VS code uses Emmet.

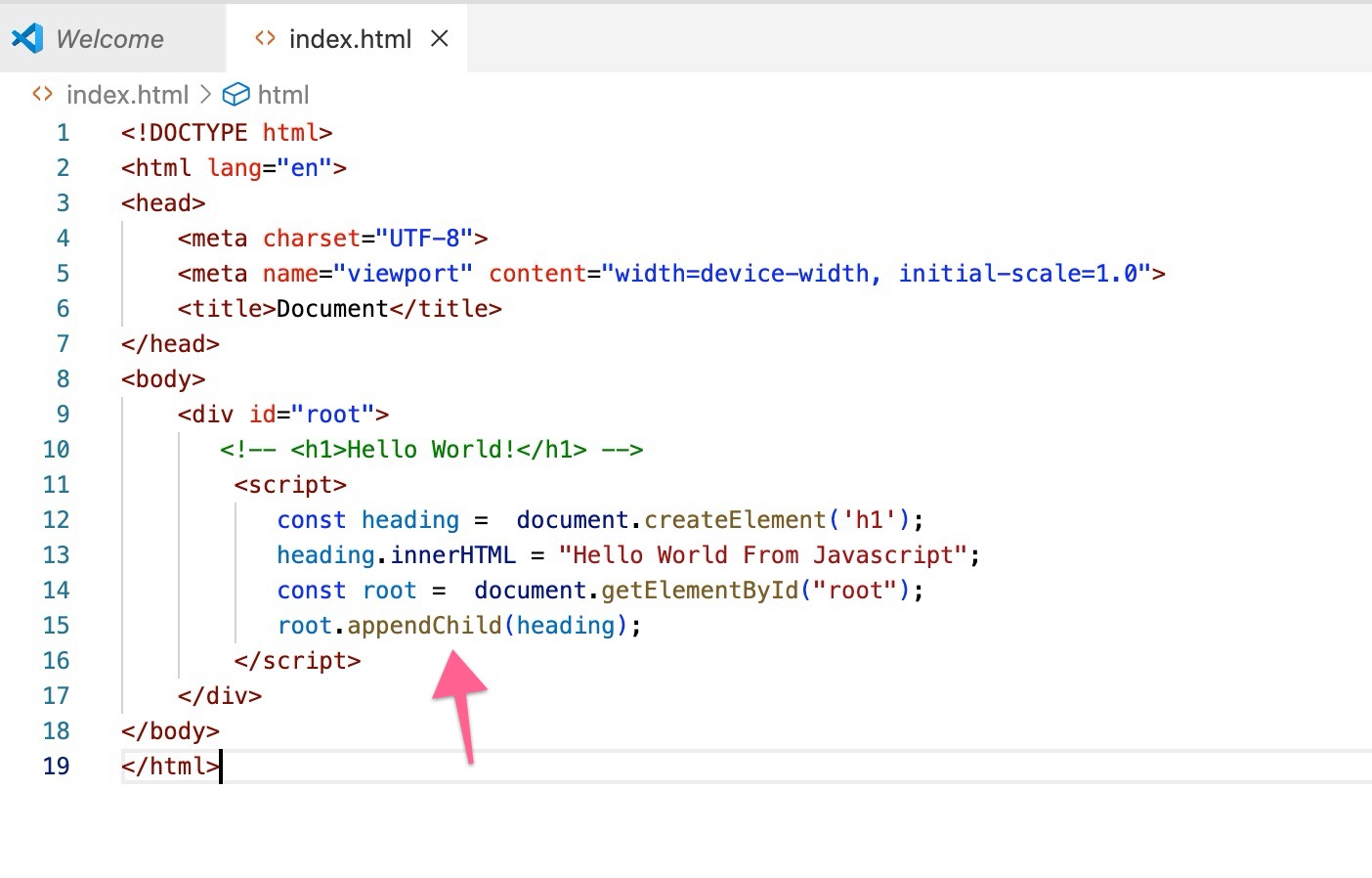
Emmet is a powerful toolkit used by web developers to speed up the process of writing HTML and CSS code. It's a plugin for text editors that allows you to type abbreviations or short expressions and expand them into full-fledged HTML or CSS code. This helps developers save time and increase productivity.

