https://www.youtube.com/playlist?list=PLFNbBlDoHwn7H43FCfZqpgr1xgRmmYkkq&jct=8PLeX-pEa7WMvrQjSjLezg

React Website Script

Thomas Brodeur, Meier Rexius, Joey Tursi, and Grace Waida

---

1. Introduction

Speaker 1:
"Hi everyone! Today, we're giving a crash course on React. React is a JavaScript library for building websites. It's fast, flexible, and used by top companies like Facebook and Instagram.

By the end, you'll:

1. Know the basics of React: components, state, and the Virtual Document Object Model or Virtual DOM which is a programming interface that allows us to manipulate HTML elements with javascript.
2. See how React ties into programming ideas like states, types, and syntax.
3. Learn why React is great for making websites.

Let's get started!"

Speaker 2:
"React simplifies web development. You don't have to tell it every little step to update the site. Instead, you tell React what the site should look like, and React figures it out."

---

2. React Fundamentals

Speaker 3:
"Let's start with the three main ideas of React:

1. **React is Declarative**:
   - Declarative means you tell React what the site should look like in different cases, and it handles the changes for you.
   - For example, when a user clicks a button, React knows what to show next — you don't write all the steps.
2. **React is Component-Based**:
   - Components are small parts of your site, like buttons, forms, or menus.

   - They help keep your code neat and let you reuse parts of your site."

Speaker 4:
"Think of a component like a puzzle piece. Each piece works on its own, and you can snap them together to build the whole site."

Speaker 1:
"The last big idea is the **Virtual DOM**:

3. **Virtual DOM**:
   - The Virtual DOM is a quick copy of the browser's real DOM.
   - When something changes, React compares the old Virtual DOM with the new one and updates only the parts that need it. This makes React fast."

Speaker 2:
"React saves time by only fixing the parts that change. It doesn't mess with parts that stay the same."

---

3. React Lifecycle

Speaker 3:
"React apps follow a lifecycle:

1. **Rendering**:
   - React starts by taking your code and building a tree of elements, like a blueprint for your site.
2. **Reconciliation**:
   - When the user does something, React figures out what's different in the Virtual DOM and updates only those parts.
3. **Commit Phase**:
   - Finally, React updates the real DOM based on those changes."

Speaker 4:
"This lifecycle makes React fast and smart. Instead of updating the whole page, React focuses on just what needs to change."

Speaker 1:
"React works like a to-do list. It checks what needs updates, then only changes those things. "

Speaker 2:

"Snack Break!!!!"

---

4. Connecting React to Programming Concepts (3 minutes)

Speaker 2:
"React reflects key ideas from programming:

1. **Mutable vs Immutable States**:
    - React state can change, so it's mutable.
    - But React doesn't edit the old state directly. Instead, it makes a new state, compares it to the old one, and updates only what's needed."

Speaker 3:
"This way, React avoids bugs and keeps track of what's new. It's faster and more reliable."

Speaker 4:
"Think of the old state like a photo. When something changes, React takes a new photo and compares it to the first one."

Speaker 1:
"2. **Abstract Syntax**:

- JSX, the code you write in React, is an abstraction.
- Under the hood, JSX turns into JavaScript calls.

3. **Type Checking**:
    - React uses type tools like PropTypes or TypeScript to check that your data is correct. This prevents errors and keeps your code clean."

Speaker 2:
"4. **Judgments and Rules**:

- React treats components like functions. If you give the same input (state and props), React gives the same output (UI). This makes components easy to test."

Speaker 3:
"React is like math: same input, same result. This makes your app behave the way you expect."

---

5. Modular UI Design with React

Speaker 4:
"React's biggest strength is how modular it is.

1. **Component-Based Design**:
    - React apps are built from small, reusable components, like building blocks."

Speaker 1:
"2. **Encapsulation**:

- Each component handles its own state and logic, so components don’t mess each other up. This keeps your code clean."

Speaker 2:
"Each part of the site takes care of itself. If one part breaks, the rest still works."

Speaker 3:
"3. **Abstraction Layers**:

- React hides the tricky details of the DOM, so you can focus on how the app looks and works."

---

6. React in Website Development

Speaker 4:
"Here’s how React helps when building real websites:

1. **Dynamic UIs**:
    - React makes it easy to show or hide parts of the page when the user interacts, like clicking a button."

Speaker 1:
"2. **Real-Time Updates**:

- Features like live chats or notifications work well with React because it only updates the parts of the page that need it."

Speaker 2:
"React can handle live updates, like showing new messages in a chat app without refreshing the page."

Speaker 3:
"3. **API Integration**:

- React can fetch data from APIs and display it on your site in real time, making it great for dashboards or live feeds."

---

7. Conclusion and Next Steps

Speaker 4:
"To wrap up, React makes websites faster and easier to build. It uses programming ideas like immutable states and reusable components to keep your code clean and reliable."

Speaker 1:
"Next steps:

- Learn React Hooks like `useEffect` to handle logic.
- Explore React Router for navigation.
- Try TypeScript for better type checking."

Speaker 2:
"React gives you a smart, simple way to build apps. Once you know the basics, you can create awesome sites."

Speaker 3:
"Thanks for listening, and happy coding!"