**Tratamento de Erros**

**Sintaxe**

***Try{***

***// bloco de código***

***}catch{***

***// bloco de código***

***}finally{***

***// bloco de código***

***}throw{***

***// bloco de código***

***}***

**Try:** tenta executar um bloco de código

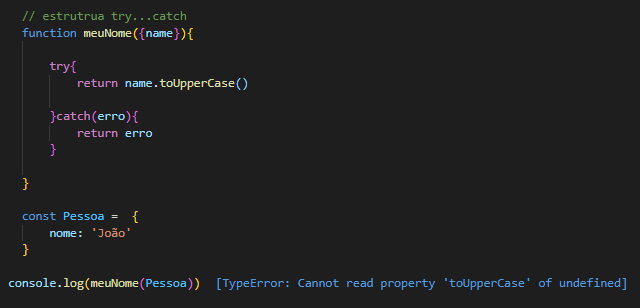
**Catch:** bloco de código para lidar com qualquer erro

**Finally:** bloco de código que será executado independetemente do resultado

**Throw:** define um errro customizado

**Estrutura Try...Catch**

**Ex 1:** tentando botar todas as letras do atributo ‘name’ do objeto ‘Pessoa’, passado para função meuNome, em maiúsculo.



**Catch vai pegar um objeto do tipo Error**

**Objeto Error**

**Proprieades**

* **Name**: nome do erro
* **Message:** descrição do erro

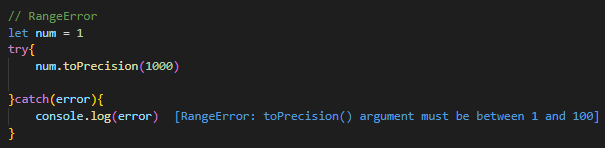
Existem 6 tipos de erros:

* **EvalError:** erro na função eval()
* **RangeError:** número fora do intervalo
* **ReferenceError:** referência ilegal
* **SytanxErro:** erro de sintaxe
* **TypeError:** erro de tipo de dado
* **URIError: erro** na função encodeURI()

**EvalError:** tipo de erro depreciado. Use SytanxError ao invés.

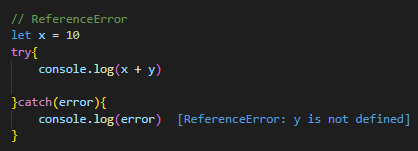
**RangeError:** ocorre quando os valores estão fora do intervalo de valores permitidos.

**Ex.:** um número só pode ter até 100 casas decimais, mas 1000 foi colocado



**ReferenceError:** quando você referencia/cita algo, como função, objeto, variável etc, que não foi declarado.

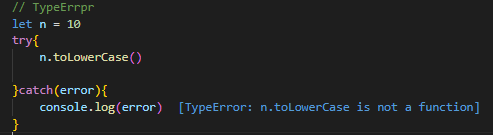
**Ex.:** tentando cita a variável y, mas ela não foi declarada ainda.



**SyntaxError:** quando você tenta executar um código escrito de forma errada

Ex.:

**TypeError:** quando o tipo do dado não é o esperado

Ex.:

URIError:

**Declaração Throw**

É possível criar erros personalizados.

**Sintaxe 1:**

***throw nome do erro***

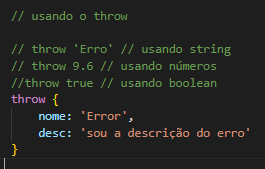
**Onde**:

* Nome do erro: pode ser uma string, boolean, número ou objeto

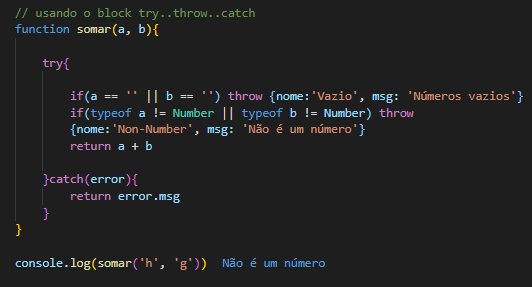
**Sintaxe 2:**

***throw new Error(mensagem)***

**Ex. 1:** tipos de erros que podem ser criados



**Ex. 2:** criando erros como objetos



**Estrutura Finally**

Bloco de códigos que será executado casa haja erro ou não, depois do bloco try...catch

**Ex.:** a função vai testar se o valor é uma string ou não. Caso não seja, cria um erro e mostra no console. Independetemente do resultado, o bloco finally será executado.

