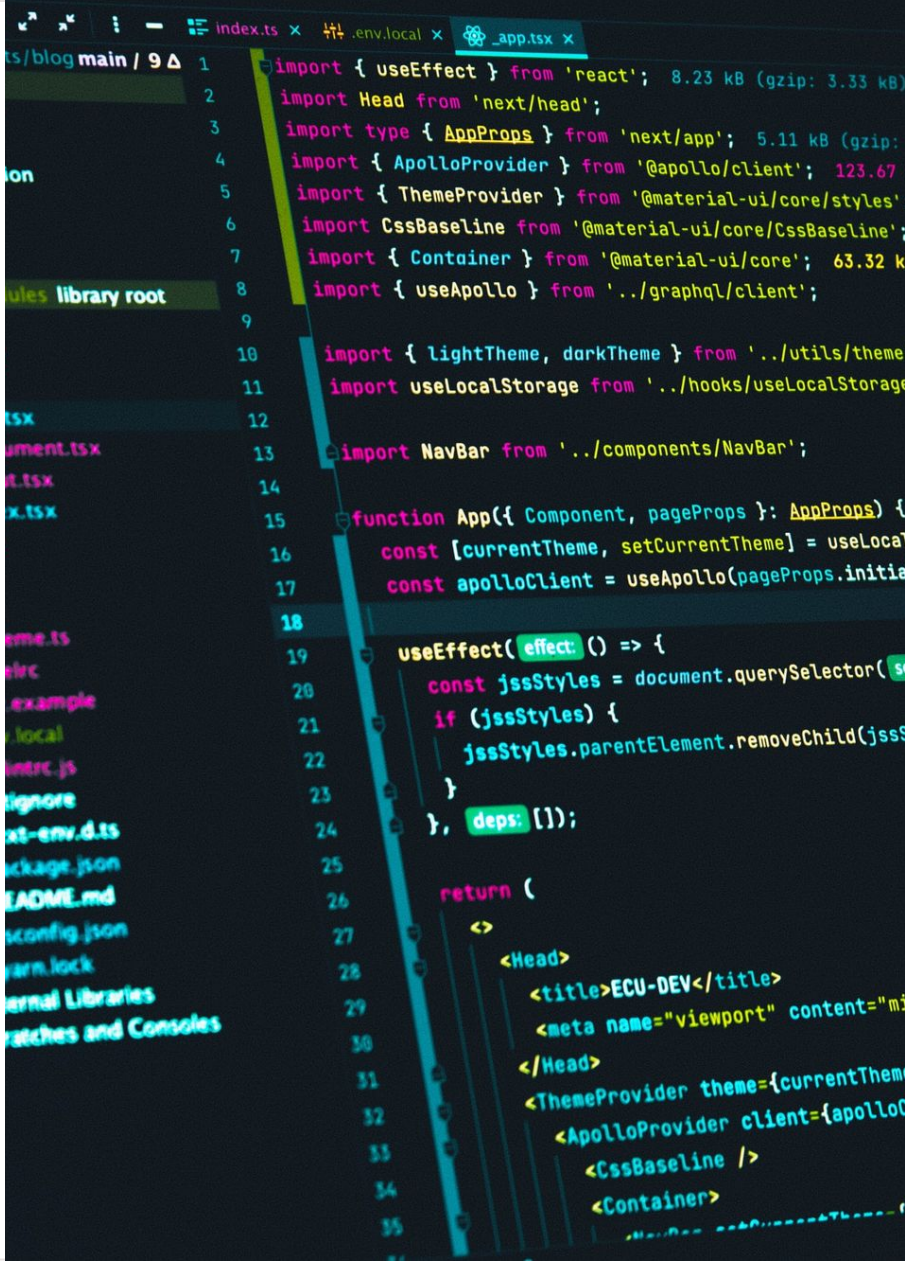
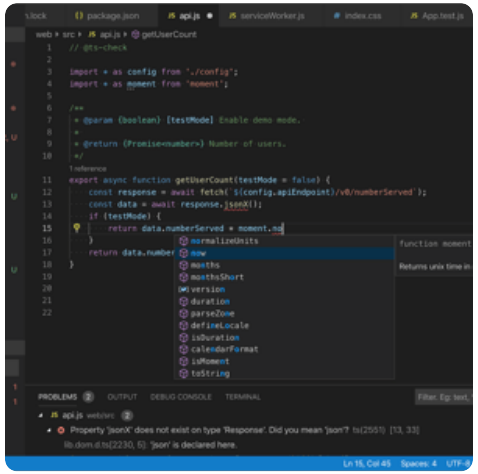
JSX and Components in React

In this presentation, we'll explore the world of JSX and Components in React and learn how they can help us build powerful web applications.

by Curious Coder (CodeWithCurious.com)



What is JSX?

