**Basic exercises jsx**

**Slot 1-2-3-4**

**Created by traltb@fe.edu.vn**

**Bước 1: Cài đặt nodejs:**

<https://nodejs.org>

Cài đặt xong kiểm tra thử node đã cài đặt ok,

Mở command line, gõ node -v

Xem phiên bản node

Tiếp tục gõ: npm -v

Xem phiên bản của npm (Node Package Manager)

Node và npm là 2 tool để phát triển, chạy ứng dụng javascript, nodejs

**Bước 2: Tạo ứng dụng react**

Mở command line, gõ lệnh:

npm install -g create-react-app

Lệnh này gõ 1 lần thôi, để tạo ứng dụng react.

**Bước 3: Bắt đầu xây dựng ứng dụng react**

Vào ổ D:\, tạo thư mục FER202, Code, Exercises để tạo ra đường dẫn trên máy tính như sau: D:\FER202\Code\Exercises

Tạo mới 1 thư mục AtClass trong thư mục này. Click phải ở thư mục này, chọn Open In Terminal

Gõ lệnh:

npx create-react-app exercise1

Sau khi chạy xong lệnh này, sẽ tạo ra 1 thư mục exercise1 trong thư mục AtClass, tiếp tục thực hiện lệnh chuyển thư mục, về làm việc với thư mục exercise1, Gõ lệnh:

Cd exercise1

Gõ lệnh sau nếu gói web-vitals nếu bị lỗi:

npm install web-vitals

Chạy ứng dụng, gõ:

npm start

Kết quả theo hình:

A blue and white symbol with a circle in the middle

AI-generated content may be incorrect.

Nếu trong quá trình chạy có bị lỗi này:

npm.ps1 cannot be loaded because running scripts is disabled on this system

Vào tìm kiếm trên máy tính, gõ PowerShell

Click phải, chạy với quyền Admin, chọn Run as Administrator:

Gõ lệnh sau:

Set-ExecutionPolicy RemoteSigned

Sau đó chạy lại lệnh: npm start

**Bước 4: Hiệu chỉnh file code jsx**

Mở Visual studio code, Chọn File->Open Folder. Duyệt đến thư mục exercises1

Vào Terminal, chọn New Terminal,

Nhấn Ctrl+ C để dừng server

Mở file App.js: hiệu chỉnh code như sau:

**Cách 1**: Thêm thuộc tính style vào thẻ h1, lưu ý dùng 2 cặp {}, và thuộc tính Viết hoa chữ cái thứ 2:

import "./App.css";

function App() {

  return (

    <div>

      <h1 style={{ color: "blue", textAlign: "center" }}>Hello world!</h1>

    </div>

  );

}

export default App;

Cách 2:

import "./App.css";

function App() {

  const h1Style = {

    color: "blue",

    textAlign: "center",

  };

  return (

    <div>

      <h1 style={h1Style}>Hello world!</h1>

    </div>

  );

}

export default App;

Cách 3: Tạo class trong file App.css

.h1Style {

  color: blue;

  text-align: center;

}

Khi đó nội dung file App.js là:

import "./App.css";

function App() {

  return (

    <div>

      <h1 className="h1Style">Hello world!</h1>

    </div>

  );

}

export default App;

Bài tập tương tự:

**Exercise 1: Display a list of names**

Cho trước 1 mảng các tên của người, hiển thị danh sách này dưới dạng danh sách không sắp xếp thứ tự (Given an array of names, display them in an unordered list)

Tạo mới 1 tập tin NamePerson.js trong thư mục src, lưu ý chỉ có 1 thẻ div cha

function NamePerson() {

  const names = ["Alice", "Bob", "Charlie"];

  return (

    <div>

      <ul>

        {names.map((name, index) => (

          <li key={index}> {name}</li>

        ))}

      </ul>

    </div>

  );

}

export default NamePerson;

Sau đó thêm component NamePerson vào trong file index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App";

import reportWebVitals from "./reportWebVitals";

import NamePerson from "./NamePerson";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

  <React.StrictMode>

    <App />

    <NamePerson />

  </React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

Reload lại trang web.

