React Overview

- ReactJS is a Open source JavaScript Library for building single page applications.
- ReactJS is a Open source JavaScript Library for building UI components in SPA.
- ReactJS was developed and maintained by Facebook
- ReactJS is one of the most popular JavaScript front-end libraries which has a strong foundation and a large community.
- ReactJS heavily relay on JavaScript
- The main objective of ReactJS is to develop User Interfaces (UI).
- It uses virtual DOM (JavaScript object), which improves the performance of the app.
- The JavaScript virtual DOM is faster than the regular DOM.

Prerequisites

- *Before learning ReactJS, you must have a good knowledge of JavaScript, HTML5, and CSS.*
- *The knowledge of ECMAScript 2015 or ES6 syntax can also be more helpful.*

Advantage of ReactJS

- React.js, developed by Facebook, is a widely used JavaScript library for building user interfaces. It has gained popularity for several reasons, and it comes with various advantages that make it a preferred choice for many developers. Here are some key advantages of React.js
- **Declarative Syntax:** React uses a declarative syntax, allowing developers to describe how the UI should look and behave. This makes it easier to understand and reason about the code, as developers can focus on the desired outcomes.
- **Component-Based Architecture:** React follows a component-based architecture, where the UI is broken down into reusable and modular components. This promotes code reusability, maintainability, and makes it easier to manage complex UIs by isolating functionality into self-contained components.
- **Virtual DOM (Document Object Model):** React uses a virtual DOM to efficiently update the actual DOM. Instead of directly manipulating the DOM for every change, React creates a lightweight virtual representation of the DOM in memory and updates only the parts that have changed. This results in better performance and a more responsive user interface.

Advantage of ReactJS Cont..

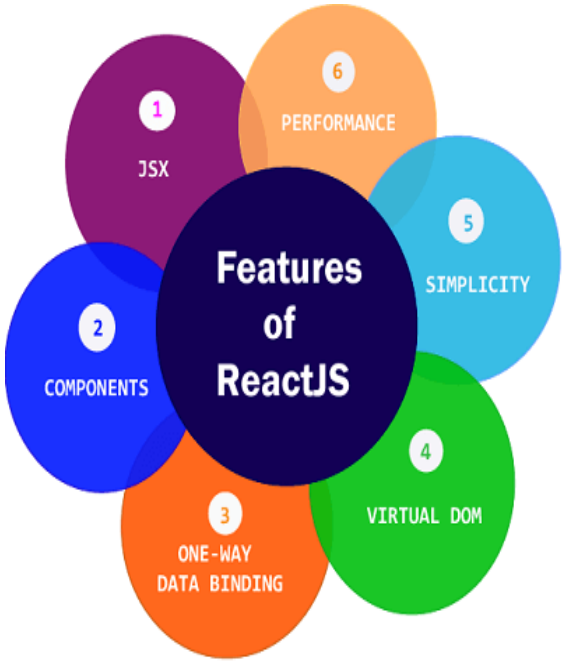
- **Efficient Rendering:** React efficiently updates the UI by only re-rendering the components that have changed, thanks to the virtual DOM. This approach minimizes the impact on performance and ensures a smoother user experience.
- **JSX (JavaScript XML):** React uses JSX, a syntax extension for JavaScript that allows HTML-like code to be written in JavaScript files. JSX makes it easier to write and understand the structure of UI components, and it is later transpiled to regular JavaScript.
- **Unidirectional Data Flow:** React follows a unidirectional data flow, which means that data in a React application flows in a single direction, from parent components to child components. This makes it easier to understand how data changes propagate through the application and helps prevent unexpected side effects.
- **React Native for Mobile Development:** React can be used with React Native to build cross-platform mobile applications. By sharing a significant portion of the codebase between web and mobile versions, developers can reduce development time and maintenance efforts.

Advantage of ReactJS Cont..

- **Strong Community Support:** React has a large and active community of developers, which means there are abundant resources, tutorials, and third-party libraries available. The community support fosters innovation and helps developers overcome challenges.
- **Backed by Facebook:** React is developed and maintained by Facebook, ensuring ongoing support and updates. The fact that it is used in several high-traffic and complex applications within Facebook itself is a testament to its scalability and reliability.
- **Ecosystem and Tooling:** React has a rich ecosystem with a variety of tools and libraries that complement its functionality. Tools like Redux for state management, React Router for navigation, and Create React App for project setup contribute to a well-rounded development experience.

ReactJS Features

- React.js is a JavaScript library for building user interfaces, and it comes with several features that contribute to its popularity and effectiveness in developing modern web applications. Here are some key features of React.js:



ReactJS Features Cont..

