**Short-circuiting** helps optimize code by stopping evaluation as soon as the result is known.

Used in logical operations like && (AND) and || (OR).

**Falsy values.**

1. false
2. 0 (zero)
3. -0 (negative zero)
4. "" (empty string)
5. null
6. undefined
7. NaN (Not-a-Number)

Truthy values

* 1. true
  2. Any non-zero number (e.g., 1, -1, 3.14, etc.)
  3. Any non-empty string (e.g., "Hello", " " (space), "0", etc.)
  4. Objects (including empty objects)
  5. Arrays (including empty arrays)

const a = false;

const b = true;

const result = a && b;  // 'b' is not evaluated because 'a' is false.

console.log(result);  // false

const a = true;

const b = false;

const result = a || b;  // 'b' is not evaluated because 'a' is true.

console.log(result);  // true

const Test = () => {

    const user = { name: 'John' }; // Object, truthy value output Welcome, John

    const user1 = {}; // Empty Object, truthy value Welcome, undefined (as no name property)

    const user2 = null; //null Falsy Value, No Output

    const user3 = undefined //undefined Falsy Value, No Output

    return (

      <div>

        {/\* Only render this if user is logged in \*/}

        {user && <h1>Welcome, {user.name}!</h1>}

      </div>

    );

}

export default Test