A white background with black and white clouds

AI-generated content may be incorrect.

Có thể viết file NamePerson.js theo cách này:

function NamePerson() {

  const names = ["Alice", "Bob", "Charlie"];

  const element = (

    <ul>

      {names.map((name, index) => (

        <li key={index}>{name}</li>

      ))}

    </ul>

  );

  return element;

}

export default NamePerson;

**Exercise 2: Display a person's details**

Given a person object, display their name, age, and occupation.

Using the above people array, do the following exercises:

**Exercise 3:** Display a list of people (PeopleList.js)

**Exercise 4:** Display a table of people

**Exercise 5:** Find the first teenager (FirstTeenager.js)

**Exercise 6:** Check if all are teenagers (AreAllTeenagers.js)

**Exercise 7:** Sort by Occupation and Then by Age

Write a function to sort the people array first by occupation alphabetically and then by age in ascending order within each occupation group.

B1: Tạo file PeopleSort.js, khởi tạo hàm PeopleSort, Destructuring Prop nhận danh sách đối tượng thông qua prop {people}



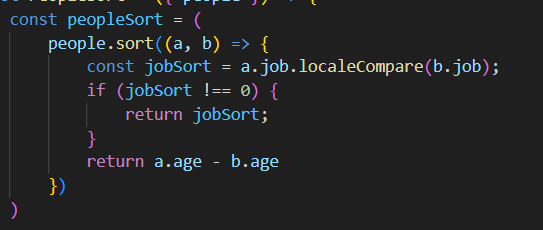
B2: Tạo biến peopleSort

Sau đó sử dụng phương thức sort với people với 2 tham số a và b

Sử dụng phương thức localeCompare để so sánh chuỗi ở đây là job

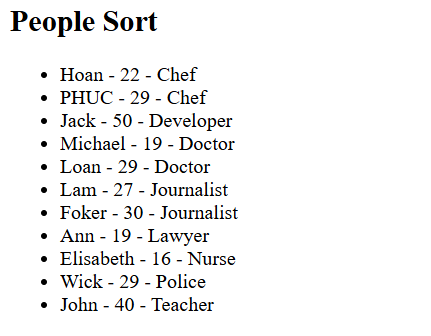
Nếu khác nhau thì sẽ trả về 1 giá trị sắp xếp theo bản chữ cái

Tiếp theo nếu như trùng job thì sẽ tiếp tục trả về sắp xếp theo tuổi



B3: trả về JSX, xuất và import gọi và truyền prop





**Exercise 8:** Group People by Occupation

Write a function that groups people by their occupation into an object. Each key is an occupation, and the value is an array of people with that occupation.

B1: Tương tự khác tên là GroupByOccupation

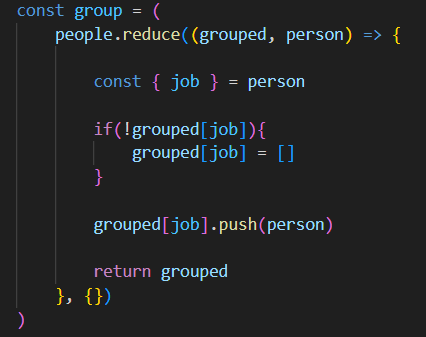


B2: sử dụng reduce để nhóm người lại theo job của họ và có 2 tham số là grouped (mảng tích lũy ) và person (một đối tượng trong people)

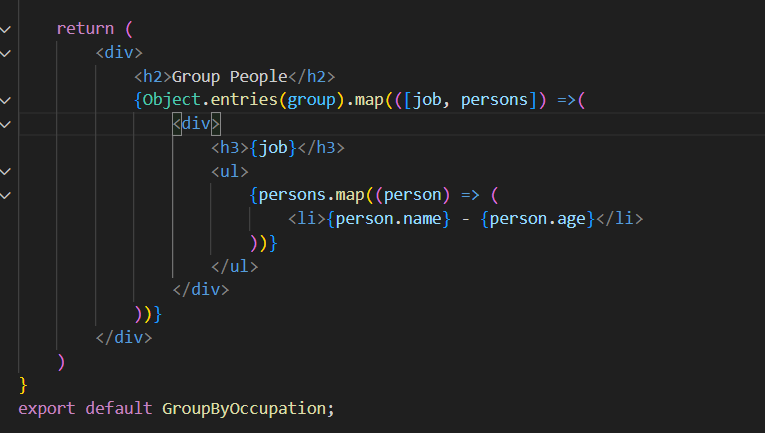
Lấy thuộc tính job từ person

Kiểm tra đối tượng grouped nếu chưa có khóa job thì khởi tạo một mảng chứa những người có job đó

