Chapter 01(class 1)

1. What is Emmet?

Emmet is a plugin for many popular text editors which greatly improves HTML & CSS

<https://code.visualstudio.com/docs/editor/emmet>

<https://docs.emmet.io/cheat-sheet/>

Example1 :

Child: >

nav>ul>li

<nav>

<ul>

<li></li>

</ul>

</nav>

Example 2:

Multiplication: \*

ul>li\*5

<ul>

<li></li>

<li></li>

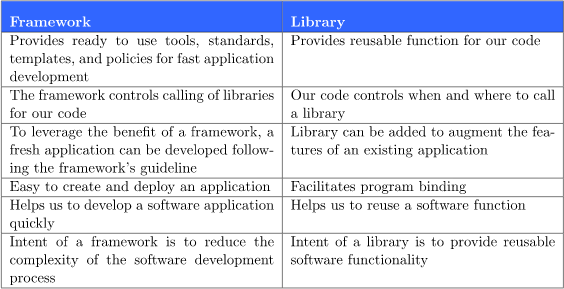
<li></li>

<li></li>

<li></li>

</ul>

1. Difference between a Library and a Framework?



| Examples of frameworks are AngularJS, Spring, NodeJS, etc. | Examples of libraries are JQuery, React JS, etc. |
| --- | --- |

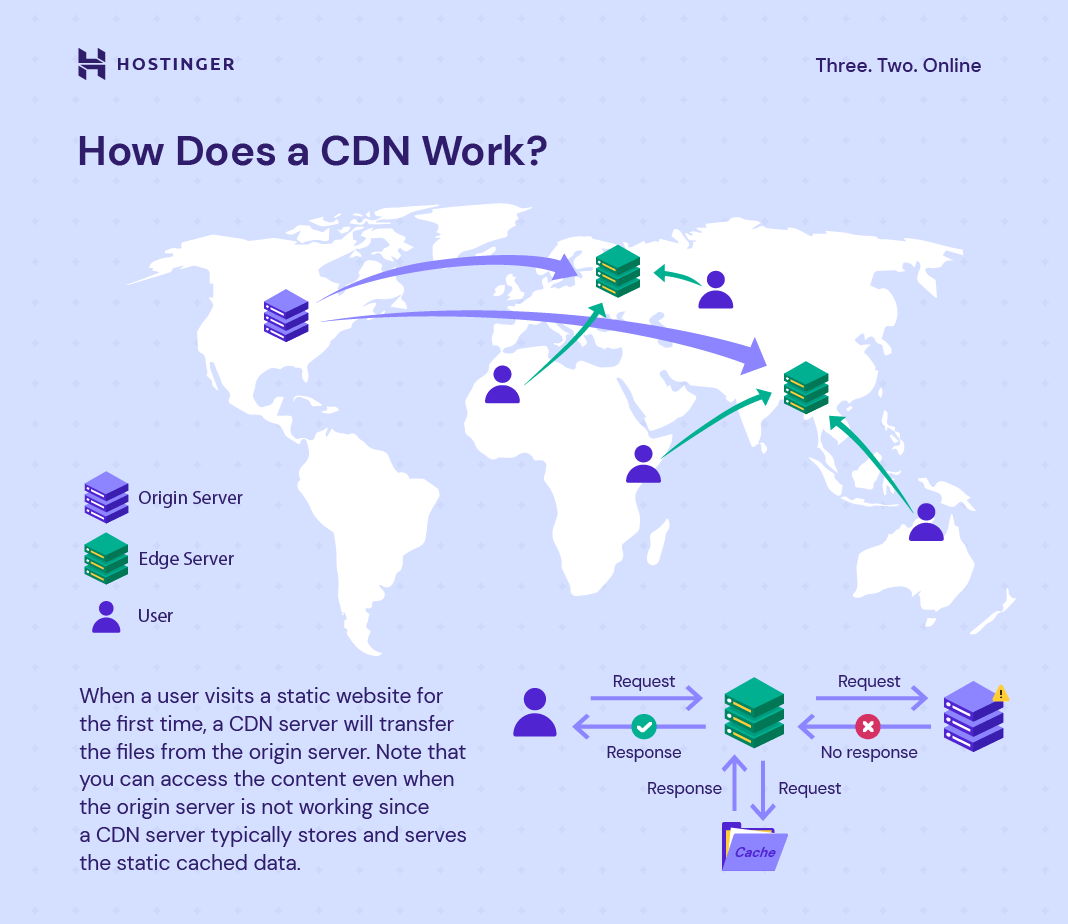
1. What is CDN? Why do we use it?

