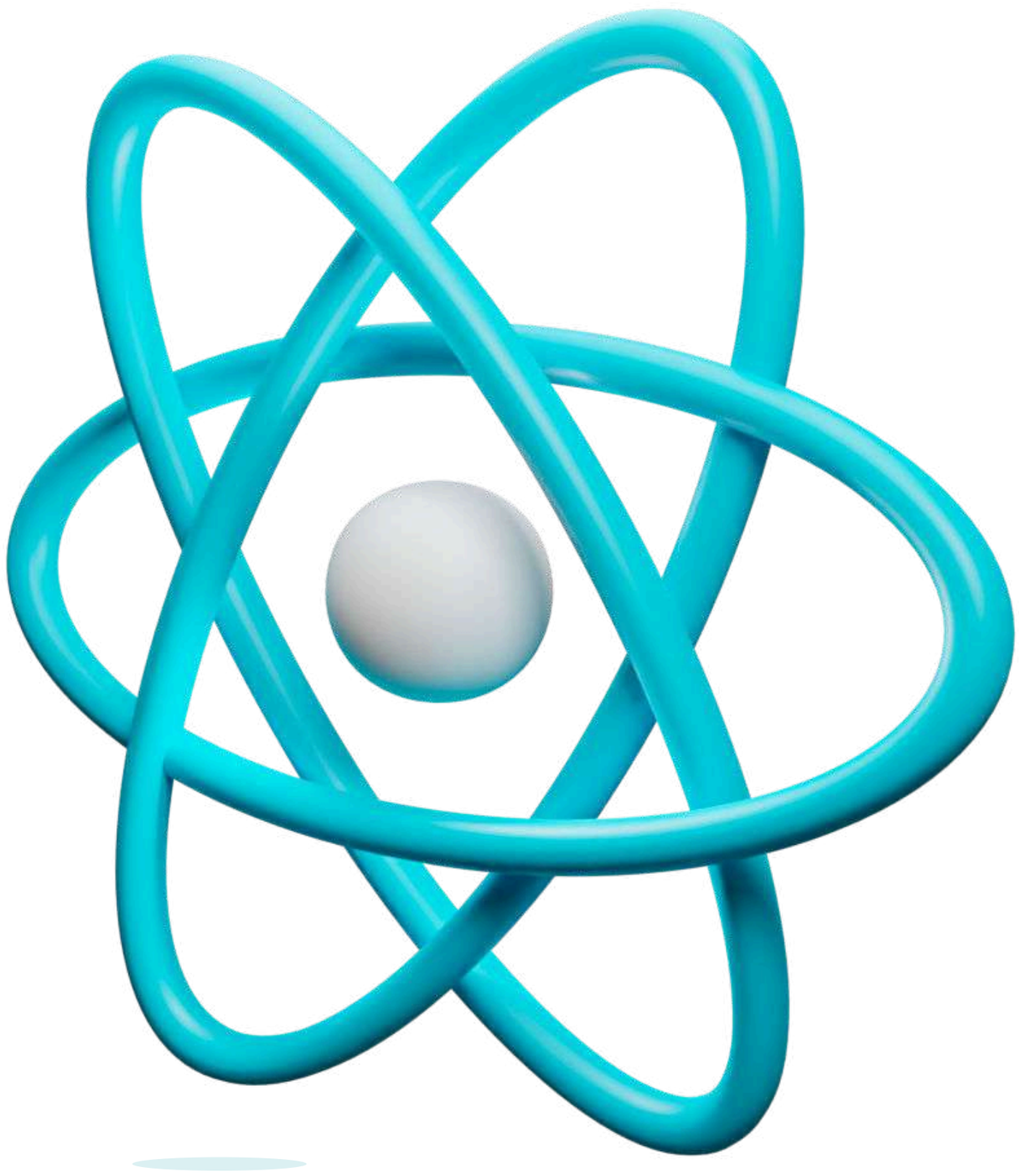
BOSSCODER ACADEMY

MASTER

REACT

IN 20 DAYS



1 DAY

Introduction to React

📝 Topic

What is React.js?

🗃️ Resources

- https://react.dev/learn
- https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs

🗃️ Questions

1. Explain the key features of React.
2. What problem does React solve in web development?
3. Is React a JavaScript library or a framework?
4. How does React make it easier to build user interfaces?

2 DAY

Setting Up React

📝 Topic

Setting up a React.js development environment.

