Error Handling Lecture Notes

ERROR HANDLING IN JAVASCRIPT

(Detailed Lecture Notes for Reading & Understanding)

---

1. Introduction to Errors in JavaScript

While writing JavaScript programs, it is very common to make mistakes. These mistakes are called **errors**. An error happens when JavaScript is unable to understand your code or cannot run it properly. When an error occurs, the program may stop suddenly, or it may behave in an unexpected way.

In simple words, **an error is JavaScript's way of saying "Something is wrong"**.

Real-life understanding:

Imagine you are giving directions to a friend and you say, "Turn left," but the road only has a right turn. Your friend will get confused because the instruction cannot be followed. This confusion is similar to an error in programming.

---

2. Why Errors Occur in Programs

Errors do not happen intentionally. They usually occur due to small mistakes made while writing the code. Even experienced programmers face errors daily.

Common reasons for errors include:

- Typing mistakes while writing code
- Writing wrong logic
- Using a variable that has not been created
- Trying to use something before it exists
- Internet or server problems while fetching data

Real-life example:

Error Handling Lecture Notes

If you try to open a door using the wrong key, the door will not open. The problem is not the door, but the wrong key. Similarly, errors occur when the instructions given to JavaScript are incorrect.

---

3. Types of Errors in JavaScript

JavaScript has different types of errors. Understanding these types helps us identify the problem faster and fix it correctly.

Some common types of errors are:

- **Syntax Error**: Happens when code grammar is wrong
- **Reference Error**: Happens when something does not exist
- **Type Error**: Happens when a value is used in the wrong way

Simple understanding:

Errors are like diseases. A headache and a stomach ache are different problems and need different treatments. Similarly, different errors need different fixes.

---

4. What is Error Handling?

Error handling is a technique used to **control errors instead of letting them crash the entire program**. Instead of stopping the program suddenly, error handling allows the program to respond in a proper and safe way.

The main goal of error handling is:

- To prevent the program from breaking
- To show meaningful messages to the user
- To continue program execution safely

Error Handling Lecture Notes

Real-life example:

If you stumble while walking and immediately hold a railing, you do not fall down completely. The railing acts like error handling, saving you from getting hurt.

---

5. try…catch Block in JavaScript

JavaScript provides a special structure called **try…catch** to handle errors.

- The try block contains code that might cause an error.
- The catch block runs only if an error occurs inside the try block.

If no error happens, the catch block is skipped.

Simple explanation:

You are saying to JavaScript:
"Try to run this code. If anything goes wrong, don't panic — I know how to handle it."

Real-life example:

You carefully carry a glass of water. If it slips, you immediately catch it before it falls and breaks. That catching action is similar to the catch block.

---

6. finally Block

The finally block is used along with try and catch. The special thing about finally is that **it always runs**, whether an error occurs or not.

The finally block is usually used for cleanup tasks such as:

- Closing files
- Resetting values
- Showing final messages

Error Handling Lecture Notes

Real-life example:

After eating food, washing hands is compulsory. It does not matter whether the food was spicy or sweet — washing hands always happens. This is exactly how the finally block works.

---

7. throw Keyword

JavaScript allows programmers to create their own errors using the **throw** keyword. This is useful when you want to stop the program if a certain condition is not met.

Using throw means you are manually telling JavaScript that something is wrong.

Simple explanation:

Instead of waiting for JavaScript to find a mistake, you shout:
“Stop! This situation is wrong!”

Real-life example:

A security guard stops people from entering a restricted area. Even if nothing is broken, the guard throws a warning to prevent a problem.

---

8. Custom Errors

Custom errors are errors created by the programmer with their own messages. These errors make programs more user-friendly because they explain clearly what went wrong.

It is important to remember that **declaring a function does not execute it**. If a function containing error logic is not called, nothing will appear in the console.

Real-life example:

Writing an alarm reminder but never turning it on. The reminder exists, but it never works because it was not activated.

Error Handling Lecture Notes

9. Error Object in JavaScript

Whenever an error occurs, JavaScript provides an **Error object** that contains useful information about the error.

The Error object includes:

- The name of the error
- The message describing the error
- The stack trace showing where the error happened

This information helps developers understand what went wrong and where.

Real-life example:

A medical report tells the disease name, symptoms, and affected area. Similarly, the Error object gives detailed information about the error.

10. Reference Error Explained

A **Reference Error** occurs when you try to use a variable or function that does not exist.

This usually happens when:

- A variable is not declared
- A variable name is misspelled
- A function is called before it is defined

Simple understanding:

Asking for a book that is not in the library. Since it does not exist there, the request fails.

Error Handling Lecture Notes

11. Error Handling in Asynchronous Code

Asynchronous code is used when JavaScript needs to wait for something, such as:

- Fetching data from a server
- Making API calls
- Waiting for internet responses

Because the internet is unpredictable, errors can happen easily. If these errors are not handled properly, the program may crash.

To avoid this, **try…catch is used with async code** to manage errors safely.

Real-life example:

You order food online and the restaurant cancels your order. Instead of closing the app, the app shows a polite message saying, “Please try again later.”

---

12. Error Handling with async and await

When using async and await, errors are handled using try…catch just like normal code.

A very important point to remember is that **defining an async function does not execute it**. If the function is not called, there will be no output.

Real-life example:

Cooking food but never serving it. The food is ready, but nobody eats it because it was never served.

---

13. Conclusion

Errors are a natural part of programming. Even expert programmers face errors every day. The difference between a beginner and a professional programmer is not the absence of errors, but **how well errors are handled**.

Good error handling makes programs:

- Stable
- User-friendly
- Easy to debug

SIMPLEST SUMMARY

Errors are mistakes in programs.
They happen due to typing errors, wrong logic, or internet problems.
Error handling prevents programs from crashing.
try…catch helps handle errors safely.
finally always runs.
throw creates custom errors.
Error objects give detailed information.
Async errors happen due to internet delays.

Handling errors calmly makes you a better programmer.
