Frontend Development

Library:

A library is a collection of pre-written code that developers can use to perform common tasks. It consists of functions, routines, and classes that can be imported and called in a program.

Developers have more control over their code and can choose when and how to use specific functions from the library. Libraries are generally used for solving specific problems or tasks, and developers can use them selectively within their projects.

jQuery is a popular JavaScript library that simplifies DOM manipulation and event handling. In a library, developers have more control and can choose when and how to use specific components.

Libraries are typically called by the developer, meaning the control flow is determined by the developer. Libraries tend to have a narrower scope, addressing specific functionalities or tasks. Libraries are generally more flexible, allowing developers to use them in diverse ways within their projects.

Framework:

A framework is a more comprehensive set of tools, rules, and conventions designed to structure and guide the development process. It provides a foundation for building an application and often dictates the overall architecture. Frameworks typically include libraries but go beyond by offering a predefined structure and flow for the application.

Developers build their applications within the framework's structure, following its guidelines and conventions. Frameworks often impose a specific architecture (e.g., MVC - Model-View-Controller) and provide a higher level of abstraction.

Angular, React, and Vue.js are popular JavaScript frameworks for building web applications. In a framework, developers follow the structure and guidelines provided by the framework, relinquishing some control in exchange for a standardized development approach.

Frameworks often follow the Inversion of Control (IoC) principle, where the framework controls the flow of the application, and developers plug in their code at predefined points.

Frameworks have a broader scope, providing a comprehensive structure for building entire applications. Frameworks may have a steeper learning curve but offer a more opinionated structure, making development faster and more standardized.

React.JS:

React.js, commonly referred to as React, is an open-source JavaScript library for building user interfaces (UIs) or user interface components. Developed and maintained by Facebook, React has gained widespread adoption in the web development community due to its declarative and efficient approach to building interactive UIs.

React is a javascript library for building user interfaces. The UI(User Interface) id the point of human-computer interaction and communication in a device. This can include display screens., keyboards, a mouse and the appearance of a desktop. React.js a client-side JavaScript library, all about building modern reactive user interface for the web, declarative, component-focused approach.

Features Of React.JS:

1. Declarative Syntax:

React uses a declarative syntax, allowing developers to describe the desired outcome of the UI rather than focusing on the step-by-step imperative process of achieving that outcome.

2. Component-Based Architecture:

React is centered around a component-based architecture. Components are self-contained, reusable pieces of code that encapsulate a part of the user interface.

3. Virtual DOM (Document Object Model):

React introduces the concept of a virtual DOM, a lightweight in-memory representation of the actual DOM. When the state of a component changes, React first updates the virtual DOM and then efficiently updates the actual DOM by identifying and applying only the necessary changes.

4. Unidirectional Data Flow:

React follows a unidirectional data flow, meaning that data flows in a single direction within the application. This helps maintain a predictable state and simplifies the debugging process.

5. JSX (JavaScript XML):

JSX is a syntax extension for JavaScript that allows developers to write UI elements using a syntax similar to XML or HTML. JSX makes the code more readable and concise. It gets transpiled to standard JavaScript before being executed in the browser.

6. Reusable Components:

React promotes the creation of reusable components. Developers can build a library of components and reuse them across different parts of the application or in different projects.

7. React Hooks:

React Hooks are functions that allow developers to use state and other React features in functional components. They enable the use of state and lifecycle features in functional components, reducing the need for class components.

8. Community & Ecosystem:

React has a large and active community, which means abundant resources, third-party libraries, and tools. The ecosystem around React has expanded to include solutions for state management (Redux), routing (React Router), and more.

9. React Native:

React extends its capabilities beyond web development with React Native. React Native allows developers to use React to build native mobile applications for iOS and Android, sharing a significant portion of the codebase between web and mobile projects.

Is React.js Library Or Framework:

React JS is not a framework. React js is a javascript library for building user interfaces. It is also known as ReactJS and react.js so don’t get confused if you read different notation in different places. react knows only one thing that is to create an awesome UI.

While React itself is a library, it's worth noting that there are frameworks built on top of React that provide more opinionated structures and additional features. For example, Next.js and Gatsby.js are frameworks that leverage React for building web applications but also provide additional functionality and structure.

1. Flexibility:

React is more focused on providing a set of tools and components for building user interfaces. It doesn't impose a strict structure or architecture on the entire application, allowing developers more flexibility in terms of project organization and technology choices.

2. Component-Based:

React is centered around a component-based architecture, where UIs are built by composing reusable and modular components.

