# Introduction

## Rendering the First React Code

```
const heading = React.createElement(
  "h1",
  { class: "Kartik" },
  "Hello World ! From React"
);
console.log(heading);

const root = ReactDOM.createRoot(document.getElementById("root"

root.render(heading);
```

**Render: Render function will take this object created by React.createElement and put it in the DOM**

> React.createElement - creates an object

How to create a nested HTML Structure inside React Js ?

```
<div>
    <div>
        <h1></h1>
    </div>
</div>
```

```
const parent = React.createElement(
  "div",
```

```
  { id: "parent" },
  React.createElement(
    "div",
    { id: "child" },
    React.createElement("h1", {}, "Hey I am a H1 Tag")
  )
);
console.log(parent);

const root = ReactDOM.createRoot(document.getElementById("root"

root.render(parent);
```

At the end of the day, it is creating a <React.Element />

What if we want to create sibling ?

```
<div>
    <div>
        <h1></h1>
        <h1></h1>
    </div>
</div>
```

```
const parent = React.createElement(
  "div",
  { id: "parent" },
  React.createElement("div", { id: "child" }, [
    React.createElement("h1", {}, "Hey I am a H1 Tag"),
    React.createElement("h1", {}, "Hey there I am a second h1 ta
  ])
);
```

We will be creating an array for creating the siblings

- But what if we want to create more siblings then the code will become very messy and will be difficult to read, so to overcome this problem we have a thing called “**JSX**”

## Now you might be doubting that the sequence matters in the code

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-s
    <title>Document</title>
    <link rel="stylesheet" href="index.css" />
  </head>
  <body>
    <div id="root"></div>
            <script src="app.js"></script>
    <script
      crossorigin
      src="https://unpkg.com/react@18/umd/react.development.js"
    ></script>
    <script
      crossorigin
      src="https://unpkg.com/react-dom@18/umd/react-dom.developr
    ></script>

if we move this script tag to the top of the react script tag it
<!-- <script src="app.js"></script> -->

  </body>
</html>
```

Now what if we have some tags inside the root attribute

```
<div id="root">
      <h1>My name is Nitin</h1>
      <h1>My name is Nitin</h1>
      <h1>My name is Nitin</h1>
      <h1>My name is Nitin</h1>
      <h1>My name is Nitin</h1>
      <h1>My name is Nitin</h1>
      <h1>My name is Nitin</h1>
      <h1>My name is Nitin</h1>
      <h1>My name is Nitin</h1>
      <h1>My name is Nitin</h1>
      <h1>My name is Nitin</h1>
    </div>
```

The root.render() will replace all the tags that are present in the root attribute with the elements that we have created.

## What is the difference between a Library and a Framework?

A library targets a specific functionality while a Framework tries to provide everything that is required to develop an application.

Advantages:

- React can be used in your old application also