Conditional Statements in JavaScript

Understanding `if`, `else`, and Nested Statements

Introduction to Conditional Statements

What are Conditional Statements?

- Control the flow of execution based on conditions.

Types of Conditional Statements:

- `if` Statement
- Nested `if-else` Statement
- `if-else` Ladder
- `if-else` Ladder with Logical Operators

The if Statement

Definition: Executes a block of code if the specified condition is true.

Syntax :

```
if (condition) {
  // Code to execute if condition is true
}
```

Nested if-else Statement

Definition: An `if-else` statement inside another `if` or `else` block.

Usage: Used to check multiple conditions.

Syntax :

```
if (condition1) {
  if (condition2) {
    // Code if both conditions are true
  } else {
    // Code if condition1 is true but condition2 is false
  }
} else {
  // Code if condition1 is false
}
```

The if-else Ladder

Definition: A series of `if-else` statements used to choose among multiple options.

Syntax:

```
if (condition1) {
  // Code if condition1 is true
} else if (condition2) {
  // Code if condition2 is true
} else if (condition3) {
  // Code if condition3 is true
} else {
  // Code if none of the conditions are true
}
```

if-else Ladder with Logical Operators

Definition: Using logical operators (&&, ||, !) to combine conditions in `if-else` ladder.

Logical Operators:

- && (AND): True if both conditions are true.
- || (OR): True if at least one condition is true.
- ! (NOT): Inverts the boolean value.

```
if (condition1 && condition2) {
  // Code if both conditions are true
} else if (condition1 || condition2) {
  // Code if at least one condition is true
} else {
  // Code if none of the conditions are true
}
```

Summary of Conditional Statements

- **if** **Statement**: Basic condition checking.
- **Nested** **if-else**: Multiple levels of conditions.
- **if-else** **Ladder**: Multiple conditions with different outcomes.
- **if-else** **Ladder with Logical Operators**: Combining conditions with logical operators.