# ASSIGNMENT 1

## EMMET

Emmet is a free add-on for your text editor. It allows you to type shortcuts that are then expanded into full pieces of code. By using Emmet, developers type less, they save both on keystrokes and time.

## CDN

CDN stands for **Content Delivery Network**. It's a network of servers distributed geographically to deliver web content more efficiently to users. In the context of React, using a CDN typically refers to including the React library directly from a CDN provider like unpkg or jsDelivr, instead of installing it locally through a package manager like npm.

## LIBRARY VS FRAMEWORK

LIBRARY : PROVIDE FUNCTIONALITY
FRAMEWORK: PROVIDE STRUCTURE

In summary, the main difference lies in the flow of control. With a library, you are in control and call the library when you need its functionality. With a framework, the control flow is more inverted, and the framework calls your code, enforcing a particular structure.

A framework is a set of pre-written code that provides a structure for developing software applications. A library, on the other hand, is a collection of pre-written code that can be used to perform specific tasks.

## REACT VS REACT-DOM

React and ReactDOM are two separate packages in the React ecosystem, each serving a specific purpose:

1. **React:**
   - The `react` package is the core library of React. It provides the fundamental functionalities for building components, managing component state, handling props, and managing the virtual DOM.
   - When you write React components, you import and use objects and functions from the `react` package, such as `React.Component`, `useState`, `useEffect`, and others.

2. **ReactDOM:**
   - The `react-dom` package is responsible for rendering React components in the DOM (Document Object Model).
   - It provides methods for rendering React elements into the DOM, updating them, and handling events.
   - The most commonly used method from `react-dom` is `ReactDOM.render()`, which takes a React element and renders it into a specified DOM container.

In summary, while the `react` package focuses on the core React functionality and the components themselves, the `react-dom` package is specifically designed for interacting with the DOM, managing the rendering of React components, and handling events within a web page. When building a web application, you typically use both packages together: `react` for component logic and structure, and `react-dom` for rendering those components into the browser's DOM.