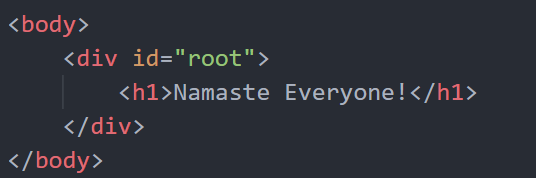
**DAY 1**

**Adding a h1 Tag using HTML**



**Adding a h1 tag using JS**

Text

Description automatically generated

A picture containing text

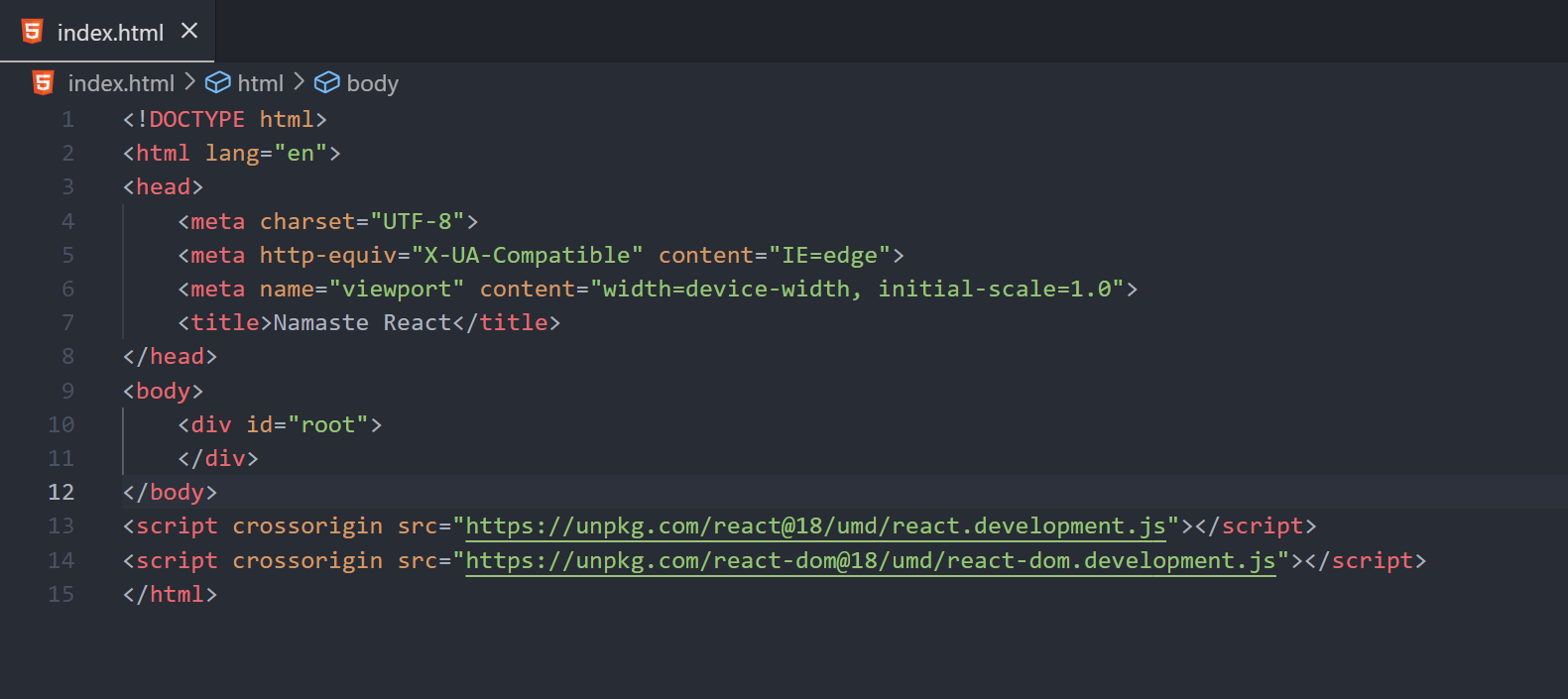
Description automatically generated

We can install react into our website by just using CDN.