A CDN (content delivery network), also called a *content distribution network*, is a group of geographically distributed and interconnected servers. They provide [cached](https://www.techtarget.com/searchstorage/definition/cache) internet content from a network location closest to a user to speed up its delivery.

The primary goal of a CDN is to improve web performance by reducing the time needed to send content and rich media to users.

The process of [accessing content cached on a CDN network edge](https://www.techtarget.com/searchcloudcomputing/feature/Edge-computing-strategies-will-determine-the-next-cloud-frontier) is almost always transparent to the user. CDN management software dynamically calculates which server is located nearest to the user making the request and delivers content based on those calculations. CDN edge servers communicate with the content's origin server to deliver cached content and new content that has not been cached to the user.

The number one reason for using a CDN is to improve your user’s experience in terms of speed



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.

React was developed by Facebook and released in 2013, and it has since become one of the most popular JavaScript libraries for building web and mobile applications. It is used by many companies and organizations around the world, and it has a large and active developer community.

1. What is cross-origin in the script tag?

The purpose of cross-origin attributes is used to share the resources from one domain to another domain. Basically, it is used to handle the CORS request. It is used to handle the CORS request that checks whether it is safe to allow for sharing the resources from other domains. The resources may include Audio, Video, Images, Links, or external script that specifies whether to support a cross-origin request or not.

CORS: It stands for cross-origin resource sharing. It is a mechanism by which one webpage requests another domain for fetching out resources like audio, video, script, etc. from the third-party server without leaking their credentials information.

Values: This attribute contains two values which are given below –

* anonymous: It has a default value. It defines a CORS request which will be sent without passing the credential information.
* use-credentials: A cross-origin request will be sent with credentials, cookies, and a certificate.

1. What is the difference between React and ReactDOM?

React:

React is an open-source JS library. to develop UI (User Interface) on web applications or sites. It's fully component base.

React Dom:

React DOM is the glue between React and DOM. When u want to show your react component on DOM u need to use this ReactDOM.render(); from React Dom.

Before v0.14 React Dom was part of ReactJs. The reason React and ReactDOM were split into two libraries was due to the arrival of React Native. React contains functionality utilized in web and mobile apps. ReactDOM functionality is utilized only in web apps.

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

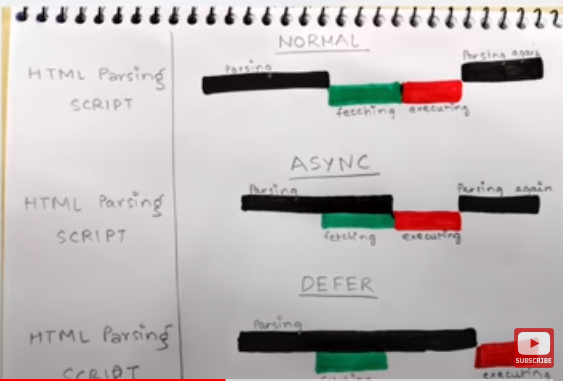
The development build is used - as the name suggests - for development reasons. You have Source Maps, debugging, and oftentimes hot reloading ability in those builds.

The production build, on the other hand, runs in production mode which means this is the code running on your client's machine. The production build runs uglify and builds your source files into one or multiple minimized files. It also extracts CSS and images and of course any other sources you're loading with Webpack. There's also no hot reloading included. Source Maps might be included as separate files depending on your webpack devtool [settings](https://webpack.js.org/configuration/devtool/).

What specifically separates production from development is dependent on your preferences and requirements, which means it pretty much depends on what you write in your Webpack configuration.

the production and development build come into the picture just because of performance impact in real-life deployed the application. Also, it happens that the location where the application is deployed is another continent altogether, so rendering development build js files on UI will take a hell of a time as compared to the production version which is very crisp, compact, compressed, uglified for better user experience and loading on UI.

1. What are async and defer?



Normal tag

* HTML parsing goes on
* When script tag is encountered scripts are fetched from network
* And executed
* Then the HTML parsing continues.

Async tag

* In Async tag HTML parsing goes on
* Scripts are fetched in an async manner parallelly
* As soon as scripts are fetched scripts are executed
* Then the HTML Parsing continues.

Defer tag

* In defer tag HTML parsing goes on
* Scripts are fetched in parallel
* Only executed once HTML parsing is completed

Async does not guarantee the order of execution that the scripts are executed in the given order.

Defer attribute maintain order of executed