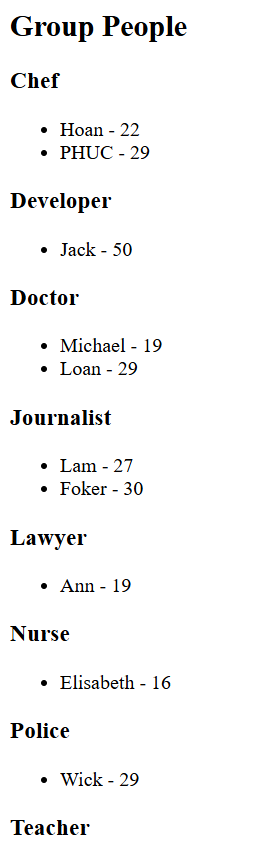
Thêm người vào trong mảng với công việc tương ứng và trả về grouped sau khi đã duyệt qua hết tất cả



B3: Trả về JSX , dùng Object.entries(group) để chuyển nó thành mảng có các cặp key value, key là job còn value là persons sau đó dùng map duyệt qua từng person và in ra theo các nhóm tương ứng







**Exercise 9:** Find the Oldest and Youngest Person

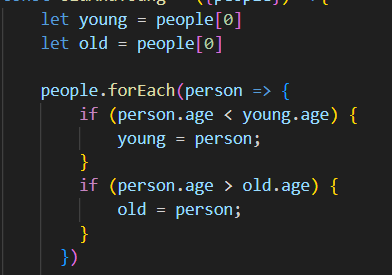
Create a function to find and display the oldest and youngest person in the array.

B1: Tương tự khác tên là OldAndYoung

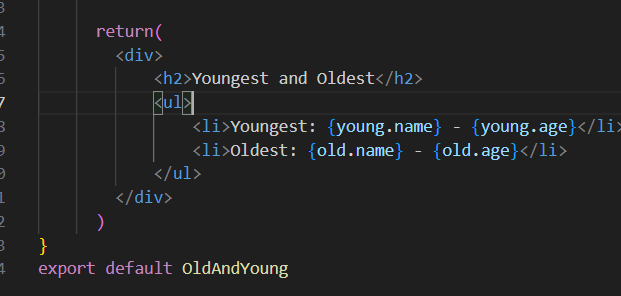


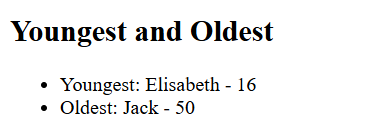
B2: khởi tạo old và yound là 2 giá trị đầu tiên trong people

Sau đó sẽ dùng forEach duyệt qua people, sử dụng if else cơ bản để tìm ra ai lớn tuổi nhất và nhỏ tuổi nhất



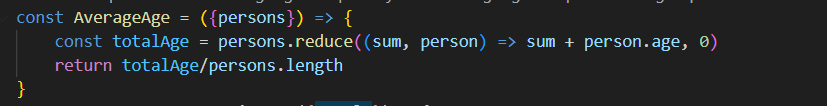
B3: Trả về JSX xuất import, gọi truyền prop và chạy





**Exercise 10:** Calculate the Average Age of Each Occupation

Write a function that calculates and displays the average age for each occupation.

Tương tự như câu 8 nhưng bây giờ nó sẽ có thêm 1 hàm để tính trung bình tuổi  


Truyền vào prop persons được phân ra thành key và value từ Object.entries ở dưới

Sau đó trả về ở bên return giá trị trung bình 

Tiếp theo xuất, import và chạy

