Nome: Mister X

Questão 1: Considere o seguinte código em JavaScript:

var x = 5;

if (true)

{

var x = 10;

}

console.log(x); // O que será impresso aqui?

Marcar apenas uma oval.

a) A variável x terá o valor 5 no console.log.

b) A variável x terá o valor 10 no console.log porque var pode ser redeclarada no mesmo escopo.

c) O código resultará em erro, pois var não pode ser redeclarada dentro de um if.

d) O valor de x dentro do if e fora do if será diferente.

Questão 2: Considere o seguinte código em JavaScript: \*

let x = 5;

if (true)

{

let x = 10; console.log(x); // O que será impresso aqui?

}

console.log(x); // O que será impresso aqui?

Marcar apenas uma oval.

a) A variável x terá o valor 10 em ambos os console.log.

b) A variável x terá o valor 5 em ambos os console.log.

c) A variável x terá o valor 10 dentro do bloco if e 5 fora do bloco.

d) O código resultará em erro porque let não permite a declaração de variáveis dentro de blocos.

Questão 3: Considere o seguinte código em JavaScript:

const numero = 10;

console.log(numero); // O que será impresso aqui? numero = 20; // O que acontecerá aqui?

Marcar apenas uma oval.

a) O valor 10 será impresso, mas tentar reatribuir numero resultará em erro.

b) O valor 10 será impresso, e numero será reatribuído para 20 sem problemas.

c) O código resultará em erro na primeira linha porque const não permite atribuições.

d) O código permitirá a reatribuição de numero para 20 e não haverá erro.

Questão 4: Considere o seguinte código em JavaScript: \*

// Declaração de função

function saudacao(nome)

{

return `Olá, ${nome}!`;

}

// Chamada da função console.log(saudacao("Maria")); // O que será impresso aqui?

Marque todas que se aplicam.

a) A função deve ser definida com um nome e pode ser chamada em qualquer parte do código, mesmo antes de sua definição.

b) A função deve ser definida dentro de um bloco de código, como uma função anônima.

c) A função deve ser criada como uma expressão de função e atribuída a uma variável.

d) A função pode ser chamada e usará o valor fornecido como argumento.

e) A função deve ser declarada dentro de outra função para ser utilizada.

Questão 5: Considere o seguinte código em JavaScript: \*

// Função anônima atribuída a uma variável

const saudacao = function(nome)

{

return `Olá, ${nome}!`;

};

console.log(saudacao("Carlos")); // O que será impresso aqui?

Marque todas que se aplicam.

a) A função anônima pode ser chamada apenas depois da sua definição no código.

b) A função anônima precisa ser atribuída a uma variável para ser utilizada.

c) A função anônima pode ser chamada antes da sua definição no código.

d) A função anônima pode ser usada em qualquer parte do código, desde que esteja dentro do escopo onde foi definida.

Questão 6: Considere o seguinte código em JavaScript( arrow function

\*

):

const soma = (a, b) => a + b;

O que a função soma faz?

Questão 7: Considere o seguinte código em JavaScript: \*

const pessoa = {

nome: "Ana",

idade: 30

};

O que é o pessoa neste código?

Questão 9: (Tipo ENADE)

Você está desenvolvendo uma aplicação em JavaScript e precisa criar funções para realizar diferentes operações. Um dos requisitos é criar uma função que calcula a área de um círculo com base no raio fornecido.

Situação:

Um desenvolvedor novo em sua equipe está tentando entender como criar e utilizar funções em JavaScript. Ele precisa saber como declarar uma função, como passar parâmetros para ela e como retornar um valor. Para garantir que a função esteja correta e bem documentada, quais das seguintes práticas são recomendadas?

Marcar apenas uma oval.

a) Declare a função com um nome claro e descritivo que indique sua finalidade, como calcularAreaCirculo.

b) Documente a função explicando o tipo de parâmetros que ela espera e o valor que será retornado.

c) Use variáveis globais para armazenar os parâmetros e resultados da função para evitar a necessidade de passar parâmetros.

d) Inclua exemplos de como chamar a função e quais resultados são esperados para diferentes entradas.

e) Utilize funções anônimas para definir o comportamento, mesmo que não sejam reutilizáveis.

Questão 10: (Tipo ENADE):

Você está desenvolvendo uma aplicação para gerenciar uma coleção de livros em uma biblioteca digital. Sua tarefa é manipular um array de números que representa o número de páginas de cada livro.

Dado o seguinte array de números de páginas:

const paginas = [150, 320, 200, 450, 180];

13. 2 pontos

Para melhorar a experiência dos usuários, você precisa realizar as

\*

seguintes operações:

Adicionar um novo livro com 275 páginas à lista.Remover o livro com o maior número de páginas