**1. Conditional rendering in React**

Conditional rendering in React refers to the practice of displaying or hiding elements or components based on certain conditions.

* This allows you to create dynamic and interactive user interfaces that adapt to changes in data, user interactions, or application state.
* React leverages standard JavaScript conditional operators like:
  + if/else statements
  + The ternary operator (condition? true: false)
  + The logical && operator
  + The logical || operator
  + switch statements
* Examples include displaying a login button for unauthenticated users, or a welcome message for authenticated users.

**2. Element variables**

* Element variables are JavaScript variables that can store JSX elements.
* They provide a way to store components or JSX fragments based on conditions, making your render method more readable and manageable.
* You can then render these variables within your JSX based on your conditional logic, avoiding repetitive return statements.
* Example: Assigning different JSX to a variable based on a user's role and then rendering that variable in your component.

**3. Preventing components from rendering**

* In cases where a component should not be rendered based on certain conditions, you can prevent it from rendering by returning null from the component's render method.
* Returning null tells React that nothing should be rendered for that specific component.
* Important Note: Even when a component returns null, its lifecycle methods (like componentDidMount, componentDidUpdate, etc.) still execute.
* This approach is useful for optimizing performance by avoiding unnecessary rendering of components that do not need to be displayed.
* Returning null is generally preferred over returning false, as returning false can potentially interfere with React's rendering process in some situations