# Chapter1 – Inception

What is React? Why is `React` known as `React`?

React is a JavaScript Library. React is named React because it quickly reacts to changes without reloading the whole page.

What is a Library?

Library is a collections of prewritten code snippets that can be used and reused to[perform certain tasks](https://generalassemb.ly/blog/how-to-easily-run-javascript-in-terminal/). A particular JavaScript library code can be plugged into application code which leads to[faster development](https://generalassemb.ly/blog/how-to-go-from-zero-to-hero-in-javascript-fast/) and fewer vulnerabilities to have errors.

Examples - React, jQuery, Underscore

What is a Framework?

Framework provides a basic foundation or structure for a website or an application.

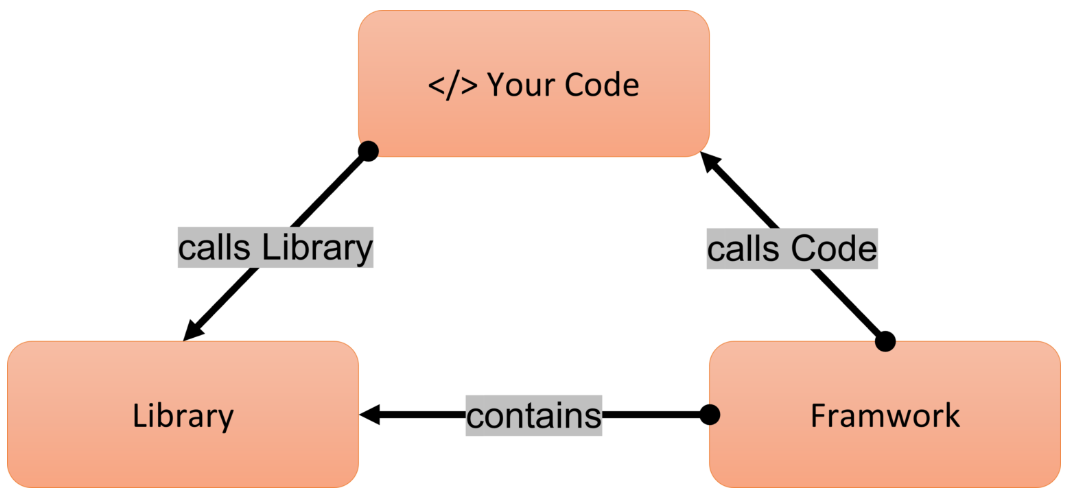
Example - Angular

Similarities between Library and Framework?

Frameworks and libraries are code written by third parties to solve regular/common problems or to optimise performance.

Difference between Library and Framework?

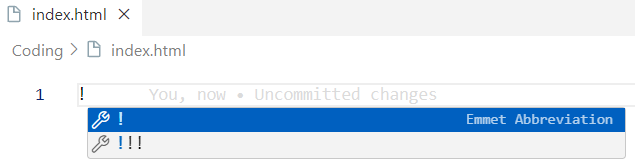
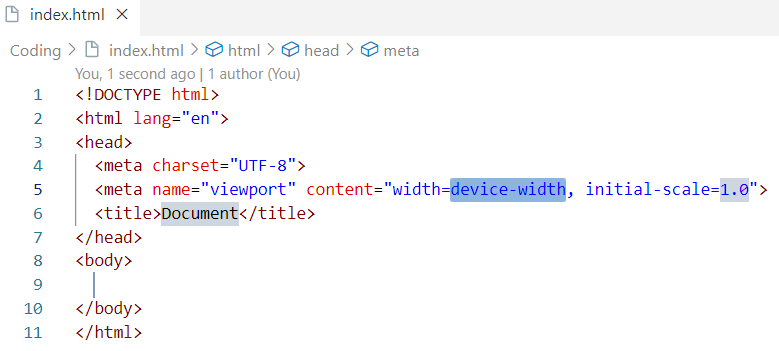
A key difference between the two is Inversion of control. When using a library, the control remains with the developer who tells the application when to call library functions. When using a framework, the control is reversed, which means that the framework tells the developer where code needs to be provided and calls it as it requires.



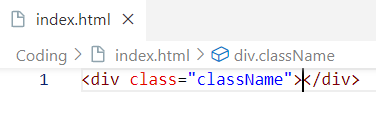
What is `Emmet`?

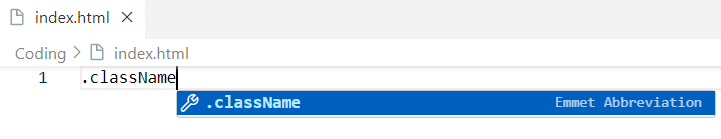
It’s a tool for IDE and text editors that help developers like us to generate full fledge boiler plate code snippets from shortcuts.

