

ERROR Handling :

→ Types of Errors :

→ Syntax error : This is the ~~err~~ error in the syntax. e.g. : If we write `console.log('JS');`, the above program throws a syntax error. The spelling of the log is a mistake here.

→ Runtime error : The runtime error occurs during the execution of the program. e.g. : calling an invalid function or a variable.

→ What is Try Catch in JavaScript?
→ Just like other programming languages, JavaScript also has exception handling capabilities. JavaScript implements the try-catch statements as well as the throw operator to handle exceptions.

SYNAX

```
TRY { // Some code that has an error.  
}
```

```
CATCH (e) {  
// This will run if the code in the try  
block errors }
```

→ With these statements, in JavaScript, we can also add a throw or a finally clause.

→ Throw: This is a block of code nested within the try statement and allows the programmer to write their own error that they want to handle.

→ Finally: This block of code runs once all the other statements have run.

TRY / THROW / CATCH

→ The throw operator generates an error.

We can define and throw their own custom errors. With the throw statement is executed, the statement present after it doesn't execute. The control will directly go to the catch block. In the following example, we create our own error ("This is a new error") in the throw block. Then try the code which throws an error which should be caught by the catch block.

```
TRY {  
    THROW NEW ERROR ("THIS IS A NEW ERROR")  
}  
CATCH (ERROR) {  
    CONSOLE.LOG (ERROR);  
    CONSOLE.LOG ("END OF TRY-CATCH BLOCK");  
}
```

→ TRY / CATCH / THROW / FINALLY

→ Finally is an optional block of statements that is executed after the execution of try and catch statements. It doesn't matter that any exception is thrown or not, finally block code will definitely execute if it is present. In this example, we will see how to use the finally statement with the other three statements. In this example, we do not ~~know~~

show the entire error. We just logged that the error has been handled in the catch block.

```
try {  
    console.log("THIS STATEMENT WORKS");  
    throw new Error("THIS STATEMENT THROWS AN ERROR");  
}  
catch (error) {  
    console.log("ERROR HAS BEEN HANDLED")  
}  
finally {  
    console.log("EVERYTHING HAS BEEN HANDLED");  
}
```