If we give html:5

Emmet plugin vs code helps to generate basic HTML structure

For example, if you type div.container>ul>li\*5 and hit the expansion shortcut, Emmet will automatically generate the following HTML code:

A white background with black dots

AI-generated content may be incorrect.It's compatible with many popular code editors like Visual Studio Code, Sublime Text, Atom, and others. Emmet can also be used to generate CSS shorthand for properties like margin, padding, and borders. It makes repetitive coding tasks much faster and more efficient.

appendChild is present on every HTML node.heading will go as child inside the root.

createElement, appendChild and innerHTML will understood by browser because browser has javascript engine in it that executes these js code.

But browser doesn’t know what is react.

**First way of adding react to our project is using CDN**

**What is CDN?**

A **Content Delivery Network (CDN)** is a network of servers strategically distributed across multiple geographic locations to deliver content to users more efficiently. Instead of fetching content (such as images, videos, JavaScript, CSS, or even entire web pages) from a single server, a CDN caches the content on multiple servers, which are closer to the user's location.

**A React CDN (Content Delivery Network) is a way to include the React library (and related packages like ReactDOM) in your web application via a remote server rather than hosting the React files locally in your project. By linking to the React library hosted on a CDN, you can easily include React in your project without needing to download or manage the React files yourself**

**Now by using CDN links**

[**https://legacy.reactjs.org/docs/cdn-links.html**](https://legacy.reactjs.org/docs/cdn-links.html)

**we have injected react library to our project using CDN links.so bowsers will understand our code.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**why are we using two cdn links and what are they**

 **React** is the core JavaScript library for building user interfaces, especially for single-page applications (SPAs). It provides the functions needed to create components and manage the application's state and rendering logic.

 The React CDN link includes the **React library** (e.g., react.development.js or react.production.min.js).

 **ReactDOM** is a separate library that provides DOM-specific methods to interact with the browser. It enables React to render components to the browser's DOM and manage the updates when state or props change.

 The ReactDOM CDN link includes the **ReactDOM library** (e.g., react-dom.development.js or react-dom.production.min.js).

**Why Are Two Links Needed?**

**1. Separation of Concerns:**

* **React** is concerned with defining the components, handling the state, and managing the logic of the user interface.
* **ReactDOM** is responsible for **rendering those React components to the actual DOM** in the browser and handling updates (when state or props change).
* These two libraries are separated because React itself does not include methods for directly interacting with the DOM. ReactDOM is what bridges the React components and the actual HTML DOM.
*  By including both the **React** and **ReactDOM** libraries, you ensure that React has both the functionality (from React) and the means to interact with the DOM (from ReactDOM).
*  You need **both** libraries for a complete React setup in the browser. You can’t just use React without ReactDOM when you're working in a web browser because ReactDOM connects React’s virtual DOM with the actual DOM.

**NOTE:**

The React CDN link includes the **React library** (e.g., react.development.js or react.production.min.js).

The ReactDOM CDN link includes the **ReactDOM library** (e.g., react-dom.development.js or react-dom.production.min.js).

For development purposes, you often use the non-minified versions of React and ReactDOM because they provide more debugging information and warnings. However, in a production environment, the minified versions are used for performance reasons.

Now we can use react to write our first program to display hello world

We use React to create elements.

But we use ReactDOM to render the elements

****

In React, the ReactDOM.createRoot function is used to **create a root for rendering** a React application in a specific DOM element.

**A computer screen shot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**It’s not good practice all the lengthy react code in one place instead we will write all the react code in separate JS file**

**The costliest operation in the webpage is when the DOM nodes need to be manipulated (Putting some nodes inside the DOM and removing some nodes from the DOM)**

**All frameworks and libraries are trying to optimize this operation.**

**React comes with a philosophy of writing or manipulating the dom using React or JavaScript. That is why react gives us helper functions to make these things work in a very performant way.**

**This is all the HTML required going forward we will work with react.**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**{} this is the object, we will add attributes of the tag**

**A screenshot of a computer

AI-generated content may be incorrect.**

**If we console the value of heading, we will see the react Element.**

**React element is nothing but the javascript object.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Props are children and the attributes that we pass in.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**We use array to create siblings as shown in below screenshot**

