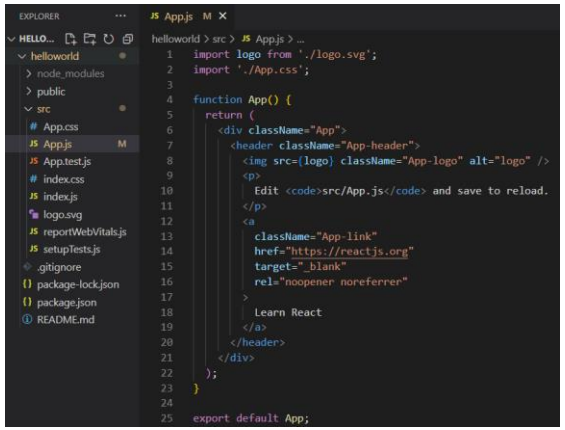
przygotowanie widoku elementu

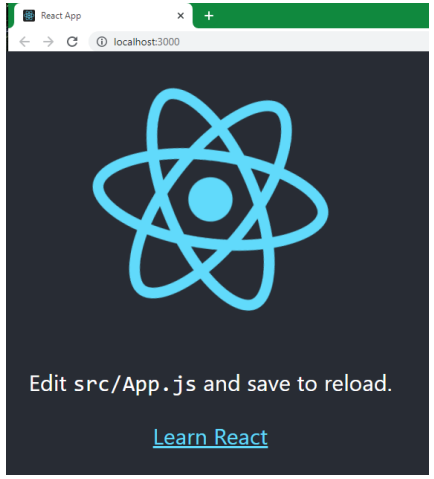


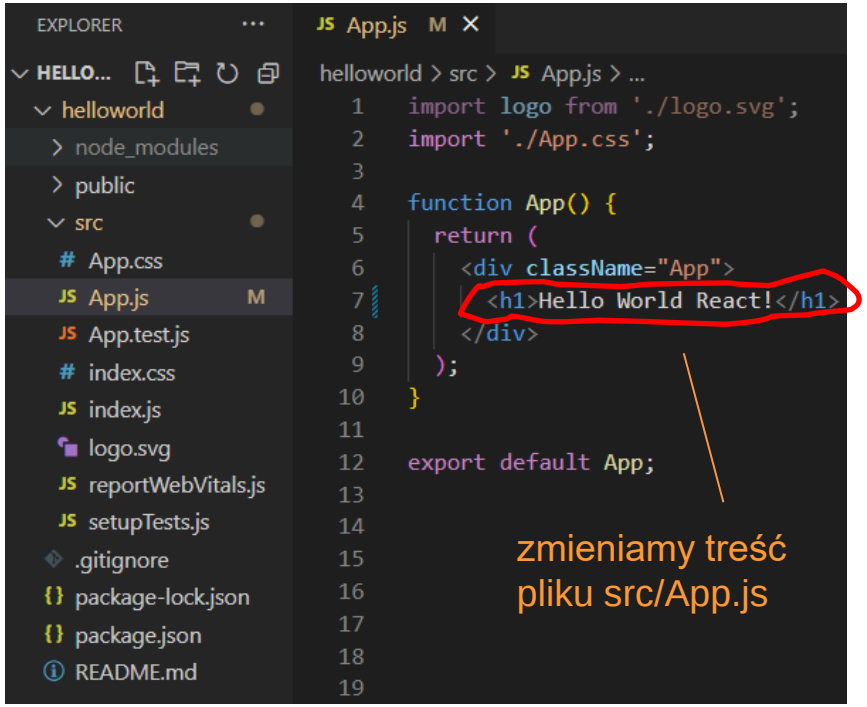
React renderowanie

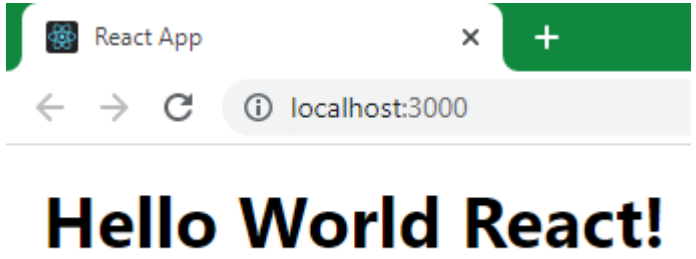


React
zmiana treści pliku App.js







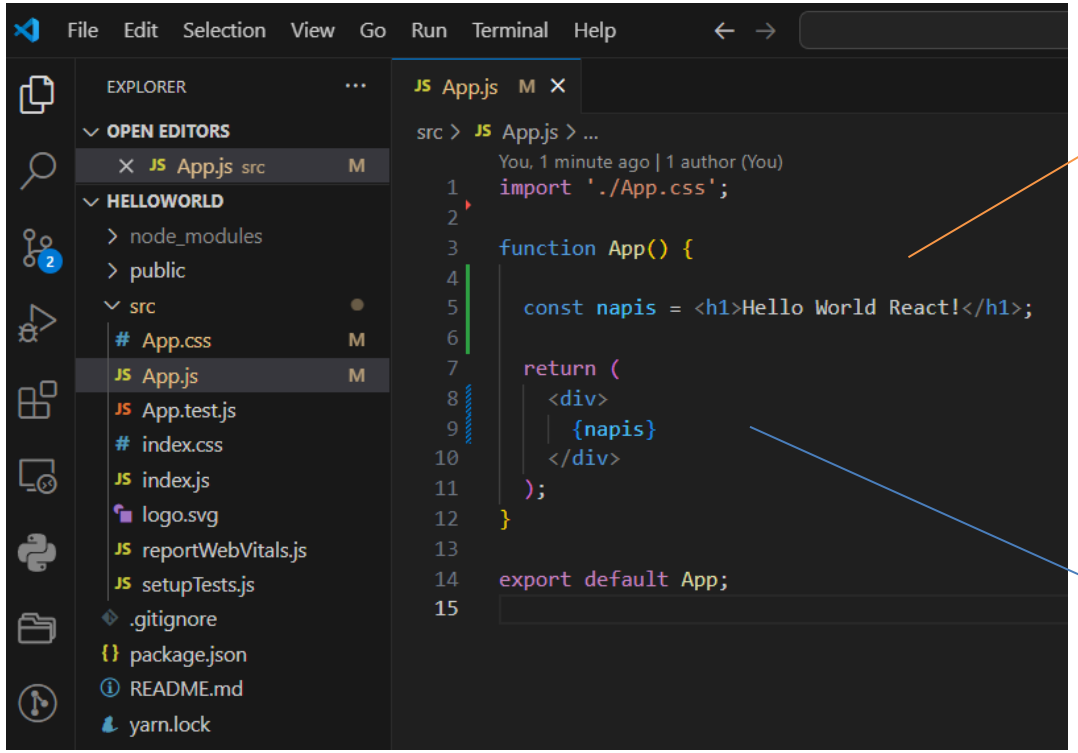


React JSX

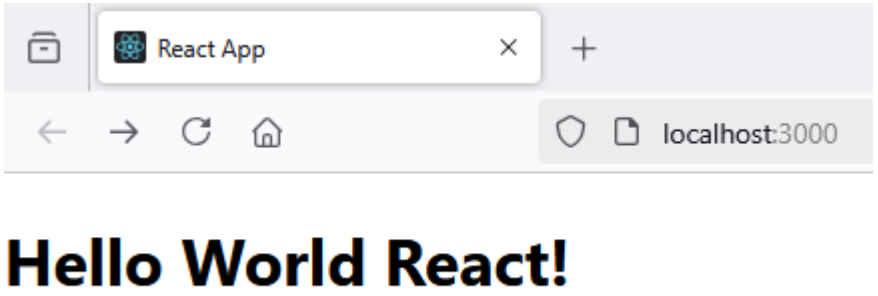
JavaScript XML

JSX umożliwia pisanie kodu HTML w React – jest to rozszerzona składnia języka JavaScript o możliwość wstawiania znaczników (React konwertuje tagi HTML na elementy Reactowe)

można nie używać składni JSX w React, ale składnia jest wtedy nieco bardziej skomplikowana

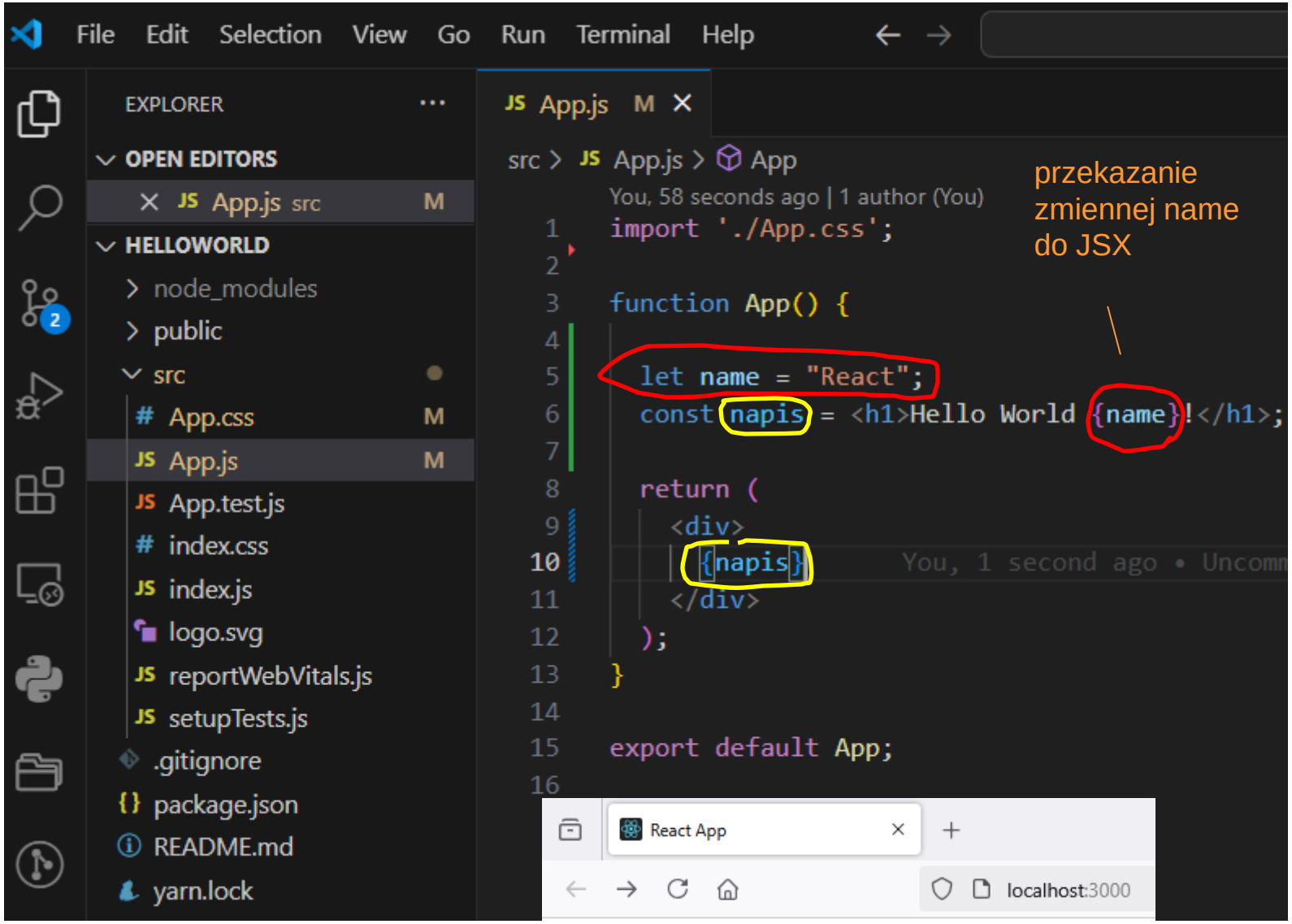


JSX umożliwia przypisanie elementów HTML do zmiennej (bez cudzysłowów)

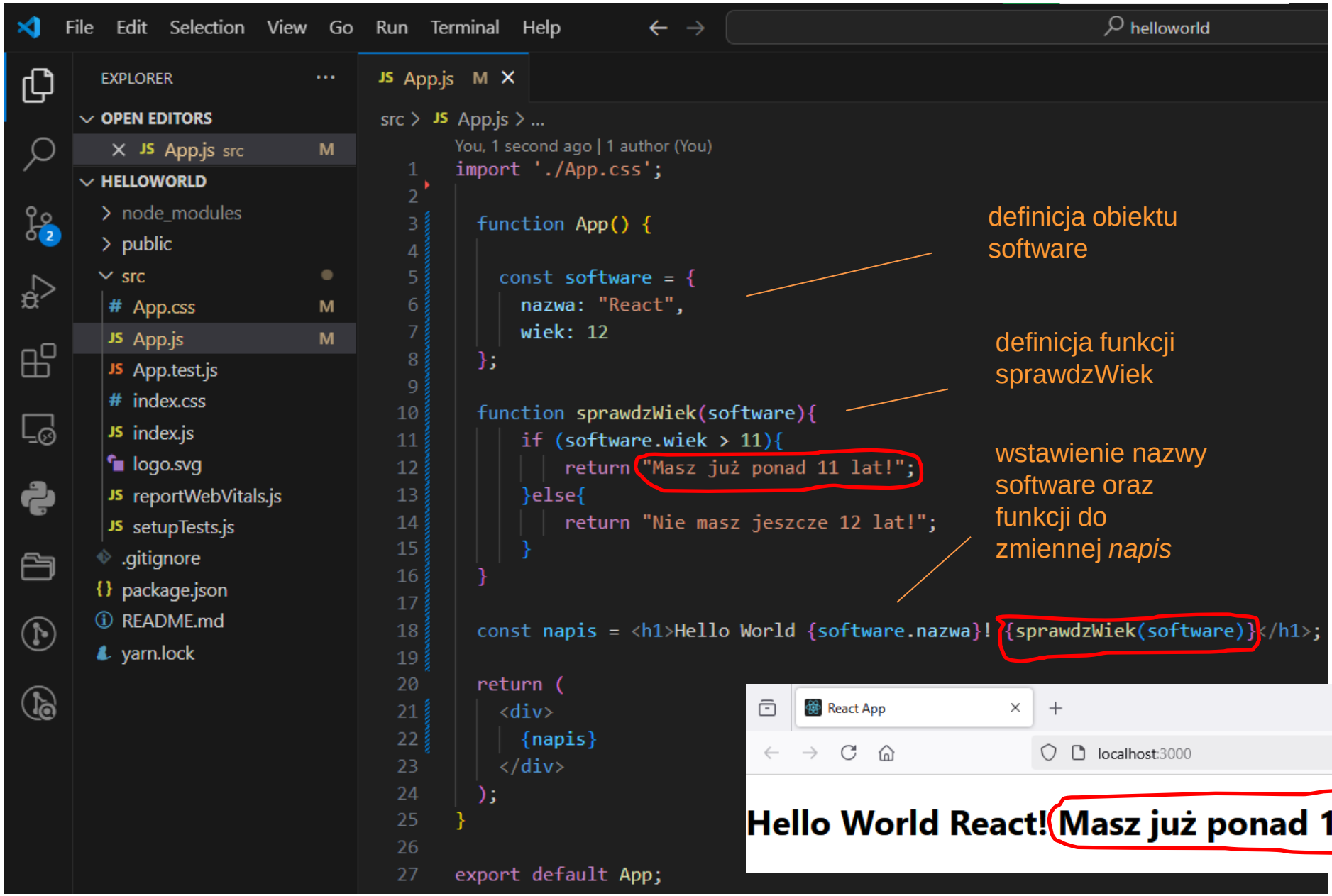


https://react.dev/learn/writing-markup-with-jsx

React JSX


File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
App.js src M
HELLOWORLD
node_modules
public
src
App.css M
App.js M
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package.json
README.md
yarn.lock
App.js M
src > App.js > App
You, 58 seconds ago | 1 author (You)
1 import './App.css';
2
3 function App() {
4
5 let name = "React";
6 const napis = <h1>Hello World {name}!</h1>;
7
8 return (
9
10 {napis}
11
12);
13 }
14
15 export default App;
16
przekazanie zmiennej name do JSX
React App
localhost:3000
Hello World React!

