

Wprowadzenie do React

Zanim zaczniemy pracę nad własnym projektem, warto odwiedzić oficjalną stronę:

<https://react.dev/learn>

React components are JavaScript functions that return markup:

```
function MyButton() {  
  return (  
    <button>I'm a button</button>  
  );  
}
```

Now that you've declared `MyButton`, you can nest it into another component:

```
export default function MyApp() {  
  return (  
    <div>  
      <h1>Welcome to my app</h1>  
      <MyButton />  
    </div>  
  );  
}
```

Notice that `<MyButton />` starts with a capital letter. That's how you know it's a React component. React component names must always start with a capital letter, while HTML tags must be lowercase.

<https://react.dev/learn/installation>

Try React

You don't need to install anything to play with React. Try editing this sandbox!

App.js

Download Reset Fork

```
1 function Greeting({ name }) {  
2   return ( <div>  
3     <h1>Hello, {name}</h1>  
4     <h2>Nice to see you!</h2>  
5   </div> );  
6 }  
7  
8 export default function App() {  
9   return <Greeting name="world" />  
10 }  
11
```

Rozpoczęcie pracy z projektem

1. Konfiguracja środowiska programistycznego

Zanim zaczniemy pisanie kodu źródłowego, musimy skonfigurować środowisko React przy użyciu **Create React App**.

Kroki:

1. Otwórz **WebStorm**.
2. Kliknij **Create New Project**.
3. Wybierz **React App**.
4. Wybierz lokalizację projektu i kliknij **Create**.
5. WebStorm automatycznie uruchomi `npx create-react-app my-app`.
6. Po zakończeniu konfiguracji przejdź do folderu projektu i uruchom serwer:
7. `cd my-app`
8. `npm start`

Spowoduje to otwarcie `http://localhost:3000/` w przeglądarce z domyślną aplikacją React.

2. Struktura projektu

- `node_modules/` – Zawiera wszystkie zainstalowane zależności.
- `public/` – Pliki statyczne (np. `index.html`).
- `src/` – Główny kod źródłowy:
 - `App.js` – Główny komponent.
 - `index.js` – Punkt wejściowy aplikacji.
 - `App.css` – Style dla `App.js`.
- `package.json` – Zarządza zależnościami i skryptami projektu.

3. Pierwszy komponent React

Utwórz nowy plik `Welcome.js` w folderze `src/`.

```
import React from 'react';

function Welcome() {
  return <h1>Witamy na pierwszych zajęciach z React!</h1>;
}

export default Welcome;
```

Teraz zaimportuj i użyj tego komponentu w App.js:

```
import React from 'react';
import Welcome from './Welcome';

function App() {
  return (
    <div>
      <Welcome />
    </div>
  );
}

export default App;
```

4. Ćwiczenia praktyczne

Ćwiczenie 1: Modyfikacja komponentu Welcome

- Zmodyfikuj Welcome.js, aby przyjmował prop name i dynamicznie go wyświetlał.

```
import React from 'react';

function Welcome({ name }) {
  return <h1>Witamy, {name}!</h1>;
}

export default Welcome;
```

Zmodyfikuj App.js, aby przekazywać prop name:

```
import React from 'react';
import Welcome from './Welcome';

function App() {
  return (
    <div>
      <Welcome name="Student" />
    </div>
  );
}

export default App;
```

Użyj komponentu wielokrotnie, przekazując różne parametry, np:

```
import React from 'react';
import Welcome from './Welcome';

function App() {
  return (
    <div>
      <Welcome name="Student" />
      <Welcome name="John" />
      <Welcome name="Anna" />
    </div>
  );
}

export default App;
```

Ćwiczenie 2: Utworzenie komponentu wyświetlającego listę języków programowania

- Utwórz nowy plik `ProgrammingLanguages.js`.
- Wyświetl prostą, statyczną listę języków programowania.

```
import React from 'react';

function ProgrammingLanguages() {
  return (
    <div>
      <h2>Lista języków programowania niezbędnych do pracy z React:</h2>
      <ul>
        <li>JavaScript</li>
        <li>CSS</li>
        <li>HTML</li>
      </ul>
    </div>
  );
}

export default ProgrammingLanguages;
```

Następnie, zaimportuj `ProgrammingLanguages.js` w `App.js`:

```
import React from 'react';
import Welcome from './Welcome';
import ProgrammingLanguages from './ProgrammingLanguages';
```

```
function App() {  
  return (  
    <div>  
      <Welcome name="Student" />  
      <Welcome name="John" />  
      <Welcome name="Anna" />  
      <ProgrammingLanguages />  
    </div>  
  );  
}  
  
export default App;
```

Zadania do samodzielnej pracy:

- 1) Spróbuj utworzyć nowy komponent Message.js, który dynamicznie wyświetla wiadomość na podstawie przekazanych props. Zaimportuj go i użyj w App.js.
- 2) Dodaj komponent odpowiedzialny za umieszczenie na stronie nagłówka z nazwą przedmiotu i linkiem do strony internetowej naszego wydziału oraz do strony głównej Reacta.
- 3) Przygotuj komponent przedstawiający listę języków programowania z którymi spotkałeś się już podczas studiów.
- 4) Dodaj komponent z listą przedmiotów na jakie uczęszczasz w tym semestrze.
- 5) Umieść na stronie przycisk „Więcej o przedmiocie” (należy umieścić jedynie sam przycisk, na razie nie wykonuje on żadnej akcji).