




Tratamento de Exceções JavaScript

Prof. Rodrigo Attique

Introdução ao Tratamento de Exceções




O que são exceções?

- Quando algo sai das regras propostas temos uma exceção.
- Por exemplo:
 - Horário de chegar ao trabalho é as 8h mas por motivos de trânsito chegou as 8:15.
 -  Situação: Você tenta pagar com o cartão, mas não tem saldo suficiente.
 -  Situação: Você digita a senha errada três vezes.
 -  Situação: Mais pessoas entram no elevador do que ele suporta.




Por que precisamos tratá-las?

- Se exceções são eventos inesperados devemos saber como lidar com o inesperado.
- Imagine que o elevador que sempre deveria subir e descer resolve emperrar bem no dia que você possui um compromisso importante no 49º andar?
- O que vão fazer para resgatar as pessoas presas nesse elevador?




Tratando exceções

-  Situação: Você tenta pagar com o cartão, mas não tem saldo suficiente.
-  Exceção: `SaldoInsuficienteError`
-  Tratamento: O sistema exibe uma mensagem "Saldo insuficiente. Escolha outra forma de pagamento."

Tratando exceções

- Senha Errada ao Fazer Login
-  Situação: Você digita a senha errada três vezes.
-  Exceção: `SenhaIncorretaError`
-  Tratamento: O sistema bloqueia temporariamente a conta e pede recuperação de senha.

Tratando exceções

- Elevador com Capacidade Excedida
-  Situação: Mais pessoas entram no elevador do que ele suporta.
-  Exceção: CapacidadeExcedidaError
-  Tratamento: O elevador exibe um alerta e impede o funcionamento até que o peso seja reduzido.

O que são erros e exceções?

O que são erros e exceções?

- **Erro:** Problema que ocorre durante a execução do código
- **Exceção:** Erro que pode ser tratado pelo programa
- Exemplos comuns de erros em JavaScript:
 - SyntaxError
 - TypeError
 - ReferenceError
 - RangeError

Bloco try...catch

- Usado para capturar erros e evitar que o programa pare de funcionar
- Sintaxe básica:

```
try {  
    // Código que pode gerar um erro  
} catch (erro) {  
    // Tratamento do erro  
}
```

Exemplo prático:

```
try {  
    let x = y + 1; // y não foi definido  
} catch (erro) {  
    console.log("Ocorreu um erro: ", erro.message);  
}
```

Uso do finally

- O bloco finally é executado sempre, independentemente de erro ou não

```
try {  
    console.log("Tentando... ");  
    throw new Error("Erro proposital");  
} catch (erro) {  
    console.log("Erro capturado: ", erro.message);  
} finally {  
    console.log("Isso será executado sempre.");  
}
```

Lançando erros com throw

- Podemos criar nossos próprios erros usando throw

```
function dividir(a, b) {  
    if (b === 0) {  
        throw new Error("Não é possível dividir por zero!");  
    }  
    return a / b;  
}  
  
try {  
    console.log(dividir(10, 0));  
} catch (erro) {  
    console.log("Erro: ", erro.message);  
}
```

Criando e Manipulando Erros Personalizados

- Podemos criar classes de erros personalizados:

Criando e Manipulando Erros Personalizados

```
class ErroPersonalizado extends Error {  
    constructor(mensagem) {  
        super(mensagem);  
        this.name = "ErroPersonalizado";  
    }  
}  
  
try {  
    throw new ErroPersonalizado("Este é um erro personalizado!");  
} catch (erro) {  
    console.log(erro.name + ": " + erro.message);  
}
```

Melhores Práticas

- Sempre tratar erros que podem ser previstos
- Evitar capturar erros genéricos sem necessidade
- Fornecer mensagens de erro claras
- Usar finally para limpar recursos, se necessário

Exercícios Propostos

1. Erro Simples:

- Escreva um código que tente acessar uma variável não definida dentro de um try...catch e exiba uma mensagem personalizada.

2. Divisão Segura:

- Crie uma função que receba dois números e trate o erro caso o denominador seja zero.

3. Validação de JSON:

- Escreva um código que tente fazer um JSON.parse() de uma string inválida e capture o erro.

4. Erro Personalizado:

- Crie uma classe de erro personalizada chamada ErroDeldade e use-a para validar se a idade informada está entre 0 e 120.