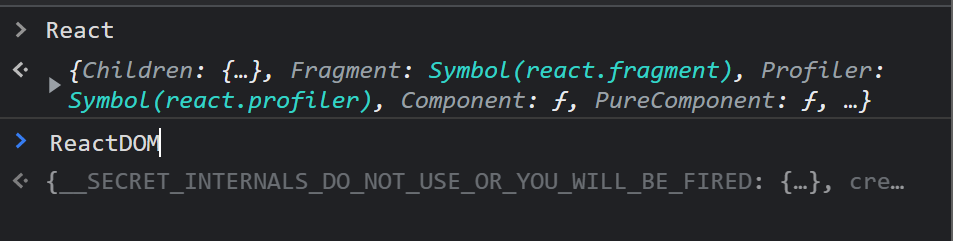
<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>

<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

Shortest program of REACT:



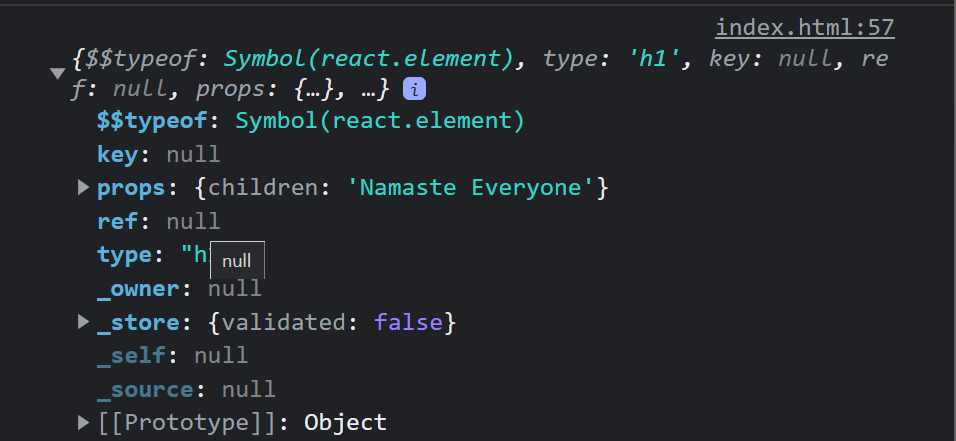
When we add these CDN links to the react app we get something called React and ReactDOM which are like global variables.



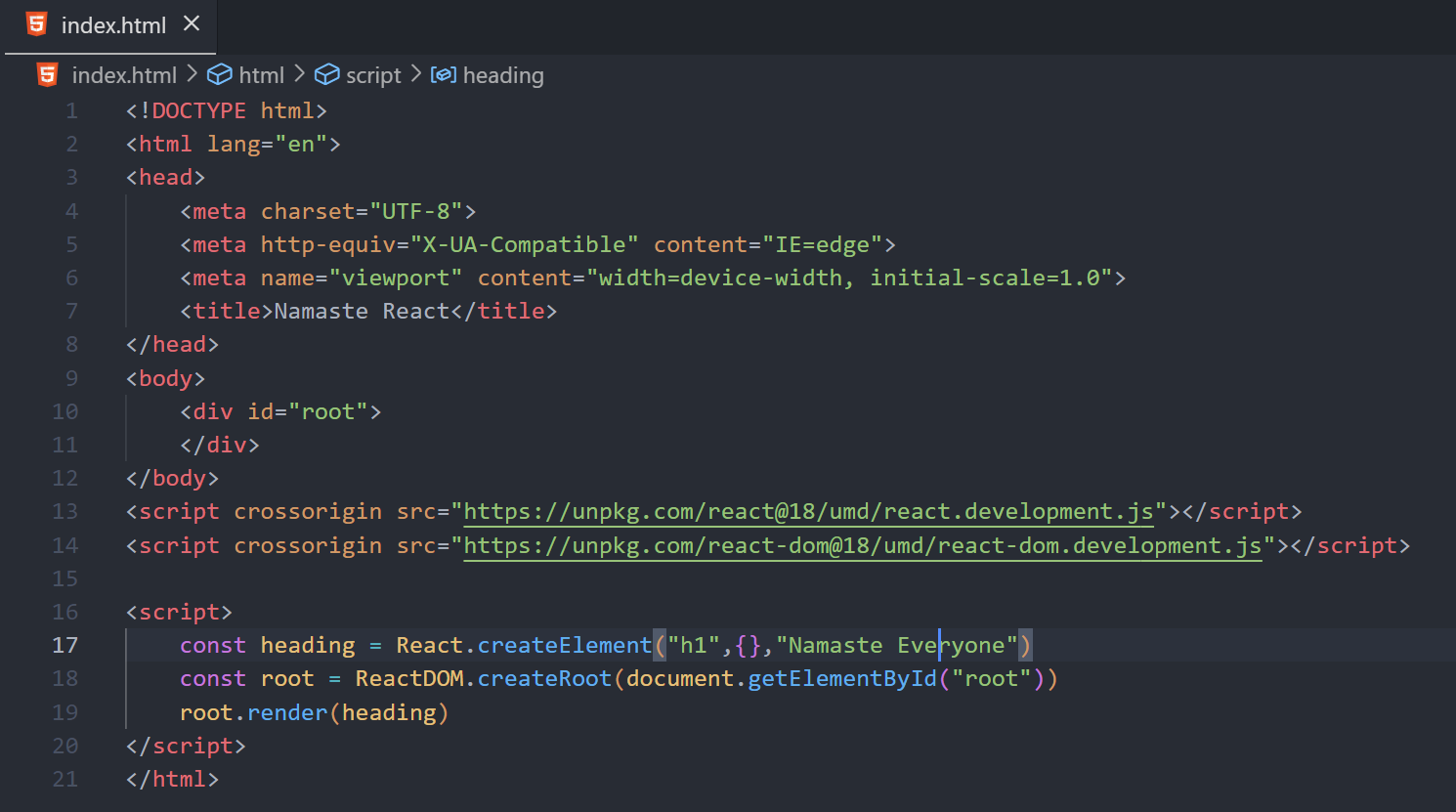
All these came by just injecting 2 CDN links. React is just a Java Script library written by Facebook developers.