🗃️ Resources

- https://reactnative.dev/docs/environment-setup
- https://www.freecodecamp.org/news/react-beginner-handbook/
- React Course - Beginner's Tutorial for React JavaScript Library [2022]

🗃️ Questions

1. List the steps to set up a basic React development environment.
2. What are the key files and folders generated when you create a new React application?
3. How do you start the development server and view your React application in a web browser?
4. Are there any specific code editors or Integrated Development Environments (IDEs) commonly used for React development?
5. What is Babel, and why is it used in React development?

3 DAY

React JSX

📝 Topic

Understanding JSX (JavaScript XML) in React.

🗃️ Resources

- https://react.dev/learn/writing-markup-with-jsx
- https://react.dev/learn/javascript-in-jsx-with-curly-braces
- https://www.w3schools.com/react/react_jsx.asp

🗃️ Questions

1. Write JSX code to render a simple element.
2. What is the key difference between JSX and HTML when it comes to element attributes?
3. Write JSX code to render a <h1> element with the text "Hello, React!".
4. Explain how you can apply CSS classes to JSX elements.
5. Add conditional rendering to a JSX component, so it displays one message if a variable is true and another message if it's false.

4-5 DAY

React Components

📝 Topic

Introduction to React components.

📂 Resources

- https://www.w3schools.com/react/react_components.asp
- https://www.geeksforgeeks.org/reactjs-components/
- https://react.dev/learn/your-first-component
- https://react.dev/learn/passing-props-to-a-component

🗃️ Questions

1. What is a React component, and why is it important?
2. Differentiate between functional and class components in React.
3. Create a functional React component called Greeting that renders a <h1> element with a greeting message of your choice.
4. Develop a class-based React component called Counter that displays a number and has two buttons, one for incrementing the number and one for decrementing it.
5. Implement a functional component called "Person" that receives a "name" prop and displays "Hello, [name]."

5-6 DAY

State and Props in React

📝 Topic

Learning about state and props in React.

🗃️ Resources

- https://www.geeksforgeeks.org/reactjs-state-vs-props/
- https://legacy.reactjs.org/docs/faq-state.html
- https://www.freecodecamp.org/news/react-js-for-beginners-props-state-explained/
- https://react.dev/learn/state-a-components-memory

🗃️ Questions

1. Explain the difference between state and props in React.
2. How can you pass data from a parent component to a child component in React? Is it done through props or state?
3. Describe a scenario in which you would use state to manage data within a React component. Provide an example use case.
4. What is the scope of state and props in a React component? Are they accessible throughout the component or limited to certain parts?
5. What happens when you modify the state of a React component? How does React handle re-rendering?

7 DAY

Lists and Keys in React

📝 Topic

Understanding lists and keys in React.

📥 Resources

- https://react.dev/learn/rendering-lists
- https://www.w3schools.com/react/react_lists.asp
- https://www.geeksforgeeks.org/reactjs-lists/

🗃️ Questions

1. Construct a component called "ToDoList" that takes an array of tasks as a prop and renders an unordered list of tasks. Ensure each list item has a unique key.
2. Explain the importance of keys when rendering lists in React.
3. Create a React component called ShoppingList that receives an array of shopping items as a prop and renders an ordered list (<ol>) with each item as a list item (<li>).
4. Build a functional React component named CommentList that accepts an array of comment objects (each with a unique ID) as a prop and displays a list of comments, including the user's name and comment text.

8 DAY

Events and Life Cycle Events in React

Topic

Events and Life Cycle Events, Responding to events in React

Resources

- https://react.dev/learn/responding-to-events
- https://www.w3schools.com/react/react_events.asp
- https://www.javatpoint.com/react-events
- https://react.dev/learn/synchronizing-with-effects

🗃️ Questions

1. What are React events, and how are they similar or different from standard HTML events?
2. Describe the concept of the React event handling system and its advantages in building interactive web applications.
3. Explain the React component lifecycle. What are the major phases, and what happens during each phase?
4. Create a React component that renders a button. Implement an event handler that logs a message to the console when the button is clicked.
5. Develop a React component that renders a list of items. Implement an event handler that allows users to remove items from the list when a "Delete" button is clicked.

9-10 DAY

Form Handling in React

📝 Topic

Form Handling and Validation in React

📥 Resources

- https://www.w3schools.com/react/react_forms.asp
- https://legacy.reactjs.org/docs/forms.html
- https://www.freecodecamp.org/news/how-to-validate-forms-in-react/

🗃️ Questions

1. Create a form with a checkbox input for "subscribe to newsletter." Implement a controlled component for the checkbox and display a confirmation message when the checkbox is checked.
2. Implement form validation for a phone number input field. Ensure that the phone number entered is in a valid format (e.g., xxx-xxx-xxxx) and display an error message if it's not.
3. Implement real-time validation for an email input field. Display a green checkmark icon next to the input when the entered email is valid and a red X icon when it's invalid.

