# React Concepts Explained

## 1. Conditional Rendering in React

Conditional rendering in React refers to the ability to render different UI elements or components based on certain conditions. It is similar to JavaScript conditions such as if-else or switch statements. In React, you can use conditional rendering to display content dynamically depending on application state or props.  
  
Common techniques for conditional rendering:  
• Using if/else statements before returning JSX.  
• Using ternary operators directly inside JSX.  
• Using logical AND (&&) operator for short-circuit evaluation.  
  
Example (Ternary Operator):  
const isLoggedIn = true;  
return (  
 <div>  
 {isLoggedIn ? <p>Welcome back!</p> : <p>Please log in.</p>}  
 </div>  
);

## 2. Element Variables

Element variables in React are used to store elements (JSX) in a variable and use them later in the render output. This is particularly helpful when you want to conditionally render different elements without using a large block of code inside JSX.  
  
Example:  
let message;  
if (isLoggedIn) {  
 message = <h1>Welcome back!</h1>;  
} else {  
 message = <h1>Please sign up.</h1>;  
}  
  
return (  
 <div>{message}</div>  
);

## 3. Prevent Components from Rendering

There are scenarios in which you might want to prevent a component from rendering at all. This can be achieved by returning null from the component’s render method. When a component returns null, nothing is rendered, but the component still exists in the React component tree.  
  
Example:  
function WarningBanner(props) {  
 if (!props.warn) {  
 return null;  
 }  
 return <div className='warning'>Warning!</div>;  
}  
  
In the above example, if the 'warn' prop is false, the component returns null and doesn't render anything on the screen.