React is not just limited to browsers. We can even make mobile apps and many others. React DOM is the web version of the React.

React element at the end is a JS object.



**Hello World using React.**

****

React.createElement() takes 3 parameters.

1st one is the element name.

3rd one is what we want to display in the element.

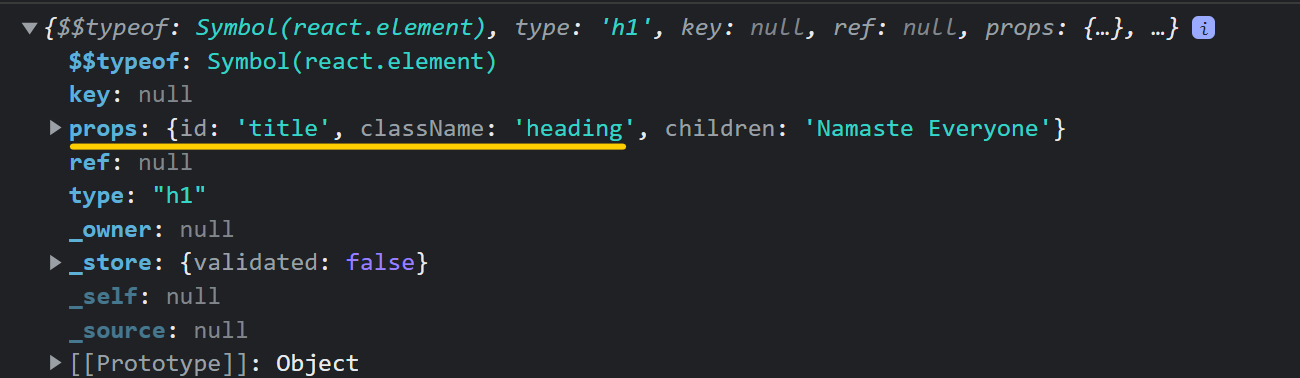
2nd one is the props which helps us to provide additional information to the element like id, class etc.

Graphical user interface, application, website

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence



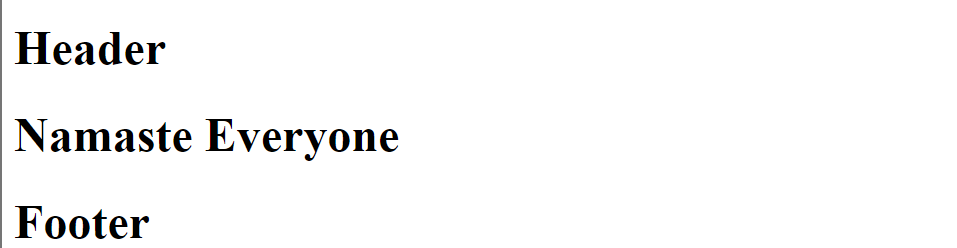
React DOM is responsible for all the DOM related operations, and it has a function called createRoot and whatever is passed into it will be the root of the DOM. Then we can render our heading using render method. We have only one root and one render method in our app.

