**NamasteDev**

Assignment 1 – Inception

Q1: What is Emmet?

Emmet is a toolkit for web developers that can significantly improve the HTML and CSS workflow.

The tags that you want to generate are entered as CSS-like expressions. These expressions are dynamically parsed, and the output is displayed on the screen.

With the help of Emmet, we can write code at a faster pace. Some examples illustrating the usage of Emmet:

Tab key press shown as 🡪

#abc 🡪 <div id="abc"></div>

p#acd 🡪 <p id="acd"></p>

p#para.fun 🡪<p id="para" class="fun"></p>

#namaste.fun.learn.react 🡪<div id="namaste" class="fun learn react"></div>

div#root>p#p2 🡪 <div id="root">

<p id="p2"></p>

</div>

#abc+p#qwe+h1#ty🡪<div id="abc"></div>

<p id="qwe"></p>

<h1 id="ty"></h1>

Q2: Difference between library and Framework

A library is a piece of code that can work in small portions of a page. It can co-exist with other components in your application. Example chart.js to render charts in your webpage.

Whereas a framework is more complex and provides everything that is required to develop the complete application. In this case, the framework takes the control of the app (and is no longer with the developer). Example Struts 2.0 framework wherein you have to abide by the set of rules laid down by the framework.

A real-world example –

I have rented a new house and I need some furniture for my workstation. I would go to Ikea and buy a table, chair etc. I could use this anywhere in my house. Just like a library.

What if there is no house and all that is available are blueprints and materials. So the contractor who is in charge of constructing takes the control and builds the house. Just like the framework.

Q3 – What is CDN? Why do we use it?

A few years ago, whenever we watched movies online, there would be a considerable amount of delay in getting the movie on the device. One main reason being that the webserver that hosted the movie was in some geographical location and the end-user in another corner of the world. To resolve this, CDN came into picture.

Content Delivery Network(CDN) is a geographically distributed group of servers that caches content close to the end users.

Without CDN

Let’s assume that the latency in this case is 5 sec. Latency is the time taken for the content to render on the end-user’s device.



Origin server



End user

With CDN

Cache server

Cache server



End user

Cache server

Origin server

In this case the latency is reduced to 2 sec.

How did this happen?

To minimize the distance between the end-user and the website’s server, CDN stores a cached version of the content in multiple geographic locations(aka points of presence ;PoP). Each PoP contains several caching servers responsible for content delivery to the user within its proximity.

Some advantages of CDN:

Improves page load speed

Handles high traffic loads

Blocks spammers thus securing the application

Reduce Bandwidth consumption

Used in: E-commerce, Gaming, Advertising, Media, Entertainment,..many more.

Q4: Why is React known as React?

React is named React because the name reflects the reactive nature of the library. The library was designed to “react” to changes in data.

Q5: What is crossorigin in script tag?

Crossorigin attribute is used for loading an external script into your domain. This external script could be from a 3rd party server or another domain with the support of HTTP CORS Request. The crossorigin attribute protects sensitive information from the 3rd party when fetching the results.

CORS – Cross origin resource sharing is a mechanism that enables an application that is loaded in one domain to interact with the resources of another domain.

Q6: What is difference between React and ReactDOM?

React is the core of react. It is the JS library for building user interfaces.

ReactDOM is a JS library that allows the elements created using React to interact with the DOM. So, ReactDOM is the bridge between React and the DOM.

Q7: What is difference between react.development.js and react.production.js files via CDN?Generally, the DEV files are not suitable in PROD setup. In the production environment, compressed and minified versions of Javascript libraries are preferred to reduce the size of the code. Performance will be much faster in PROD when compared to DEV.

Q8: What is async and defer?

The <script> tag can be placed either in the <head> or in the <body>

Download HTML and parse –



Download script –



Execute script -



1. Head section



When a script is encountered while HTML parsing, the parsing is paused and the script is downloaded and executed. Once completed, the HTML parsing is resumed. So if there are DOM elements that are accessed by the script before the document is ready, errors will be thrown.

1. At the end of body section



In this case, the HTML is parsed completely, and the document is ready before the script is downloaded and then executed. Therefore, no errors.

Note that the order in which the <script> tags are written will define the order of script download and execution.

For better efficiency we have 2 attributes – async and defer

Async and defer are Boolean attributes, if present value is true , otherwise false.

Async ensures that other script downloads don’t wait while a script is being downloaded. The browser does not block the HTML parsing (DOM content creation) when a script is encountered. The async script downloads in the background and executes once done.



The async scripts execute in the ‘load-first’ order. There could be a case when there is a smaller file placed in the end of script tags that is downloaded first, and hence executes first. This can be a problem and cause errors as there is no guarantee that the relevant DOM is loaded into the browser. It is here that ‘defer’ comes to rescue.

Use async attribute when the script does not manipulate the DOM. When accessing external scripts

When accessing libraries, chatbots etc.

Like async defer also downloads the scripts in the background parallelly, but the HTML parsing will not be interrupted. The script execution is only after the HTML parsing and loading of the complete document.



In case there are multiple scripts, they all are executed in sequence before the ‘DOMContentLoaded’ event is fired.

defer is used when the script manipulates the DOM. Improves page loading by downloading all the scripts in the background and executes the scripts only when the document is ready.