



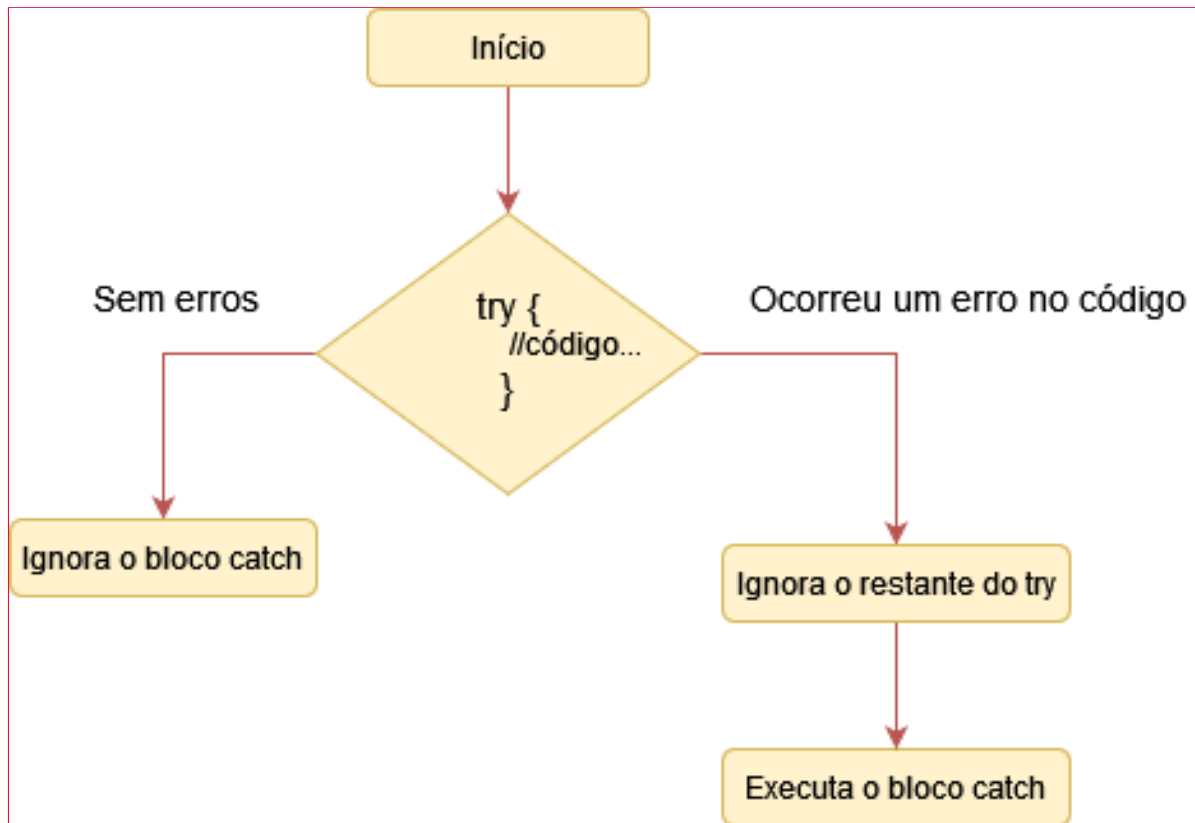
Front End II

Tratamento de erros

Erros que ocorrem em um programa podem ocorrer devido ao nosso descuido, entrada inesperada do usuário, resposta errônea do servidor, entre outros motivos. Normalmente, um script é interrompido e parado quando isto acontece. Mas podemos evitar isto com `try...catch` que nos permite "encurrular" os erros para que o script ainda possa funcionar.

- A declaração **try** permite que um bloco de código seja testado para detectar erros.
- A declaração **catch** permite o tratamento de erros.
- A declaração **throw** permite a criação de erros personalizados.
- A declaração **finally** permite que o código seja executado, após tentativa e captura, independentemente do resultado.

Fluxo



Sintaxe

```
try {  
  Block of code to try  
}  
catch(err) {  
  Block of code to handle errors  
}  
finally {  
  Block of code to be executed regardless of the try / catch result  
}
```



Exemplo:

```
function myFunction() {  
  let message, x;  
  message = document.getElementById("intro");  
  message.innerHTML = "";  
  x = document.getElementById("demo").value;  
  try { //Executamos um try com condicionadores enviando(throw) uma  
mensagem:  
  
    if(x == "") throw "Conteúdo vazio";  
    if(isNaN(x)) throw "Não é um número";  
    x = Number(x);  
    if(x > 10) throw "Número muito alto";  
    if(x < 5) throw "Número muito baixo";  
  
  }  
  catch(err) { //Executamos um catch para tratar do erro exibindo-o no  
navegador  
    message.innerHTML = "Error: " + err + ".";  
  }  
  
  finally { //Executamos a ação que encerra a função para devolver o valor  
requerido  
    document.getElementById("demo").value = "";  
  }  
}
```

Nota-se que um erro pode ter origem em valores diferentes:

Nome de erro	Descrição
RangeError	Ocorreu um número "fora de alcance".
ReferenceError	Ocorreu uma referência ilegal.
Erro de sintaxe	Ocorreu um erro de sintaxe.
Erro de teclado	Ocorreu um erro de digitação.
URIError	Ocorreu um erro em encodeURIComponent ().