

Documentation of react.js

we'll take a quick dive into React, a popular JavaScript library for building user interfaces. Whether you're sipping your morning coffee or taking a quick break, this guide aims to equip you with a basic understanding of React's fundamentals.

What is React?

Developed by Facebook in 2011, React has become a cornerstone in web development. It's known for its efficiency and flexibility in creating user interfaces. At its core, React breaks down complex UIs into small, reusable components. These components are the building blocks of React applications, making them easier to manage and maintain.

React Components

Components are the heart of React. They can be either functional or class-based. Functional components are simple JavaScript functions that return JSX (JavaScript XML), while class components are ES6 classes that extend `React.Component` and have a `render` method.

Jsx

```
// Functional Component
```

```
function Name() {  
  return <h2>Hi, my name is Joe!</h2>;  
}
```

```
// Class Component
```

```
class Person extends React.Component {  
  render() {  
    return <h2>Hi again from Joe!</h2>;  
  }  
}
```

JSX

JSX allows you to write HTML-like syntax within JavaScript. It's a powerful feature of React that makes code more readable and intuitive. JSX can be used to compose complex UIs with ease.

State and Props

State allows React components to manage their own data. It represents the current state of a component and can be updated over time. Props, short for properties, are used to pass data from parent to child components. They are immutable and help in creating dynamic and reusable components.

Component Lifecycle

React components have a lifecycle consisting of various phases such as mounting, updating, and unmounting. Lifecycle methods provide opportunities to perform actions at specific points in a component's lifecycle. For example, `componentDidMount` is called after a component is rendered for the first time.

Why Choose React Over Vanilla JavaScript?

React simplifies complex UI development by breaking it down into reusable components. Unlike vanilla JavaScript, React handles both display and functionality in a more organized manner. It employs a virtual DOM for efficient updates, making it ideal for building large-scale applications.

What Can You Build with React?

React powers some of the most popular web applications, including Facebook, Instagram, Netflix, and WhatsApp. As a beginner, you can start with simple projects like todo lists, weather apps, or even a basic calculator. The possibilities are endless!

In conclusion, React provides a straightforward approach to building modern web applications. With its component-based architecture and JSX syntax, React simplifies UI development and offers a seamless developer experience. So grab your coffee and start exploring the world of React!

Table of Contents

[What Is React?](#)[ReactJS History](#)[Why React?](#)[ReactJS Keys](#)[ReactJS Advantages](#)[View More](#)

The world can't live without mobile and web applications in this day and age. Everything is digitized, from booking cabs to ordering food to make bank transactions. Thanks to the efficient frameworks that provide a seamless user experience. One such robust frontend library is React. This tutorial on 'what is React' will help you understand the library's fundamentals and work with a simple demo.

What Is React?

React is a framework that employs Webpack to automatically compile React, JSX, and ES6 code while handling CSS file prefixes. React is a [JavaScript](#)-based UI development library. Although React is a library rather than a language, it is widely used in web development. The library first appeared in May 2013 and is now one of the most commonly used frontend libraries for web development.

React offers various extensions for entire application architectural support, such as Flux and React Native, beyond mere UI.

ReactJS History

When compared to other technologies on the market, React is a new technology. Jordan Walke, a software engineer at Facebook, founded the library in 2011, giving it life. The likes of XHP, a straightforward HTML component framework for PHP, have an influence on React. React's newsfeed was its debut application in 2011. Later, Instagram picks it up and incorporates it into their platform.

Why React?

React'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.

The above reasons more than justify the popularity of the React library and why it is being adopted by a large number of organizations and businesses. Now let's familiarize ourselves with React's features.

