

<b>Programme</b>	: Bachelor of Science
<b>Subject</b>	: Computer Science
<b>Semester</b>	: VI
<b>Course Code</b>	: CSC109
<b>Course Title</b>	: Full Stack Web Development
<b>Unit III</b>	: React
<b>Module Name</b>	: <b>React elements and JSX</b> : What is react? advantages and disadvantages, Overview of JSX.

---

## **Notes**

### **Introduction to React**

#### **What Is React?**

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It is an open-source, component-based front end library responsible only for the view layer of the application. It lets you compose complex UI's from small and isolated pieces of code called "components".

Today, most of the websites are built using MVC (model view controller) architecture. In MVC architecture, react is the 'V' which stands for view, whereas the architecture is provided by the Redux or Flux.

A ReactJS application is made up of multiple components, each component

responsible for outputting a small, reusable piece of HTML code. The components are the heart of all React applications. These Components can be nested with other components to allow complex applications to be built of simple building blocks.

ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time.

## **DOM and Virtual DOM**

### **DOM**

DOM stands for Document Object Model and is an abstraction of a structured text. For web developers, this text is an HTML code, and the DOM is simply called HTML DOM. Elements of HTML become nodes in the DOM. So, while HTML is a text, the DOM is an in-memory representation of this text. The HTML DOM provides an interface (API) to traverse and modify the nodes. It contains methods like `getElementById` or `removeChild`. Since we are more and more pushed towards dynamic web apps we need to modify the DOM tree incessantly and a lot and this is a real performance and development pain.

### **Virtual DOM**

The Virtual DOM is an abstraction of the HTML DOM. It is lightweight and detached from the browser-specific implementation details. Perhaps it's better to think of the virtual DOM as React's local and simplified copy of the HTML DOM. It allows React to do its computations within this abstract world and skip the "real" DOM operations, often slow and browser-specific.

## React “Hello World” Example

```
ReactDOM.render(  
  <h1>Hello, world!</h1>,  
  document.getElementById('root')  
>);
```

It displays a heading saying “Hello, world!” on the page.

## React Features

ReactJS gaining quick popularity as the best JavaScript framework among web developers. It is playing an essential role in the front-end ecosystem. The important features of ReactJS are as following:

### **1] JSX**

It stands for JavaScript XML. It is a JavaScript syntax extension. It's an XML or HTML like syntax used by ReactJS. This syntax is processed into JavaScript calls of React Framework.

### **2] Components**

ReactJS is all about components. ReactJS application is made up of multiple components, and each component has its own logic and controls. These components can be reusable which help you to maintain the code when working on larger scale projects.

### **3] Simplicity**

ReactJS uses JSX file which makes the application simple and to code as well as understand. ReactJS is a component-based approach which makes the code reusable as your need. This makes it simple to use and learn.

#### **4] One-way Data Binding**

ReactJS is designed in such a manner that follows unidirectional data flow or one-way data binding. The benefits of one-way data binding give you better control throughout the application. If the data flow is in another direction, then it requires additional features. Flux is a pattern that helps to keep your data unidirectional. This makes the application more flexible that leads to increase efficiency.

#### **5] Virtual DOM**

A virtual DOM object is a representation of the original DOM object. It works like a one-way data binding. Whenever any modifications happen in the web application, the entire UI is re-rendered in virtual DOM representation. Then it checks the difference between the previous DOM representation and new DOM. Once it has done, the real DOM will update only the things that have actually changed. This makes the application faster, and there is no wastage of memory.

### **Advantages and Disadvantages**

#### **Advantages**

##### **1. Easy to learn and use**

ReactJS is much easier to learn and use. It comes with a good supply of documentation, tutorials, and training resources.

##### **2. Creating Dynamic Web Applications Becomes Easier**

It provides less coding and gives more functionality. It makes use of the JSX (JavaScript Extension), which is a particular syntax letting HTML quotes and HTML tag syntax to render particular subcomponents.

### **3. Reusable Components**

A ReactJS web application is made up of multiple components, and each component has its own logic and controls. These components are responsible for outputting a small, reusable piece of HTML code which can be reused wherever you need them.

### **4. Performance Enhancement**

ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time.

### **5. The Support of Handy Tools**

ReactJS has also gained popularity due to the presence of a handy set of tools. These tools make the task of the developers understandable and easier.

### **6. The Benefit of Having JavaScript Library**

It offers a very rich JavaScript library. The JavaScript library provides more flexibility to the web developers to choose the way they want.

## **Disadvantages**

### **1. View Part**

ReactJS Covers only the UI Layers of the app and nothing else. So you still need to choose some other technologies to get a complete tooling set for development in the project.

### **2. Learning curve**

Being not full-featured framework it is required in-depth knowledge for integration user interface free library into MVC framework.

3. Lots of developers' dislike JSX React's documentation, **manuals are difficult** for newcomers' understanding.

4. React's **large size library**.

## **JSX**

### **What is JSX?**

JSX stands for JavaScript XML. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React. JSX allows us to write HTML elements in JavaScript and place them in the DOM without any createElement() and/or appendChild() methods. JSX converts HTML tags into react elements.

### **Why JSX?**

1. React embraces the fact that rendering logic is inherently coupled with other UI logic: how events are handled, how the state changes over time, and how the data is prepared for display.
2. Instead of artificially separating technologies by putting markup and logic in separate files, React separates concerns with loosely coupled units called “components” that contain both.
3. React doesn't require using JSX, but most people find it helpful as a visual aid when working with UI inside the JavaScript code. It also allows React to show more useful error and warning messages.

Consider this variable declaration:

```
const element = <h1>Hello, world!</h1>;
```

This tag syntax is neither a string nor HTML. It is called JSX, and it is a syntax extension to JavaScript. Use JSX with React to describe what the UI should look like. JSX may remind you of a template language, but it comes with the full power of JavaScript. JSX produces React “elements”.

### Example:

```
import React from 'react';

import ReactDOM from 'react-dom';

const myelement = <h1>This is JSX!</h1>;

ReactDOM.render(myelement, document.getElementById('root'));
```

### Explanation of the above example:

Import React from ‘react’ is needed even though there is no reference to the word ‘React’ in the above code, this is because during the build process, each JSX element is transformed into React.createElement call by the Babel compiler.