For example, when we type ! in an html file and press enter –

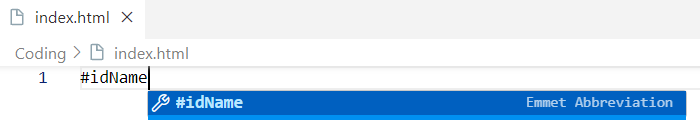
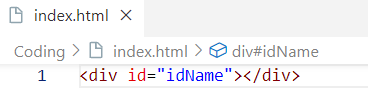


when we type .<className> in an html file and press enter –



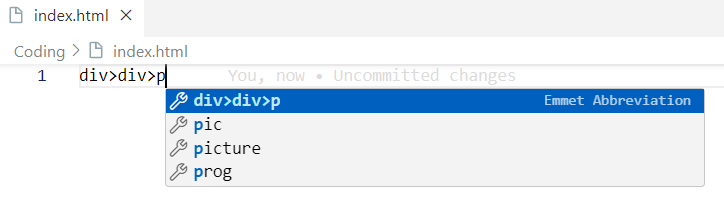
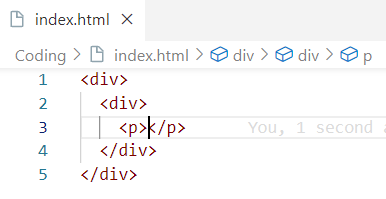


when we type #<idName> in an html file and press enter –



when we want to generate a nested html elements structure we need below syntax.

Parent > child 1 > child’s child > and so on …



Before learning react, let’s create a basic hello world program using html.





Now let’s create a basic hello world program using JavaScript.



Now let’s create a basic hello world program using react. Our browser does not know what react is, we need to add react into our project to use all of its superpowers. There are two ways of adding react into our project.

Way1: Using CDNs

Way 2: Install react library via NPM and then import react into your project and use it.

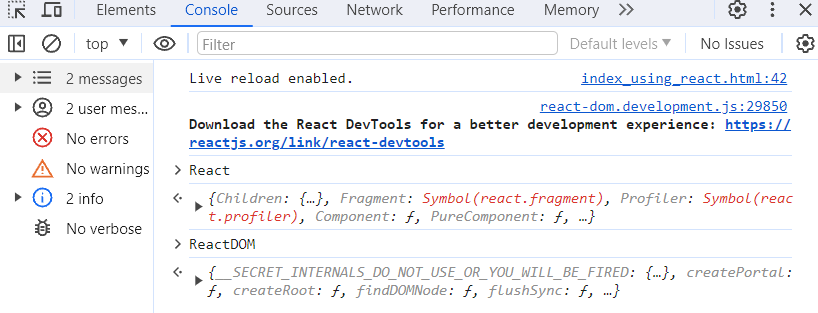
In this chapter we are going discuss about adding react into our project using CDN.

What is `CDN`? Why do we use it?

CDN stands for Content Delivery Network which refers to a geographically distributed group of servers that work together to provide fast delivery of Internet content. The main use of a CDN is to deliver content through a network of servers in a secure and efficient way. CDN has 100% uptime.



Using these two CDN links we made react available to the browser. This is the shortest react program. why? because If we go to the browser and type React and ReactDOM we will get objects like -



These are coming from the CDNs that we have just imported.

The first CDN link <https://unpkg.com/react@18/umd/react.development.js> is the core file of react.

The second CDN link <https://unpkg.com/react-dom@18/umd/react-dom.development.js> is the file which we need for DOM Operations.

What is difference between react and react-dom/client packages?

The react package contains React.createElement(), React.Component, React.Children, and other helpers related to elements and component classes. We can use these helper APIs to create react components or elements. The react-dom/client package contains ReactDOM.render() which takes a react element as argument and it defines where the react element has to be rendered inside DOM.

