Introduction to React

Before we start working on our first project, it good to look into the official react website:

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:



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

Working on react app project

1. Setting Up the Development Environment

Before we start coding, we need to set up our React environment using **Create React App**.

Steps:

- 1. Open WebStorm.
- 2. Click on **Create New Project**.
- 3. Select React App.
- 4. Choose the project location and click **Create**.
- 5. WebStorm will automatically run npx create-react-app my-app.
- 6. Once the setup is complete, navigate to the project folder and start the development server:
- 7. cd my-app
- 8. npm start

This will open http://localhost:3000/ in your browser with a default React app running.

2. Understanding the Project Structure

- node_modules/ Contains all installed dependencies.
- public/ Static files (like index.html).
- src/ Main source code:
 - App.js Main component.
 - index.js Entry point of the application.
 - App.css Styling for App.js.
- package.json Manages project dependencies and scripts.

3. First React Component

Create a new component named Welcome.js in the src/ folder.

```
import React from 'react';
```

function Welcome() {

return <h1>Welcome to your first React class!</h1>;

```
}
```

```
export default Welcome;
```

Now, import and use this component inside App.js:

import React from 'react';

import Welcome from './Welcome';

```
function App() {
```

return (

<div>

```
<Welcome />
```

</div>

```
);
```

}

```
export default App;
```

4. Hands-On Exercises

Exercise 1: Modify the Welcome Component Welcome.js to accept a name prop and display it dynamically.

```
import React from 'react';
```

function Welcome({ name }) {

```
return <h1>Welcome, {name}!</h1>;
```

```
}
```

export default Welcome;

```
Modify App.js to pass a name prop:

import React from 'react';

import Welcome from './Welcome';

function App() {

return (

<div>

<Welcome name="Student" />

</div>

);

}

export default App;
```

Use it in App.js multiple times using different name values:

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;

Exercise 2: Create a Component with list - Displaying a List of Programming Languages

- Create a new file ProgrammingLanguages.js.
- Display a simple, static list of programming languages required to know to write code with React.

```
import React from 'react';
```

```
function ProgrammingLanguages() {
```

```
return (
```

<div>

```
<h2>List of Programming Languages:</h2>
```

```
JavaScript
```

CSS

```
HTML
```

</div>

```
);
```

}

export default ProgrammingLanguages;

Then, import ProgrammingLanguages.js inside App.js:

import React from 'react';

import Welcome from './Welcome';

import ProgrammingLanguages from './ProgrammingLanguages';

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

Tasks For Practice:

- 1) Try to create a new component Message.js that dynamically displays a message based on props. Import and use it in App.js.
- 2) Add a component responsible for displaying a header on the page with the course name and links to the website of our department as well as the official React homepage.
- 3) Create a component that presents a list of programming languages you have learned during your studies so far.
- 4) Add a component with a list of subjects you are attending this semester.
- 5) Place a **"More about the course"** button on the page (for now, it should only be a button without any functionality).