- **Virtual DOM:**
- React uses a virtual DOM to improve rendering performance. The virtual DOM is a lightweight copy of the actual DOM in memory, and React compares it with the real DOM to determine the minimal set of changes needed for an update.
- **JSX (JavaScript XML):**
- JSX is a syntax extension for JavaScript that allows HTML-like code to be written directly in JavaScript files. It enhances code readability and enables developers to express UI components more naturally.
- **Component-Based Architecture:**
- React follows a component-based architecture, where UIs are built by composing individual, self-contained components. This promotes reusability, maintainability, and a modular approach to building interfaces.
- **Unidirectional Data Flow:**
- React enforces a unidirectional data flow, meaning that data in a React application flows in a single direction—from parent components to child components. This makes it easier to manage and understand how data changes propagate through the application.

ReactJS Features Cont..

- **Reusability and Composability:**
- Components in React can be reused across different parts of an application, leading to a more efficient and maintainable codebase. Composing complex UIs becomes easier by combining smaller, reusable components.
- **State Management:**
- React provides a mechanism for managing the state of components. State is used to store and manage data that can change over time, and when the state changes, React efficiently updates the UI to reflect those changes.
- **Lifecycle Methods:**
- React components have lifecycle methods that allow developers to execute code at specific points in the component's lifecycle, such as when it is first mounted, updated, or unmounted. This provides hooks for developers to perform actions like data fetching or cleanup.
- **React Hooks:**
- Introduced in React 16.8, hooks allow functional components to manage state and side effects previously only possible in class components. Hooks, such as **useState** and **useEffect,** , make it easier to write and organize code in functional components.

ReactJS Features Cont..

- **Declarative Syntax:**
- React uses a declarative syntax, allowing developers to describe how the UI should look and behave, and React takes care of updating the DOM to match the desired state. This makes the code more expressive and easier to understand.
- **React Router:**
- React Router is a popular library for handling navigation in React applications. It allows developers to define and manage different routes in a single-page application.
- **Ecosystem and Tooling:**
- React has a rich ecosystem of libraries and tools that complement its functionality. Tools like Redux for state management, Jest for testing, and webpack for bundling contribute to a comprehensive development experience.
- **Community and Documentation:**
- React has a large and active community of developers. The community provides support, shares best practices, and contributes to a wealth of documentation and tutorials, making it easier for developers to learn and stay updated.

Applications build by React

- There are numerous applications and websites built using React. Here are some well-known applications and websites that were built using React:
- **Facebook**: React was developed by Facebook, and it's widely used in various parts of the Facebook website, including the main interface and dynamic components.
- **Instagram**: Instagram, also owned by Facebook, uses React extensively for its user interface to provide a seamless and responsive experience.
- **WhatsApp Web**: The web version of WhatsApp uses React for its user interface components.
- **Twitter**: Certain parts of Twitter's web interface are built using React, providing a more dynamic and interactive experience for users.
- **Netflix**: While not the entire Netflix platform, certain sections of the user interface, especially those requiring dynamic updates, are built using React.
- **Uber**: React is used in some parts of Uber's web applications to create a responsive and efficient user interface.
- **GitHub**: GitHub, a platform for version control and collaboration, uses React for its web interface.
- **Dropbox**: React is employed in certain components of Dropbox's web application.
- **Slack**: Slack, a popular collaboration tool, uses React for its frontend to deliver a smooth and responsive user experience.
- **WhatsApp Desktop**: The desktop version of WhatsApp is built using React for its user interface.

React Versions

- React.JS was first used in 2011 for Facebook's Newsfeed feature.
- Facebook Software Engineer, Jordan Walke, created it.
- Initial Release to the Public (V0.3.0) was in July 2013.
- Current version of React.JS is V18.2.0 (June 2022).
- Click here for complete version history https://legacy.reactjs.org/

React Building Blocks

- In React, building blocks refer to the fundamental elements and concepts used to create user interfaces. These building blocks are the core concepts that developers work with when building React applications. Here are the key building blocks in React:
- **Components:**
- Components are the fundamental building blocks in React. They are reusable, self-contained pieces of code that represent a part of the user interface. Components can be class-based or functional, and they encapsulate both the UI and the logic related to that UI.
- **JSX (JavaScript XML):**
- JSX is a syntax extension for JavaScript that allows HTML-like code to be written within JavaScript files. It is used to describe the structure of React components in a more concise and readable way.
- **Props (Properties):**
- Props are short for properties and are used to pass data from a parent component to a child component. Props are immutable (read-only) and are a way to make components dynamic by configuring them with different values.

React Building Blocks

- **State:**
- State represents the mutable data within a component. Unlike props, state is managed internally by the component and can change over time. When state changes, React automatically re-renders the component to reflect the updated state.
- **Virtual DOM (Document Object Model):**
- The Virtual DOM is a lightweight, in-memory representation of the actual DOM. React uses the virtual DOM to optimize updates by comparing the current virtual DOM with a previous version and determining the minimal changes needed to update the actual DOM.
- **Lifecycle Methods:**
- Lifecycle methods are methods that are invoked at different points in the life cycle of a React component. These methods allow developers to perform actions such as setting up or cleaning up resources, making API calls, or updating the component when certain events occur (e.g., component mounting, updating, unmounting).
- **Events:**
- React supports a synthetic event system that wraps native browser events. Event handlers in React follow a camelCase naming convention(e.g **onClick**, **onChange** etc). Events are used to handle user interactions and trigger changes in the application.

