**[NOTES] React fundamentals - most important definitions**

The name **React** comes from the English word "react," meaning to respond. **React** is a JavaScript library that allows you to create web pages that **react** to user actions, such as clicks or typing. With **React**, we can create dynamic, interactive applications that respond immediately to changes.

**JSX:** JSX stands for **"JavaScript XML."** The name reflects the combination of JavaScript and XML (which has a structure similar to HTML). **JSX** allows you to write code that looks like HTML but works in JavaScript, making it easier to create user interfaces.

**Component:** The word **"component"** comes from the English word "component," meaning a part or element of a larger whole. In **React**, a **component** is a part of a larger entity, such as a web page. Each **component** is a separate element that can be reused and combined with others, like LEGO bricks.

**Hook (useState):** The name **"hook"** means a hook or anchor, because **hooks** in **React** are used to "hook" functions to a component. **useState** is a **hook** that not only hooks but also connects a function to a specific state of a variable. This allows us to not only remember a value (state) but also easily change it and update the user interface when that state changes.

When you use **useState**, you do two things:

* Declare a state — **useState(0)** creates a state variable that starts with a value of 0.
* Connect a function to change this state to the variable.

**Rendering Components:** **Rendering** comes from the word "render," which means to present or display something. In **React**, **rendering** is the process by which **components** are displayed on the page, turning code into visible elements.

**Event Handling:** **"Event"** means an occurrence, and **"handling"** means managing. **Event handling** is the process by which **React** manages different events (e.g., clicks) on the page and responds to them appropriately.

**ReactDOM:** **"DOM"** stands for **"Document Object Model,"** which is the structure of a web page in the browser. **ReactDOM** is a tool that connects **React** to this structure, enabling the display of components on the page.

**State:** **"State"** refers to a state or condition. In **React**, **state** is a value that a component "remembers" and that can change over time, such as the number of button clicks.

**Arrow Function (()=>{}):** This is a modern way of writing functions in JavaScript, more concise and readable.

**Destructuring (const [a, b] = [1, 2];):** This is a way to quickly "unpack" values from arrays or objects, making it easier to work with data.

**React.createRoot:** **"CreateRoot"** means to create a root. It’s a function that creates the main point from which **React** begins to build the component tree on the page.

**Virtual DOM:** **"Virtual"** means virtual (something that exists only in memory), and **"DOM"** is the structure of the page. The **Virtual DOM** is a virtual version of the page that **React** uses to quickly respond to changes, updating only what’s necessary without reloading the entire page.