

Introduction to React.js

React.js is a powerful JavaScript library used to create interactive and dynamic user interfaces. It enables developers to build complex applications with ease and is a popular choice for both beginners and seasoned web developers.



by Curious Coder (CodeWithCurious.com)

Key Features of React.js

Virtual DOM

React uses a Virtual DOM, a lightweight replica of the browser's DOM, which enables fast and efficient rendering.

Reusable Components

Components in React enable developers to create reusable interfaces, which can be easily modified and reused throughout the application.

Unidirectional Data flow

React features a unidirectional data flow, which makes it easier to debug and understand the application's data flow.

JSX Syntax

React uses JSX - JavaScript syntax extension, which enables the creation of component hierarchies in a readable and streamlined way.

How React.js Compares to Other Frameworks



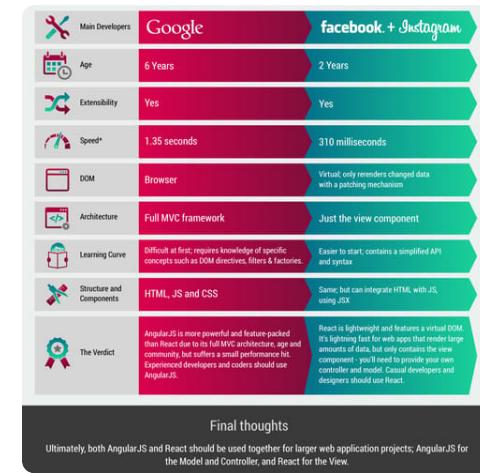
React.js vs Angular

While both frameworks are popular, Angular is based on a MVC pattern while React.js uses an MVVM pattern. React tends to be more flexible and easier to learn thanks to its component-based approach.

	ANGULAR	REACT
Lead	Miško Hevery	Jordan Walke
Release Date	May 2016	March 2013
Script	TypeScript	JavaScript
Complexity	high	average
Implementation	very complex	complex

React.js vs Vue.js

Both frameworks are popular but Vue.js is considered easier to learn and use, making it a good choice for small to medium-sized applications. However, React is more popular and is often preferred for large-scale applications.



React.js vs jQuery

React.js is a full-fledged front-end library, whereas jQuery is a JavaScript library that focuses on DOM manipulation and event handling. React.js is ideal for building complex single-page applications, while jQuery is better suited for smaller websites.

Using Components in React.js

1 Class Components

These components are JavaScript Classes that extends React Component and have their own state and lifecycle methods.

2 Functional Components

These components are simpler and more lightweight since they don't have state and lifecycle methods. They only take in props as parameters and render the component.



React.js Lifecycle Methods

The render method is called each time a component is updated, it's the main method that uses JSX to create your components.

render()



constructor()

This is the first method that is called when a component is created, and it's used for initializing state and binding event listeners.

componentDidMount()

This method is called after the component is mounted and can be used to execute AJAX calls, modifications to state or any other setup that requires changes to the DOM.

How to Get Started with React.js

Step 1

Install Node.js, which comes with npm, the package manager for JavaScript libraries such as React.js.

Step 2

Create a new project using `create-react-app`, the official command-line tool for creating React.js applications.

Step 3

Start building your application, by creating components, handling state, and adding other libraries or resources if needed.

Conclusion

React.js is a powerful and popular front-end library that enables developers to build complex web applications with ease. Its component-based architecture, unidirectional data flow, and JSX syntax make it a top choice for both beginners and seasoned developers.