**Chapter 01 – Inception :**

**✅ 1. What is Emmet?**

**Answer**:  
Emmet is a **shortcut tool** used in VS Code (and other editors) to write HTML faster using abbreviations.

**Example**:  
Typing div>h1+p → expands to:

🧠 Think of it as **autofill for HTML**.

**✅ 2. Difference between a Library and a Framework?**

🔹 What is a Library?

A library is a collection of reusable code/functions that you can call when you need them.  
You are in control — you decide when and how to use it.

📌 React is a Library.

🔹 What is a Framework?

A framework gives you a structure to build applications.  
It is in control — it calls your code at the right time and handles flow for you.

📌 Angular is a Framework.

| **Aspect** | **Library (React)** | **Framework (Angular)** |
| --- | --- | --- |
| **Control** | You control the flow (you call the library). | Framework controls the flow (it calls your code). |
| **Flexibility** | More flexible, less opinionated. | Less flexible, more opinionated. |
| **Example** | React, Lodash | Angular, Django |

🧠 Analogy:

* **Library** = Buffet (you choose what you eat)
* **Framework** = Fixed thali (you eat what’s served)

**✅ 3. What is CDN? Why do we use it?**

**Answer**:  
CDN = **Content Delivery Network**. It's a network of servers distributed globally that helps in **faster delivery** of static files (like JS, CSS, images, videos).

**Why use it?**

* Faster loading time
* Reduces server load
* Easily access libraries like React without downloading

**Example**:

A black screen with green text

AI-generated content may be incorrect.

**✅ 4. Why is React known as React?**

**Answer**:  
React is named so because it **“reacts” to data changes** in UI efficiently. It **re-renders** only the parts of UI that changed.

🧠 Think: **React** = **Responsive** updates to the UI.

**✅ 5. What is crossorigin in script tag?**

**Answer**:  
crossorigin is used when loading resources from a **different domain** (like CDN).

It controls how **browsers handle CORS (Cross-Origin Resource Sharing)** and **error visibility**.

**Example**:

A screen shot of a computer

AI-generated content may be incorrect.

Use it to avoid **CORS errors** in browser and enable better error debugging.

**✅ 6. Difference between React and ReactDOM?**

✅ **What is React?**

React is a JavaScript library used to build user interfaces (UI) using components.  
It handles the logic, state, and UI of components — but not how or where they appear in the browser.

✅ **What is ReactDOM?**

ReactDOM is a package that helps connect React to the actual browser DOM.  
It provides methods like createRoot() and render() to display React components on the screen.

| **React** | **ReactDOM** |
| --- | --- |
| Core library for building UI | Handles rendering UI to browser DOM |
| React.createElement(), hooks, components | ReactDOM.render(), ReactDOM.createRoot() |
| Doesn’t interact with browser | Talks to the actual browser DOM |

🧠 Think:

* **React** builds components
* **ReactDOM** places them on the screen

**✅ 7. Difference between react.development.js and react.production.js?**

**✅ What is react.development.js?**

It’s the development version of the React library (via CDN) that includes extra debugging features, warnings, and detailed error messages.  
Used while building and testing apps.

**✅ What is react.production.js?**

It’s the production version of the React library that is optimized and minified for performance.  
Used when deploying your app live.

| **Development JS** | **Production JS** |
| --- | --- |
| For debugging during development | For deploying live websites |
| Bigger in size | Minified and optimized |
| Has helpful warnings & logs | Removes logs to improve performance |

Use **development.js** while learning or debugging,  
Use **production.js** for faster, cleaner live apps.

**✅ 8. What is async and defer in <script> tag?**

| **Attribute** | **Behavior** |
| --- | --- |
| **async** | Loads script **asynchronously**, executes **as soon as loaded**. Doesn't wait for HTML. |
| **defer** | Loads script in background, executes **after HTML is parsed**. Keeps order of scripts. |

**Prefer defer** for loading JS that should run after HTML is ready (like React).

**Example**:

A screen shot of a computer

AI-generated content may be incorrect.

✅**Q: Difference between Virtual DOM and Real DOM?**

* **Real DOM** → The actual structure of the web page in the browser (HTML elements).
* **Virtual DOM** → A lightweight copy of the Real DOM kept in memory by React.

**🏷️ Virtual DOM vs Real DOM**

| **Feature** | **Real DOM 🏛️** | **Virtual DOM 🪞** |
| --- | --- | --- |
| **Definition** | Actual DOM structure in the browser | In-memory lightweight copy of the Real DOM |
| **Update Speed** | Slow – updates re-render the whole page | Fast – updates happen in memory first |
| **Performance** | Expensive (causes reflows & repaints often) | Efficient (only necessary changes applied) |
| **Usage** | Directly managed by browser | Managed by React for optimization |
| **Efficiency** | Not optimized for frequent updates | Optimized for frequent UI changes |