**A picture containing background pattern

Description automatically generated**

React can just be added to certain parts of the website. Like if we just need a search bar which uses react, we can make it.



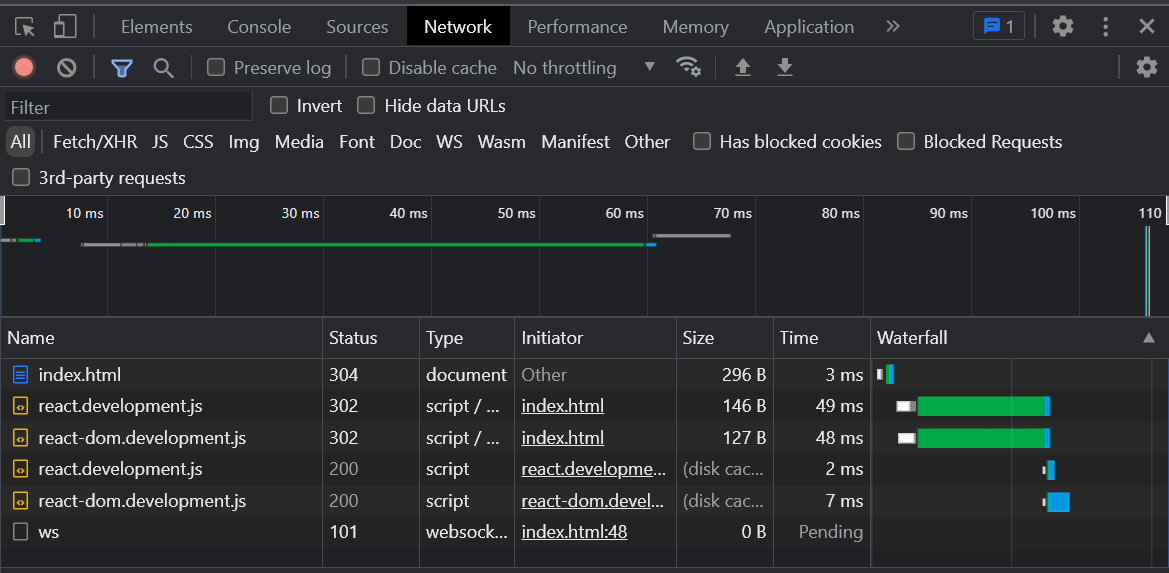


NOTE:

If we already have some elements in the root and then using root render, we render something else the existing elements will get overwritten.

