# Erros JavaScript - Jogue e tente pegar

[❮ Anterior](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_regexp.asp&usg=ALkJrhhUmvXKu92_bnuhRFq3hMcpiZuSew)[Próximo ❯](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_debugging.asp&usg=ALkJrhjtWTqsZkqzXfzCzRFVGT6sTloInQ)

A declaração **try** permite testar um bloco de código para erros.

A declaração **catch** permite manipular o erro.

A declaração de **lançamento** permite criar erros personalizados.

A declaração **final** permite que você execute o código, depois de tentar e pegar, independentemente do resultado.

## Erros acontecerão!

Ao executar o código JavaScript, podem ocorrer diferentes erros.

Os erros podem ser erros de codificação feitos pelo programador, erros devido à entrada incorreta e outras coisas imprevisíveis.

### Exemplo

Neste exemplo, escrevemos um alerta como adddlert para produzir deliberadamente um erro:

<p id="demo"></p>  
  
<script>  
try {  
    adddlert("Welcome guest!");  
}  
catch(err) {  
    document.getElementById("demo").innerHTML = err.message;  
}  
</script>

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_try_catch&usg=ALkJrhjMFZQ-AQF_MiUWV2mQfY_0RcsXWg)

JavaScript captura **adddlert** como um erro e executa o código de captura para lidar com isso.

## JavaScript tenta pegar

A instrução **try** permite que você defina um bloco de código a ser testado para erros enquanto ele está sendo executado.

A instrução **catch** permite que você defina um bloco de código a ser executado, se ocorrer um erro no bloco try.

As declarações de JavaScript **tentam** **pegar** em pares:

try {  
    Block of code to try}  
catch(err) {  
    Block of code to handle errors}

## JavaScript lança erros

Quando ocorre um erro, o JavaScript normalmente irá parar e gerar uma mensagem de erro.

O termo técnico para isso é: JavaScript **lançará uma exceção (lançar um erro)** .

O JavaScript criará um **objeto de erro** com duas propriedades: **nome** e **mensagem** .

## A declaração de lançamento

A declaração de **lançamento** permite que você crie um erro personalizado.

Tecnicamente, você pode **lançar uma exceção (lançar um erro)** .

A exceção pode ser uma Cadeia de JavaScript, um Número, um Booleano ou um Objeto:

throw "Too big";    // throw a text  
throw 500;          // throw a number

Se você usar **jogar** junto com **tentar** e **pegar** , você pode controlar o fluxo do programa e gerar mensagens de erro personalizadas.

## Exemplo de validação de entrada

Este exemplo examina a entrada. Se o valor for errado, uma exceção (err) é lançada.

A exceção (erro) é captada pela instrução catch e uma mensagem de erro personalizada é exibida:

<!DOCTYPE html>  
<html>  
<body>  
  
<p>Please input a number between 5 and 10:</p>  
  
<input id="demo" type="text">  
<button type="button" onclick="myFunction()">Test Input</button>  
<p id="message"></p>  
  
<script>  
function myFunction() {  
    var message, x;  
    message = document.getElementById("message");  
    message.innerHTML = "";  
    x = document.getElementById("demo").value;  
    try {   
        if(x == "") throw "empty";  
        if(isNaN(x)) throw "not a number";  
        x = Number(x);  
        if(x < 5) throw "too low";  
        if(x > 10) throw "too high";  
    }  
    catch(err) {  
        message.innerHTML = "Input is " + err;  
    }  
}  
</script>  
  
</body>  
</html>

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_throw_error&usg=ALkJrhhvoUkHq8_6gyLEcYQwOrnIVrAsjw)

## Validação HTML

O código acima é apenas um exemplo.

Os navegadores modernos costumam usar uma combinação de JavaScript e validação HTML integrada, usando regras de validação predefinidas definidas em atributos HTML:

<input id="demo" type="number" min="5" max="10" step="1"

Você pode ler mais sobre a validação de formulários em um capítulo posterior deste tutorial.