React Building Blocks

- **Conditional Rendering:**
- React allows components to conditionally render content based on certain conditions. This can be achieved using conditional statements, such as **'if'** or ternary operators, in combination with JSX.
- **Lists and Keys:**
- Rendering lists of elements is a common requirement in React applications. Keys are used to help React identify which items have changed, are added, or are removed in a list, improving performance and efficiency.
- **Forms:**
- React provides a controlled component approach for handling forms. State is used to control the form elements, and event handlers are used to update the state based on user input.
- **Hooks:**
- Introduced in React 16.8, hooks are functions that allow functional components to use state and lifecycle features that were previously only available in class components. Commonly used hooks include '**useState**' for managing state and '**useEffect**' for handling side effects.
- These building blocks collectively provide the foundation for creating dynamic, interactive, and maintainable user interfaces in React. Understanding how these concepts work together is essential for developing efficient and scalable React applications.

create react app

- "Create React App" (often referred to as CRA), which is a command-line interface (CLI) tool for bootstrapping and setting up new React projects/app.
- It helps configure the necessary configuration for the react app.
- Create React App (CRA) is not part of React itself but is maintained by the React team. It allows developers to quickly set up a new React project with a predefined project structure, build configuration, and development server. Using CRA eliminates the need for manual configuration and setup, making it easier to start working on React projects.
- For example, setting up Babel to transpile JSX into the browser-ready code, and configuring web pack to bundle your project assets.
- create-react-app includes built tools such as webpack, Babel, and ESLint.
- Current version of create-react-app is v5.0.1 (April 2022).

- **What is Babel?**
  - Babel is a JavaScript compiler that is commonly used in React applications. It allows developers to write code using the latest ECMAScript standards (such as ES6 and beyond) and other syntax extensions, and then transforms or compiles that code into a version of JavaScript that can run in environments that may not support the latest features.

```
// ES6 code
const myFunction = (param) => {
  console.log(`Hello, ${param}!`);
};

// Compiled JavaScript (using Babel)
var myFunction = function myFunction(param) {
  console.log('Hello, ' + param + '!');
};
```

Configuring React CLI

Prerequisites

- Install Node and NPM
- Install create-react-app(React CLI) globally
  - npm install -g create-react-app

How to create ReactJS APP

using npx

- npx is npm package runner
- it gets installed when you install npm
- npx create-react-app app-name

using npm

- create-react-app my-app

Run the React App

- Run this command to run the React application
- npm start my-app

- **What is Webpack?**
  - Webpack is a popular open-source JavaScript module bundler that is commonly used in React applications (as well as in many other JavaScript projects). It helps manage and bundle various assets, such as JavaScript files, CSS stylesheets, and images, into a more optimized format for deployment in web applications. Webpack simplifies the development workflow by automating tasks like code splitting, minification, and dependency management.

ESLint

- **What is ESLint?**
- ESLint is a static code analysis tool for identifying and fixing problems in JavaScript code. It is commonly used in React projects to enforce coding standards, catch common programming errors, and promote consistent and maintainable code. ESLint can be configured to check for a variety of issues, ranging from basic syntax errors to more complex patterns and potential pitfalls in your code.
- Key features and uses of ESLint in React projects include:
- **Coding Standards:** ESLint allows you to define and enforce a set of coding standards and best practices for your project. This ensures that all developers on a team follow the same guidelines, leading to a more consistent and readable codebase.
- **Error Prevention:** ESLint helps catch potential errors and bugs during the development process. It can identify issues such as undeclared variables, unused variables, and other common mistakes that might lead to runtime errors.

ESLint Cont..

- **Code Formatting:** ESLint can be configured to enforce a specific code style, including indentation, line spacing, and other formatting rules. Consistent code formatting improves code readability and makes it easier for developers to collaborate.
- **React-Specific Rules:** ESLint has a set of rules specifically tailored for React. These rules help identify common mistakes and anti-patterns in React code, ensuring that developers adhere to best practices when working with React components.
- **Integration with Development Tools:** ESLint can be integrated with development tools and text editors, providing real-time feedback to developers as they write code. This immediate feedback helps catch and fix issues early in the development process.

React App Files and Folder Structure

- **package.json** - include packages and its versions info
- **node_module** - contain packages installed by npm
- **package-lock.json** - lock dependencies to a specific version number
- **.gitignore** - contains list of ignored file types and folers
- **Readme.MD** - used for providing some instuctions
- **src** - source code
- **index.js** -entry point for react app
- **index.css** -global css file.
- **app.js**(component file),**app.css**(write css),**app.test.js**(write test cases)
- **public** - public files include index.html