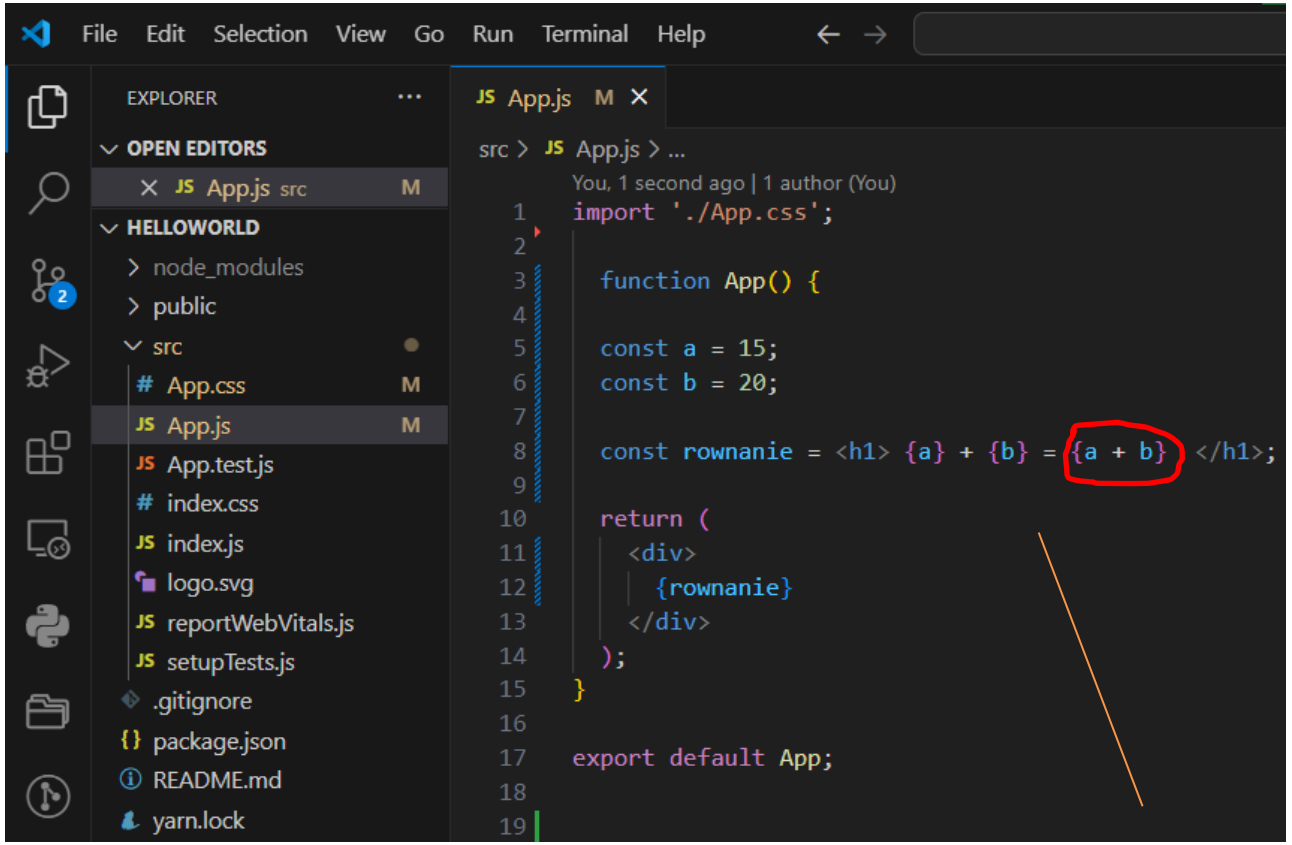
React JSX

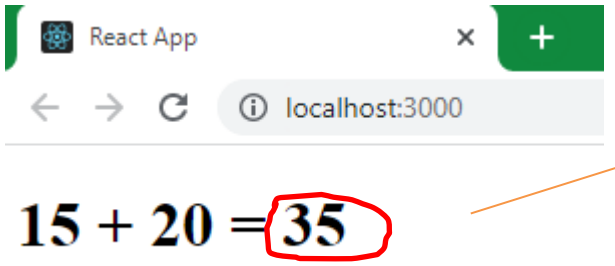

File Edit Selection View Go Run Terminal Help
helloworld
EXPLORER
OPEN EDITORS
App.js src M
HELLOWORLD
node_modules
public
src
App.css M
App.js M
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package.json
README.md
yarn.lock
App.js M
src > App.js > ...
You, 1 second ago | 1 author (You)
import './App.css';
function App() {
const software = {
nazwa: "React",
wiek: 12
};
function sprawdzWiek(software){
if (software.wiek > 11){
return "Masz już ponad 11 lat!";
}else{
return "Nie masz jeszcze 12 lat!";
}
}
const napis = <h1>Hello World {software.nazwa}! {sprawdzWiek(software)}</h1>;
return (

{napis}

);
}
export default App;
definicja obiektu software
definicja funkcji sprawdzWiek
wstawienie nazwy software oraz funkcji do zmiennej napis
React App
localhost:3000
Hello World React! Masz już ponad 11 lat!

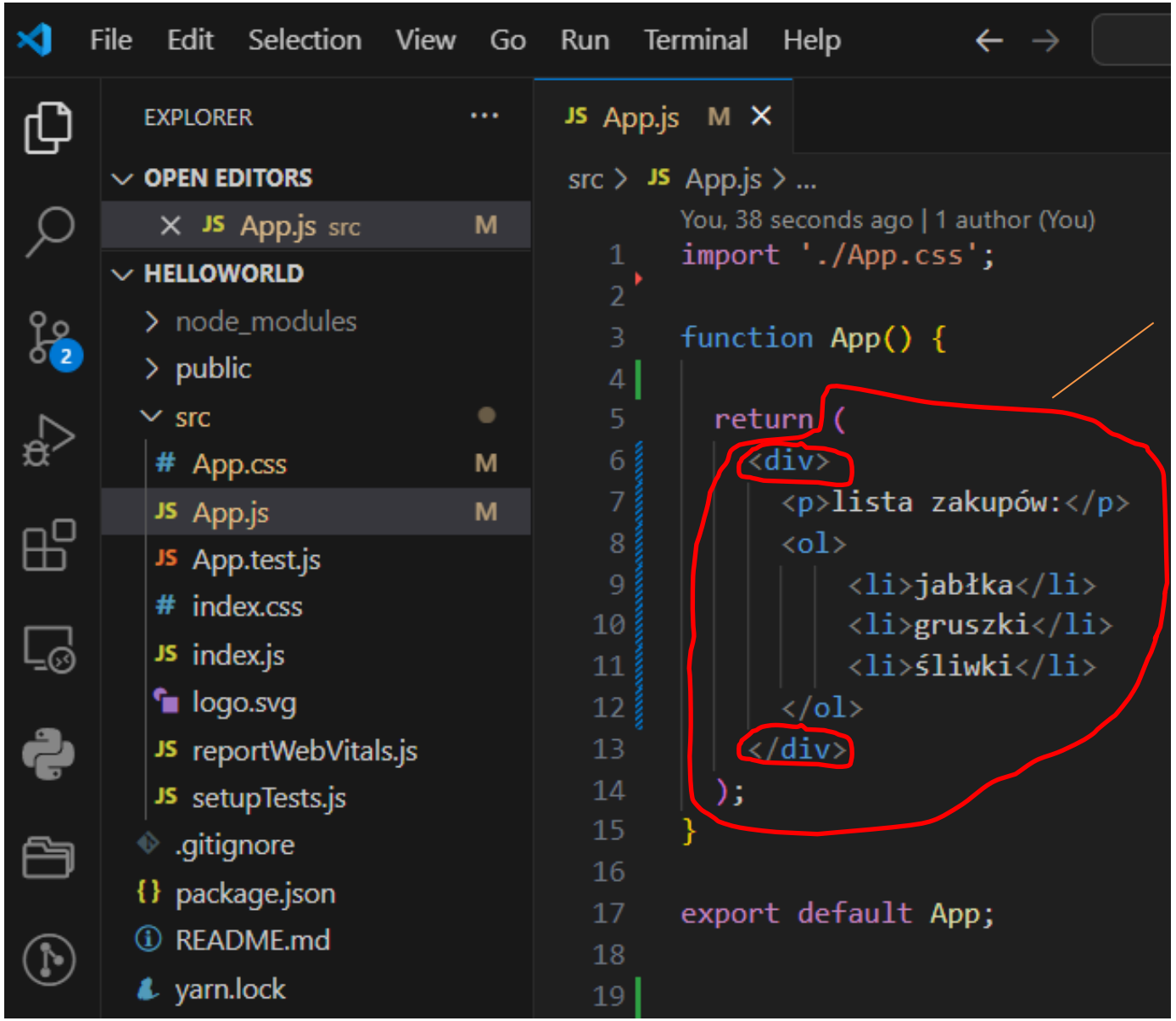
React JSX

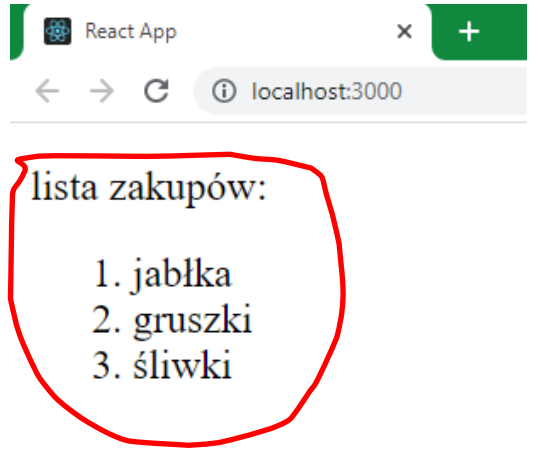




w JSX można stosować wyrażenia

React JSX



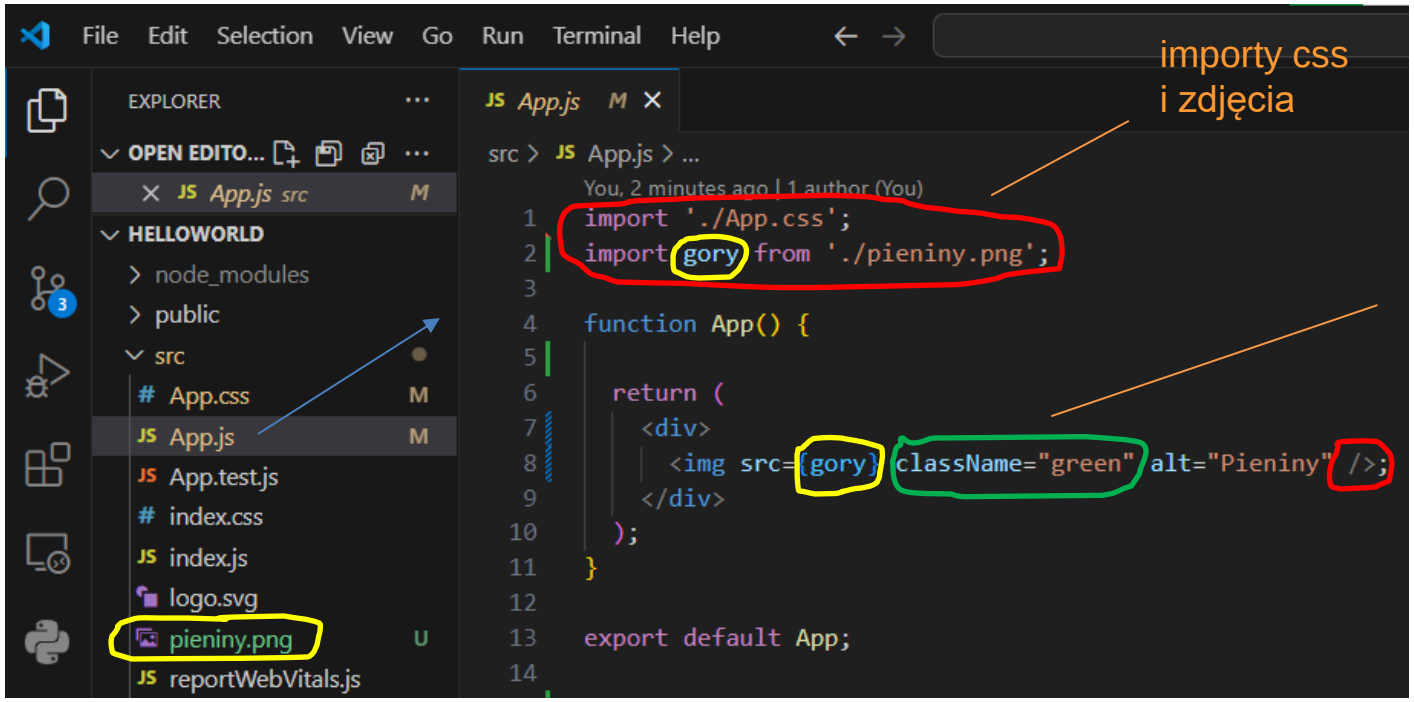


kilka elementów HTML musi się znaleźćw pojedynczym elemencie top level np. (lub pustym <></>)

```
function App() {

  return (
    <>
      <p>lista zakupów:</p>
      <ol>
        <li>jabłka</li>
        <li>gruszki</li>
        <li>śliwki</li>
      </ol>
    </>
  );
}
```

