**React JS**

1. React is a JavaScript library used for building User Interfaces.
2. React is used to build Single Page Applications.
3. React allows us to create reusable UI components.

**Library Vs Framework**

Graphical user interface, application

Description automatically generated

**Library:** Libraries are collections of prewritten code snippets that can be used and reused to perform common functions. Ex: ReactJS, jQuery etc…

**Framework:** Framework is a collection of libraries that provide a developer with prewritten code snippets for routine programming. Ex: AngularJS, VueJS, EmberJS etc…

The key difference between the Library and Framework is something known as **“Inversion of Control”.**

|  |  |
| --- | --- |
| By using a library, you can control the flow of the application and call the library where and when you want. **Your code will be in-charge of flow.** | By using a framework, framework will control the flow of the applications and calls your code. **Framework will be in-charge of flow.** |

**React Vs ReactDOM**

|  |  |
| --- | --- |
| **React** | **ReactDOM** |
| The “React” package holds the react source for the components, state, props and all the code that is React. | ReactDOM package is the glue between React and DOM. |
| React library is responsible for creating views. | ReactDOM library is responsible to render UI in the browser. |

<script>

*// Using React - Create HTML Element*

*// createElement has 3 parameters. (type, props, children)*

*const* heading = React.createElement(

"h1",

{

id: "title",

className: "myTitle",

},

"Hello React"

);

*// heading holds the react objectof type h1*

console.log(heading);

*// Get the root using ReactDOM.createRoot*

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

console.log(root);

*// Render the HTML created using createElement*

*// If any element exists in root the render will replace.*

root.render(heading);

</script>

**Regular Vs Async Vs Defer – Attributes in script tag.**

**End of the body –** Scripts are fetched and executed “after the HTML is completely parsed”

Async – Scripts are fetched asynchronously and executed immediately.

**Defer –** Scripts are fetched asynchronously and executed “after the HTML is completely parsed”

**Crossorigin:** The attribute crossorigin in the script tag specifies that CORS - Cross Origin Resource Sharing is supported when loading an external script file from third party server or domain. The crossorigin attribute valid on the <audio>, <video>, <img>, <link>, <script> elements.