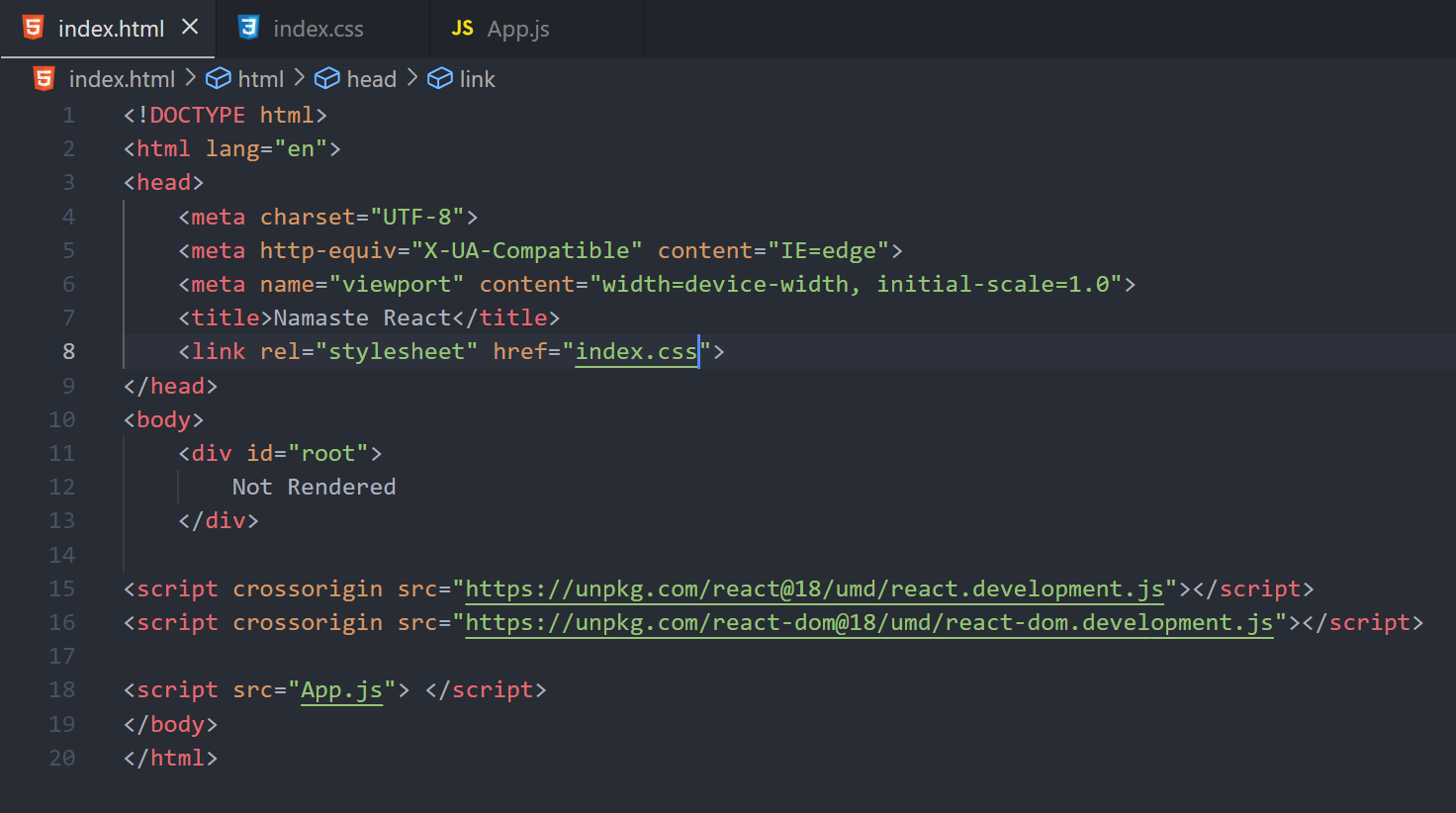
Some developers use this and put not rendered initially in the root. So, if the React could not load due to some reasons not rendered is displayed.

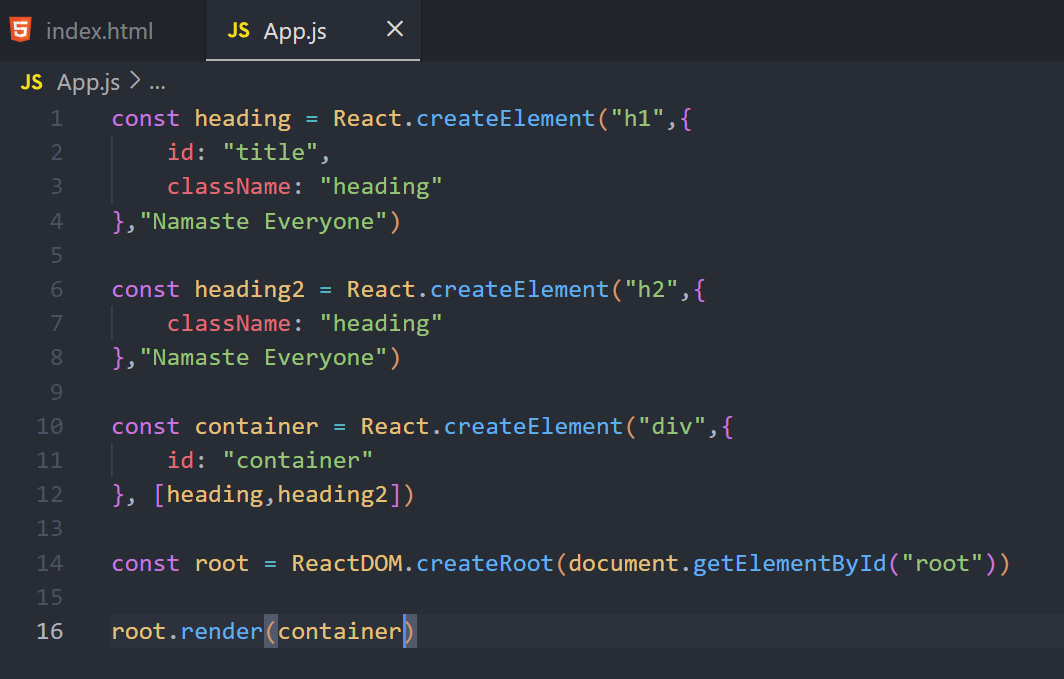
React takes some time to load in the mean while not rendered is displayed on the screen.



