**React JS Tutorials By Rahul**

1. **React js introduction:**

* **ReactJS**, also known as **React**, is a popular JavaScript library for building user interfaces. It is also referred to as a front-end JavaScript library. It was developed by Facebook and is widely used for creating dynamic and interactive web applications.

1. **What is React?**

* **React** is a **JavaScript library** for building **user interfaces** (UIs) on the web.
* React is a declarative, component based library that allows developers to build reusable UI components and It follows the [Virtual DOM](https://www.geeksforgeeks.org/reactjs-virtual-dom/) (Document Object Model) approach, which optimizes rendering performance by minimizing DOM updates.
* React is **fast** and works well with other tools and libraries.

1. **Prerequisite of React :**

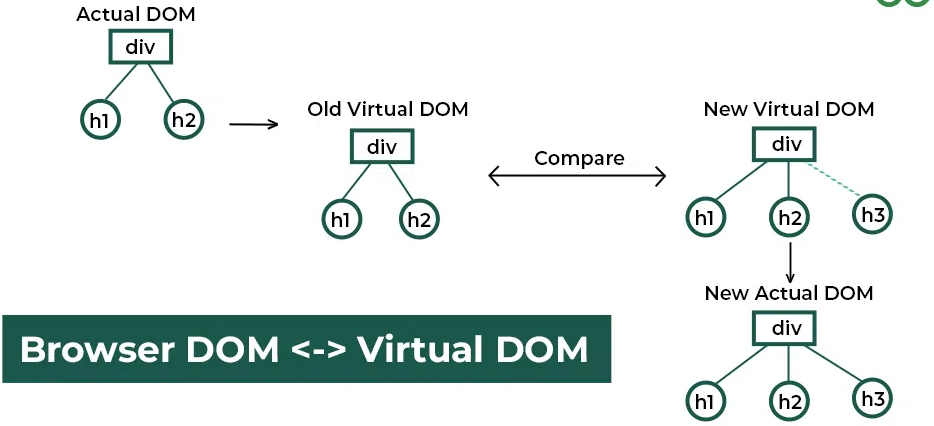
* [HTML](https://www.geeksforgeeks.org/html-tutorial/) and [CSS](https://www.geeksforgeeks.org/css-tutorial/)
* [JavaScript](https://www.geeksforgeeks.org/javascript/) and ES6
* [JSX (JavaScript XML)](https://www.geeksforgeeks.org/reactjs-jsx-introduction/) & Babel
* [Node](https://www.geeksforgeeks.org/nodejs/)+Npm
* [Git](https://www.geeksforgeeks.org/ultimate-guide-git-github/) and CLI (Command Line Interface).

1. **History of React:**

* React was invented by Facebook developers who found the traditional DOM slow.
* Designed by Jordan Walke and first initial release in May29 2013
* By implementing a virtual DOM, React addressed this issue and gained popularity rapidly.
* The library continues to evolve, introducing new features with each update.
* React was first deployed on Facebook's [News Feed](https://en.wikipedia.org/wiki/News_Feed) in 2011 and subsequently integrated into [Instagram](https://en.wikipedia.org/wiki/Instagram) in 2012.
* In May 2013, at JSConf US, the project was officially open-sourced, marking a significant turning point in its adoption and growth.

1. **How Does React Work?**

* React operates by creating an in-memory virtual DOM rather than directly manipulating the browser’s DOM.
* It performs necessary manipulations within this virtual representation before applying changes to the actual browser DOM.
* React is efficient, altering only what requires modification.



1. **Features of React**

### Component-Based Architecture: React provides the feature to break down the UI into smaller, self-contained components. Each component can have its own ****state and props****

### JSX (JavaScript Syntax Extension): JSX is a syntax extension for JavaScript that allows developers to write HTML-like code within their JavaScript files. It makes React components more readable and expressive.

### Virtual DOM: React maintains a lightweight representation of the actual DOM in memory. When changes occur, React efficiently updates only the necessary parts of the DOM.

### One-way Data Binding: [One-way data binding](https://www.geeksforgeeks.org/reactjs-data-binding/), the name itself says that it is a one-direction flow. The data in react flows only in one direction i.e. the data is transferred from top to bottom i.e. from parent components to child components.

### ****Performance:**** As we discussed earlier, react uses virtual DOM and updates only the modified parts. So , this makes the DOM to run faster. DOM executes in memory so we can create separate components which makes the DOM run faster.

### ****Components:**** React divides the web page into multiple [components](https://www.geeksforgeeks.org/reactjs-components/) as it is component-based. Each component is a part of the UI design which has its own logic.

### ****Single-Page Applications (SPAs):**** React is recommended in creating SPAs, allowing smooth content updates without page reloads. Its focus on reusable components makes it ideal for real-time applications.

### ReactJs Life Cycle

### Initialization: This is the stage where the component is constructed with the given Props and default state. This is done in the constructor of a Component Class.

### Mounting Phase:

* **Constructor**: The constructor method initializes the component. It’s where you set up initial state and bind event handlers.
* **render ():**This method returns the JSX representation of the component. It’s called during initial rendering and subsequent updates.
* **componentDidMount ():** After the component is inserted into the DOM, this method is invoked. Use it for side effects like data fetching or setting timers.

### 

### Updating Phase:

* **componentDidUpdate(prevProps, prevState)**: Called after the component updates due to new props or state changes. Handle side effects here.
* **shouldComponentUpdate(nextProps,nextState):** Determines if the component should re-render. Optimize performance by customizing this method.
* **render():** Again, the render() method reflects changes in state or props during updates.

### Unmounting Phase:

* **componentWillUnmount()**: Invoked just before the component is removed from the DOM. Clean up resources (e.g., event listeners, timers).

### What is JSX?

### JSX, which stands for JavaScript XML, is a syntax extension for JavaScript.

### ReactJS uses an XML or HTML-like syntax, which is then transformed into React Framework JavaScript calls.

### Essentially, JSX expands ES6 to allow HTML-like text to coexist with JavaScript React code.

### Although it is not mandatory to use JSX in ReactJS, it is highly recommended.

### What is ReactJSX?

### React JSX is a syntax extension of JavaScript for writing React Code in a simple way. Using JSX it is easier to create reusable UI components with fewer lines of code in a template-type language with the power of JavaScript.

### JSX creates an element in React that gets rendered in the UI. It is transformed into JavaScript functions by the compiler at runtime. Error handling and warnings become easier to handle when using JSX.

### Sy : Sample code is

### const ele = <h1>This is sample JSX</h1>;

### Why JSX?

* It is faster than normal JavaScript as it performs optimizations while translating to regular JavaScript.
* It makes it easier for us to create templates.
* Instead of separating the markup and logic in separate files, React uses *components* for this purpose. We will learn about components in detail in further articles.
* As JSX is an expression, we can use it inside of if statements and for loops, assign it to variables, accept it as arguments, or return it from functions.

### Expression in JSX.

### In React we are allowed to use normal JavaScript expressions with JSX.

### To embed any JavaScript expression in a piece of code written in JSX we will have to wrap that expression in curly braces {}.

### Ex :

### const example = "JSX"

### const ele = <div>This component uses {example} </div>

### Programm :

### // Filename - App.js

### import React from "react";

### const name = "Learner";

### const element = (

### <h1>

### Hello,

### {name}.Welcome to GeeksforGeeks.

### </h1>

### );

### ReactDOM.render(element, document.getElementById("root"));