# Tipos de Erros

Error do ECMAScript – Erros que ocorrem em tempo de execução. Composto por:

* Mensagem
* Nome
* Linha
* Call Stack

DOMException – Erro do DOM (Document Object Model) erros relacionados com erros que ocorrem quando uma página da web está sendo executada.

# Tratando os Erros (Throw e Return), Try/Catch e Finaly

## Return

Somente retornará um valor de acordo com a lógica de programação feita, enquanto o Throw irá exibir uma mensagem de erro de acordo com a lógica de programação feita.

Ex:

function verificaPalindromo(string) {

if (!string) return "String Inválida";

if (String.prototype.toUpperCase(string) === String.prototype.toUpperCase(string.split('').reverse().join(''))){

return `A palavra ${string} é um palíndromo`;

} else {

return `A palavra ${string} não é um palíndromo`;

}

}

console.log(verificaPalindromo('Ava'));

function verificaPalindromo2(string) {

if (!string) throw "String Inválida";

if (String.prototype.toUpperCase(string) === String.prototype.toUpperCase(string.split('').reverse().join(''))){

return `A palavra ${string} é um palíndromo`;

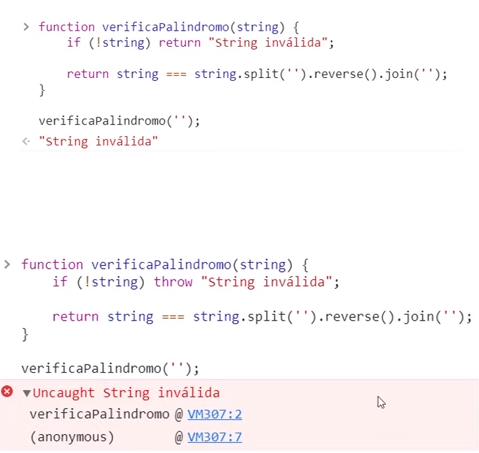
} else {

return `A palavra ${string} não é um palíndromo`;

}

}

console.log(verificaPalindromo2(''));



## Try / Catch

A função try/catch, ele tenta executar um determinado trecho de código, caso dê certo, morre no try, caso contrário, ele entra no catch (que será para a exceção) para tratarmos o erro ocorrido. Dentro do catch também podemos utilizar o throw para exibir no navegador o erro que ocorreu.

function verificaPalindromo2(string) {

if (!string) return "String Inválida";

console.log( (String.prototype.toUpperCase(string) === String.prototype.toUpperCase(string.split('').reverse().join(''))));

if (String.prototype.toUpperCase(string) === String.prototype.toUpperCase(string.split('').reverse().join(''))){

console.log(`A palavra ${string} é um palíndromo`);

return `A palavra ${string} é um palíndromo`;

} else {

console.log(`A palavra ${string} não é um palíndromo`);

return `A palavra ${string} não é um palíndromo`;

}

}

function tryCatchExemplo (string2){

try {

verificaPalindromo2(string2);

} catch(e){

throw(e);

}

}

tryCatchExemplo('ava');

## Try / Catch / Finally

O finally servirá para a execução de uma mensagem caso entre no try ou no catch.

## O Objeto Error

O objeto Error serve para criarmos uma mensagem de erro padrão para tratamento de nossos erros.

Exemplo:

const MeuErro = new Error(‘Mensagem Inválida’);

throw MeuErro;

Podemos alterar também a mensagem depois de já definida:

const MeuErro = new Error(‘Mensagem Inválida’);

MeuErro.name = ‘InvalidMessage’;

throw MeuErro;