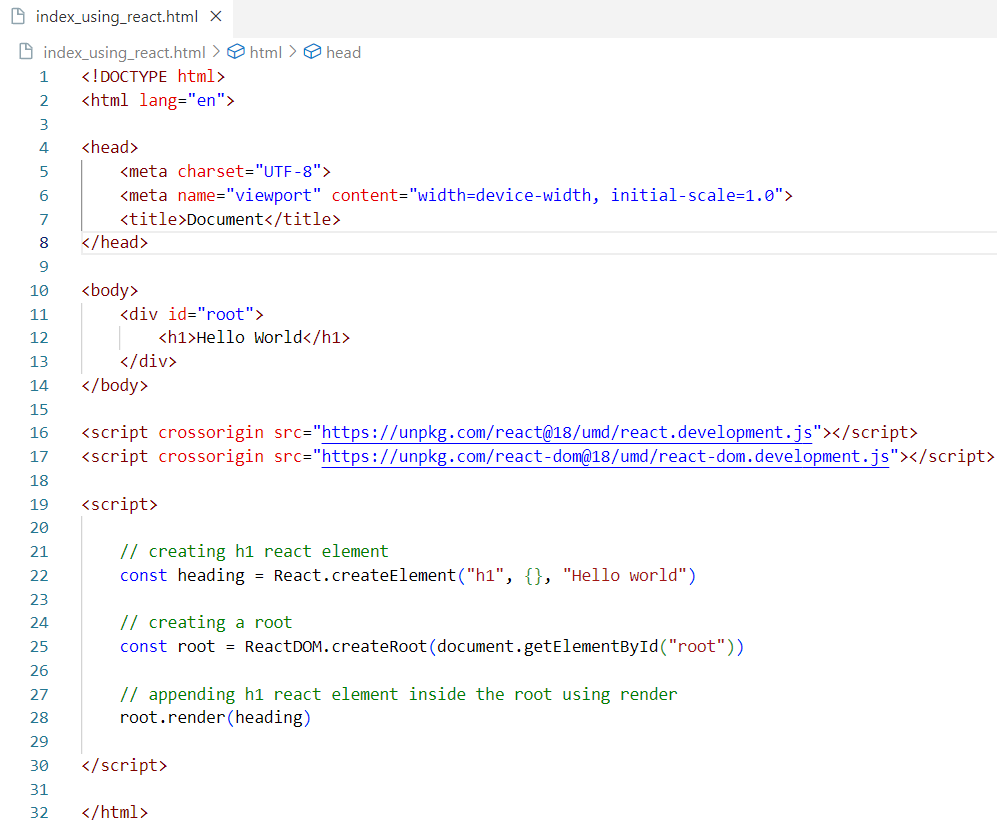
What is crossorigin in script tag?

The crossorigin attribute sets the mode of the request to an HTTP CORS Request. Web pages often make requests to load resources on other servers. Here is where CORS comes in.

A cross-origin request is a request for a resource (e.g. style sheets, iframes, images, fonts, or scripts) from another domain. CORS is used to manage cross-origin requests.

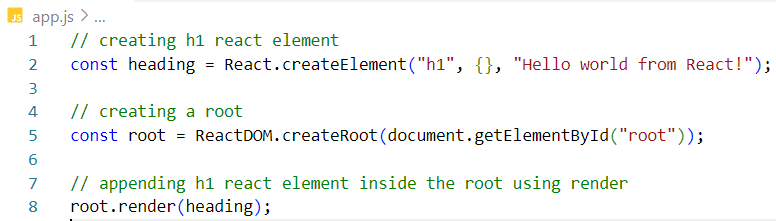
CORS stands for Cross-Origin Resource Sharing, and is a mechanism that allows resources on a web page to be requested from another domain outside their own domain. It defines a way of how a browser and server can interact to determine whether it is safe to allow the cross-origin request. CORS allows servers to specify who can access the assets on the server, among many other things.

If we navigate to these CDN links, we get source code of React which is written using JavaScript by Facebook developers. So, react in the end of the day is JavaScript. Now let’s continue writing hello world program using react.



Let’s keep this script code in an external JavaScript file and put a reference to this file in html code.





The application behaves the same way it was behaving before.

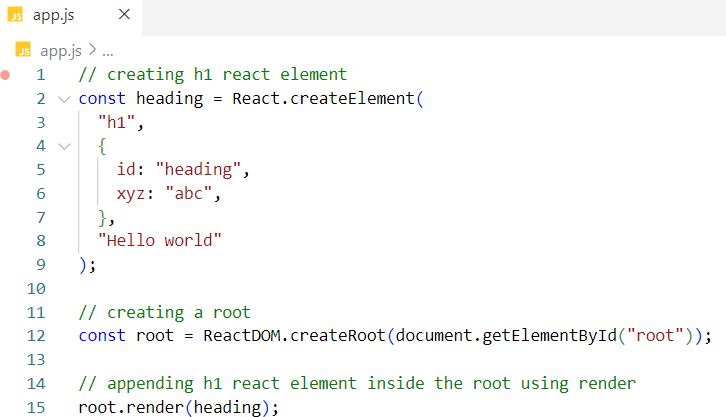
React.createElement

createElement is a top-level API that React uses to create a React element. This API takes three arguments such as tagName, props & children and returns a JavaScript object AKA react element.