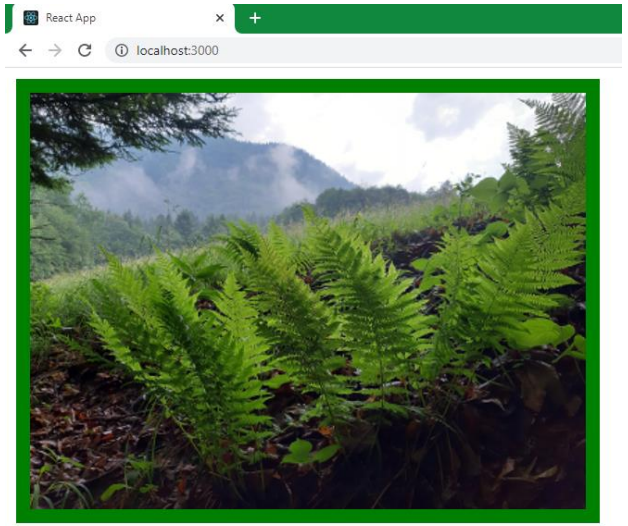
top level element może być pusty

React JSX className

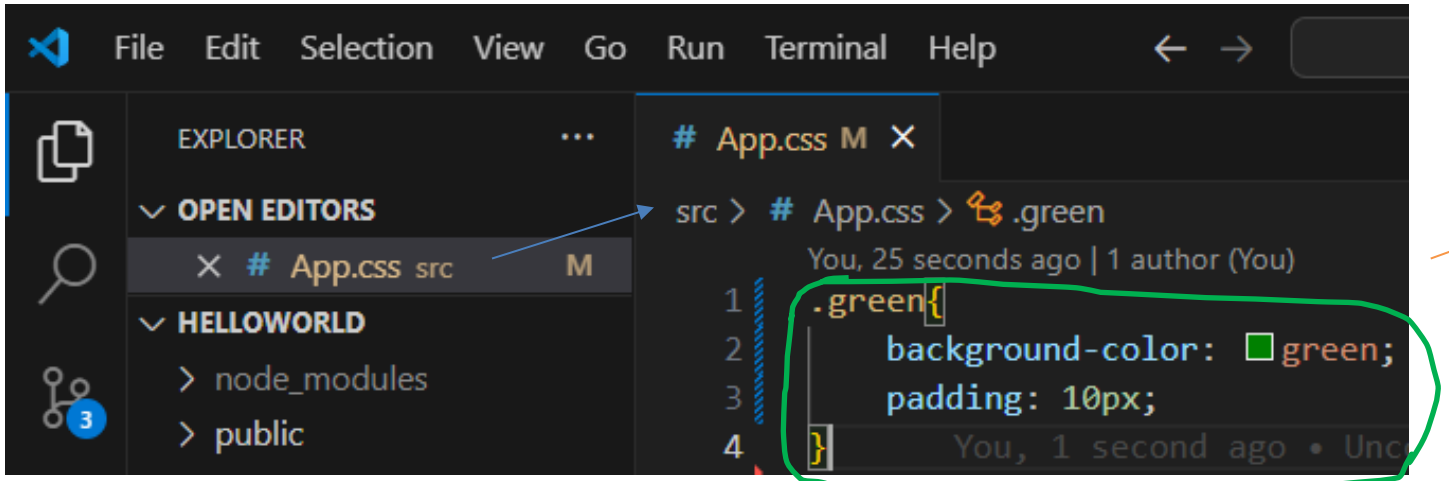


importy css i zdjęcia

zamiast słowa class w React stosujemy className



element bez tagu kończącego musi być zamknięty

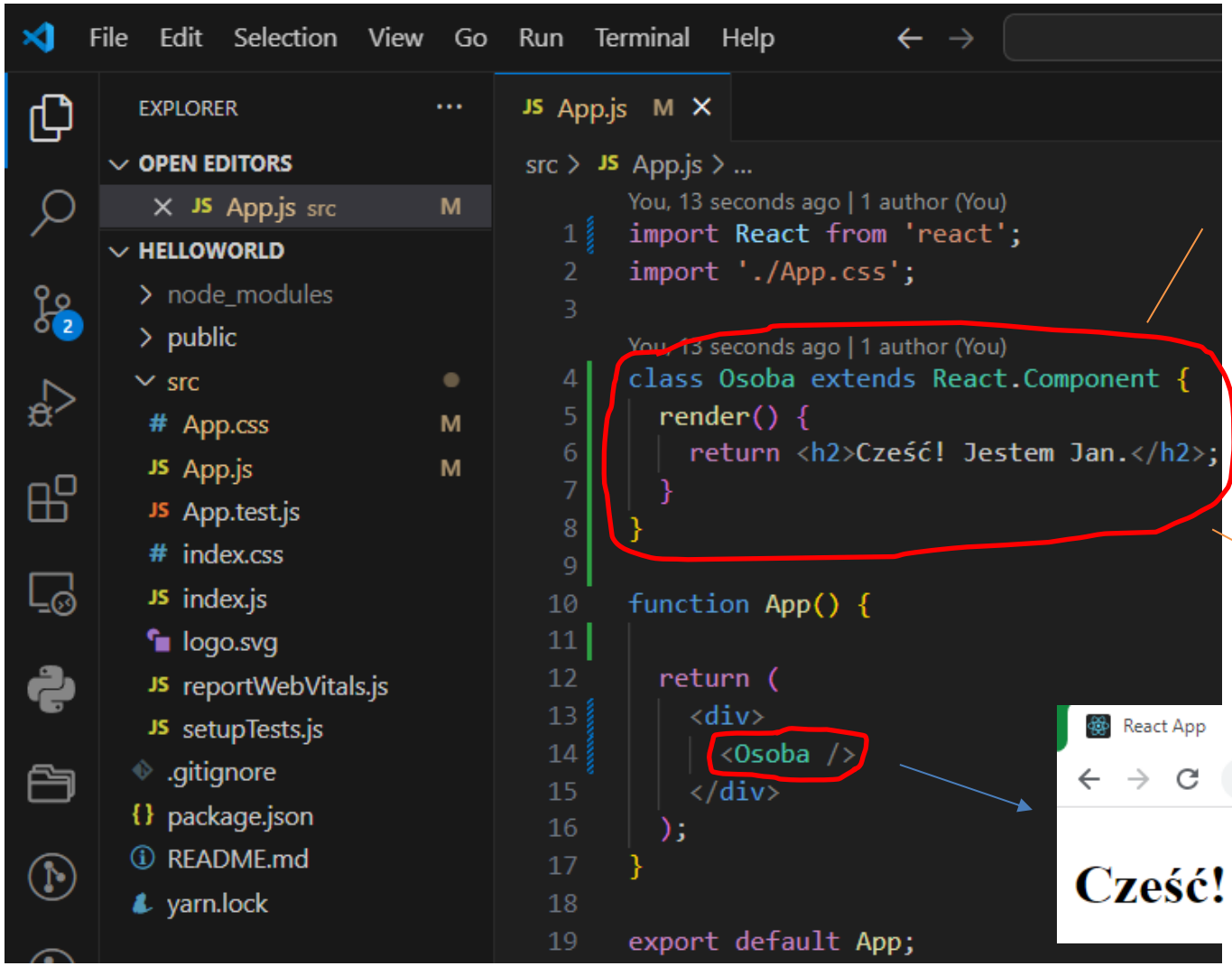


App.css ze stylem klasy green

React
Class Component

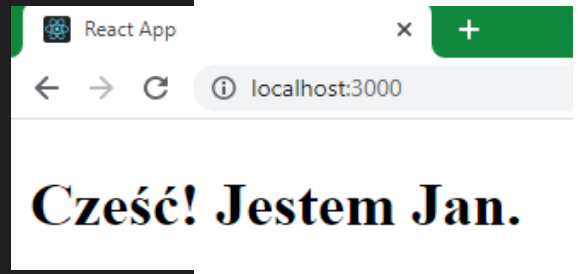
Komponenty zwracają elementy HTML

komponent klasowy Osoba

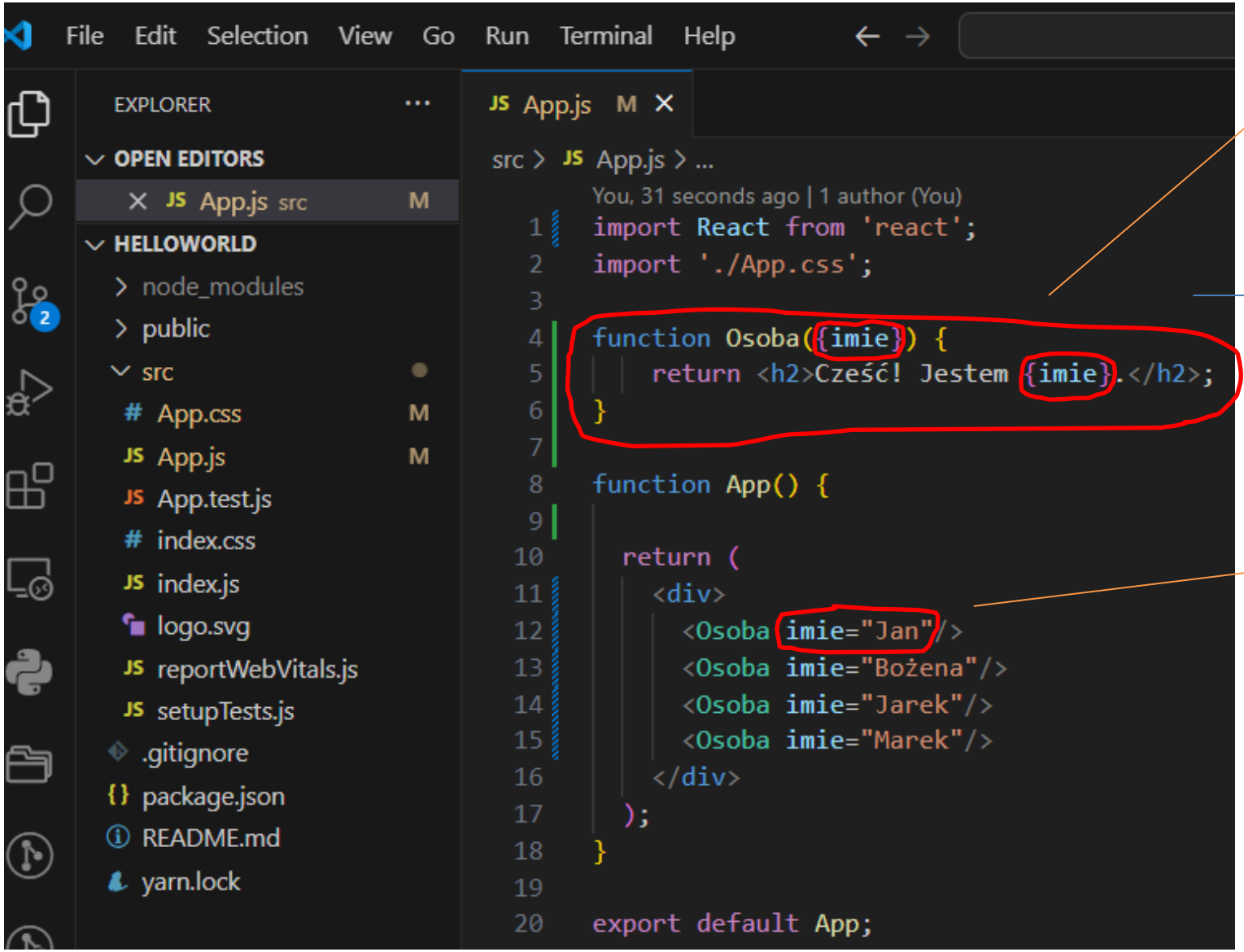


klasa Osoba (nazwa z dużej litery!) dziedziczy z klasy React.Component

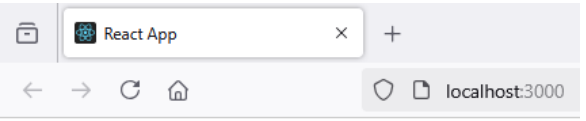
metoda render zwraca kod HTML



React
Function Component



komponent funkcyjny Osoba z właściwością *imie*



Cześć! Jestem Jan.

Cześć! Jestem Bożena.

Cześć! Jestem Jarek.