11 DAY

Context API in React

📝 Topic

Context API, Server Side Rendering

📂 Resources

- https://react.dev/learn/passing-data-deeply-with-context
- https://react.dev/reference/react/useContext
- https://www.freecodecamp.org/news/server-side-rendering-your-react-app-in-three-simple-steps-7a82b95db82e/

🗃️ Questions

1. Create a simple React application that uses the Context API to manage the theme of the application (e.g., light and dark mode). Allow users to toggle between the two themes using a button. Use the useContext hook to access and apply the theme context.
2. Explain the role of the useContext hook in React. How does it enable components to access values from the context?
3. What is the purpose of the Context API in React, and in what situations would you use it? Provide an example scenario where using context makes sense.
4. What are the key steps involved in implementing Server-Side Rendering (SSR) in a React application? Briefly outline the process.

12 DAY

Custom Hooks in React

📝 Topic

Custom Hooks

🗃️ Resources

- https://react.dev/learn/reusing-logic-with-custom-hooks
- https://www.w3schools.com/react/react_customhooks.asp
- https://www.freecodecamp.org/news/how-to-create-react-hooks/

🗃️ Questions

1. What is a custom hook in React, and why would you use one in your application?
2. Explain the difference between custom hooks and built-in hooks in React. When would you choose to create a custom hook instead of using a built-in one?
3. Can custom hooks have dependencies on other hooks or external libraries? How can you manage these dependencies effectively when creating custom hooks?
4. Build a custom hook called useKeyPress that listens for a specific key press (e.g., the "Enter" key) and returns a boolean indicating whether that key is currently pressed. Use this hook in a component to trigger an action when the user presses the specified key.

13-14 DAY

Styling in React

📝 Topic

CSS in React, SASS in React, Styled Components

🗃️ Resources

- https://www.w3schools.com/react/react_css.asp
- https://create-react-app.dev/docs/adding-a-stylesheet
- https://create-react-app.dev/docs/adding-a-sass-stylesheet
- https://www.freecodecamp.org/news/style-react-apps-with-css/

🗃️ Questions

1. Explain the concept of CSS Modules in React. How do they help prevent class name conflicts and improve encapsulation of styles?
2. Provide an example of how to create and use a Styled Component in a React application. Describe the benefits of using Styled Components for styling.
3. Build a React component that uses Styled Components to style a custom button. Create variations of the button (e.g., primary, secondary) using props and conditional styling.
4. Implement a dark mode toggle in a React application using Styled Components. Create a styled component for a toggle switch, and use it to switch between light and dark themes.

15 DAY

React Router

📝 Topic

React Router

🗃️ Resources

- https://reactrouter.com/en/main/start/tutorial
- https://www.geeksforgeeks.org/reactjs-router/
- https://www.w3schools.com/react/react_router.asp

🗃️ Questions

1. How can you implement a "404 Not Found" page in a React Router application to handle routes that do not match any defined routes?
2. What is the purpose of the history object in React Router, and how can you programmatically navigate between routes using it?
3. Explain the concept of nested routes in React Router. When and why might you use them in your application?

16 DAY

Deploying a React Application

📝 Topic

Deploying a React Application

🗃️ Resources

- https://create-react-app.dev/docs/deployment
- https://create-react-app.dev/docs/proxying-api-requests-in-development
- https://www.netlify.com/with/react/

🗃️ Questions

1. Walk through the steps to create a production build of a React application using Create React App (CRA). Discuss how this build differs from a development build.
2. Choose a hosting platform (e.g., Netlify, Vercel, or GitHub Pages) and deploy a simple React application to it. Describe the steps you took and any configuration required.

DAY 17- 18

Work on a project!

Now that you have a solid understanding of React concepts and have explored various app ideas, it's time to put your knowledge into action by working on your React project.

Ideas of Apps you can work on:

1. To-Do List/Notes App
2. Weather App
3. Recipe Finder
4. Expense Tracker

17 DAY

Project Setup and Planning

1. Set up your project environment. If you're using Create React App or a similar tool, initialize your project.
2. Plan your project's structure and components. Sketch out a rough design or wireframe for your app.
3. Create a GitHub repository to version control your project.

18 DAY

Building Components and State Management

1. Build the foundational components of your app based on your project plan.
2. Implement component logic, including state management using React's state and props as appropriate.
3. Set up any forms or user input components required for your app.

