Exception Handling

Overview

An exception is a strange code that breaks the normal flow of the code. Such exceptions require specialised programming constructs for its execution.

Exception handling is a process or method used to handle the abnormal statements in the code and execute them. It also enables handling the flow control of the code/program. For handling the code, various handlers are used that process the exception and execute the code.

Example: 5/0 = infinity always, and it is an exception.
Thus, with the help of exception handling, it can be executed and handled.

Method

A **throw statement** is used to raise an exception. It means when an abnormal condition occurs, an exception is thrown using throw.

The thrown exception is handled by wrapping the code into the **try...catch block**. Statements in which there is a doubt that an error can occur, those statements are kept in the try block, and when the error occurs, the statements that need to be run are kept into the catch block.

Types of Errors

1. **Compile Time Error:** When a user makes a mistake in the pre-defined syntax of a programming language, a syntax error may appear.
2. **Runtime Error:** When an error occurs during the execution of the program, such an error is known as a Runtime error. **The codes which create runtime errors are known as Exceptions.** Thus, exception handlers are used for handling runtime errors.
3. **Logical Error** An error occurs when there is any logical mistake in the program that may not produce the desired output and may terminate abnormally. Example : Division by zero.

Exception Handling Statements

- try...catch statements
- throw statement
- try...catch...finally statements

try catch

The **try statement** is used to define a block of code to be tested for errors while it is being executed.
The **catch statement** is used to define a block of code to be executed if an error occurs in the try block.

Syntax :

```
try {
      //  Block of code to try
}
catch(err) {
      //Block of code to handle errors
}
```

Example :

```
var a = [1,2,3];

try{
    console.log(b[4]);
}catch(e){
    console.log("Error");
}
```

Output : Error

❖ Because b array was not defined anywhere

throw Statement

Throw statements are used for throwing user-defined errors. Users can **define and throw their own custom errors.** When a throw statement is executed, the statements present after it will not execute. The control will directly pass to the catch block.

It does the same work of console.log( ), but the output shown on the console is like an error i.e. in red colour

Syntax : throw Exception ;

Example : throw “Custom Error” ;
Output :
```
Uncaught Custom Error
```

Example :
```
var a = [1,2,3];
try{
   console.log(b[4]);
}catch(e){
   throw "Error in code";
}
```
Output :
```
Uncaught Error in code
```

try catch finally

Finally is an optional block of statements which is executed after the execution of try and catch statements. Finally, the ..block does not hold for the exception to be thrown. Any exception is thrown or not, finally block code, if present, will execute. It does not care for the output too.

Syntax :
```
try {
   //Block of code to try
}
catch(err) {
   //Block of code to handle errors
}
finally {
   // Block of code to be executed regardless of the try / catch result
}
```

- It is like the **default** in switch case

Example : var password = 1234 ;

```
try{
  if(password == 1234)
    console.log("Correct password");
}catch(err){
    throw "Incorrect password";
}finally{
     console.log("Welcome to Coding Ninjas");
}
```

Output :

```
Correct password
Welcome to Coding Ninjas
```

NOTE : The **err** parameter passed into the catch statement is called the error Object.

Error Object

JavaScript has a built-in error object that provides error information when an error occurs.
The error object provides two useful properties: name and message.

Error Name	Description
EvalError	An error has occurred in the eval( ) function
RangeError	A number "out of range" has occurred
ReferenceError	An illegal reference has occurred
SyntaxError	A syntax error has occurred
TypeError	A type error has occurred
URIError	An error in encodeURI() has occurred