1. What is Emmet?

When writing JSX, there are some rules to follow to prevent unnecessary console errors:

* HTML attributes and CSS properties must be named using camelCase
* JavaScript code must be wrapped in curly braces {} inside JSX
* For every opening tag of an HTML element, there must be a corresponding closing tag

With the Emmet plugin in VS Code, it’s easy to follow these rules, helping you type HTML in React faster and more efficiently.

To Configure Emmet in VS Code,, add this in Setting.json file:

"emmet.includeLanguages": {

"javascript": "javascriptreact",

"typescript": "typescriptreact"

}

With the code above, Emmet is now enabled for files that are recognized as javascriptreact or typescriptreact in VS Code, which are .jsx and .tsx files. You would need to reload the IDE to experience the updated changes.

**Basic abbreviations and expressions are:**

Attribute operators allows you to easily define the class and ID for a particular element:

* div.demo => <div className="demo"></div>
* div#demo => <div id="demo"></div>

The attributes can also be combined to form an expression as follows:

* div#headerId.headerClass => <div id="headerId" className="headerClass"></div>

**Nesting operators**

Nesting operators allows us to position how elements are placed and the order they follow.

**Child >**

Child is used to nest elements inside each other following the nav>ul>li structure:

<nav>

<ul>

<li></li>

</ul>

</nav>

**Sibling +**

Sibling places elements on the same level, following p+span:

<p></p>

<span></span>

**Climb up ^**

Climb up ^ moves the following element one level up the tree, header+main>div^footer:

<header></header>

<main>

<div></div>

</main>

<footer></footer>

**Multiplication \***

Multiplication \* defines the number times an element should be created li\*2:

<li></li>

<li></li>

**Item numbering $**

The item numbering $ operator allows us to assign unique values to repeated elements. It can be used alongside the multiplication operator to output the current number of the repeated element div.group$\*5:

<div className="group1"></div>

<div className="group2"></div>

<div className="group3"></div>

<div className="group4"></div>

<div className="group5"></div>

**Text formating {}**

Text formatting {} is used to add text to elements as follows:

p.demo{test} => <p className="demo">test</p>

2. Difference between a Library and Framework?

* Frameworks and libraries are both code written by someone else that helps you perform some common tasks in a less verbose way.
* framework inverts the control of the program. It tells the developer what they need. A library doesn’t. The programmer calls the library where and when they need it.

Example: A library is like going to Ikea. You already have a home, but you need a bit of help with furniture. You don’t feel like making your own table from scratch. Ikea allows you to pick and choose different things to go in your home. You are in control.

A framework, on the other hand, is like building a model home. You have a set of blueprints and a few limited choices when it comes to architecture and design. Ultimately, the contractor and blueprint are in control. And they will let you know when and where you can provide your input.

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

*Content Delivery Network or* content distribution network

A CDN, or [content delivery network](https://www.fastly.com/products/cdn), is a network or collection of servers in locations all over the world. Also known as a content distribution network, a CDN can refer to many types of content delivery services, such as load balancing and video streaming.

A CDN’s [network of servers](https://www.fastly.com/network-map/) allows companies to deliver content from their websites and mobile applications to people more quickly and efficiently, based on their geographic location. In short, a CDN moves data and applications closer to the end user — increasing speed, enhancing security, and improving the user experience.

Benefits:

* **Reduced page load time**
* **Improved availability**
* **Increased security**
* **Increased scalability**
* **Reduced bandwidth costs**

Reference: <https://www.fastly.com/learning/what-is-a-cdn>

4. Why is React known as React?

5. What is crossorigin in script tag?

The crossorigin attribute sets the mode of the request to an HTTP CORS Request.Web pages often make requests to load resources on other servers. Here is where CORS comes in.A cross-origin request is a request for a resource (e.g. style sheets, iframes, images, fonts, or scripts) from another domain.CORS is used to manage cross-origin requests.

CORS stands for Cross-Origin Resource Sharing, and is a mechanism that allows resources on a web page to be requested from another domain outside their own domain. It defines a way of how a browser and server can interact to determine whether it is safe to allow the cross-origin request. CORS allows servers to specify who can access the assets on the server, among many other things.

**Tip:** The opposite of cross-origin requests is same-origin requests. This means that a web page can only interact with other documents that are also on the same server. This policy enforces that documents that interact with each other must have the same origin (domain).

6. What is diference between React and ReactDOM?

React library is responsible for creating views and ReactDOM library is responsible to actually render UI in the browser.

Ref: [How React and ReactDOM works? - GeeksforGeeks](https://www.geeksforgeeks.org/how-react-and-reactdom-works/)

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

Later one is for production application where the overall bundle size of app is reduced because of minified files.

8. What is async and defer?

These are Boolean attributes which are use along with script tag to load the external scripts efficiently into our web page.

Async – Doesn’t guarantee the order of execution of the script.

Defer-- Guarantee the order of execution of the script.( While HTML parsing is going on, the scripts are fetched from the network and only executed when the HTML parsing is completed)