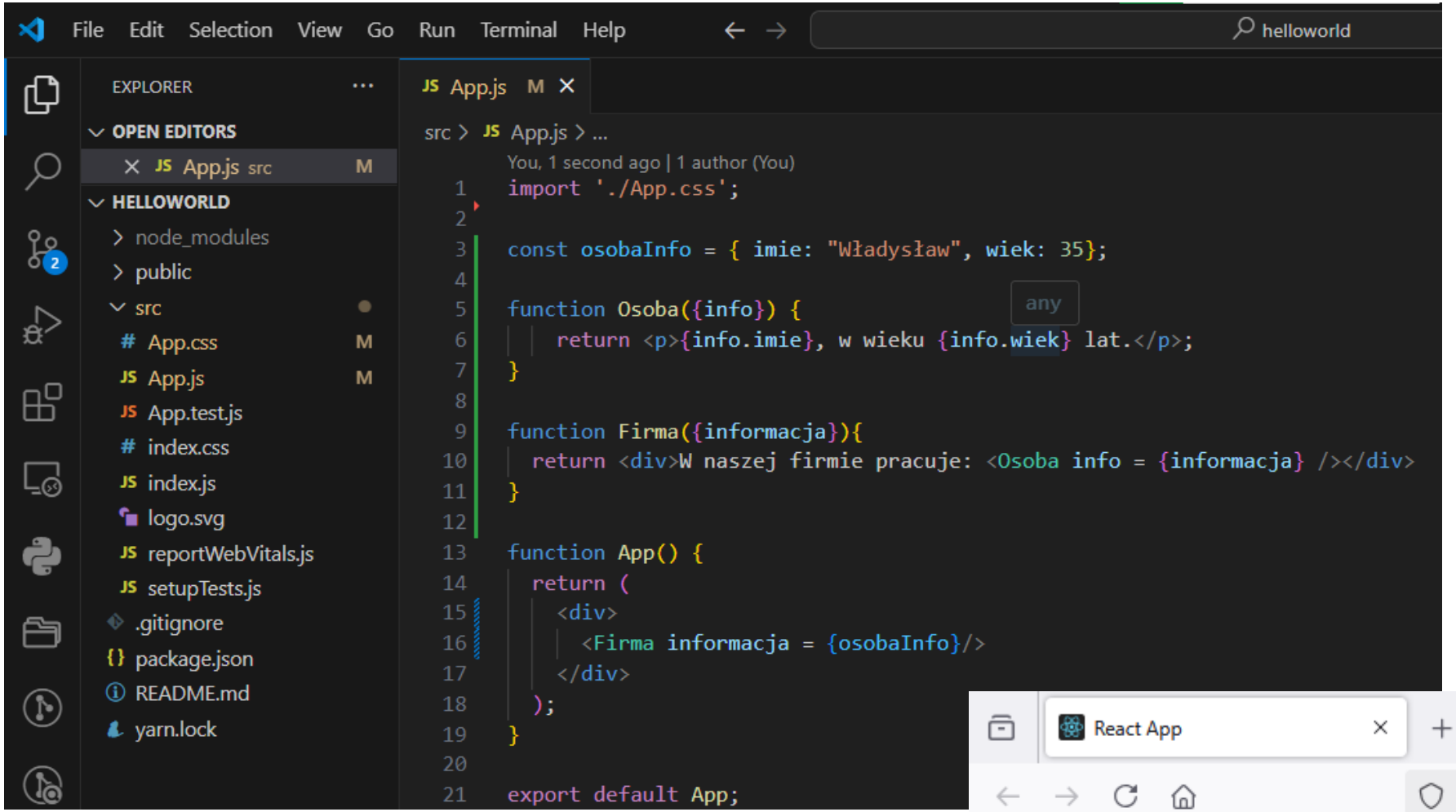
Cześć! Jestem Marek.

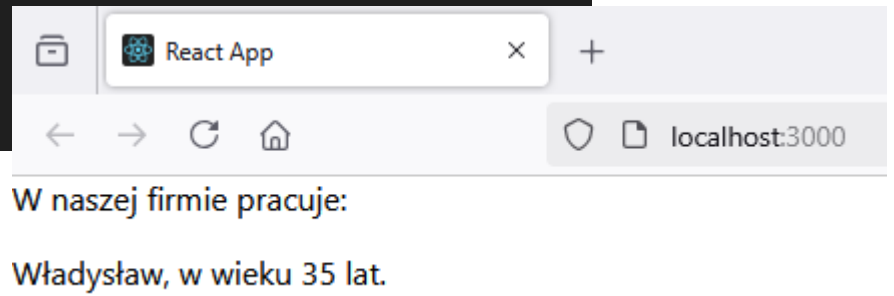
właściwość (atrybut) komponentu z przypisanym imieniem

właściwość komponentu to po angielsku *properties*, stąd potoczna nazwa ***props***

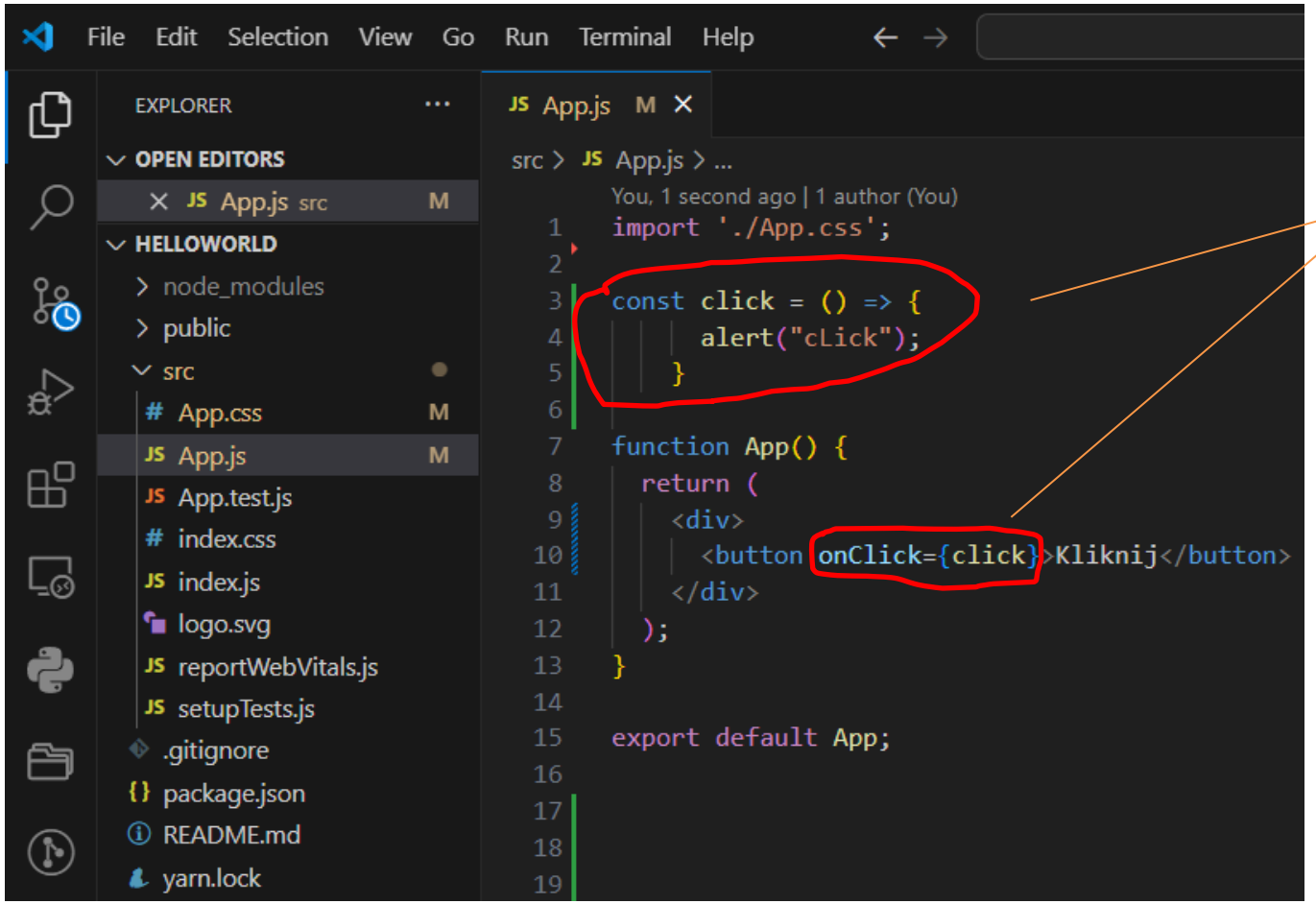
React
Function Component

komponent funkcyjny Osoba zagnieżdżony w komponencie funkcyjnym Firma

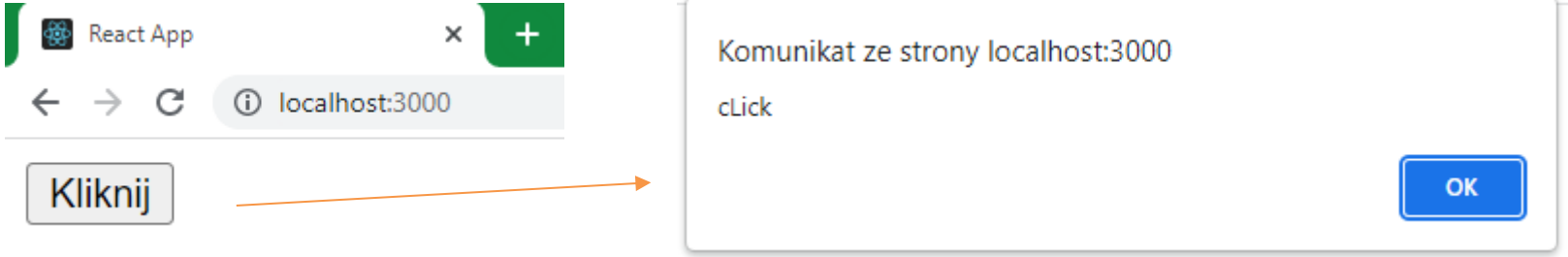




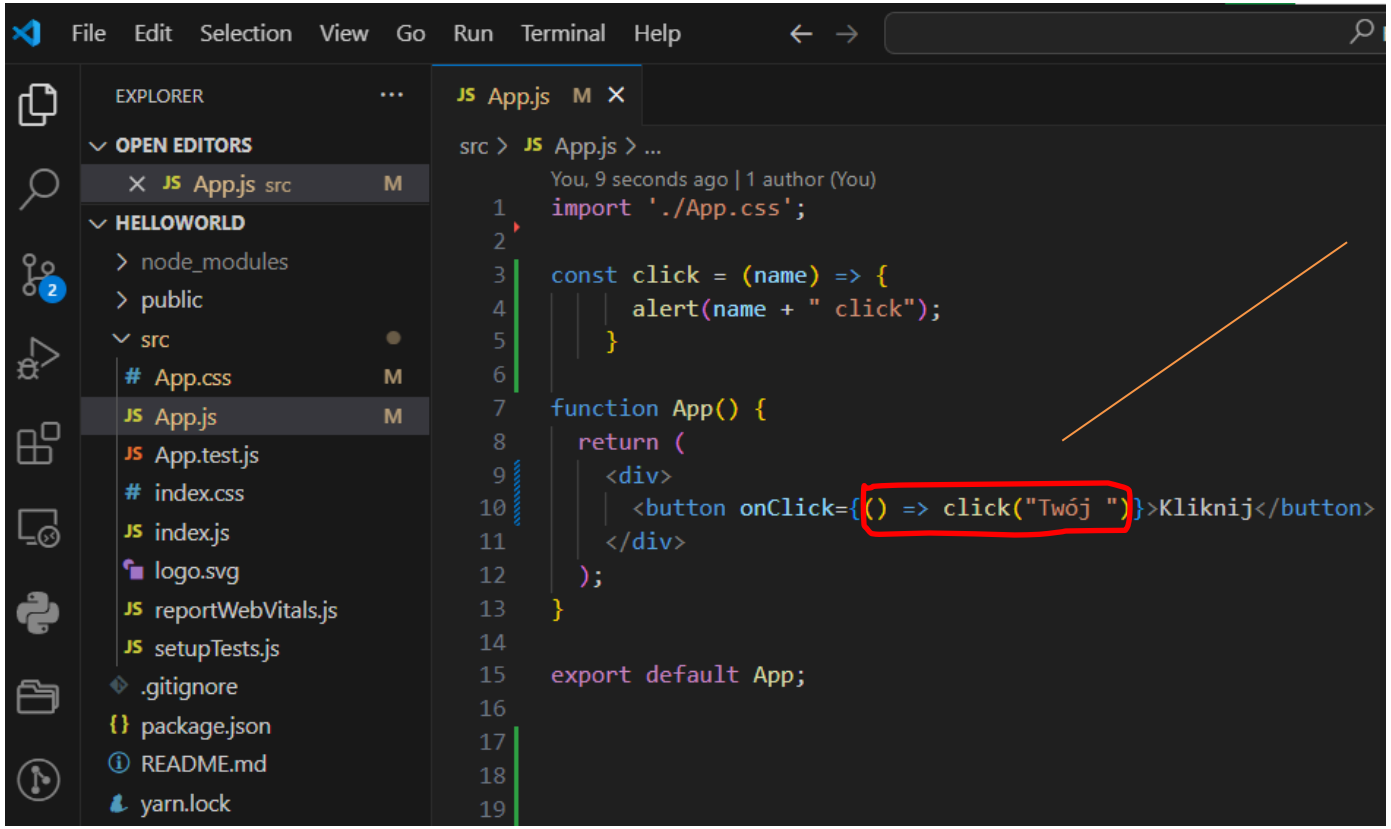
React events



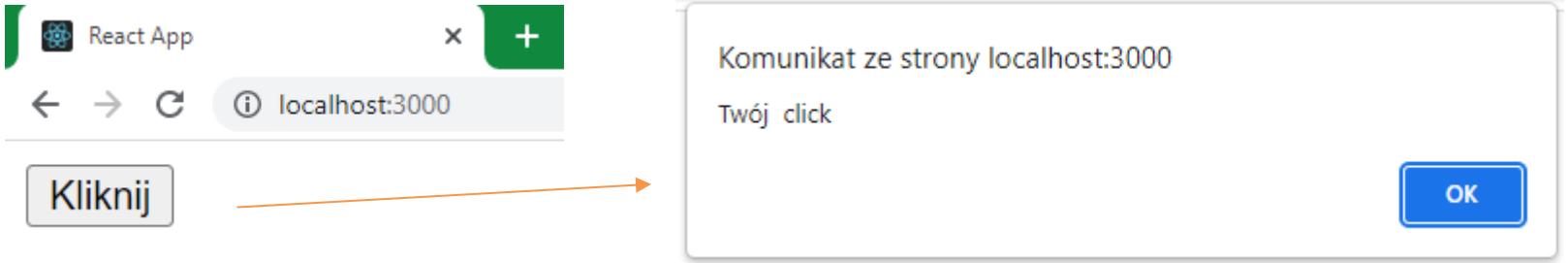
przypisanie funkcji *click* do zdarzenia buttona *onClick*



