# OBJECTIVE

## Various Ways of Conditional Rendering in React

* Conditional rendering means rendering different UI elements based on conditions. Common approaches include:  
    
  Using if statements:  
  function Greeting({ isLoggedIn }) {  
   if (isLoggedIn) {  
   return <h1>Welcome back!</h1>;  
   }  
   return <h1>Please log in.</h1>;  
  }  
  + **Using the Ternary Operator (condition ? expr1 : expr2):**  
    <div>{isLoggedIn ? <Dashboard /> : <Login />}</div>
  + **Using Logical AND (&&):**  
      
    <div>{notifications.length > 0 && <NotificationList />}</div>
  + **Using Switch Statements:**  
    switch(status) {  
     case 'loading': return <Loading />;  
     case 'error': return <Error />;  
     default: return <Content />;  
    }

## How to Render Multiple Components

* We can render multiple components by including them within a parent component:  
    
  function App() {  
   return (  
   <div>  
   <Header />  
   <Main />  
   <Footer />  
   </div>  
   );  
  }

## Define List Component

* A List Component is a component that takes an array of data and renders a list dynamically:

function ListComponent({ items }) {  
 return (  
 <ul>  
 {items.map((item, index) => (  
 <li key={index}>{item}</li>  
 ))}  
 </ul>  
 );  
}

## Explain about Keys in React Applications

* Keys are special attributes used by React to identify which items in a list have changed, been added, or removed. They help React optimize rendering.  
    
  Rules:  
   Keys must be unique among siblings.  
   Avoid using array index if items can reorder.  
    
  Example:  
    
  items.map(item => <li key={item}>{item}</li>);

## Explain how to Extract Components with Keys

* When extracting a list item into a separate component, the key should be assigned to the component in the list:  
    
    
  function ListItem({ value }) {  
   return <li>{value}</li>;  
  }  
    
  function ItemList({ items }) {  
   return (  
   <ul>  
   {items.map(item => <ListItem key={item.id} value={item.name} />)}  
   </ul>  
   );  
  }

## Explain React Map and map() Function

* The map() function in React is used to iterate over arrays and return JSX elements:  
    
    
  const names = ['Alice', 'Bob', 'Charlie'];  
  function NameList() {  
   return (  
   <ul>  
   {names.map((name, index) => <li key={index}>{name}</li>)}  
   </ul>  
   );  
  }