



# < Aula 1: Tipos de erros >

## Tratamento de erros

## Objetivos



- < 1. Apresentar o objeto Error do ECMAScript >
- 2. Apresentar a DOMException

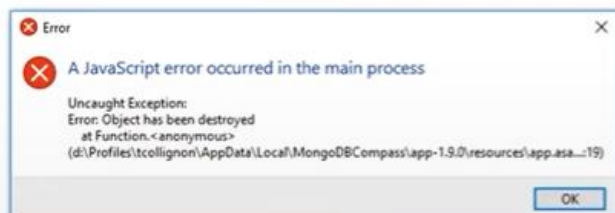
# ECMAScript Error



Erros que ocorrem em tempo de execução.



```
> let sum = a + 2  
✖ ▶ Uncaught ReferenceError: a is not defined  
   at <anonymous>:1:11
```



Composto por:

- Mensagem
- Nome
- Linha
- Call Stack



Aula 1 | Etapa 2:



## DOMException

Tratamento de erros





# DOMException



Erros relacionados ao Document Object Model (DOM).

```
DOMException: String contains an invalid character index.js:1
e      index.js:1
consoleOnDev      utils.js:120
errorOnDev      utils.js:136
Document      Document.js:216
_callee$      Document.js:114
tryCatch      runtime.js:63
invoke      runtime.js:293
defineIteratorMethods      runtime.js:118
  Babel 2
    asyncGeneratorStep
      _throw
Source error: -InvalidCharacterError: String contains an invalid character
```

```
DOMException: "Node cannot be inserted at the specified point in the hierarchy"
  VueJS 5
    logError
    globalErrorHandler
    handleError
    nextTick
    flushCallbacks
  run      es6-promise.js:75
  notify      es6-promise.js:92
  flush      microtask.js:18_
```



## Aula 2: Tratando erros

Tratamento de erros



- < 1. Explicar a diferença entre “throw” e “return” >
2. Apresentar a implementação da declaração try/catch



Aula 1 | Etapa 1:

< **Throw** >

Tratamento de erros



```
function verificaPalindromo(string) {
  if (!string) return "String inválida";

  return string === string.split('').reverse().join('');
}

verificaPalindromo('cat');
```

```
> function verificaPalindromo(string) {
  if (!string) return "String inválida";

  return string === string.split('').reverse().join('');
}

verificaPalindromo('');
< "String inválida"
```

```
function verificaPalindromo(string) {
  if (!string) throw "String inválida";

  return string === string.split('').reverse().join('');
}

verificaPalindromo('cat');
```

```
> function verificaPalindromo(string) {
  if (!string) throw "String inválida";

  return string === string.split('').reverse().join('');
}

verificaPalindromo('');
✖ Uncaught String inválida
verificaPalindromo @ VM302:2
(anonymous) @ VM302:7
```



## Aula 1 | Etapa 2:

# < Try...catch

## Tratamento de erros





# Try...catch



```
function verificaPalindromo(string) {  
  if (!string) throw "String inválida";  
  
  return string ===  
    string.split('').reverse().join('');  
}  
  
function tryCatchExemplo(string) {  
  try {  
    verificaPalindromo(string)  
  }  
  catch(e) {  
    console.log(e)  
  }  
}  
  
tryCatchExemplo('');
```

```
> function verificaPalindromo(string) {  
  if (!string) throw "String inválida";  
  
  return string ===  
    string.split('').reverse().join('');  
}  
  
function tryCatchExemplo(string) {  
  try {  
    verificaPalindromo(string)  
  }  
  catch(e) {  
    console.log(e)  
  }  
}  
  
tryCatchExemplo('');  
String inválida VM324:12
```



# Try...catch



```
function verificaPalindromo(string) {  
  if (!string) throw "String inválida";  
  
  return string === string.split('').reverse().join('');  
}  
  
function tryCatchExemplo(string) {  
  try {  
    verificaPalindromo(string)  
  }  
  catch(e) {  
    throw e;  
  }  
}  
  
tryCatchExemplo('');
```

```
> function verificaPalindromo(string) {  
  if (!string) throw "String inválida";  
  
  return string ===  
    string.split('').reverse().join('');  
}  
  
function tryCatchExemplo(string) {  
  try {  
    verificaPalindromo(string)  
  }  
  catch(e) {  
    throw e;  
  }  
}  
  
tryCatchExemplo('');  
Uncaught String inválida VM494:12
```





## Aula 1 | Etapa 3:

# Finally

## Tratamento de erros



# Finally



```
function verificaPalindromo(string) {  
  if (!string) throw "String inválida";  
  
  return string === string.split('').reverse().join('');  
}  
  
function tryCatchExemplo(string) {  
  try {  
    return verificaPalindromo(string)  
  }  
  catch(e) {  
    throw e;  
  }  
  finally {  
    console.log('A string enviada foi: ' + string);  
  }  
}  
  
tryCatchExemplo('');
```

```
> function verificaPalindromo(string) {  
  if (!string) throw "String inválida";  
  
  return string === string.split('').reverse().join('');  
}  
  
function tryCatchExemplo(string) {  
  try {  
    return verificaPalindromo(string)  
  }  
  catch(e) {  
    throw e;  
  }  
  finally {  
    console.log('A string enviada foi: ' + string);  
  }  
}  
  
tryCatchExemplo('');  
A string enviada foi: VM1096:15  
▶ Uncaught String inválida VM1096:12
```



```
function verificaPalindromo(string) {  
  if (!string) throw "String inválida";  
  
  return string === string.split('').reverse().join('');  
}  
  
function tryCatchExemplo(string) {  
  try {  
    return verificaPalindromo(string)  
  }  
  catch(e) {  
    throw e;  
  }  
  finally {  
    console.log('A string enviada foi: ' + string);  
  }  
}  
  
tryCatchExemplo('ala');
```

```
> function verificaPalindromo(string) {  
  if (!string) throw "String inválida";  
  
  return string === string.split('').reverse().join('');  
}  
  
function tryCatchExemplo(string) {  
  try {  
    return verificaPalindromo(string)  
  }  
  catch(e) {  
    throw e;  
  }  
  finally {  
    console.log('A string enviada foi: ' + string);  
  }  
}  
  
tryCatchExemplo('ala');  
A string enviada foi: ala  
VM904:15  
< true
```



## Aula 3: Criando erros personalizados

### Tratamento de erros





# Objetivos



1. Explicar como manipular o objeto Error



Aula 3 | Etapa 1:



## O objeto Error

Tratamento de erros



# O objeto Error



## Código:

```
new Error(message, fileName, lineNumber)

// todos os parâmetros são opcionais

const MeuErro = new Error('Mensagem Inválida');

throw MeuErro;
```

## Resultado:

```
✖ Uncaught Error: Mensagem Inválida
  at <anonymous>:1:17
  (anonymous) @ VM60:1
```

# O objeto Error



## Código:

```
const MeuErro = new Error('Mensagem Inválida');
MeuErro.name = 'InvalidMessage';

throw MeuErro;
```

## Resultado:

```
✖ Uncaught InvalidMessage: Mensagem Inválida
  at <anonymous>:3:17
  (anonymous) @ VM237:3
```



# O objeto Error



```
> MeuErro.name
< "InvalidMessage"
> MeuErro.stack
< "InvalidMessage: Mensagem Inválida\n    at <anonymous>:3:17"
> MeuErro
< InvalidMessage: Mensagem Inválida
    at <anonymous>:3:17
```