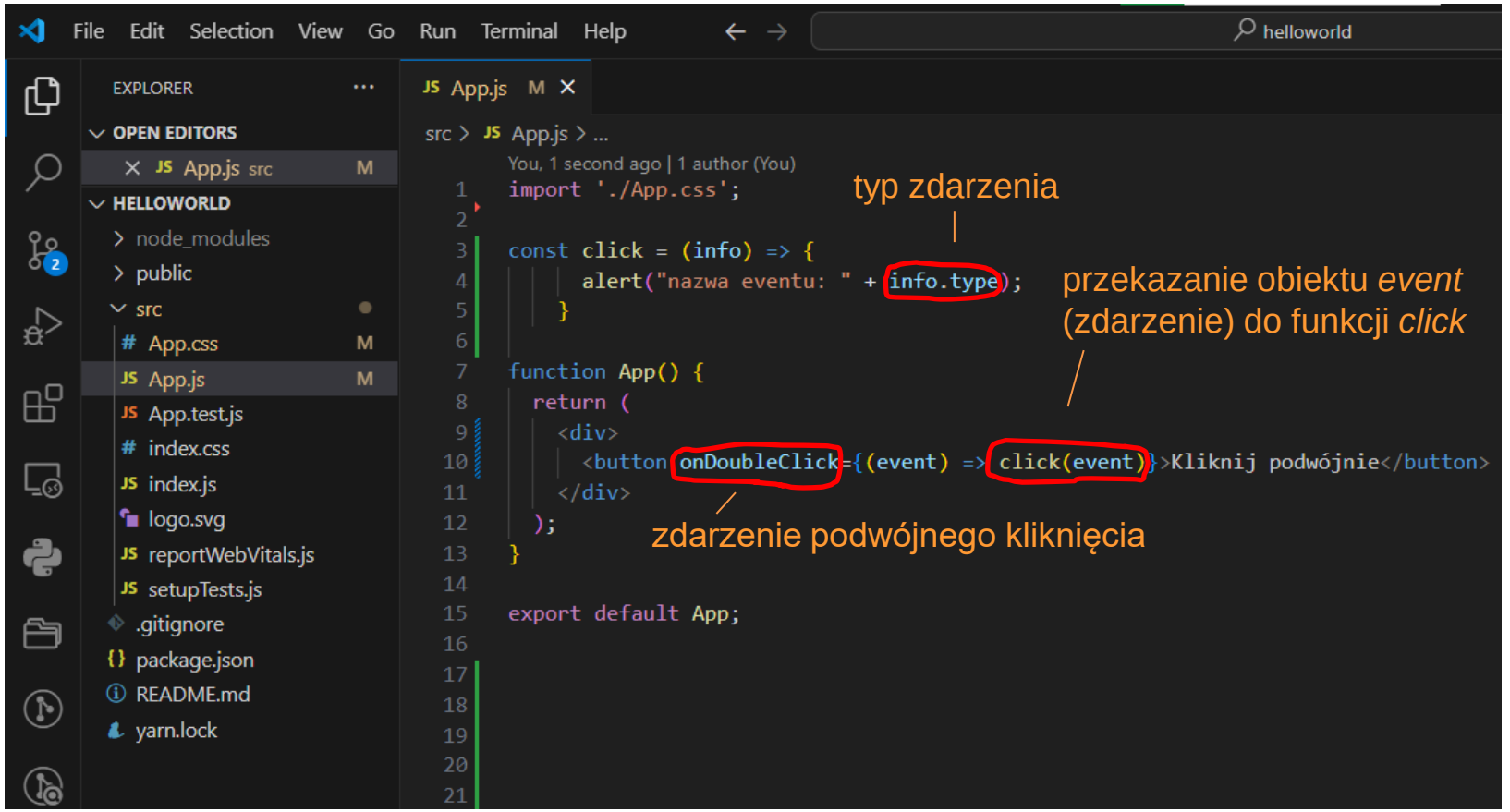
React events

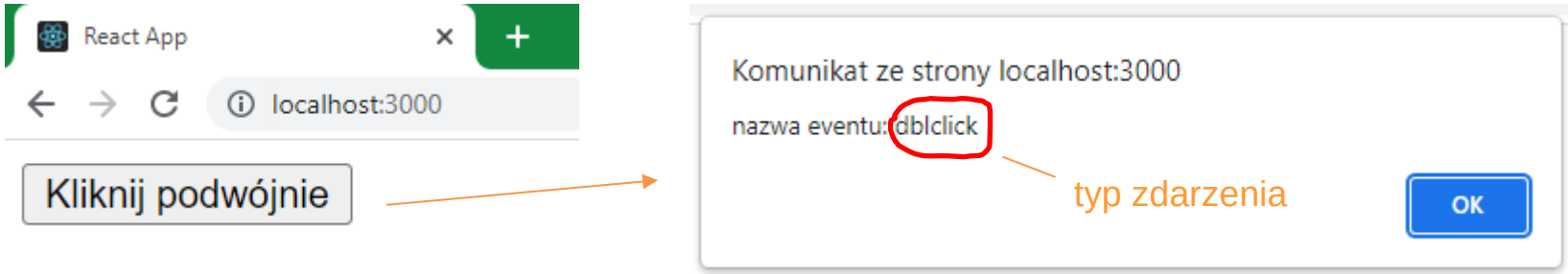


aby przekazać argument do funkcji *click* trzeba użyć funkcji strzałkowej (array function)



React events





useState Hook

useState is a React Hook that lets you add a state variable to your component

useState

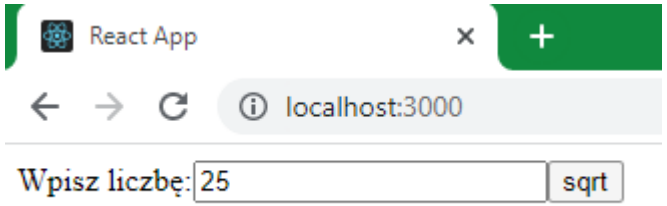
State generally refers to data or properties that need to be tracking in an application.

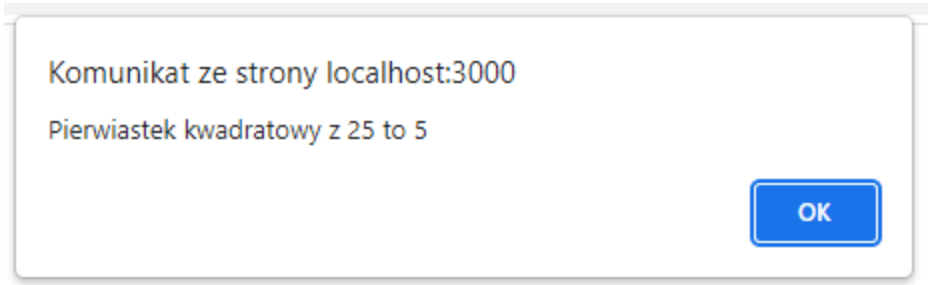
https://react.dev/reference/react/useState

https://www.w3schools.com/REACT/react_usestate.asp

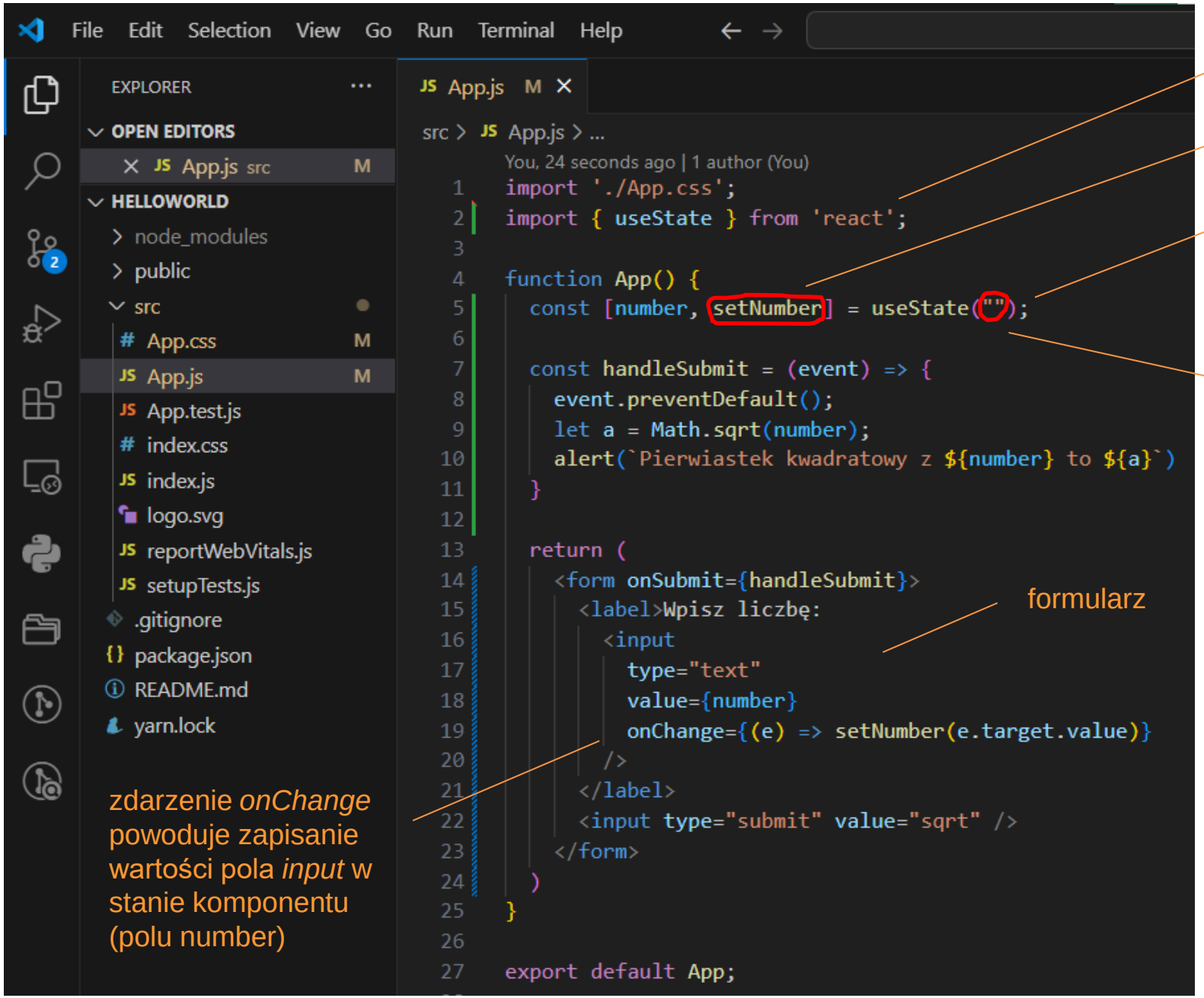
React form

komponent funkcyjny obsługujący formularz ma swój stan, w którym przechowuje dane (wartość z pola input)





React form



import useState Hook

funkcja zmieniająca stan komponentu

stan komponentu (pole number)

wartość początkowa stanu komponentu

formularz

zdarzenie *onChange* powoduje zapisanie wartości pola *input* w stanie komponentu (polu number)

https://www.w3schools.com/REACT/react_forms.asp

React form

Ankieta

imię

nazwisko

płeć

mężczyzna kobieta

zainteresowania

sport film elektronika informatyka

szkoła

Podstawowa

krótko o sobie

wyświetl dane resetuj ankietę

napisz ankietę

wpisane dane
wyświetl w dowolny
sposób

useEffect Hook

useEffect is a React Hook that lets you synchronize a component with an external system.

useEffect

Effects are an escape hatch from the React paradigm. They let you “step outside” of React and synchronize your components with some external system like a non-React widget, network, or the browser DOM

https://react.dev/reference/react/useEffect

useEffect Hook

useEffect(setup, dependencies?)

setup: The function with your Effect’s logic. Your setup function may also optionally return a cleanup function. When your component is added to the DOM, React will run your setup function. After every re-render with changed dependencies, React will first run the cleanup function (if you provided it) with the old values, and then run your setup function with the new values. After your component is removed from the DOM, React will run your cleanup function.

optional dependencies: The list of all reactive values referenced inside of the setup code. Reactive values include props, state, and all the variables and functions declared directly inside your component body.

