

10:18

Quiz- Criando erros

Sobre o bloco catch, é correto afirmar que:

É executado apenas quando ocorre um erro e possibilita a manipulação de um objeto Error

É executado apenas quando ocorrem erros, mas não possibilita a manipulação de um objeto Error

Faz parte da sintaxe do javascript assíncrono

É executado apenas quando não ocorrem erros

Não é obrigatório seu uso após um bloco try

PRÓXIMA PERGUNTA

8:09

Quiz- Criando erros

Qual a composição de um ECMAScript Error?

Nome e linha

Mensagem, nome, linha e opções

Mensagem, nome, linha e call stack

Mensagem e nome

Linha e call stack

FINALIZAR

10:29

Quiz- Criando erros

Selecione a opção que possui o instanciamento de um erro personalizado de forma correta:

const MeuErro = error

const MeuErro = Exception()

const MeuErro = throw

const MeuErro = new Err ()

const MeuErro = new Error()

PRÓXIMA PERGUNTA

8:32

Quiz- Criando erros

Qual a diferença entre os objetos ECMAScript Error e DOMException?

Um se refere a erros de tipo e outro se refere a erros de referência

Não há diferenças

O ECMAScript error é um tipo de erro em tempo de execução e o DOM Exception é um erro referente ao DOM

Ambos são erros referentes à integração do javascript com o HTML

O DOMException é um tipo de erro em tempo de execução e o ECMAScript Error é um erro referente ao DOM

11:50

Quiz- Criando erros

Sobre a declaração try...catch, é correto afirmar que:

Dentro do bloco try são executadas instruções e caso ocorra algum erro durante o processo, ele será tratado no bloco catch

É uma novidade do ECMAScript 6

As instruções de uma função são executadas e, caso sejam bem sucedidas o código passa para o bloco try, e, caso contrário, para o bloco catch

Apenas podem ser utilizados para tratar de DOM Exceptions

Podem ser utilizadas apenas em funções assíncronas

PRÓXIMA PERGUNTA

8:41

Quiz- Criando erros

Sobre a declaração finally, é correto afirmar que:

É executada sempre, independente de erros ocorrerem ou não

É executada apenas quando não ocorrem erros

É executada apenas quando ocorrem erros, atuando como auxiliar da declaração catch

É executada apenas quando ocorrem erros

Não é executada, a menos que seja invocada

PRÓXIMA PERGUNTA

10:51

Quiz- Criando erros

Qual é a grande vantagem do bloco catch?

A possibilidade de ver erros no console

Ela esconde os erros dos usuários

Fazer o uso da declaração finally em conjunto

A possibilidade de manipular o tratamento de um erro com maior flexibilidade

Encerrar a função

PRÓXIMA PERGUNTA

Dado o código

const NovoErro = new Error();

NovoErro.name = "Novo Erro";

NovoErro.message = "Este erro nunca ocorreu"

Qual será o retorno quando ele for lançado?

Este Erro nunca ocorreu

Novo Erro: Este erro nunca ocorreu

Novo Erro: Este erro nunca ocorreu!

Novo Erro

New TypeError: Novo Erro: Este erro nunca ocorreu

PRÓXIMA PERGUNTA

```
return e;
}
}

validaArray();
```

Qual será o retorno?

[]

RangeError

Outro tipo de erro

O tamanho do array é inválido

RangeError: O tamanho do array é inválido

PRÓXIMA PERGUNTA

Quiz- Criando erros

Qual a diferença entre throw e return?

Não há diferença

Throw lança um erro, enquanto o return retorna um valor.

Ambos finalizam uma função, mas o return também lança um erro

Return lança um erro, enquanto throw retorna um valor

Throw não finaliza a função

PRÓXIMA PERGUNTA