**A computer code with text

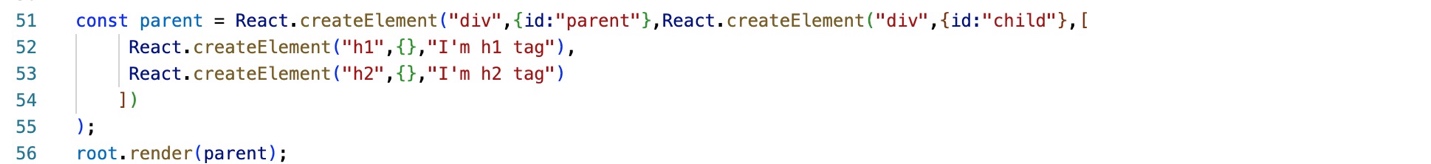
AI-generated content may be incorrect.**

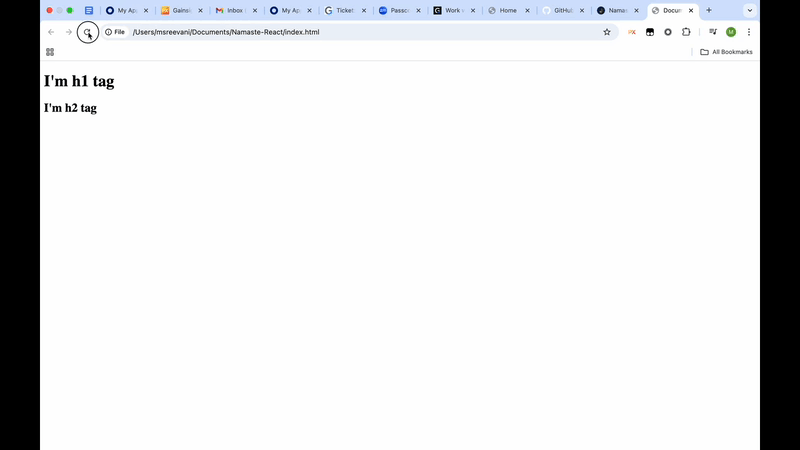
**If we’ve already content inside root, when we use render ().**

**The content inside render () will replaces the existing content inside root.**

**A screenshot of a computer

AI-generated content may be incorrect.**

****

****

**If we have other html code, that will remain same on the page. only the root element content of react gets updated by react.**

**A close-up of a web page

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**We call react as library. Because we can apply react to small portion of the page.**

**Igniting Our App**

**Let’s create a new repository in git and configure that remote to our local repository.**

**Our local repository is not a git repository.**

**So first we need to initialize using git init to make it as git repository.**

**Local git does not know what is remote, so we have to configure that using below command**

git remote add origin <https://github.com/msreevani2109/namaste-react.git>

-this step we have to do once to setup local repository to git repository.

git branch -m Main – means we are creating default branch as main instead of master

git add . - adding all our files which needs to be go inside commit

Git and GitHub are related but serve different purposes. Here's a breakdown of the differences:

**1. Git:**

* **What it is:** Git is a distributed version control system (VCS) used to track changes in source code during software development.
* **Functionality:** It allows you to manage and keep track of code changes, revert to previous versions, and collaborate on code with others. Git is a tool that works on your local machine, allowing you to create repositories, commit changes, and merge branches.
* **Where it's used:** Git is used on your local computer or a server. It can be used independently of any online platform.

**. GitHub:**

* **What it is:** GitHub is a cloud-based platform that hosts Git repositories online. It is a web-based service that provides a collaborative space for developers to store and share code.
* **Functionality:** GitHub offers a web interface to interact with Git repositories, providing features like issue tracking, pull requests, project boards, and more. While Git is the underlying version control system, GitHub adds a layer of collaboration, visibility, and management tools.
* **Where it's used:** GitHub is used for hosting and collaborating on Git repositories, allowing teams to share code, track issues, and work together more efficiently.

**Key Differences:**

* **Git** is the tool you use to manage version control locally.
* **GitHub** is a service that hosts Git repositories and adds collaborative features like online editing, pull requests, and issue tracking.