1. **What is Emmet?**

   A built-in plugin in Visual Studio Code (VS Code) that helps developers write HTML and CSS more efficiently.
   Generate boilerplate code for writing HTML and CSS.

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

   A key difference between the two is Inversion of control.

   Libraries are collections of pre-written code that you can use to perform specific tasks. You can easily link libraries into existing programs to add specific functionality.
   When using a library, the control remains with the developer who tells the application when to call library functions.
   You control the application flow, including when and where to contact the library.

   When using a framework, the control is reversed, which means that the framework tells the developer where code needs to be provided and calls it as it requires.
   The framework controls the flow of the application. Frameworks are sets of pre-written code that provide a structure for developing software applications.
   The framework sets the rules for how your project is organised, and your code has to follow those rules. It's difficult to incorporate a framework seamlessly into an existing project.

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

   It is a Content Delivery Network.
   CDN is a system of servers that speeds up the delivery of web content to users by storing it closer to them.

   Benefits of using a CDN -

   Performance
   CDNs reduce load times by storing content on servers closer to the user.
   This can also reduce file sizes and optimise server infrastructure.

Reliability
CDNs can provide uninterrupted service even when servers go down, networks are congested, or connections are interrupted.

Security
CDNs can help protect websites from malicious attacks, such as Distributed Denial of Service (DDOS) attacks.

Cost-effectiveness
CDNs can make businesses more cost-effective by reducing infrastructure bills.

4. **Why is React known as React?**
   Because it reacts to changes in data without reloading the entire page.

   The name 'React' was chosen because the library was designed to allow developers to react to changes in state and data within an application, and to update the user interface in a declarative and efficient manner

5. **What is crossorigin in script tag?**
   The crossorigin attribute in the script tag enables CrossOrigin Resource Sharing (CORS).
   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.

6. **What is difference between React and ReactDOM**
   React is a JavaScript library for building User Interfaces and ReactDOM is the JavaScript library that allows React to interact with the DOM.

   For example, we create react elements like heading(h1) using some core fundamentals of react i.e React.createElement("h1",{},"I am h1 Element");
   To display this element on the browser we use ReactDOM.

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

   When you're building a React application, you'll encounter two primary versions of the React library available via CDN: react.development.js and react.production.js. While both serve the same fundamental purpose, they are optimised for different environments:

   React.development.js:

   Purpose: Designed for development environments.

   Features:

   1. **Source Maps:** Enables detailed debugging by mapping minified code back to its original source.

   2. **Warnings and Error Messages:** Provides helpful warnings and error messages to aid in development.

   3. **Additional Checks:** Performs extra checks for potential issues, such as invalid props or state updates.

   Drawbacks:

   1. **Larger File Size:** Includes additional code for debugging and development features, resulting in a larger file size.

   2. **Slower Performance:** The extra checks and debugging information can impact performance.

   React.production.js:

   Purpose: Optimised for production environments.

   Features:

   1. **Minified and Compressed:** The code is minified and compressed to reduce file size and improve load times.

   2. **Optimised for Performance:** Prioritises performance by removing unnecessary code and optimising for speed.

3. **Reduced Debugging Information:** Provides minimal debugging information to avoid performance overhead.

Drawbacks:

1. **Less User-Friendly for Debugging:** Lacks detailed error messages and source maps, making debugging more challenging.

When to Use Which:

1. **Development:** Use react.development.js to take advantage of debugging tools and warnings during development.

2. **Production:** Use react.production.js to deploy your application with optimal performance and security.

8. **What is async and defer?**

In web development, `async` and `defer` are attributes you can add to `<script>` tags to optimise the loading and execution of JavaScript files, improving page load performance.

<script src="script1.js" async />
</script> <script src="script2.js" defer />

When to Use Which:

Async:
- Independent scripts
- Scripts that don't modify the DOM
- Scripts that can execute immediately

Defer:
- Scripts that modify the DOM
- Scripts that depend on other scripts
- Scripts that need to execute after the page is fully parsed

By using `async` and `defer` effectively, you can significantly improve the perceived performance of your web pages, especially on slower connections.