

1. What is emmet?

-> Emmet is a free add-on for your text editor that allows you to type shortcuts that are then expanded into full pieces of code. Emmet is primarily independent from any text editor, as the engine works directly with text rather than with any particular software.

2. Difference between a Library and Framework ?

Both frameworks and libraries are code written by someone else that is used to help solve common problems

Library	Framework
<ul style="list-style-type: none">• A developer calls the library when and where it is needed	<ul style="list-style-type: none">• Framework inverts program control. It informs the developer of what they require.
<ul style="list-style-type: none">• You are in control. 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.	<ul style="list-style-type: none">• A framework, on the other hand, is like building a model home. You have a set of blueprints and a few <i>limited</i> 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.
<ul style="list-style-type: none">• When you use a library, you are in charge of the flow of the application. You are choosing when and where to call the library	<ul style="list-style-type: none">• When you use a framework, the framework is in charge of the flow. It provides some places for you to plug in your code, but it calls the code you plugged in as needed.
<ul style="list-style-type: none">• Example: jQuery, React	<ul style="list-style-type: none">• Example: Vue.js, Angular

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

A CDN is a network of servers that distributes content from an "origin" server throughout the world by caching content close to where each end user is accessing the internet via a web-enabled device. The content they request is first stored on the origin server and is then replicated and stored elsewhere as needed.

By caching content physically close to where a user is and reducing the distance it has to travel, latency is reduced. This process also decreases stress on origin servers by distributing the load geographically across multiple servers.

4. Why is React called React?

React was developed for applications (Facebook) that have constantly changing data. Since React is a front-end framework this means that as the user clicks around and changes the app's data, the view should "react" or change with those user events. User events being mouse clicks, typing, submitting a form.

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.

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.

6. Difference between React and ReactDOM.

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

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

The development file contains code which is easy to read and understand. Thus, making it more bulkier. Whereas, in production mode, code is minified so that it run quicker in client machine. This also comes with a fact that the code file becomes a little difficult to read.

8. What is async and defer?

Async	Defer
<ul style="list-style-type: none">• async scripts load in the background and run when ready. The DOM and other scripts don't wait for them, and they don't wait for anything. A fully independent script that runs when loaded.• Async scripts are great when we integrate an independent third-party script into the page: counters, ads and so on, as they don't depend on our scripts	<ul style="list-style-type: none">• The defer attribute tells the browser not to wait for the script. Instead, the browser will continue to process the HTML, build DOM. The script loads "in the background", and then runs when the DOM is fully built.
<ul style="list-style-type: none">• The browser doesn't block on async scripts	<ul style="list-style-type: none">• Scripts with defer never block the page.

<ul style="list-style-type: none">• DOMContentLoaded may happen both before and after async, no guarantees here.	<ul style="list-style-type: none">• Scripts with defer always execute when the DOM is ready (but before DOMContentLoaded event).
--	--