useEffect Hook

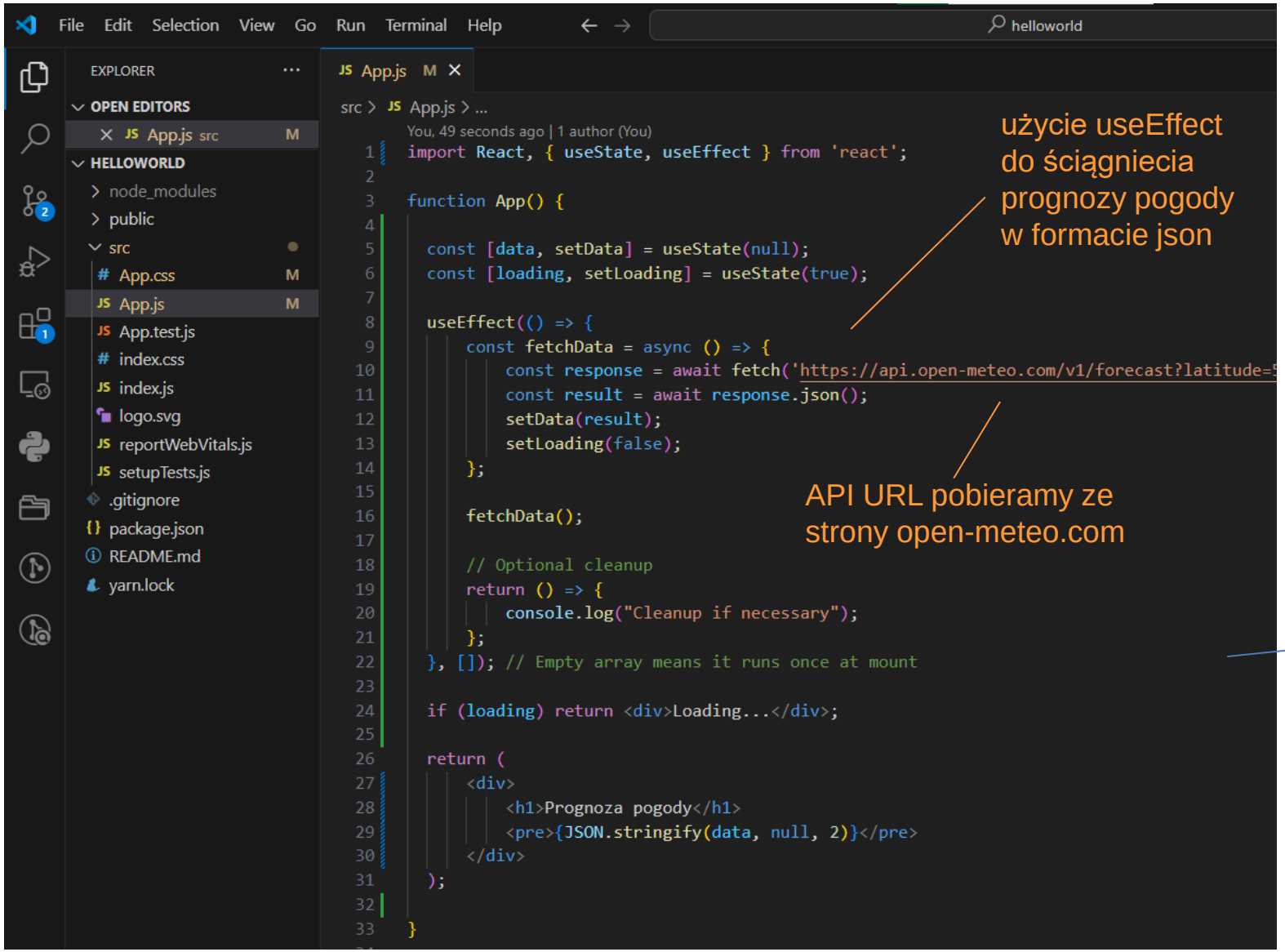
1. **Running the Effect:**
    - The function passed to `useEffect` runs after every render by default.
    - By specifying a second argument, an array of dependencies, you can control when the effect runs.
        - `[]`: the effect runs only once after the initial render (similar to `componentDidMount`).
        - `[dependency]`: the effect runs whenever the specified dependency changes.
        - No array: the effect runs after every render.

2. **Cleanup:**
    - If your effect returns a function, that function will be called when the component unmounts or before the effect runs next. This is used to clean up subscriptions, timers, etc.

3. **Multiple Effects:**
    - You can have multiple `useEffect` hooks in a single component, which allows you to separate different concerns.

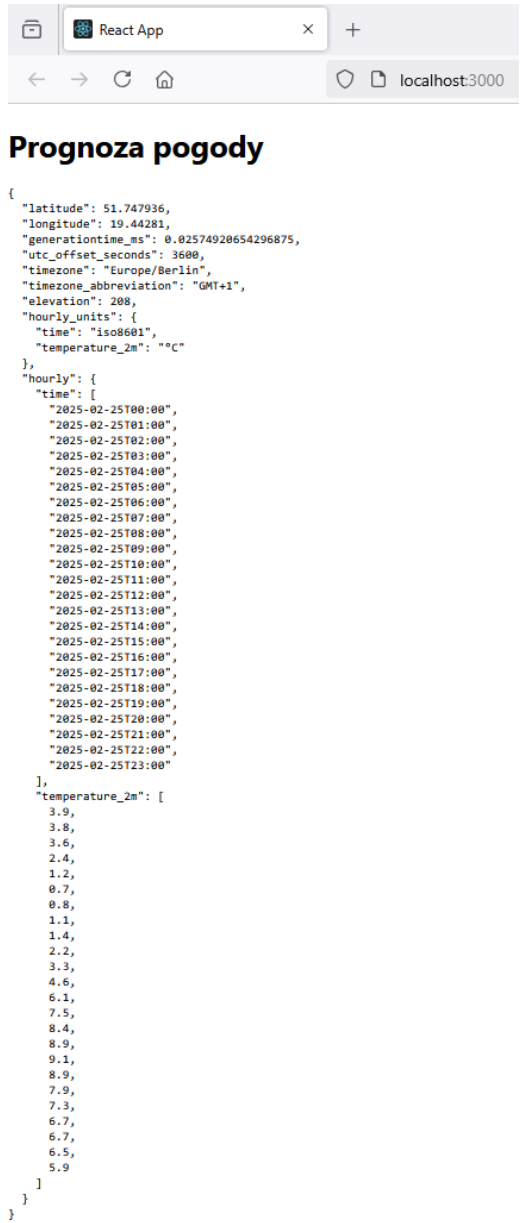
https://deepai.org/chat

prognoza pogody



użycie useEffect do ściągniecia prognozy pogody w formacie json

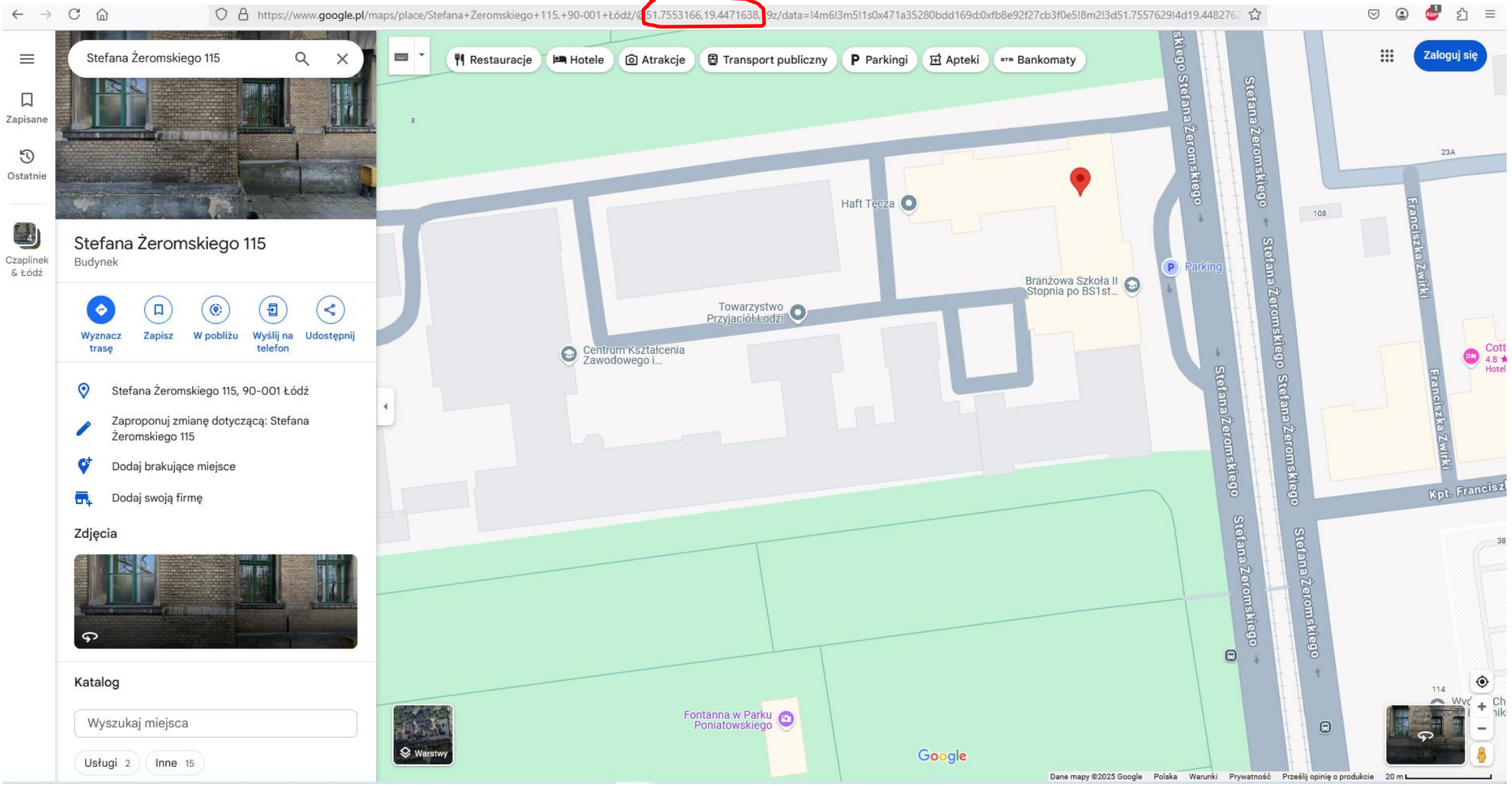
API URL pobieramy ze strony open-meteo.com



