Title:- JavaScript try…catch

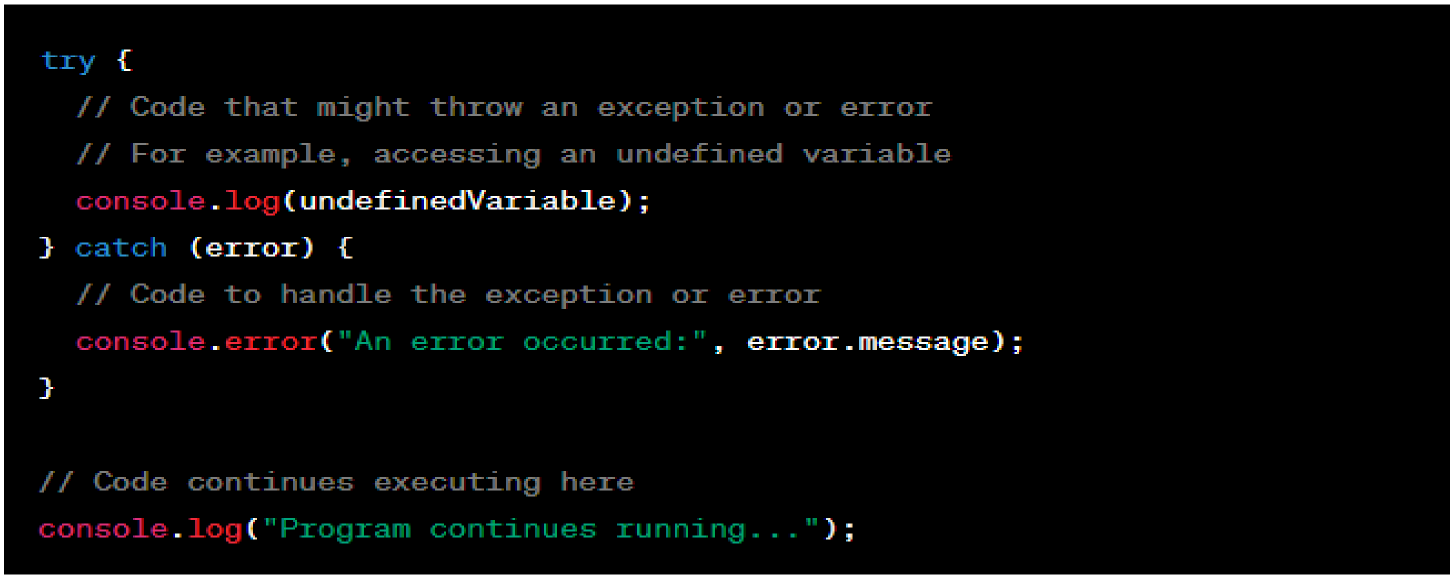
In JavaScript, the try...catch statement is used to handle exceptions or errors that might occur during the execution of your code. It provides a structured way to prevent your entire program from crashing due to unexpected issues and allows you to gracefully handle errors by providing alternative paths of execution.

Let's discuss each block of statement individually:

try{} statement: Here, the code which needs possible error testing is kept within the try block. In case any error occur, it passes to the catch{} block for taking suitable actions and handle the error. Otherwise, it executes the code written within.

catch{} statement: This block handles the error of the code by executing the set of statements written within the block. This block contains either the user-defined exception handler or the built-in handler. This block executes only when any error-prone code needs to be handled in the try block. Otherwise, the catch block is skipped

Here's a basic introduction to how the try...catch statement works:



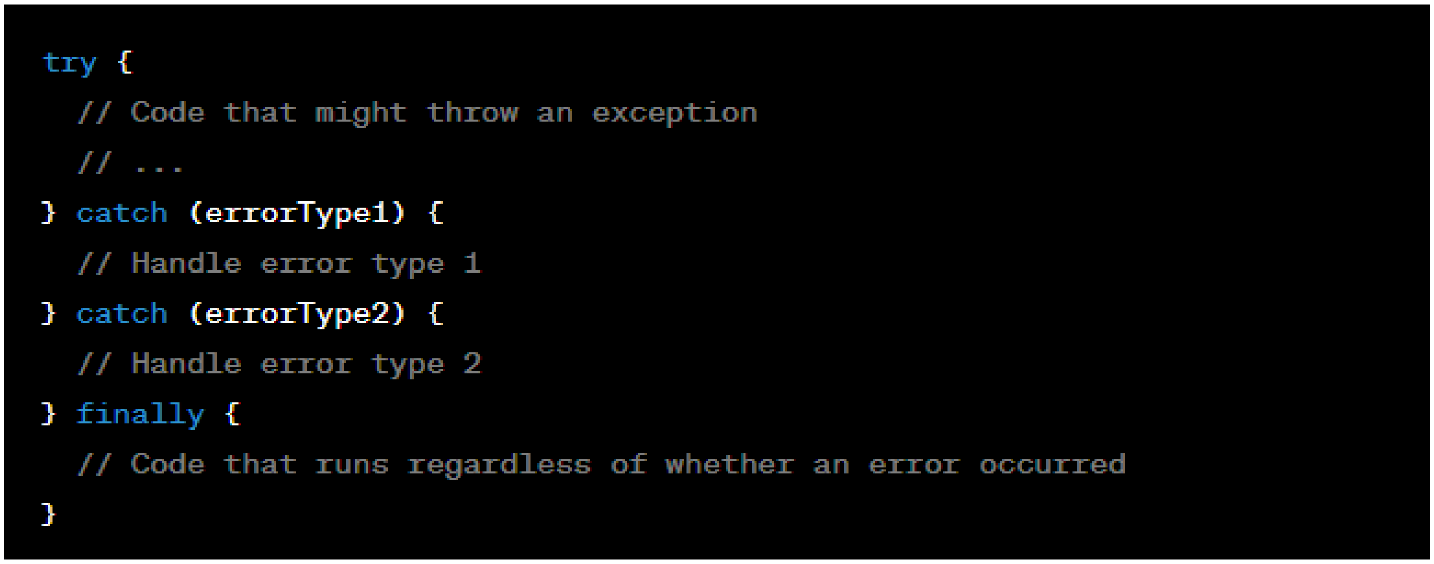
```javascript
try {
  // Code that might throw an exception or error
  // For example, accessing an undefined variable
  console.log(undefinedVariable);
} catch (error) {
  // Code to handle the exception or error
  console.error("An error occurred:", error.message);
}

// Code continues executing here
console.log("Program continues running...");
```

In the above example, if the variable undefined Variable is not defined, it will throw a reference error. However, because it's wrapped in a try block, the program won't crash. Instead, it will jump directly to the corresponding catch block. The error object (error) in the catch block contains information about the error, including its type and message.

It's important to note that try...catch only catches runtime errors that occur within the corresponding try block. Compile-time errors (syntax errors) will not be caught by the try...catch construct.

You can also have multiple catch blocks to handle different types of errors:



```
try {
  // Code that might throw an exception
  // ...
} catch (errorType1) {
  // Handle error type 1
} catch (errorType2) {
  // Handle error type 2
} finally {
  // Code that runs regardless of whether an error occurred
}
```

The finally block, if provided, will execute regardless of whether an error occurred. It's often used to perform cleanup operations, such as closing resources or releasing memory.

Overall, using try...catch in JavaScript is a crucial technique for writing robust and error-tolerant code, as it helps prevent unexpected crashes and allows you to handle errors in a controlled manner.