

JavaScript Errors Throw and Try to Catch

Last Updated : 1 Aug, 2025

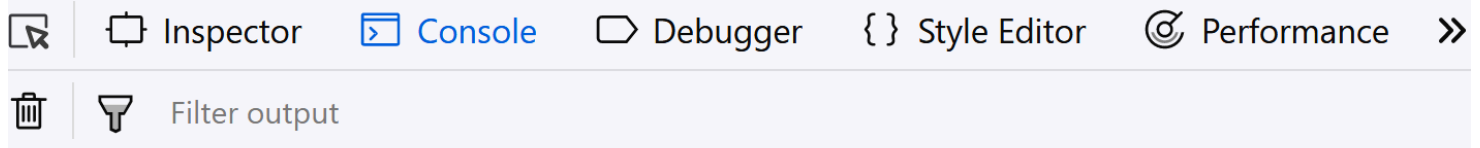


JavaScript uses throw to create custom errors and try...catch to handle them, preventing the program from crashing. The finally block ensures that code runs after error handling, regardless of success or failure.

- **throw:** Creates custom errors and stops code execution.
- **try...catch:** Catches and handles errors to prevent crashes.
- **finally:** Runs code after try and catch, regardless of errors.
- **Custom Errors:** Create custom error types by extending the Error class.

```
try {  
    dadalert("Welcome Fellow Geek!");  
}  
catch (err) {  
    console.log(err);  
}
```

Output: In the above code, we make use of 'dadalert' which is not a reserved keyword and is neither defined hence we get the error.



Syntax

The **try** statement allows you to check whether a specific block of code contains an error or not. The **catch** statement allows you to display the error if any are found in the try block.

```
try {  
    Try Block to check for errors.  
}  
catch(err) {  
    Catch Block to display errors.  
}
```

Output: In the above code, our catch block will not run as there's no error in the above code and hence we get the output 'Value of variable a is: 10'.

Javascript Throws BlockThe *throw* Statement

The throw statement allows you to create custom error messages and throw exceptions manually.

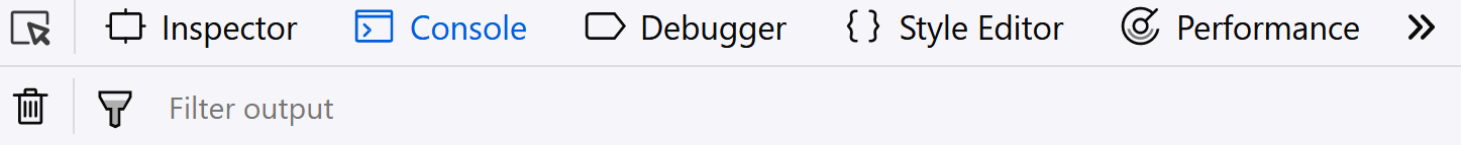
```
throw new Error("Error message");
```

Key Point:

- `throw` creates a custom error.
- It can be used to stop the execution of a program and pass control to error handling mechanisms.
- You can throw any type of object (string, number, object, etc.), but it's common to throw an instance of `Error`.

```
try {  
  throw new Error('Yeah... Sorry');  
}  
catch (e) {  
  console.log(e);  
}
```

Output:



Error: "Yeah... Sorry"

The finally Block

The finally Statement runs unconditionally after the execution of the try/catch block. Its syntax is

```
try {  
  console.log('try');  
} catch (e) {  
  console.log('catch');  
} finally {  
  console.log('finally');  
}
```

Output: The Finally Block can also override the message of the catch block so be careful while using it.

Creating Custom Error Types

You can create custom error types by extending the built-in `Error` class. This can be useful for more specific error handling.

```
function validateInput(input) {  
  if (input < 0) {  
    throw new ValidationError("Input must be a positive number");  
  }  
  return "Valid input";  
}
```

Output:



Error Handling - try and catch in JavaScript

[Visit Course ↗](#)[Comment](#)**Nishan...**[+ Follow](#)

16



Article Tags:

[JavaScript](#)[Web Technologies](#)[JavaScript-Errors](#)

Explore

[JavaScript Basics](#)[Array & String](#)[Function & Object](#)[OOP](#)[Asynchronous JavaScript](#)[Exception Handling](#)[DOM](#)[Advanced Topics](#)

Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)



Registered Address:

K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

Company

[About Us](#)[Legal](#)[Privacy](#)[Policy](#)[Contact Us](#)[Advertise with us](#)[GFG](#)[Corporate](#)[Solution](#)

Explore

[POTD](#)[Job-A-Thon](#)[Blogs](#)[Nation](#)[Skill Up](#)

Tutorials

[Programming](#)[Languages](#)[DSA](#)[Web](#)[Technology](#)[AI, ML & Data Science](#)[DevOps](#)[CS Core](#)[Subjects](#)

Courses

[ML and Data](#)[Science](#)[DSA and](#)[Placements](#)[Web](#)[Development](#)[Programming](#)[Languages](#)[DevOps & Cloud](#)[GATE](#)

Videos

[DSA](#)[Python](#)[Java](#)[C++](#)[Web](#)[Development](#)[Data Science](#)[CS Subjects](#)

Preparation Corner

[Interview](#)[Corner](#)[Aptitude](#)[Puzzles](#)[GfG 160](#)[System Design](#)

Campus
Training
Program

Interview
Preparation
Software and
Tools

Trending
Technologies