3. One Part Of The Ecosystem:

React is often just one part of a broader ecosystem of tools and libraries used in a modern web development stack.

4. No Prescribed Project Structure:

Unlike frameworks that often prescribe a specific project structure and dictate the flow of control, React leaves certain aspects, such as project structure and state management, to the discretion of the developer. This aligns with the characteristic of a library.

History Of React.JS:

React was first designed by Jordan Walke, a software engineer at Facebook. It was first developed and deployed for facebook news feed around 2011. In 2013, react was open sourced at JS conference.

An open source project is where th code to a certain project is completely open source. That means anybody can readily see the code that went into a project. Open source projects also are usually community-based and accept help from other programmers.

About React:

Component based approach. A component is one of the core building blocks of react. In other words we can say that every application you will develop in react will made up of pieces called components. Components make the task of building UIs much easier.

Uses a declarative approach declarative programming paradigm that expresses the logic of a computation without describing its control flow. Dom updates and handled gracefully. Reusable code. React is designed for speed, speed of implementing the application simplicity and scalability.

Why React.JS:

Created and maintained by Facebook. It has huge community on github. Component based architecture.

Prerequisites For React.js:

1. Basic HTML, CSS, & JS

2. Basic Understanding Of ES6 features.

3. Basic Understanding Of How To Use npm(Node Package Manager)

React.JS Installation Process:

1. Install ` **VS Code(Visual Studio Code)** ` code editor from Internet

2. Add ` **Live server** `, ` **JavaScript(ES6)** ` code snippets, ` **Babel JavaScript** ` Extensions in VS Code.

3. Download & Install ` **npm(Node Package Manager)** ` from internet from node.org website.

4. Open ` **CMD (Command Prompt)** ` and add ` **npm –v** ` to see the version and for check whether it is working properly or not.

5. Download & install ` **NodeJS** ` from ` **node.org** ` website.

6. Open ` **CMD(Command Prompt)** ` and add ` **node -v** ` to see the version of node check whether it is working properly or not.

7. In ` **CMD(Command Prompt)** ` add ` **npm install -g create-react-app** ` it will install the react library in your system

8. In ` **CMD(Command Prompt)** ` add ` **create-react-app -version** ` it will show your current react version.

9. In ` **CMD(Command Prompt)** ` add ` **create-react-app <projectname>** ` it will create a react project for you. ` **It Will Take Time Max: 3-5 Minutes!** `

10. After creation of project in ` **CMD(Command Prompt)** ` add ` **cd <projectname>** ` so you will enter inside the project.

11. In ` **CMD(Command Prompt)** ` add ` **npm start** ` so it will show the output in your default browser(recommended: Chrome).

React can be used to control parts of HTML pages or entire pages. ` **Widget** ` approach on a multi-page application. Some pages are still rendered on and served by a backend server.

React can also be used to control the entire frontend of a web application. Single page application approach. Server only sends one HTML page, thereafter, react takes over and control the UI.

React.js creates Component-Driven User Interfaces & Building Interactive & Scalable UIs. React js surrounding things are ` **React.js Core Syntax & JSX** `, ` **Working With Components** `, ` **Working With Data** `.

What Is Component?

"***React JS is a javascript library for building user interfaces.***" React makes building complex, interactive and reactive user interfaces simpler. React Is All About Components. React is all about components because all user interfaces in the end are made up of components.

Why Components?

Reusability(Don't repeat Yourself), Separation Of Concerns(Don't do too many things in one and the same place (funciton)), split big chunks of code into multiple smaller functions.

How Is A Component Built?

We combined HTML , CSS, & JS in all this component. React allows you to create re-usable and reactive components consisting of HTML and JavaScript and CSS. For building component we will use Declarative Approach. Declarative approach define the desired target state(s) and let react figure out the actual javascript DOM instructions. ` **Build your own custom HTML elements.** `

NPM:

NPM (Node Package Manager) is the world's largest software registry. The registry contains over 800,000 code packages.

React Folder Structure:

node_modules:

in node modules folder there are all the features and import codes are there which were useful for react coding.

public:

in public folder there is only html file is there which is used for to show the UI output in browser.

src:

in the src folder you can create component folder for your customized component. Inside the src folder. You can see the root component folder which is in "App.js" which is connected to the root js file "Index.js". The "index.js" file is connected to the "public/index.html" for to transfer the js operations and JSX code.

package.json:

this a json file in which you can see the metadata of your entire project.

index.css:

this is the connected CSS file for index.js