19 DAY

Implement Features and Functionality

- Implement the core features and functionality of your app. This may involve fetching data from APIs, handling user interactions, and managing application state.
- Test your app's features as you go, fixing any bugs or issues that arise.

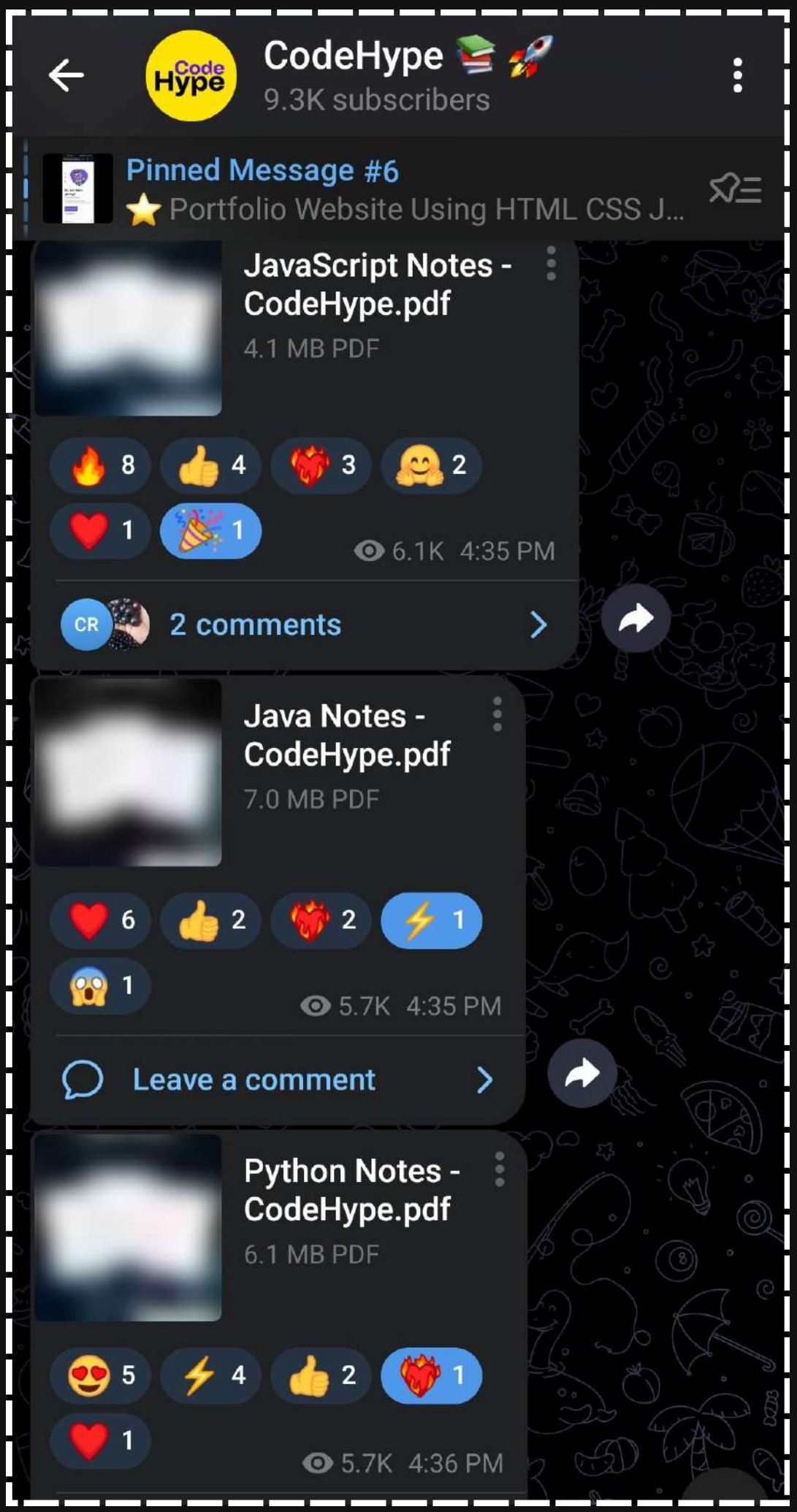
20 DAY

Styling, Testing, and Deployment

- Apply CSS or styling frameworks to make your app visually appealing and user-friendly.
- Conduct thorough testing and debugging to ensure your app is robust and functional.
- Once you're confident in your project's functionality and design, deploy it to a hosting platform of your choice. Document the deployment process for future reference.

Follow for more useful content

Telegram @codehype

Join Telegram for FREE Coding Resources

CodeHype