We can see this stuff in the network tab.

In order to make a div named container and put 2 heading tags into it we need to do it in the following way.





A screenshot of a computer

Description automatically generated with medium confidence

HOME WORK

1. What is Emmet?

Emmet is a plugin for text editors that provides powerful shortcuts for writing HTML and CSS code quickly. It can be used with a wide variety of text editors, including Sublime Text, Visual Studio Code, and others. Emmet uses abbreviations and expansion to automate the tedious and repetitive parts of web development, making it easier for developers to write HTML and CSS. For example, typing "ul>li\*5" and then pressing the expansion key (usually the Tab key) will generate a HTML unordered list with 5 list items. This can save a significant amount of time compared to manually writing out the same HTML code. Emmet supports a wide range of syntax, allowing developers to generate more complex HTML and CSS code with ease.

1. Difference between a Library and Framework?

A library and a framework are both collections of code that are used to solve specific problems, but there is a key difference between the two.

A library is a collection of pre-written code that can be called by a program to perform specific tasks. A library is essentially a tool that you can use to accomplish a specific task, but you have to write the code to use the library yourself. The library provides a set of functions that you can call, but you have to specify the order in which they are called and how they are used.

A framework, on the other hand, provides a set of rules and guidelines that dictate how the code should be organized and how it should work. When you use a framework, you write your code within the framework, following the rules and guidelines that it provides. The framework is responsible for coordinating the flow of control in your application, and it calls the code that you write at the appropriate times.

In other words, a library is a tool that you use to get a job done, whereas a framework is a set of rules and guidelines that you follow to get the job done.

It's important to note that libraries and frameworks are not mutually exclusive. Some libraries can also be used as frameworks, and some frameworks include libraries as part of their implementation.

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

A Content Delivery Network (CDN) is a system of distributed servers that delivers web content to users based on their geographic location. The main purpose of a CDN is to reduce the latency and improve the performance of a website by caching its content on multiple servers in different locations.

When a user requests a web page, the CDN routes the request to the nearest server in its network, which delivers the cached content to the user. This is faster than having the user request the content directly from the origin server, which might be located far away from the user. By reducing the distance that the data has to travel, the CDN reduces latency and improves the user's experience.

We use CDNs for several reasons:

* Speed: CDNs help to speed up the delivery of web content by caching it closer to the user.
* Scalability: CDNs can handle large amounts of traffic, making it easier to scale a website as it grows.
* Reliability: CDNs can provide a backup in case the origin server goes down, ensuring that the content is still available to the user.
* Security: CDNs can help protect a website from attacks, such as DDoS attacks, by distributing the load across multiple servers.
* Global Reach: CDNs can help deliver content to users all over the world, even in regions where the origin server might not have a strong presence.

Overall, CDNs are an important tool for improving the performance and reliability of a website, and they are widely used by websites of all sizes, from small personal blogs to large e-commerce sites.

1. Why is React known as React?

React, a JavaScript library for building user interfaces, was created and is maintained by Facebook. The name "React" was chosen to reflect the library's key feature: its ability to react quickly and efficiently to changes in user interactions and data.

In a user interface, changes in data and user interactions can happen frequently and unpredictably. React was designed to efficiently update the user interface in response to these changes, without having to reload the entire page. This is accomplished through the use of a virtual DOM (Document Object Model), which allows React to update only the parts of the user interface that have changed, rather than redrawing the entire interface from scratch.

The name "React" was chosen to reflect this key feature, and it has since become synonymous with the fast, efficient, and dynamic user interfaces that can be built with the library. The name has been adopted and embraced by the development community, and it is now widely recognized as one of the most popular and widely-used libraries for building user interfaces.

