**ARROW FUNCTION**

// Utilizando arrow function (na arrow function pode omitir o return, pois é implicito e por ser uma linha, podemos remover as chaves)

const maxNumero2 = (*x*, *y*) => *x* > *y* ? `${*x*} > ${*y*}` : `${*y*} > ${*x*}`;

Cria uma variável com que recebe os valores da função e com o símbolo => aponta para a função que ele trata

**TRATANDO E LANÇANDO ERROS**

Utilizamos o try catch para realizar o tratamento de erros em nosso código, podendo também levantar erros com o throw.

function soma (*x*, *y*) {

    if (typeof *x* !== 'number' || typeof *y* !== 'number') {

        throw **new** *TypeError*('x e y precisam ser números');

    }

    return *x* + *y*;

}

try {

    console.log(soma(1, 2));

    console.log(soma('1', 2));

} catch(error) {

    // console.log(error)

    console.log('Alguma coisa mais amigavel para o usuário')

}

Junto com o try e catch pode ser usado tbm o finally, que sempre será executada, independente de ter erro ou não:

try {

    // É executado quando não há erros

    console.log('Abri um arquivo')

    console.log('Manipulei um arquivo e gerou erro')

} catch (e) {

    // É executado quando há erros

    console.log('Tratando o erro')

} finally {

    // Sempre é executado

    console.log('FINALLY: Fechei o arquivo')

}

trim() -> remove os espaços em branco antes e depois da string