google maps

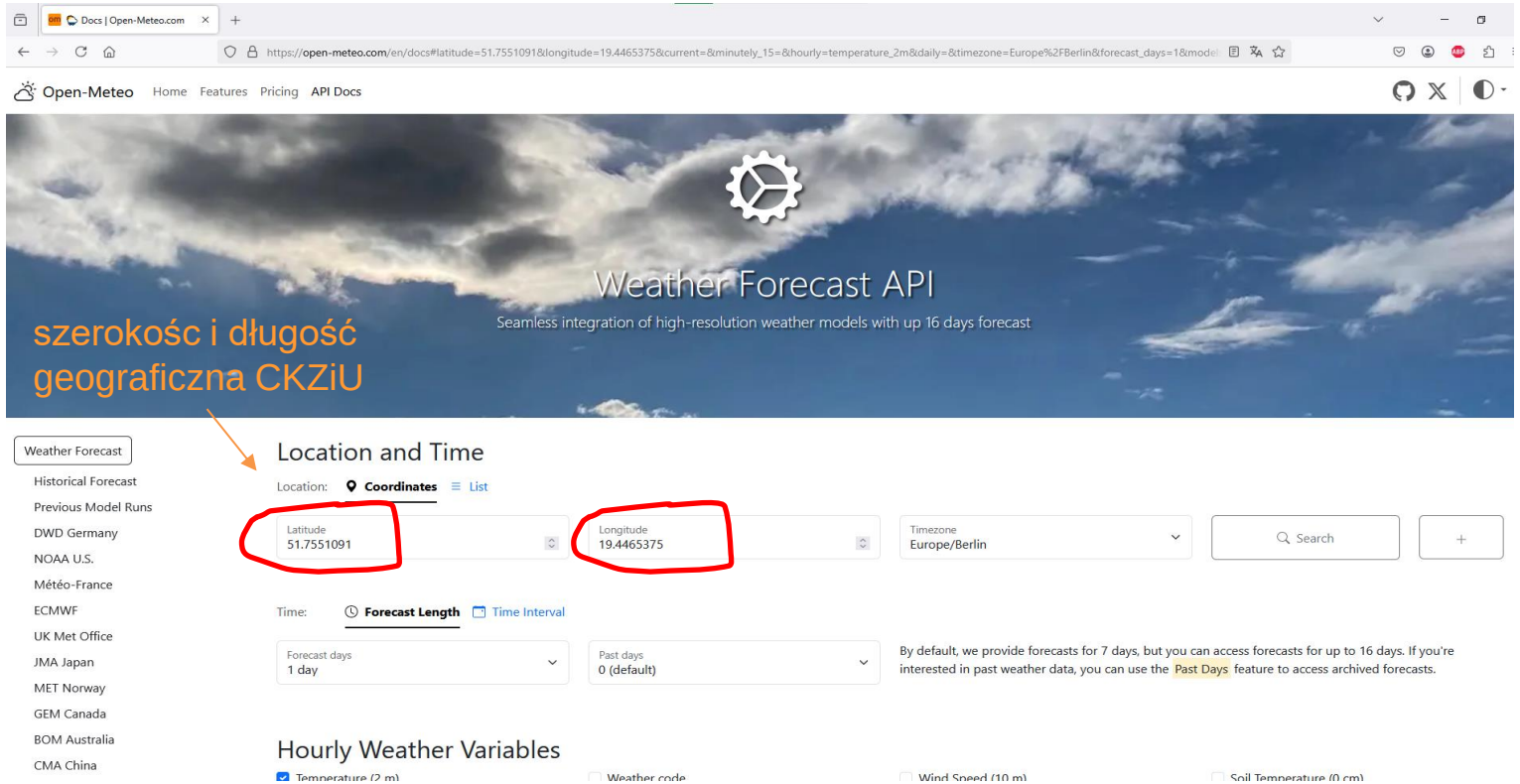
prognoza pogody

szerokośc i długość geograficzna CKZiU



prognoza pogody

open-meteo.com



https://open-meteo.com/en/docs

Download XLSX | Download CSV

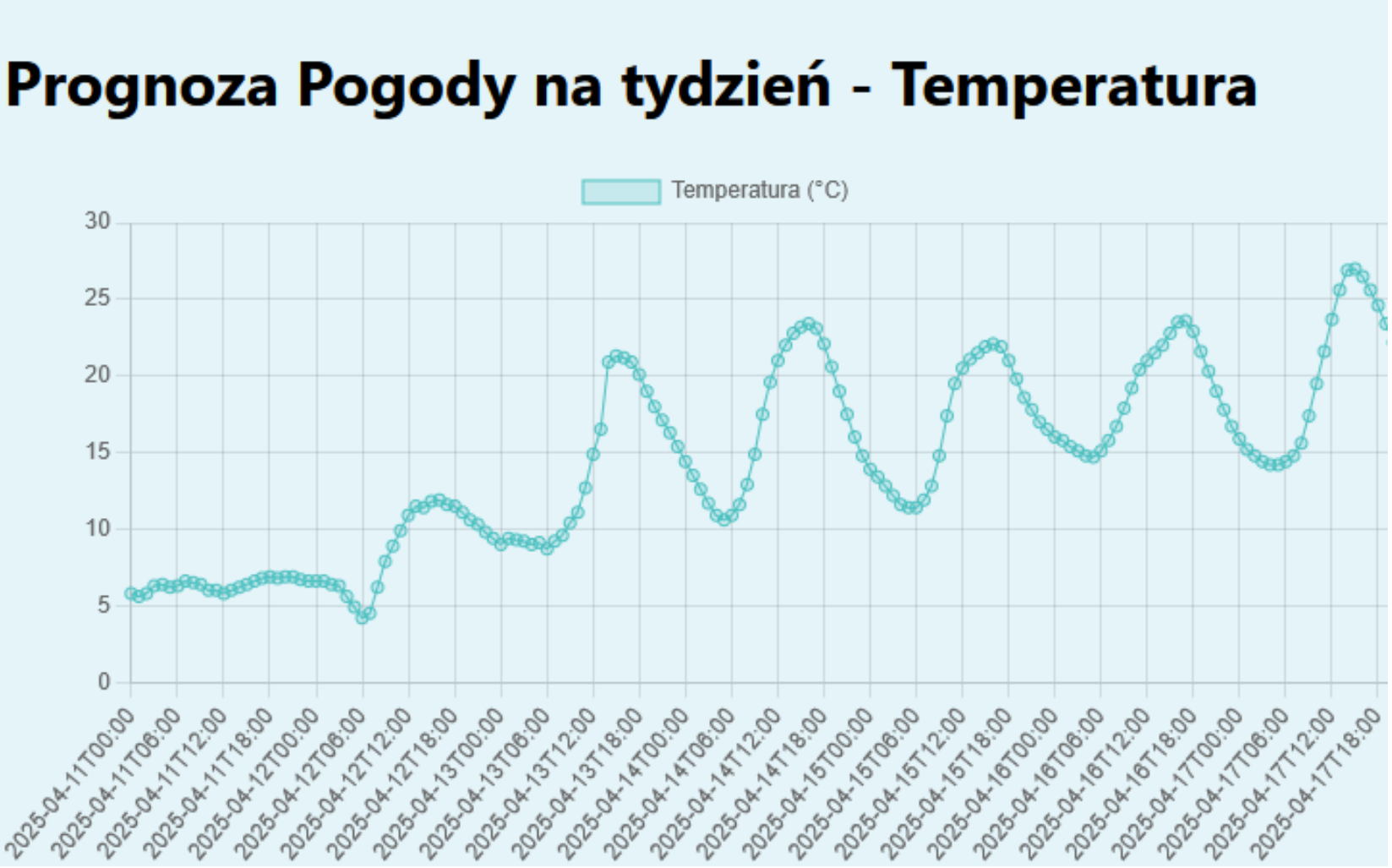
API URL

API URL (Open in new tab or copy this URL into your application)

https://api.open-meteo.com/v1/forecast?latitude=51.7551091&longitude=19.4465375&hourly=temperature_2m&timezone=Europe%2FBerlin&forecast_days=1

prognoza pogody

korzystając z bibliotek graficznych w js zaprezentuj
przesłane dane z open-meteo.com w postaci wykresu



useRef Hook

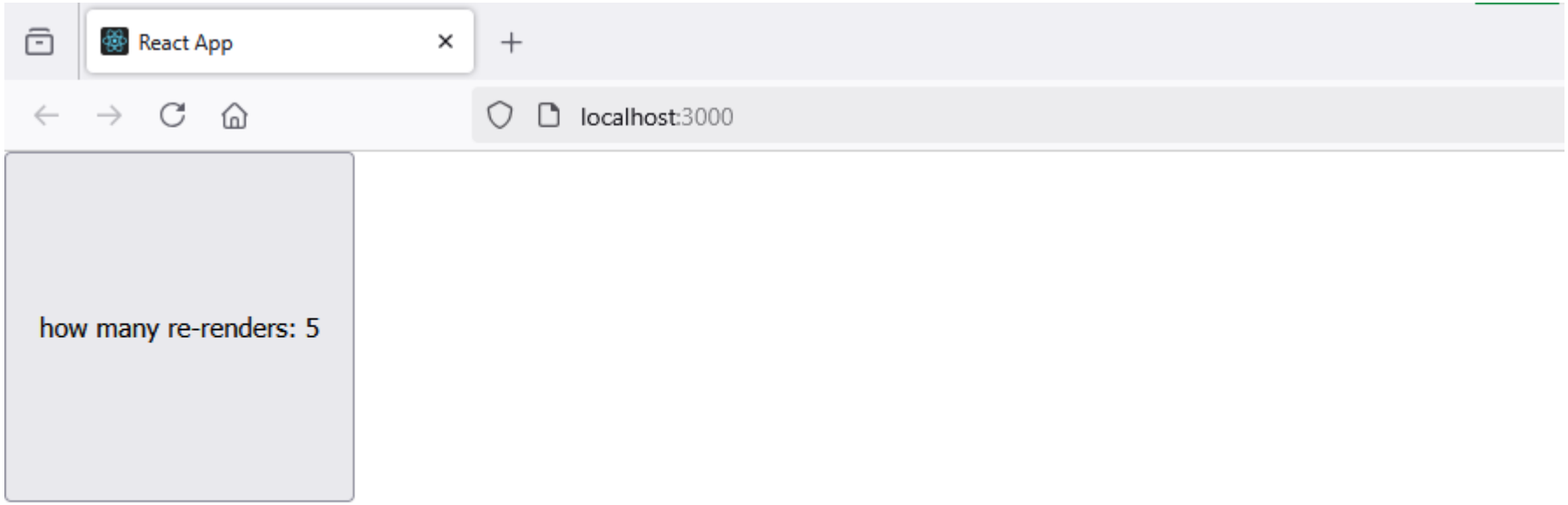
The useRef Hook allows you to persist values between renders.

https://www.w3schools.com/react/react_useref.asp

https://react.dev/reference/react/useRef

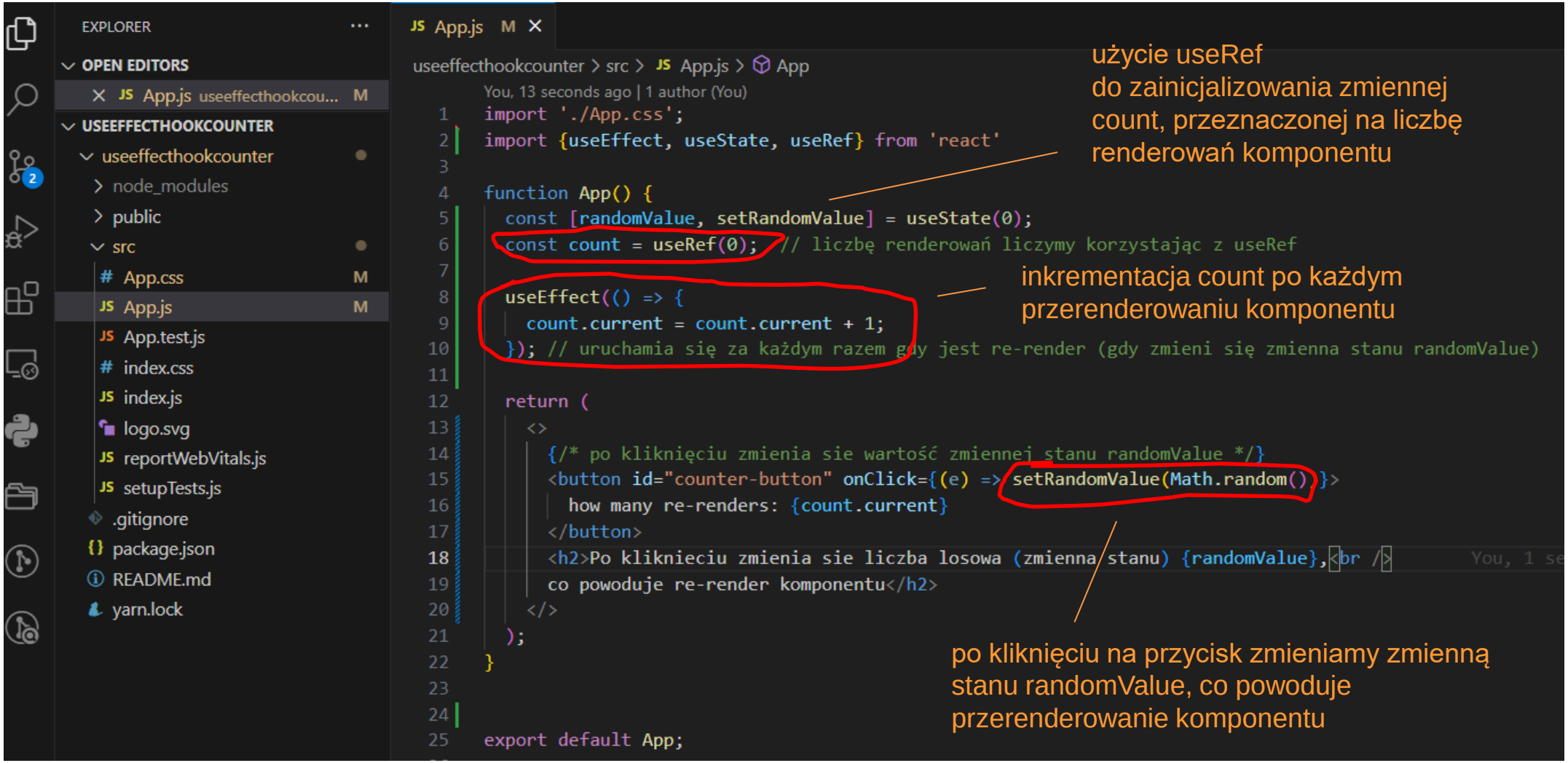
useRef - zapisywanie wartości pomiędzy renderowaniami

liczbę renderowań
liczymy
korzystając z
useRef

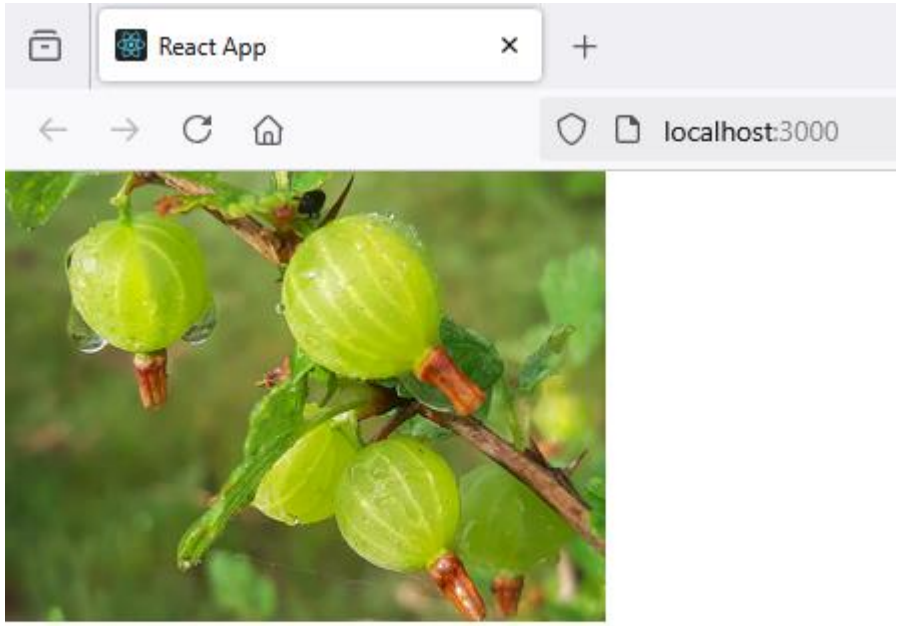


Po kliknieciu zmienia sie liczba losowa (zmienna stanu) 0.4162854043184033, co powoduje re-render komponentu

