

Learn JavaScript Syntax: Error Handling

Runtime Error in JavaScript

A JavaScript runtime error is an error that occurs within code while its being executed. Some runtime errors are built-in objects in JavaScript with a name and message property. Any code after a thrown runtime error will not be evaluated.

The throw Keyword in JavaScript

The JavaScript `throw` keyword is placed before an `Error()` function call or object in order to construct and raise an error. Once an error has been thrown, the program will stop running and any following code will not be executed.

```
// The program will raise and output this  
Error object with message 'Something went  
wrong'
```

```
throw Error('Something went wrong');
```

```
//The program will stop running after an  
error has been raised, and any following  
code will not be executed.
```

```
console.log('This will not be printed');
```

Javascript Error Function

The JavaScript `Error()` function creates an error object with a custom message. This function takes a string argument which becomes the value of the error's `message` property. An error created with this function will not stop a program from running unless the `throw` keyword is used to raise the error.

```
console.log(Error('Your password is too  
weak.')); //Error: Your password is too  
weak.
```

A JavaScript `try ... catch` statement can anticipate and handle thrown errors (both built-in errors as well as those constructed with `Error()`) while allowing a program to continue running. Code that may throw an error(s) when executed is written within the `try` block, and actions for handling these errors are written within the `catch` block.

// A try...catch statement that throws a constructed `Error()`

```
try {
  throw Error('This constructed error
will be caught');
} catch (e) {
  console.log(e); // Prints the thrown
Error object
}
```

// A try...catch statement that throws a built-in error

```
const fixedString = 'Cannot be
reassigned';
try {
  fixedString = 'A new string'; //
A TypeError will be thrown
} catch (e) {
  console.log('An error occurred!'); //
Prints 'An error occurred!'
}
```

```
console.log('Prints after error'); //
Program continues running after the error
is handled and prints 'Prints after
error'
```