



# JavaScript Errors

March 2023



JavaScript Errors can fit into two categories:

1. **Syntax Errors** (parsing errors, happens at interpretation time)

`SyntaxError`

2. **Runtime Errors** (exceptions, happens during execution)

`ReferenceError`

`TypeError`

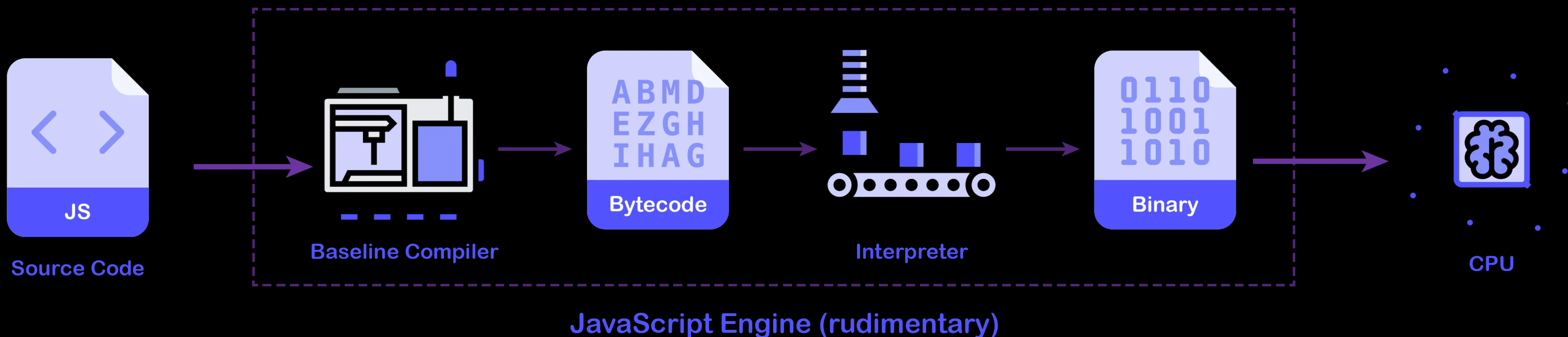
`RangeError`

`URIError`

`EvaluationError`



# Syntax Errors



# SyntaxError



Represents an error when trying to interpret syntactically invalid code. It is thrown when the JavaScript engine encounters tokens or token order that does not conform to the syntax of the language when parsing code.

```
var animal = "cat
```



# ReferenceError

The ReferenceError object represents an error when a variable that doesn't exist (or hasn't yet been initialized) in the current scope is referenced.

```
console.log(undefinedVariable);
```

# TypeError



Represents an error when an operation could not be performed, typically (but not exclusively) when a value is not of the expected type.

```
console.logg("cat");
```



# RangeError

The RangeError object indicates an error when a value is not in the set or range of allowed values.

```
const animal = "cat";  
const normalizedCat = animal.normalize("abc");
```



# Exercise 1





# Creating & Throwing Errors



# Error objects



The `Error()` constructor creates an error object.

`Error()` can be called with or without `new`.

Both create a new `Error` instance.

`new Error()`

`Error()`

`Error(message)`

`Error(message, options)`

`Error(message, fileName)`

`Error(message, fileName, lineNumber)`

`message`

A human-readable description of the error.

`options`

An object that has the `cause` property. Often used to pass the original error.

`fileName` ⚠

Non-standard Mozilla property

`lineNumber` ⚠

Non-standard Mozilla property

# Instance Properties & Methods



Error.prototype.toString()

Error.prototype.constructor

Error.prototype.name

Error.prototype.stack ▲

message

cause

fileName ▲

lineNumber ▲

columnNumber ▲

▲ = Non-standard property

# Static Methods

(Non-standard V8 functions)



`Error.captureStackTrace()`

`Error.stackTraceLimit()`

`Error.prepareStackTrace()`



# Exercise 2

# throw

## expression



- The **throw** statement throws a user-defined exception.
- Execution stops and controls passed to first **catch** block.
- In no **catch** blocks, the program will terminate.
- You can throw other JavaScript data types.

# try...catch



```
try {  
    // the code to try  
} catch(error) {  
    // runs if the code throws any exceptions  
} finally {  
    // any code in here is executed regardless  
}
```





# Exercise 3

You can create an Error from  
the built in Error objects



```
new SyntaxError()  
new TypeError()  
new RangeError()  
new ReferenceError()
```

And check if an error is an instance of a built-in Error object

```
const MyError = new TypeError("Oops!");  
const isTypeError = MyError instanceof TypeError; // true
```



# Exercise 4



# Debugging Errors

# Debugging JavaScript



1. `console.log()`
2. Breakpoints
3. The **debugger** keyword
4. Error Reporting Services

Sentry

LogRocket

Rollbar

DataDog



# Exercise 5



# Error Reporting Services

Sentry

LogRocket

Rollbar

DataDog

<https://sentry.io/welcome/>

<https://logrocket.com/>

<https://www.datadoghq.com/dg/apm/benefits/>



# Exercise 6

Creating a custom error



# Resources



[https://developer.mozilla.org/en-US/docs/web/javascript/reference/global\\_objects/error](https://developer.mozilla.org/en-US/docs/web/javascript/reference/global_objects/error)

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Errors>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/throw>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/try...catch>

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control flow and error handling](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling)

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/debugger>

[https://www.w3schools.com/js/js\\_errors.asp](https://www.w3schools.com/js/js_errors.asp)

[https://www.w3schools.com/js/js\\_debugging.asp](https://www.w3schools.com/js/js_debugging.asp)