

### 1. What is Emmet ?

**Emmet** is a plugin built into many modern code editors like Visual Studio Code. It allows developers to write shorthand abbreviations that expand into full HTML (or CSS) code snippets. For example, typing `html:5` and hitting Tab generates the complete HTML5 boilerplate. Emmet significantly speeds up frontend development by reducing repetitive typing.

### 2. Difference between a Library and Framework?

**A library** is a collection of reusable functions or modules that you can call when needed, giving you full control over when and how to use it. **A framework**, on the other hand, provides a structured foundation with built-in rules and flow; it calls your code at specific points. The main difference between the two is “Inversion of control”. In short: *with a library, you call the code; with a framework, the code calls you.*

### 3. What is CDN? Why do we use it?

**CDN (Content Delivery Network)** is a network of distributed servers that deliver static assets like scripts, images, and videos to users based on their geographical location. It improves load times and performance by serving content from the server closest to the user. We commonly use CDNs by including their URLs in the `<script>` or `<link>` tags in HTML. Technologies like **Anycast** and **Geolocation** help route requests to the nearest CDN node, ensuring faster content delivery.

### 4. Why is React known as React?

**React** is named so because it “reacts” to changes in data or state. When the state of a component changes, React efficiently updates and re-renders only the parts of the UI that need to change. This reactive behaviour is a core principle of the library, making UI updates seamless and dynamic.

### 5. What is crossorigin in script tag?

**corsorigin** is an attribute used in the `<script>` tag (and other resource-loading tags like `<link>` and `<img>`) when loading resources from a different origin (domain). It controls how the browser handles cross-origin requests and whether credentials like cookies or HTTP authentication are included.

There are three common values:

- **anonymous** – Loads the resource without credentials (cookies, HTTP auth). This is the default behavior for many CDNs.
- **use-credentials** – Sends credentials with the request. The server must respond with the appropriate CORS headers.

It's primarily used for **CORS (Cross-Origin Resource Sharing)** compliance and to allow things like **Subresource Integrity (SRI)** to function properly when loading external scripts.

6. What is difference between React and ReactDOM ?

**React** is the core library that provides the fundamental building blocks for creating user interfaces — things like components, state, props, and the virtual DOM. It is platform-agnostic, meaning it can be used to build apps for various environments such as web, mobile (React Native), or even desktop.

**ReactDOM** is a specific package that enables React to work in the **web browser**. It provides DOM-specific methods (like ReactDOM.render) that allow React components to be rendered to the actual browser DOM.

7. What is difference between react.development.js and react.production.js files via CDN?

The difference between react.development.js and react.production.js lies in how they are optimized for different environments:

- **react.development.js** is the unminified version of React used during development. It includes helpful warnings, detailed error messages, and developer tools to aid debugging and development.
- **react.production.js** is the optimized, minified version used in production environments. It strips away all warnings, error messages, and extra debugging information to reduce file size and improve performance.

**In short:**

Use react.development.js for **development** (easier debugging),

Use react.production.js for **production** (faster performance).

**async vs defer — Key Differences:**

Both async and defer are used in <script> tags to load JavaScript **without blocking** the HTML parsing, but they work differently:

The async and defer attributes in a <script> tag are used to load JavaScript without blocking HTML parsing, but they work differently. Scripts with async are downloaded in parallel and executed immediately after they finish downloading, which can pause HTML parsing and may cause scripts to run out of order. In contrast, defer also downloads scripts in parallel, but defers execution until after the HTML document has been completely parsed, ensuring scripts execute in the order they appear. Use async for independent scripts like analytics, and defer when script order matters or when scripts depend on the DOM.