Syntax - React.createElement(tagName, {props}, [children])

Where tagName represents Name of html tag like h1, div etc, Props represents attribute to html element, props could also be null & children represents the content of html element or an array list of react elements.

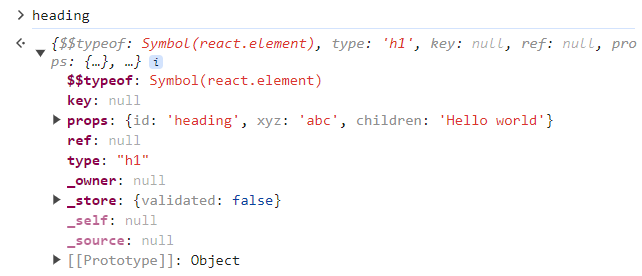
In the above example lets add some attributes to the h1 react element. We configure these attributes inside props object.



We see these attributes got added to the h1 element in the DOM.



In above JavaScript code snippet, heading is a plain JavaScript object which is a react element.



How to create nested elements using react?

Representation of 2 level Nested elements using html

<div id="parent">

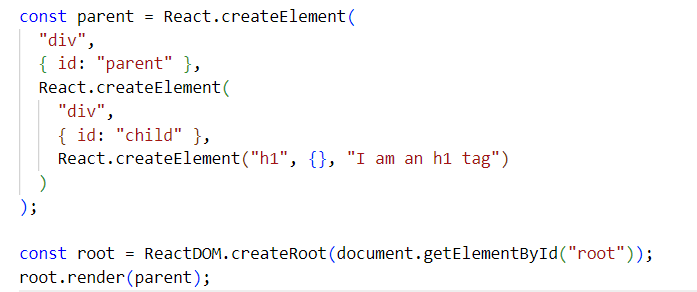
  <div id="child">

    <h1>I am an h1 tag</h1>

  </div>

</div>

Representation of 2 level Nested elements using react





Representation of sibling elements using html – (in below example h1 and h2 are siblings)

<div id="parent">

  <div id="child">

    <h1>I am an h1 tag</h1>

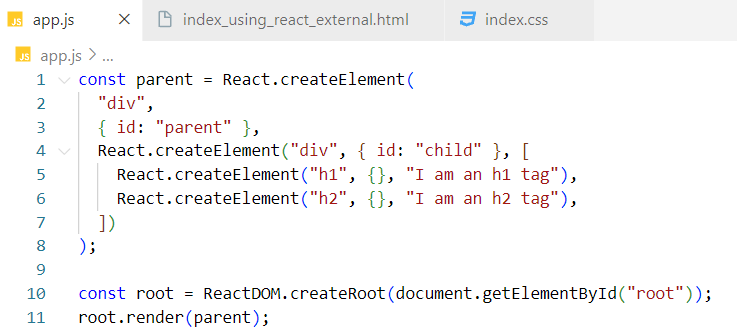
    <h2>I am an h2 tag</h1>

  </div>

</div>

Representation of sibling elements using react - react wraps these two siblings inside an array

For one child it does not use array as we have seen above.



Representation of Nested sibling elements using html

<div id="parent">

  <div id="child1">

    <h1>I am an h1 tag</h1>

    <h2>I am an h2 tag</h1>

  </div>

  <div id="child2">

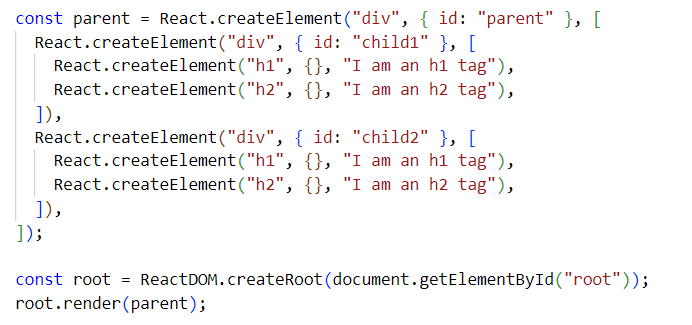
    <h1>I am an h1 tag</h1>

    <h2>I am an h2 tag</h1>

  </div>

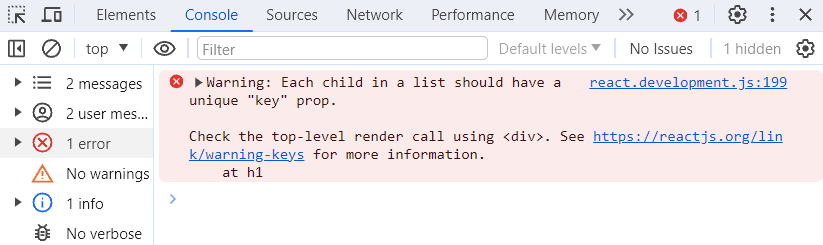
</div>

Representation of Nested sibling elements using react



The code works fine but we get and warning in the console when multiple children were introduced

This error is genuine because we have not provided Key for each of the children, because react tracks these elements by their Key. We will discuss more on keys in later chapters.

