Module 8) JavaScript

JavaScript Error Handling

Question 1: What is error handling in JavaScript? Explain the try, catch, and finally blocks with an example.

Answer :

Error handling in JavaScript helps manage errors in code execution to prevent crashes. The **`try`**, **`catch`**, and **`finally`** blocks are used for handling errors safely.

- **`try`**: Code inside this block is tested for errors.
- **`catch`**: If an error occurs, this block runs and handles the error.
- **`finally`**: This block always runs, whether an error occurs or not.

Example:

```
try {

    let result = x + 10; // x is not defined (error)

} catch (error) {

    document.write("An error occurred: " +
error.message);

} finally {

    document.write("<br>Execution completed.");

}
```

If **x** is undefined, the **`catch`** block will handle the error instead of stopping the script.

Question 2: Why is error handling important in JavaScript applications?

Answer :

Error handling is important because it prevents the program from crashing and ensures a smooth user experience. It helps developers detect, debug, and fix issues without breaking the entire application. Using **`try...catch`** makes JavaScript applications more reliable and user-friendly.

Example:

```
try {
    let num = prompt("Enter a number:");
    if (isNaN(num)) {
        throw "Not a valid number!";
    }
    document.write("You entered: " + num);
} catch (error) {
    document.write("Error: " + error);
} finally {
    document.write("<br>Thank you for using our app.");
}
```

Explanation:

- The user enters a value.
- If the input is not a number, an error is thrown and caught in **`catch`**.
- The `finally` block always runs, displaying a thank-you message.

This ensures that the application doesn't break due to invalid input.