Intro to React

1. What is React?

React is a JavaScript library for building user interfaces. It focuses on:

- **Component-based architecture**: Breaking the UI into small, reusable pieces called components.
- **Efficient rendering**: Using a concept called the Virtual DOM to update the user interface without reloading the entire page.
- **Declarative programming**: Developers specify what they want the UI to look like, and React handles the how.

Why Use React?

- **Ease of Development**: Write reusable code using components.
- **Performance**: React updates only the parts of the UI that need changes.
- **Flexibility**: React can be used for web, mobile (via React Native), and even desktop apps.
- **Large Ecosystem**: Thousands of tools and libraries support React development.

Who Created React?

React was developed by Facebook (now Meta) and open-sourced in 2013. It powers many popular applications like Facebook, Instagram, Netflix, and Airbnb.

The Virtual DOM

The Virtual DOM is like a copy of the real DOM (the structure of the web page) that React uses behind the scenes.

Here's a simple explanation:

1. What is the DOM?
   The DOM (Document Object Model) is the structure of the web page that the browser reads and shows to users. Whenever you update the DOM (like adding or removing elements), it takes time because the browser has to redraw parts of the page.

2. What is the Virtual DOM?
   The Virtual DOM is a lightweight version of the real DOM that exists in memory. React uses this to figure out the smallest, most efficient way to update the real DOM.

3. How Does it Work?
   - When you change something in your React app, React updates the Virtual DOM first.
   - React compares the new Virtual DOM to the old one.
   - React calculates the smallest changes needed and updates only those parts of the real DOM.
4. Why is it Fast?
   Instead of updating the whole page, React updates just the specific parts that changed. This makes it much faster and smoother.

Example:
Imagine a to-do list app where you add an item.

- Without the Virtual DOM, the browser would check and update the entire page.
- With the Virtual DOM, React sees that only one new item was added and tells the real DOM to update just that one part.

2. What is JSX (JavaScript XML)?

JSX is a **syntax extension for JavaScript** that looks like HTML. It allows you to write HTML-like code directly inside JavaScript.

Why JSX?

1. **Readability**: It's easier to understand UI structure compared to raw JavaScript.
2. **Declarative**: You can define how the UI should look and React will handle rendering.
3. **Integration with JavaScript**: You can use JavaScript logic (like variables or functions) directly in your UI code.

How JSX Works

- Behind the scenes, JSX is converted to plain JavaScript.
- For example:

```
function App() {
    const name = "Victor";
    return <h1>Hello, {name}!</h1>; // this is jsx (html like syntax)
}
```

`{name}` embeds a JavaScript variable inside JSX.
The curly braces `{}` allow you to insert dynamic JavaScript expressions.

3. Prerequisites for React

Before starting with React, you should be familiar with:

1. **JavaScript Fundamentals**:
    - Variables (`let`, `const`)
    - Functions (especially Arrow Functions)
    - ES6 Features (like Destructuring and Spread Syntax)
    - Array Methods (like `.map()` and `.filter()`)
    - Modules (`import`/`export`)

2. **HTML and CSS**:
    - Basic understanding of building web pages.

Tools Needed

1. **Node.js** (and npm or yarn):
    - React uses Node.js to run its development server and manage dependencies.
    - Install it from Node.js official website.
2. **A Code Editor**:
    - (VS Code).

3. **Browser**:
    - A modern browser like Chrome or Firefox (dev edition) to test your app.

N.B:

1. Run the following command in your terminal to create a new React app (replace my-app with your desired app name):

    npx create-react-app my-app

    cd my-app

2. Refer to the React Quick Start guide: https://react.dev/learn