## A Declaração final

A declaração **final** permite que você execute o código, depois de tentar e pegar, independentemente do resultado:

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() {  
    var message, x;  
    message = document.getElementById("message");  
    message.innerHTML = "";  
    x = document.getElementById("demo").value;  
    try {   
        if(x == "") throw "is empty";  
        if(isNaN(x)) throw "is not a number";  
        x = Number(x);  
        if(x > 10) throw "is too high";  
        if(x < 5) throw "is too low";  
    }  
    catch(err) {  
        message.innerHTML = "Error: " + err + ".";  
    }  
    finally {  
        document.getElementById("demo").value = "";  
    }  
}

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_finally_error&usg=ALkJrhgEPtS5IfCm5kDl-__pAYTI0tMYRA)

## O objeto de erro

O JavaScript possui um objeto de erro incorporado que fornece informações de erro quando ocorre um erro.

O objeto de erro fornece duas propriedades úteis: nome e mensagem.

## Propriedades do objeto de erro

|  |  |
| --- | --- |
| **Propriedade** | **Descrição** |
| nome | Define ou retorna um nome de erro |
| mensagem | Define ou retorna uma mensagem de erro (uma string) |

## Valores de nome de erro

Seis valores diferentes podem ser retornados pela propriedade do nome do erro:

|  |  |
| --- | --- |
| **Nome do erro** | **Descrição** |
| EvalError | Ocorreu um erro na função eval () |
| RangeError | Um número "fora do alcance" ocorreu |
| ReferenceError | Uma referência ilegal ocorreu |
| Erro de sintaxe | Ocorreu um erro de sintaxe |
| TypeError | Ocorreu um erro de tipo |
| URIError | Ocorreu um erro no encodeURI () |

Os seis valores diferentes estão descritos abaixo.

## Eval Error

Um **EvalError** indica um erro na função eval ().

As versões mais recentes do JavaScript não lançam nenhum EvalError. Use o SyntaxError em vez disso.

## Erro de intervalo

Um **RangeError** é jogado se você usar um número que esteja fora do alcance de valores legais.

Por exemplo: você não pode definir o número de dígitos significativos de um número para 500.

### Exemplo

var num = 1;  
try {  
    num.toPrecision(500);   // A number cannot have 500 significant digits  
}  
catch(err) {  
    document.getElementById("demo").innerHTML = err.name;  
}

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_error_rangeerror&usg=ALkJrhg8I7cSqK5WVsc8wIbagy5xhzJtQQ)

## Erro de referência

Um **ReferenceError** é lançado se você usar (referência) uma variável que não tenha sido declarada:

### Exemplo

var x;  
try {  
    x = y + 1;   // y cannot be referenced (used)  
}  
catch(err) {  
    document.getElementById("demo").innerHTML = err.name;  
}

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_error_referenceerror&usg=ALkJrhh5c-kqgN2wE7o50kO21g3n6PUgMA)

## Erro de sintaxe

Um **SyntaxError** é lançado se você tentar avaliar o código com um erro de sintaxe.

### Exemplo

try {  
    eval("alert('Hello)");   // Missing ' will produce an error  
}  
catch(err) {  
    document.getElementById("demo").innerHTML = err.name;  
}

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_error_syntaxerror&usg=ALkJrhjEqwZ0SuBimj9Jb5w54gEiKEStRQ)

## Tipo Erro

Um **TypeError** é lançado se você usar um valor que esteja fora do intervalo de tipos esperados:

### Exemplo

var num = 1;  
try {  
    num.toUpperCase();   // You cannot convert a number to upper case  
}  
catch(err) {  
    document.getElementById("demo").innerHTML = err.name;  
}

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_error_typeerror&usg=ALkJrhgvgw81_YD-R9ekto6ayz8pPKP85Q)

## Erro URI

Um **URIError** é lançado se você usar caracteres ilegais em uma função URI:

### Exemplo

try {  
    decodeURI("%%%");   // You cannot URI decode these percent signs  
}  
catch(err) {  
    document.getElementById("demo").innerHTML = err.name;  
}

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_error_urierror&usg=ALkJrhj-olebok1I8_Z793qYOl3XW7p5Qw)