Conditional Rendering

Conditional rendering in React works the same way conditions work in JavaScript.

Use JavaScript operators like if or the conditional (ternary) operator to create elements representing the current state, and let React update the UI to match them.

if and if-else statements don't work inside JSX. This is because JSX is just syntactic sugar for function calls and object construction.

```
<div id={if (condition) { 'msg' }}>Hello</div>
```

```
React.createElement("div", {id: if (condition) { 'msg' }}, "Hello");
```

if Statement

```
if(true){
    return something;
}
```

Inline if with Logical && Operator

You may embed any expressions in JSX by wrapping them in curly braces. This includes the JavaScript logical && operator.

Operand 1 &&	Operand 2	Result
True	True	True
True	False	False
False	True	False
False	False	False
True	Expression	Expression
False	Expression	False

Ex:- purchase && <Payment />
If *purchase* evaluates to *true*, the <Payment /> component will be return

Ex:- purchase && <Payment />
If *purchase* evaluates to *false*, the <Payment /> component will be ignored

true && expresion1 && expression2 = expression2

if else Statement

```
if(true){
     return something_1;
} else {
    return something_2;
}
```

Inline if-else with Conditional Operator

Syntax: -

Condition ? Expression_1 : Expression_2

If the condition is true it will return expression_1 else it will return expression_2.

IIFE

```
return (
    <div>
     {
          (  () => {

               // Your Code
               }  )  ()
     }
   </div>
);
```

In React, we use curly braces to wrap an IIFE, put all the logic you want inside it (if/else, switch, ternary operators, etc), and return whatever you want to render.