**React**

**Ce este ReactJS?**

ReactJS este o librarie JavaScript, care permite programatorilor cu experienta, precum ai nostri, sa creeze o Interfata (UI) intr-un timp mult mai scurt. Mai putin timp folosit pentru dezvoltare inseamna mai putini bani investiti. Mai mult decat atat, lucrand cu Roweb, vei primi o [echipa dedicata](https://www.roweb.ro/ro/servicii/echipa-dedicata) care se va ocupa strict de proiectul tau.  
Poti folosi ReactJS pentru a dezvolta interfata aplicatiei atat intr-o aplicatie de tip Web cat si intr-o aplicatie de tip Mobile. Datorita abilitatii librariei de a refolosi componente din interfata, poti scuti si mai mult timp pentru dezvoltarea aplicatiilor dorite. Folosind ReactJS pentru o aplicatie Web sau Mobile ce necesita o actualizare dinamica a paginii prezinta urmatoarele avantaje:

**React** (cunoscută și sub numele de React.js sau ReactJS) este o bibliotecă [JavaScript](https://ro.wikipedia.org/wiki/JavaScript) [open-source](https://ro.wikipedia.org/wiki/Open-source)[[6]](https://ro.wikipedia.org/wiki/React.js#cite_note-react-6) pentru construirea de interfețe de utilizator. Este întreținută de [Facebook](https://ro.wikipedia.org/wiki/Facebook) și de o comunitate de dezvoltatori și companii individuale.[[7]](https://ro.wikipedia.org/wiki/React.js" \l "cite_note-7)[[8]](https://ro.wikipedia.org/wiki/React.js#cite_note-8)[[9]](https://ro.wikipedia.org/wiki/React.js#cite_note-9) React poate fi folosită ca bază pentru dezvoltarea aplicațiilor mobile cu o singură pagină sau mobile.

De ce folosim ReactJS?

ReactJS offers graceful solutions to some of front-end programming’s most persistent issues, allowing you to build dynamic and interactive web apps with ease. It’s fast, scalable, flexible, powerful, and has a robust developer community that’s rapidly growing. There’s never been a better time to learn React.

eact’s popularity today has eclipsed that of all other front-end development frameworks. Here is why:

* Easy creation of dynamic applications: React makes it easier to create dynamic web applications because it requires less coding and offers more functionality, as opposed to JavaScript, where coding often gets complex very quickly.
* Improved performance: React uses Virtual DOM, thereby creating web applications faster. Virtual DOM compares the components’ previous states and updates only the items in the Real DOM that were changed, instead of updating all of the components again, as conventional web applications do.
* Reusable components: Components are the building blocks of any React application, and a single app usually consists of multiple components. These components have their logic and controls, and they can be reused throughout the application, which in turn dramatically reduces the application’s development time.
* Unidirectional data flow: React follows a unidirectional data flow. This means that when designing a React app, developers often nest child components within parent components. Since the data flows in a single direction, it becomes easier to debug errors and know where a problem occurs in an application at the moment in question.
* Small learning curve: React is easy to learn, as it mostly combines basic HTML and JavaScript concepts with some beneficial additions. Still, as is the case with other tools and frameworks, you have to spend some time to get a proper understanding of React’s library.
* It can be used for the development of both web and mobile apps: We already know that React is used for the development of web applications, but that’s not all it can do. There is a framework called React Native, derived from React itself, that is hugely popular and is used for creating beautiful mobile applications. So, in reality, React can be used for making both web and mobile applications.
* Dedicated tools for easy debugging: Facebook has released a Chrome extension that can be used to debug React applications. This makes the process of debugging React web applications faster and easier

**Caracteristici generale**

**JSX**

JSX is a syntax extension to JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code. This makes the code easier to understand and debug, as it avoids the usage of complex JavaScript DOM structures.

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.

Virtual DOM

React keeps a lightweight representation of the “real” DOM in the memory, and that is known as the “virtual” DOM (VDOM). Manipulating real DOM is much slower than manipulating VDOM because nothing gets drawn on the screen. When the state of an object changes, VDOM changes only that object in the real DOM instead of updating all of the objects.

It may all seem a bit overwhelming for now, so let’s first understand what DOM is, and then we’ll go through how VDOM and real DOM interact with each other.

* What is the Document Object Model (DOM)?

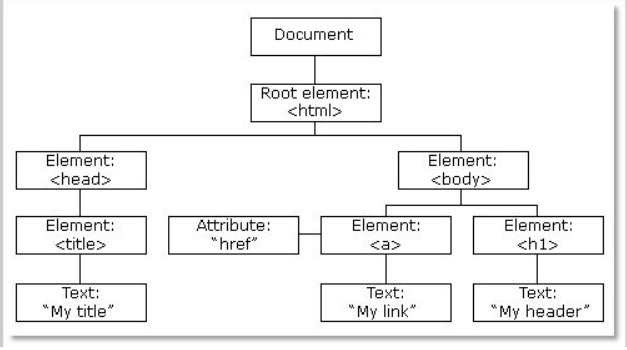


Fig: DOM of a Webpage

DOM (Document Object Model) treats an XML or HTML document as a tree structure in which each node is an object representing a part of the document.

When the state of an object changes in a React application, VDOM gets updated. It then compares its previous state and then updates only those objects in the real DOM instead of updating all of the objects. This makes things move fast, especially when compared to other front-end technologies that have to update each object even if only a single object changes in the web application.

### Performance

React uses VDOM, which makes the web applications run much faster than those developed with alternate front-end frameworks. React breaks a complex user interface into individual components, allowing multiple users to work on each component simultaneously, thereby speeding up the development time.

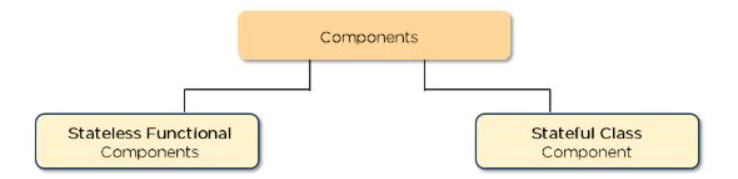
### One-way Data Binding

React’s one-way data binding keeps everything modular and fast. A unidirectional data flow means that when a developer designs a React app, they often nest child components within parent components. This way, a developer knows where and when an error occurs, giving them better control of the whole web **application.**

### Components

Components are the building blocks of any React application, and a single app usually consists of multiple components. A component is essentially a piece of the user interface. React splits the UI into independent, reusable parts that can be processed separately.

There are two types of components in React:

****

* Functional Components: These components have no state of their own and only contain a render method, so they are also called stateless components. They may derive data from other components as props (properties).
* Class Components: These components can hold and manage their state and have a separate render method for returning JSX on the screen. They are also called stateful components, as they can have a state.

### State

The state is a built-in React object that is used to contain data or information about the component. A component’s state can change over time; whenever it changes, the component re-renders. The change in state can happen as a response to user action or system-generated events, and these changes determine the behavior of the component and how it will render.

### Props

Props are short for properties. It is a React built-in object which stores the value of a tag’s attributes and works similar to the HTML attributes. It provides a way to pass data from one component to other components in the same way as arguments are passed in a function.

**https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs?source=sl\_frs\_nav\_playlist\_video\_clicked**