1. What is crossorigin in script tag?

The crossorigin attribute in a script tag is used to specify the type of CORS (Cross-Origin Resource Sharing) request that should be made when the script is loaded. This attribute is used to control whether the browser should include credentials (such as cookies) in the request.

There are two possible values for the crossorigin attribute:

* anonymous: The browser will not include credentials in the request, and the server must not include a "Access-Control-Allow-Credentials" header in the response.
* use-credentials: The browser will include credentials in the request, and the server must include a "Access-Control-Allow-Credentials" header in the response, set to "true".

By default, the crossorigin attribute is not set, and the browser will include credentials in the request if the server is not on a different origin (i.e., domain, protocol, and port) from the current page.

The use of the crossorigin attribute can help to improve the security and performance of a web page by controlling the type of CORS request that is made. However, it's important to note that setting crossorigin to anonymous may prevent some scripts from functioning correctly, as they may rely on the browser including credentials in the request.

1. What is difference between React and ReactDOM?

React and ReactDOM are two separate libraries that are both maintained by Facebook.

React is a JavaScript library for building user interfaces. It provides a set of abstractions for building and updating the user interface in a flexible and efficient way, using a virtual DOM and a concept of "components" to define reusable UI elements. React provides a way to build complex UI interfaces using a declarative approach, which makes it easier to understand and debug.

ReactDOM, on the other hand, is a specific implementation of React that is optimized for rendering to the DOM, which is the tree-like structure that represents an HTML document. ReactDOM provides a set of functions that allow you to render React components into the DOM, and to update the UI efficiently in response to changes in data and user interactions.

In other words, React is the core library for building user interfaces, while ReactDOM is the specific implementation of React that is optimized for rendering to the DOM. If you're building a web-based user interface using React, you'll typically use React and ReactDOM together, as they complement each other to provide a complete solution for building dynamic and interactive web UIs.

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

The react.development.js and react.production.js files are different builds of the React library that are optimized for different environments.

The react.development.js file is a development build of the library, and is intended to be used during the development process. This build is typically larger in size and includes additional features, such as error checking and debugging information, that are designed to make it easier to develop and debug your React applications.

The react.production.js file, on the other hand, is a production build of the library that is optimized for performance and size. This build is typically smaller in size and has had all of the development-specific features and error checking removed, making it faster and more efficient. This file is intended to be used when deploying your application to production.

When you include the React library in your HTML using a CDN, you can choose which of these files you want to use by selecting the appropriate URL. For example, to include the development build, you would use a URL like https://unpkg.com/react@16.13.1/umd/react.development.js, and to include the production build, you would use a URL like https://unpkg.com/react@16.13.1/umd/react.production.js.

It's important to use the correct build of the React library for the environment in which it is being used, as using the wrong build can lead to performance and stability issues. For example, if you use the development build in production, your application may be slower and less efficient than it would be if you used the production build. Similarly, if you use the production build during development, it may be harder to debug and diagnose issues with your application.

1. What is async and defer?

The async and defer attributes are used to specify the behavior of a script when it is loaded by a web page. They are typically used in the script tag to control the order in which scripts are executed and the timing of their execution.

async: The async attribute tells the browser to load the script asynchronously, which means that the script will be loaded in the background while the rest of the page continues to load. When the script has finished loading, it will be executed as soon as possible, without blocking the parsing of the rest of the page. This can improve the loading performance of a page, as the script is not blocking the rendering of other elements on the page. However, because scripts with the async attribute are executed as soon as they are loaded, there is no guarantee of the order in which they will be executed.

defer: The defer attribute tells the browser to load the script asynchronously and to defer its execution until the page has finished parsing. This means that the script will be loaded in the background, but will not be executed until after all other elements on the page have been parsed. This can be useful in situations where you want to ensure that a script is executed after all other elements on the page have been processed, but you don't need the script to block the parsing of the rest of the page.

In general, the async attribute is used for scripts that don't need to be executed in a specific order, and the defer attribute is used for scripts that need to be executed in a specific order, but don't need to block the parsing of the rest of the page. It's important to choose the appropriate attribute based on the requirements of your script and the behavior you want to achieve, as using the wrong attribute can lead to unpredictable results.



In case of CSS files even if it is not downloaded it goes to the rest of the document. But incase of JS it waits till it is loaded.



This is the 1st way to load JS so we observe in most of the websites the JS is loaded at the bottom of the body tag so the website loads all the images and text before loading JS.