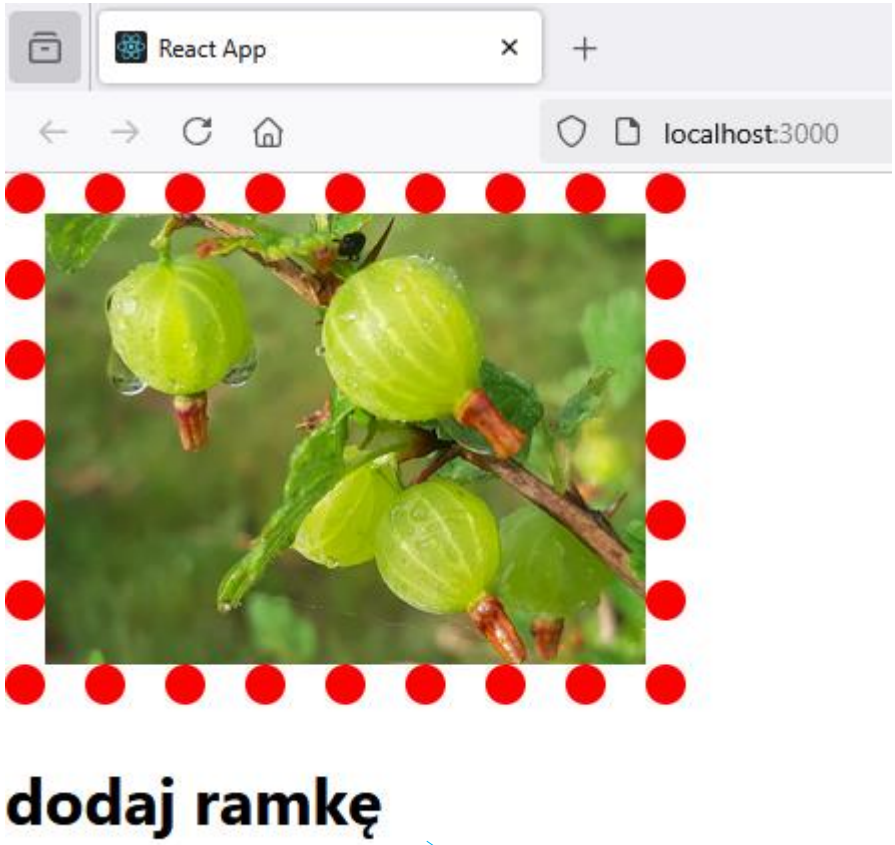
liczba renderowań



useRef – przypisanie referencji do elementu DOM

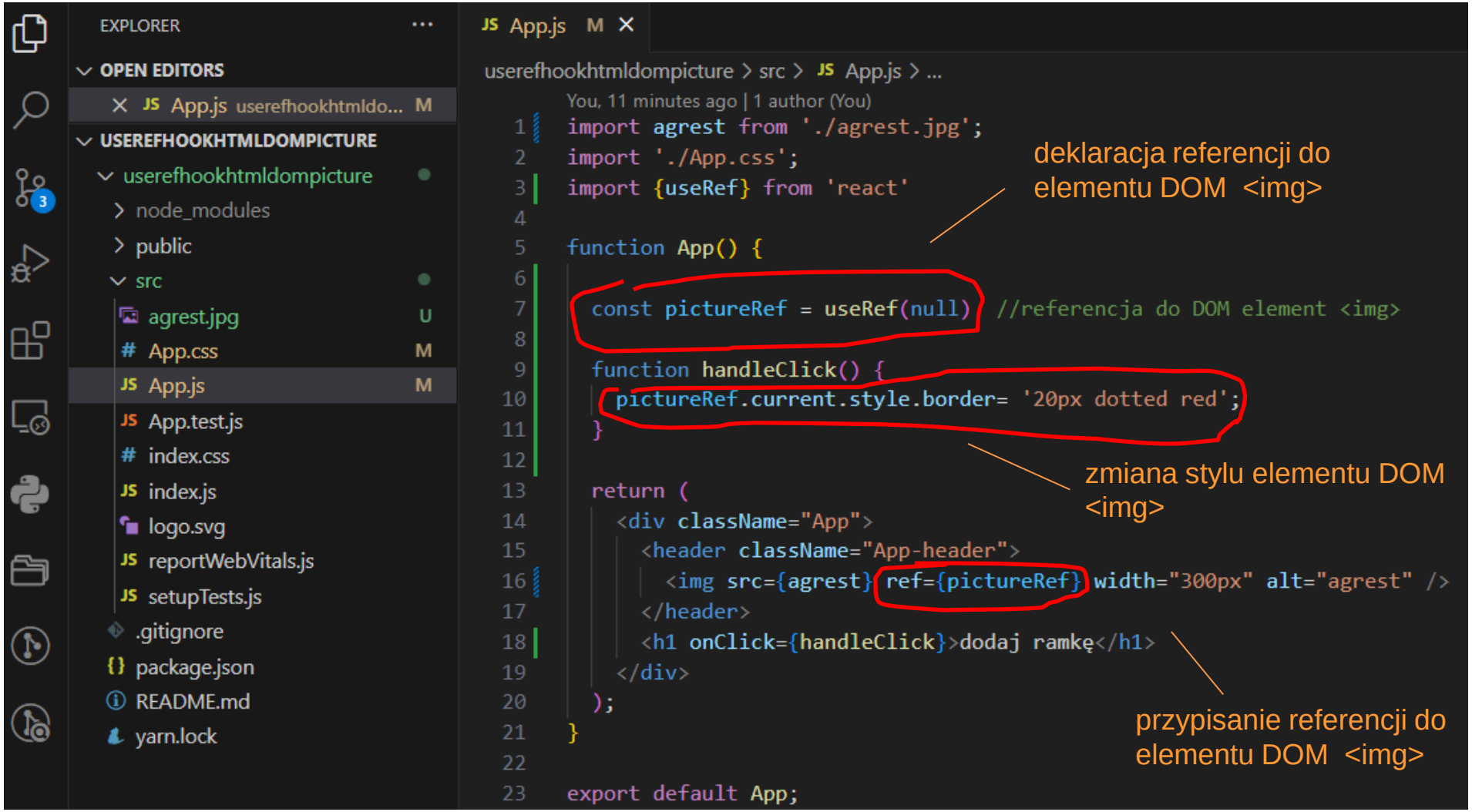


zdjęcie ma atrybut ref dzięki czemu możemy mu przypisać styl

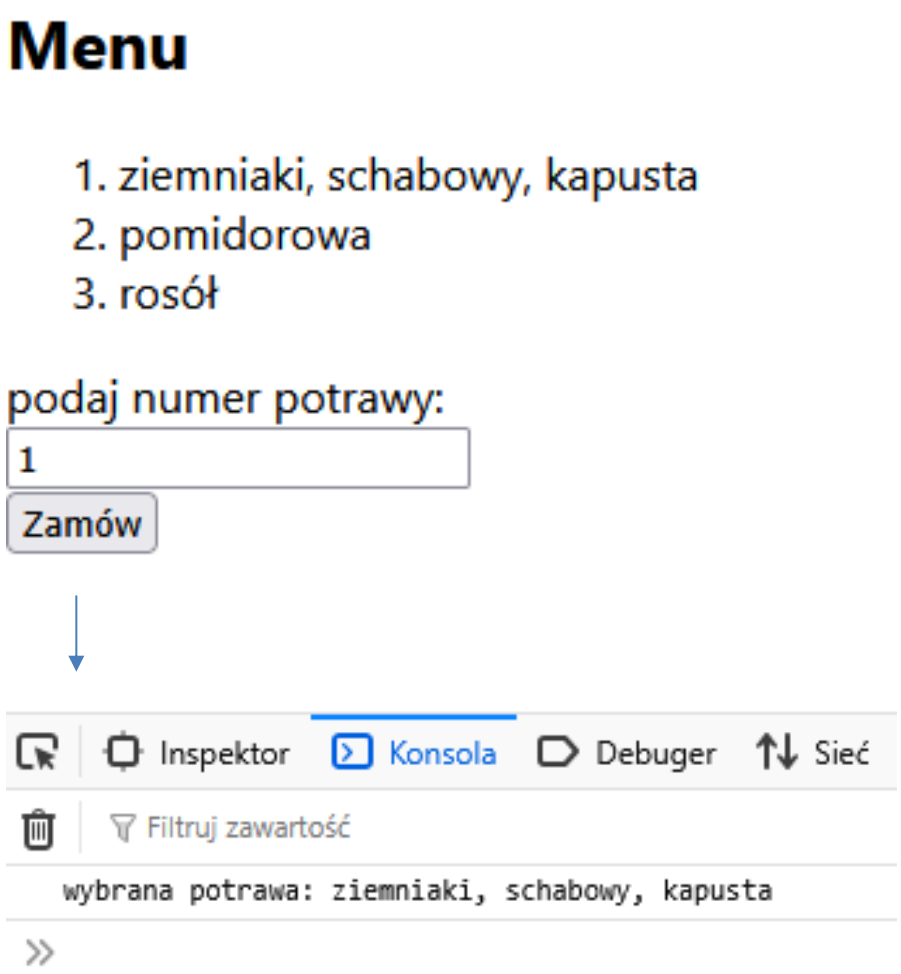


klikając na napis dodajemy do zdjęcia ramkę

ramka


EXPLORER
OPEN EDITORS
App.js userefhookhtmldo... M
USEREFHOOKHTMLDOMPICTURE
userefhookhtmldompicture
node_modules
public
src
agrest.jpg U
App.css M
App.js M
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package.json
README.md
yarn.lock
App.js M
userefhookhtmldompicture > src > App.js > ...
You, 11 minutes ago | 1 author (You)
1 import agrest from './agrest.jpg';
2 import './App.css';
3 import {useRef} from 'react'
4
5 function App() {
6
7 const pictureRef = useRef(null) //referencja do DOM element <img>
8
9 function handleClick() {
10 pictureRef.current.style.border= '20px dotted red';
11 }
12
13 return (
14
15 <header className="App-header">
16
17 </header>
18 <h1 onClick={handleClick}>dodaj ramkę</h1>
19
20);
21 }
22
23 export default App;
deklaracja referencji do elementu DOM <img>
zmiana stylu elementu DOM <img>
przypisanie referencji do elementu DOM <img>

formularz


Menu
1. ziemniaki, schabowy, kapusta
2. pomidorowa
3. rosół
podaj numer potrawy:
1
Zamów
Inspektor
Konsola
Debuger
Sieć
Filtruj zawartość
wybrana potrawa: ziemniaki, schabowy, kapusta

formularz

```jsx
import { useRef } from 'react';

function App() {
  // tablica z menu
  const menu = ['ziemniaki, schabowy, kapusta', 'pomidorowa', 'rosół'];

  // referencje do input
  const nrPotrawyRef = useRef("");

  let wybranaPotrawa = "";
  let nrPotrawy = 0;

  const handleSubmit = (event) => {
    event.preventDefault();
    nrPotrawy = nrPotrawyRef.current.value;
    wybranaPotrawa = menu[nrPotrawy - 1];
    console.log("wybrana potrawa: " + wybranaPotrawa);
  }

   return (
    <div>
      <h2>Menu</h2>
      <ol>
        {menu.map((nazwa) => <li> {nazwa}</li> )}
      </ol>
      <form onSubmit={handleSubmit}>
        <div >
          <label htmlFor='nrPotrawy'>podaj numer potrawy:</label><br />
              <input
                id="nrPotrawy"
                type="text"
                ref={nrPotrawyRef}
              />
          </div>
          <input type="submit" value="Zamów" />
        </form>
      </div>
    );
}

export default App;
```

deklaracja referencji do elementu <input>

pobranie wartości z pola tekstowego formularza po naciśnięciu przycisku „Zamów”

przypisanie referencji do elementu <input>

napisz clickera, który będzie zmieniał prędkość kręcenia się logo

clicker

czas animacji skracamy o 0.1 sekundy po każdym kliknięciu

9.700000000000001s

WTF!!!

!?

Change spin

clicker

useContext Hook

React Context is a way to manage state globally.

It can be used together with the useState Hook to share state between deeply nested components more easily than with useState alone.

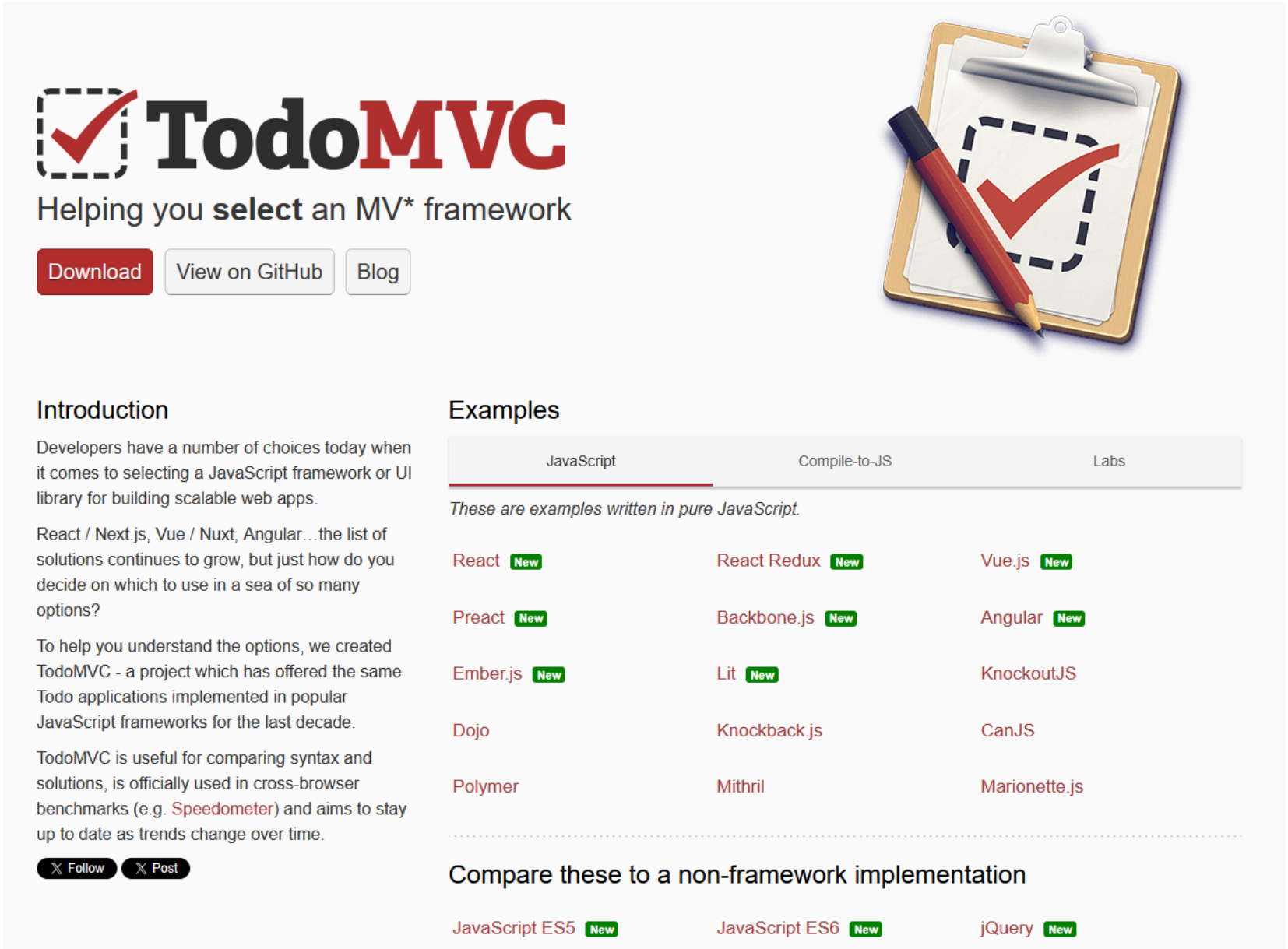
https://www.w3schools.com/REACT/react_usecontext.asp

https://react.dev/reference/react/useContext

useContext Hook

po zapoznaniu się z przykładem
na w3schools: useContext
napisz dowolną aplikacje
korzystającą z useContext Hook

porównanie różnych frameworków do budowy aplikacji webowych



https://todomvc.com/

How to deploy a React App on Apache web server

Firstly, in your react project go to your **package.json** file and adjust this line of code to match your destination domain address + folder:

```
"homepage": "https://yourwebsite.com/your_folder_name/",
```

Secondly, go to **terminal** in your react project and type:

```
npm run build
```

Now, **take all files** from that newly created **build** folder and upload them into **your_folder_name**, with filezilla in subfolder like this:

```
public_html/your_folder_name
```

Check in the browser!

https://stackoverflow.com/questions/42461279/how-to-deploy-a-react-app-on-apache-web-server