1. **What is Emmet?**

* Emmet uses different abbreviations and short expressions depending on what's passed, and then dynamically converts the abbreviations into the full code.
* Emmet is mostly used for HTML, XML, and CSS, but it can also be used with programming languages.
* Cheatsheet - <https://docs.emmet.io/cheat-sheet/>

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

* The following are some other fundamental differences between Frameworks and Libraries:

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Library** | **Framework** |
| Definition | Libraries provide developers with predefined functions and classes to make their work easier and boost the development process. | Framework, on the other hand, is like the foundation upon which developers build applications for specific platforms. |
| Inversion of Control | By using a library, you can control the flow of the application and call the library. | In contrast, when you use a framework, the control is inverted, i.e., the framework controls the flow and calls your code. |
| Collection | Generally, libraries are a collection of helper modules, objects, classes, functions, message templates, pre-written code, etc. | Frameworks consist of a lot of APIs, compilers, toolsets, support programs, libraries, etc. |
| Code Modification | Codes in libraries are geared toward a particular program or to solve a specific development problem. Therefore, developers must modify library code to meet their needs. | Despite the fact that frameworks generate new codes for developers. These codes cannot be altered or modified later. Unlike libraries, frameworks do not allow users to modify their pre-written codes, so you don’t have to worry about deleting or changing them. |
| Scope | It is possible to call a library out of context. You may use the library wherever you see fit in your code. | On the other hand, you can only call and use what belongs to a Framework within the same Framework. |
| Function | In the program linking and binding process, they play an important role. | Using them, you can build and deploy applications in a standard way as the framework already provides code to perform common tasks and uses code provided by a developer for custom functionality. |
| Complexity | Having a library means understanding the functionality of each method, and it isn’t easy to create complex interactions since you need to call many methods to get the desired results. | Frameworks, on the other hand, embody the basic flow, and since plugins need to be added to code, it is easier to do the right modification. |
| Extensibility | Generally, libraries aren’t designed for extensibility; they are designed to accomplish a specific purpose. | Frameworks provide general functionality. Because of this, they are built to be extensible, which allows developers to incorporate app-specific features without modifying the framework’s source code. |
| Replaceable | It is easy to replace a library with another library. For instance, if you do not like the jQuery date picker library, you can use another date picker like a bootstrap date picker or pick date. | Frameworks are difficult to replace. If, for instance, you were using AngularJS to build your product, you cannot simply swap it out for another framework. It requires rewriting the entire codebase. |
| Performance | Less code is required to build libraries, which leads to faster loading times and better performance. | Developing a framework requires a lot of coding, which increases loading times and decreases performance. |
| Usage | The purpose of libraries is to perform a defined and specific task. Eg: Image manipulation, network protocols, math operations, etc. | Frameworks can be used for performing a wide range of tasks. Among these are Web application systems, plug-in managers, GUI systems, and so on. |
| Existing Projects | You can integrate libraries seamlessly into existing projects to add functionality. | Incorporating frameworks seamlessly into an existing project is impossible. Instead, frameworks should be used when starting a new project. |
| Benefits | Good code quality, reusability, and control, enhanced speed and performance of the program, etc. | Faster programming, support from the community, great support for MVC (Model View Controller) pattern, etc. |
| Examples | JQuery, React JS, etc. | Spring, NodeJS, AngularJS, Vue JS, etc. |

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

* A content delivery network (CDN) is a group of geographically distributed servers that speed up the delivery of web content by bringing it closer to where users are.
* Data centers across the globe use caching, a process that temporarily stores copies of files, so that you can access internet content from a web-enabled device or browser more quickly through a server near you.
* CDNs cache content like web pages, images, and video in proxy servers near to your physical location.
* This allows you to do things like watch a movie, download software, check your bank balance, post on social media, or make purchases, without having to wait for content to load.
* CDN services were created to solve the problem of network congestion caused by delivering rich web content, such as graphics and video over the internet — much like a traffic jam.
* Getting content from centrally located servers to individual users simply took too long.
* CDNs have now grown to include everything from text, graphics, scripts, and media files to software downloads, documents, portals, ecommerce, live streaming media, on-demand [video streaming](https://www.akamai.com/our-thinking/streaming-media/live-video-streaming) media, and social media sites.

1. **Why is React known as React?**

* React is called React because it was designed to be a declarative, efficient, and flexible JavaScript library for building user interfaces.
* 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.

1. **What is crossorigin in script tag?**

* The crossorigin attribute, valid on the [<audio>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio),  [<img>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img),  [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link), [<script>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script), and [<video>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video) elements, provides support for [CORS](https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS), defining how the element handles cross-origin requests, thereby enabling the configuration of the CORS requests for the element's fetched data.
* Depending on the element, the attribute can be a CORS settings attribute.The crossorigin content attribute on media elements is a CORS settings attribute.

1. **What is difference between React and ReactDOM?**

* React: React is a javascript library, designed for building better user interfaces.
* React-DOM: React-DOM is a complimentary library to React which glues React to the browser DOM

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

* react.development.js provides us extra features like debugging, hmr(Hot module reloading) and lots of other stuffs that you might use while developing app with the help of bundlers like webpack, parcel, vite. This bundler bundles and minifies our code to be deployed on production.
* These minified files will be deployed on production which removes lots of unnecessary files which will not be used by our app for this we have react.production.js to make our much faster(as bundlers and lots of other files have done their work and are not required now)

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

* 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.
* 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.
* In practice, defer is used for scripts that need the whole DOM and/or their relative execution order is important.
* And async is used for independent scripts, like counters or ads. And their relative execution order does not matter.