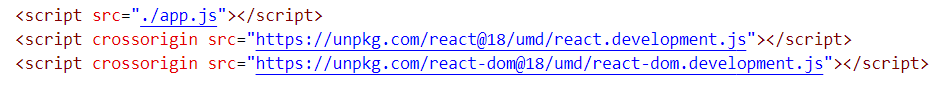


If we have deep level nested elements, writing code like this will not be useful as it is difficult to read, maintain and test. To overcome this problem JSX was introduced. We will discuss JSX in later chapters.

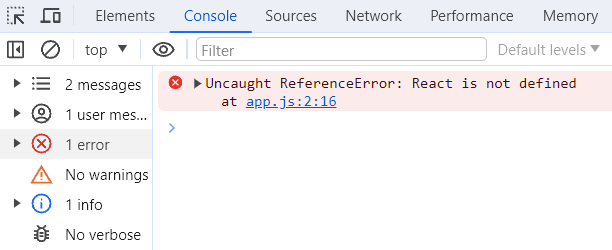
Important Observations

1. Order of the script tag matters.

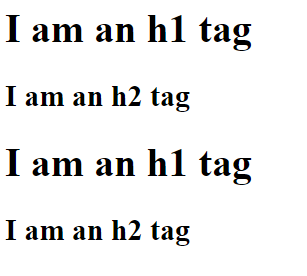
We have moved our script tag from bottom to two levels up that is before CDN fetches react code into our browser.



In this case we get an error that React is not defined. We are trying to access React before react source code is available to our browser.



2.While rending react element into the dom React overrides the children elements present inside root element.



Not rendered will be replaced by the react element that we are pushing into the root. So, when we load the page Not rendered will be displayed for a fraction of seconds until react source code is fetched from the CDNs and available to the browser. Once react code is available, this will be replaced by the react element which is two children div elements under one parent div.

3. We can use react to a specific component or part of our application.

In next example, we are using react on root component, we are not using react on header and footer component. This is why we call react as a library. because we can use react in a small portion of our app and we as developers have control on our application.



ReactDOM.createRoot

createRoot is a ReactDOM API which creates a root in the DOM where JS engine runs react application. It returns an object with 2 methods render and unmount

# Syntax - ReactDOM.createRoot(document.getElementById(<rootID>))

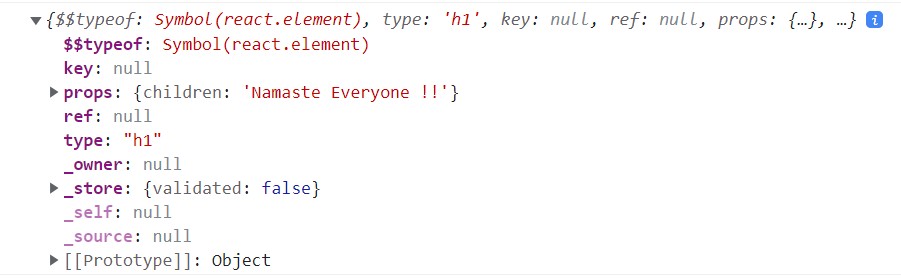
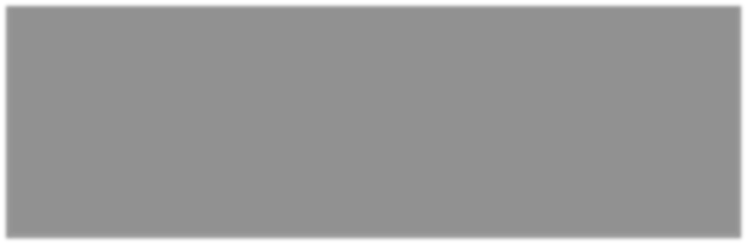
render

render method takes react element or JSX as argument and renders it inside the root element inside DOM.This method returns undefined.

Syntax - <root>.render(<JSX/React element>)

If we log heading in the console, we will get a plain JavaScript object.

-



What is difference between `Development` and `production` stage of an application?

In `Development` stage, application is not public meaning we don’t deploy the application into production. while `production` is the term used for the same application when it's made `public`. `Development build` is several times `slower` than the `production build`.