Javascript basic functions for HTML

- document.createElement( )
- document.getElementById( )
- document.appendChild( )

React Introduction **(React is a Library)**

Namaste React (Episode 01 Inception)

Create element using

React.createElement(“tag name”, { attributes dictionary } , “children”)
React.createElement(“h1”, { id: “heading” } , “Hello from React”)

- createElement does not return an HTML element.
- createElement returns a React Element (Javascript object)

- The object returned has a field : props
- props are children that go inside the tag.

React needs root for rendering and DOM manipulation.

ReactDOM.createRoot(document.getElementById(“root”))

To render inside root, use .render( ) function

root.render(whatever you want to render)

Converts ReactElement (Javascript obj) => HTMLElement (in DOM)

root.render( ) **replaces the entire content of root** with passed content.

Only affects the root. Other elements are untouched.

Note: Most costly operation in browser: DOM Manipulation
React comes with philosophy of manipulating DOM using Javascript.

Order of script tag matters ? Yes

- **Basic Principle:** HTML documents are parsed from top to bottom.

- **Implication:** Scripts are loaded and executed in the order they appear in the HTML document.

- **Example:** If script1.js is placed before script2.js, script1.js will be loaded and executed first.

- **Issue:** Some scripts depend on others to function correctly.

- **Consequence:** If a dependent script is loaded before its dependency, it may result in errors.

- **Best Practice:** Place dependent scripts after their dependencies to ensure correct functionality.

Add script tag in <head> ? No.

- **Reason**: By default, script tags block HTML parsing while they load and execute. Delay page rendering.

- **Solution:** Place scripts at the bottom of the <body> tag to improve page load times.

Jsx

JSX is syntactic sugar to create React Elements in simple way.