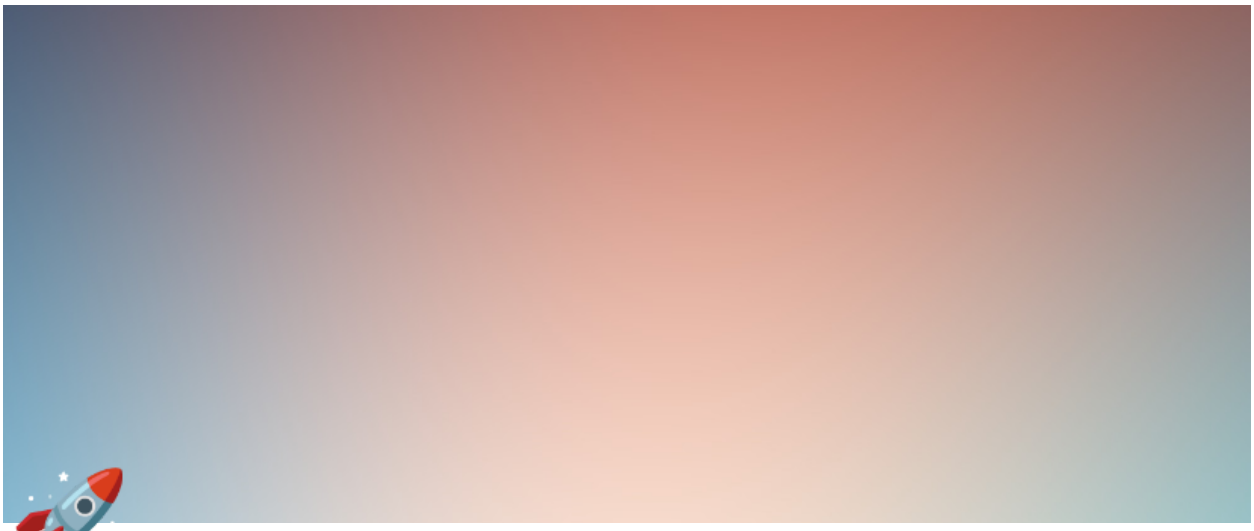


Inception! (Namaste-React)

💡 Please make sure to follow along with the whole ***"Namaste React"*** series, starting from Episode-1 and continuing through each subsequent episode. The notes are designed to provide detailed explanations of each concept along with examples to ensure thorough understanding. Each episode builds upon the knowledge gained from the previous ones, so starting from the beginning will give you a comprehensive understanding of React development.

💡 I've got a quick tip for you. To get the most out of these notes, it's a good idea to watch **Episode-1** first. Understanding what ***"Akshay"*** shares in the video will make these notes way easier to understand.

> **So Let’s begin our Namaste React Journey**
>
> - In this course we will study how the React concepts are actually applied into the industry i.e. into the real world projects.
> - So are you ready to fall in love with React????

Introducing React.

Q) What is React? Why React is known as ‘React’?

React is a JavaScript Library. The name ‘React’ was chosen because the library was designed to allow developers to react to changes in state and data within an application, and to update the user interface in a declarative and efficient manner.

Q) What is Library?

Library is a collections of prewritten code snippets that can be used and reused to perform certain tasks. A particular JavaScript library code can be plugged into application code which leads to faster development and fewer vulnerabilities to have errors.

Examples: React, jQuery

Q) What is Framework?

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

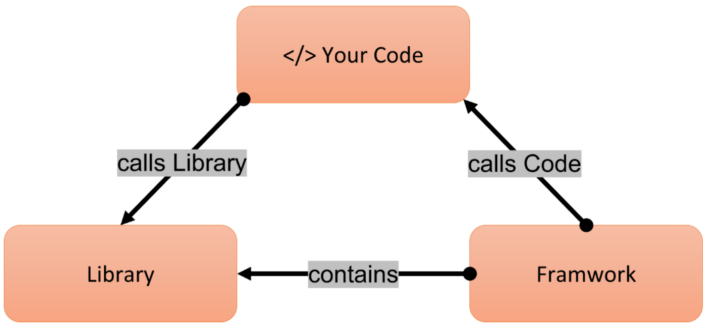
Examples: Angular

Q) Similarities between Library and Framework?

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

Q) 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.



* Emmet:

Emmet is the essential toolkit for web-developers. It allows you to type shortcuts that are then expanded into full-fledged

boiler plate code for writing HTML and CSS.

Q) Create Hello World Program using only HTML?

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <div id="root">
      <h1>Hello World! using only HTML</h1>
    </div>
  </body>
</html>
```

Q) Create Hello World Program using only JavaScript?

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>

  <body>
    <div id="root"></div>
  </body>
  <script>
    const heading = document.createElement("h1");
    heading.innerHTML = "Hellow World! From Javascript";

    const divElement = document.getElementById("root");

    divElement.appendChild(heading);
  </script>
</html>
```

Q) Create Hello World Program using only React?

```html
<body>
  <div id="root"></div>

  <script
    crossorigin
    src="https://unpkg.com/react@18/umd/react.development.js"
  ></script>
  <!--This file i.e. react.development.js contains the React Code which is plain JS-->

  <script
    crossorigin
    src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"
  ></script>
  <!--This file i.e. react-dom.development.js is useful for DOM operations or use to modify the DOM-->

  <script>
    const heading = React.createElement("h1", {}, "Hello World from React!");
    const root = ReactDOM.createRoot(document.getElementById("root"));
    root.render(heading);
  </script>
</body>
```

* Cross Origin:

The crossorigin attribute in the script tag enables Cross-Origin Resource Sharing (CORS) for loading external JavaScript files from different origin than the hosting web page. This allows the script to access resources from the server hosting the script, such as making HTTP requests or accessing data.

Q) What is {} denotes in above code?

```
<div id="root">
  <h1 id="title">Hello World!</h1>
</div>
```

This (id='title'), classes, etc should come under {}. Whenever I'm passing inside {}, will go as tag attributes of h1.

📢 **NOTE: React will overwrite everything inside "root" and replaces with whatever given inside render.**

Q) Do the below HTML code in React?

```
<div id="parent">
  <div id="child1">
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
  </div>
  <div id="child2">
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
  </div>
</div>
```

To build the structure like this in React

```
const parent = React.createElement("div", { id: "parent" }, [
  React.createElement("div", { id: "child1" }, [
    React.createElement("h1", { id: "heading_1" }, "Heading 1"),
    React.createElement("h2", { id: "heading_2" }, "Heading 2"),
  ]),
  React.createElement("div", { id: "child2" }, [
    React.createElement("h1", { id: "heading_1" }, "Heading 1"),
    React.createElement("h2", { id: "heading_2" }, "Heading 2"),
  ]),
]);

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

root.render(parent);

```