Assignment 1

Owner - Mohit R

1. What Is Emmet?

- Emmet is a plugin for text editors and Integrated development environment's (IDE) that enhances the efficiency of writing HTML and CSS code.
- It helps developers to write code quickly and with fewer keystrokes.
- Emmet also enables users to use shorthand notations and expand them to HTML and CSS code.
- VS code has a built-in emmet.
- Example if we type "div.container" and pressing tab will expand to:
  ** **

2. Difference between Library and Framework?

- **Library** are more flexible and allow you to use a specific component as needed.
- **Framework** provide a comprehensive structure and enforce a specific way to build an application.
- The choice between a library and framework lies completely between the specific requirements of the project and the development preferences.
- In summary, the main difference lies in the "Level of control" and the "structure they provide".

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

- CDN (Content Delivery Network) are used by a wide range of websites and applications including, e-commerce sites, media and entertainment platforms, and Software-as-a-service(SaaS) providers.
- **Improving the speed of webpages**: By serving the content from a server closer to the users location, CDNs help in significantly improving the speed of the website.
- **Reducing bandwidth costs**: CDNs help is reducing the bandwidth cost for websites by serving cached copies of the content closer to the user.
- **Enhancing security**: CDNs provide additional security measures such as DDoS (Distributed Denial of Service) protection and SSL (Secure Socket Layer) encryption, helping to protect the website and applications from attacks.
- **Improved availability**: CDNs can help improve the availability of websites by providing multiple copies to the user if one server goes down.

4. Why is React known as react?

- It encapsulates the core principles of a library which include:
  - Efficiently updating the UI in response to data change
  - Organizing the UI into reusable components
  - Using a declarative syntax
  - Handling user interactions
- This makes react a very powerful to build dynamic and responsive web and mobile applications.

5. What is cross origin in the script tag?

- Cross origin attribute sets the mode of the request to the HTTP CORS (**C**ross **O**rigin **R**esource **S**haring) request.
- Cross origin attribute is used to share resources from one domain to the other. Basically, it is used to handle CORS request. It is used to handle the CORS request that checks if it is safe or not to allow the sharing of resources from other domains.
- The Cross origin attribute in the <script> tag specifies that CORS is supported the loading an external script from other third party servers or domains. It is a standard mechanism to retrieve files from other domains.
- **<script crossorigin="anonymous | use-credentials">**

6. What is the difference between React and ReactDOM?

- React and ReactDOM are two separate libraries that are used together in the development of web applications in React.
- React is a javascript library used to build User Interfaces, whereas ReactDOM is also a javascript library which enables React to interact with the DOM.
- ReactDOM on the other hand, it is a library that provides an interface between React and DOM (Document Object Model). DOM is a tree like structure that represents HTML of a webpage, and ReactDOM provides a set of functions that allows react components to be rendered in the DOM and updated efficiently.
- React package contains React.createElement( ), React.Component, React.Children and other such helpers related to the component classes. ReactDOM contains ReactDOM.render( ) and on the ReactDOM/server-side we server side rendering

support with ReactDOMServer.renderToString( ) and ReactDOMServer.renderToSTaticMarkup( ).

- Example:

```
const header = React.createElement(
    “div”,
    {id: “parent”},
    React.createElement(
    “h1”,
    { },
    “Welcome to React!”
     )
    );
ReactDOM.render(header, document.getElementById(“root”);
```

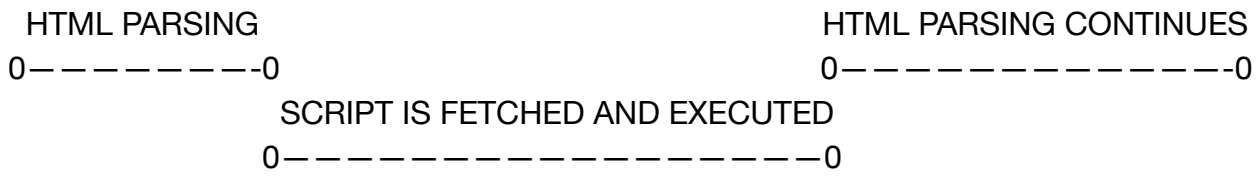
7. What is the difference between React.development.js and React.production.js files via CDN?

- Development is the stage of an application before it is made public. Production is termed for the same application which is made public.
- Development JS is for development reasons which may include, source maps, debugging and at times hot reloading ability in their builds.
- Production JS runs in production which means the code is running on the client server.
- Development build is generally (maybe 2-3x) slower than the production build.

8. What is async and defer?

- Async and Defer are boolean attributes that are used with our script tag to load the scripts efficiently into our browser.
- When we load a webpage, there are two main things that are happening in the browser. One is HTML parsing and the other is loading of the script.
- The loading of the script contains two things:
    - Fetching the script from the network
    - Executing the script
- **Example 1:**

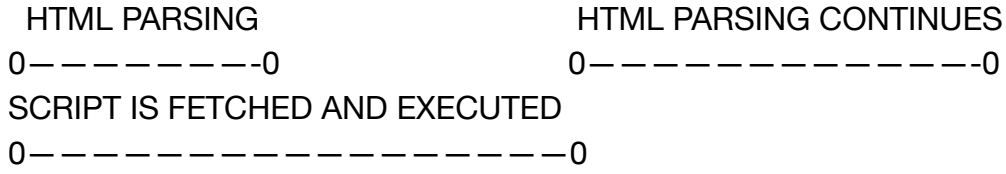
```
<script src=“ “/>
```



- When HTML is parsing, the browser encounters the src of the script and it stops parsing and gets the script from the network.
- When fetching and executing the script, HTML parsing is stopped.

- **Example 2:**

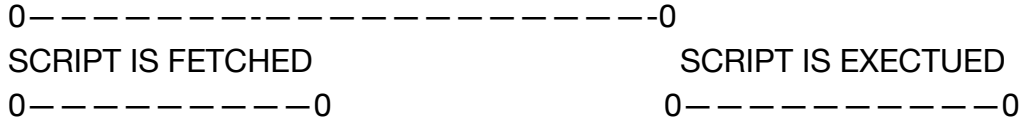
  <script async src=“ “ />



- While using async, HTML parsing doesn’t stop while the script is being fetched asynchronously from the network (the fetching is done while HTML parsing is going on).
- When the script is fetched, HTML parsing stops until the script is executed. Once the script is executed, HTML parsing resumes.

- **Example 3:**

  <script defer src=“ “ />



- Incase of defer, HTML parsing goes on while the script is being fetched. The script is executed only after HTML parsing is completed
- ASYNC doesn’t guarantee the order of execution of the script. But defer does, incase of interdependent scripts, async should be used.