

Mastering ReactJS: From Beginner to Advanced Lecture 1: Introduction to ReactJS

Overview ReactJS is a popular JavaScript library used to build interactive and dynamic user interfaces for web applications. Developed by Facebook, React simplifies the process of creating reusable UI components and efficiently updating the DOM with minimal effort.

In this lecture, you will learn: - What ReactJS is and why it's widely used. - How React differs from other JavaScript frameworks. - The benefits of component-based architecture.

Core Concepts 1. Components - Components are the building blocks of a React application. - They can be functional or class-based. - Components allow for reusability and modular development.

1. JSX (JavaScript XML)
2. JSX is a syntax extension for JavaScript that looks similar to HTML.
3. It allows you to write UI elements directly in JavaScript.
4. Virtual DOM
5. React uses a virtual DOM to optimize rendering.
6. It updates only the parts of the DOM that have changed, improving performance.

Setup To start building React applications, you need to set up your development environment:

1. Install Node.js and npm
2. Node.js is required to run JavaScript outside the browser.
3. npm (Node Package Manager) is used to manage project dependencies.
4. Create a React Project
5. Using Create React App:

```
npx create-react-app my-app  
cd my-app  
npm start
```

6. Using Vite:

```
npm create vite@latest my-app  
cd my-app  
npm install  
npm run dev
```

7. Understand Folder Structure

8. src/ – Contains all the React components.

9. public/ – Contains static files like index.html.

10. package.json – Contains project dependencies.

Your First Component 1. Create a functional component

```
import React from 'react';

const Welcome = () => {
  return <h1>Welcome to ReactJS!</h1>;
};

export default Welcome;
```

1. Render the component

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import Welcome from './Welcome';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Welcome />);
```

Key Takeaways - React uses components to build reusable UI elements. - JSX allows writing HTML-like code in JavaScript. - The virtual DOM efficiently updates only changed elements. - Your development environment is ready to build React apps.

Next Steps - Lecture 2: Working with Props and State – Learn how to pass data to components and manage dynamic content.