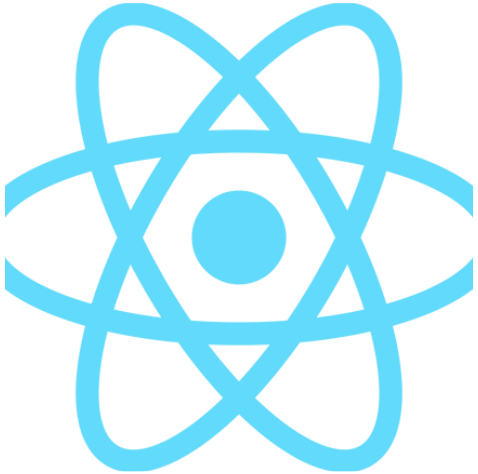


Javascript Syntax

JSX is a syntax extension for JavaScript that allows us to write HTML-like code that can be transformed into JavaScript.

JSX Syntax Example

With JSX, we can write code that looks like HTML, and render it using React.



JSX + React

JSX is a key feature of React, and is used to build reusable components and UI elements.

Benefits of Using JSX

1 Improved Code Readability

JSX helps us write code that's easier to read and understand, especially when working with nested components.

2 Faster Development

By allowing us to write HTML-like code, JSX makes it faster to build and prototype UI components.

3 Better Performance

Since JSX helps us avoid unnecessary DOM updates, it can lead to improved performance in our applications.

Introduction to Components

What are Components?

Components are reusable UI elements that can be composed together to build complex user interfaces.

Why Use Components?

By breaking our UI down into smaller components, we can make it easier to reason about and maintain our code.

Types of Components

There are two types of components in React: Class Components and Functional Components.

Class Components vs. Functional Components

Functional Components

Functional components are defined using JavaScript functions, and are simpler and easier to test than class components.

1

Class Components

Class components are defined using ES6 Classes, and can hold state and lifecycle methods.

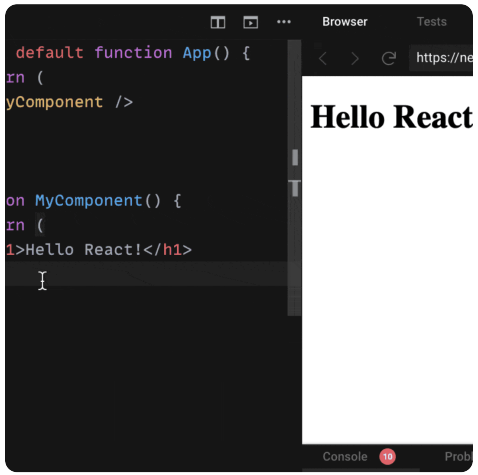
2

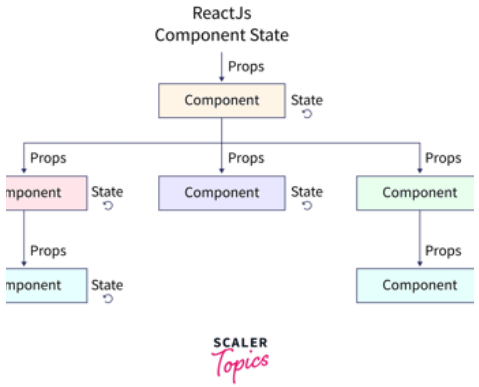
3

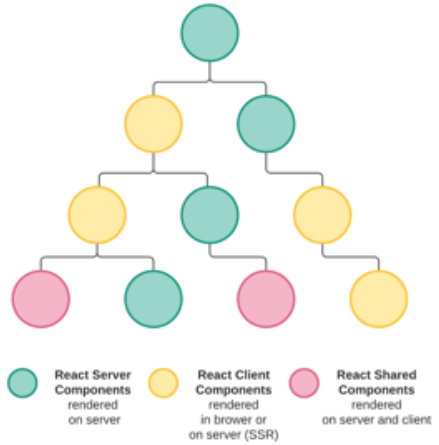
Choosing the Right Component Type

When deciding between class and function components, consider the complexity of your UI and whether you need to manage state.

Props and State in Components







Props

Props are read-only values that are passed from a parent component to a child component, and can be used to customize component behaviour.

State

State is an internal value that can be updated by a component, and is used to manage user interface interactions and component behaviour.

Props vs. State

Props and state are both important concepts in React, and understanding how they work is essential for building robust and maintainable components.

Rendering Components

Creating Components

To create a new component, define a function or class that returns some JSX.

Rendering Components

To use a component, render it in another component or in the main application.

Composing Components

By combining our components together, we can build complex user interfaces made up of simple, reusable building blocks.

Conclusion and Resources for Learning More

1 What We've Learned

Today, we've learned what JSX is, the benefits of using it, and how to create and use components in React.

2 Next Steps

If you want to learn more about React components and JSX, check out the official React documentation or some online tutorials.

3 Resources

React Documentation, FreeCodeCamp, Udemy, PluralSight and Codecademy are some platforms for React JS learning.