

JOY - CASOS 1:

1) Você está trabalhando em um projeto de desenvolvimento web e foi encarregado de escrever um script em JavaScript para implementar uma funcionalidade específica. O script precisa calcular a média de um conjunto de números armazenados em um array e exibir essa média na tela. Para cumprir essa tarefa, é essencial que você compreenda e utilize corretamente os símbolos da sintaxe de JavaScript, como parênteses, chaves, operadores de atribuição e de comparação.

Enquanto escreve o código, você se depara com a necessidade de entender a diferença entre os símbolos `==` e `===`, bem como o uso correto de `;` no final das instruções. Além disso, é fundamental saber quando usar colchetes `[]` para acessar elementos de um array e como as chaves `{}` são utilizadas para definir o corpo de funções e estruturas de controle de fluxo.

O desafio inclui garantir que as funções e loops do seu script estejam sintaticamente corretos, evitando erros comuns, como a confusão entre `=` (atribuição) e `==` (comparação), ou o esquecimento de chaves `{}` em estruturas condicionais e de repetição. A precisão na utilização desses símbolos é essencial para o funcionamento adequado do seu script.

No contexto do cenário acima, qual das seguintes opções representa a melhor maneira de calcular a média dos números em um array e exibi-la, levando em conta o correto uso dos símbolos da sintaxe de JavaScript?

a) ``var total = 0; for(i = 0; i < numeros.length; i++) { total += numeros[i]; }console.log('A média é: ' + total / numeros.length);``

b) ``var total; for(i == 0; i < numeros.length; i++) { total += numeros[i]; }console.log('A média é: ', total / numeros.length);``

c) ``var total = 0; for(i = 0; i <= numeros.length; i++) { total = total +numeros[i]; }; console.log('A média é: ' + total / numeros.length);``

d) ``var total = 0; for(i = 0; i < numeros.length; i++) { total = total + numeros[i];} console.log('A média é: ' + total / numeros.length);``

<https://jseditor.io/>

// Defina o array de números

var numeros = [10, 20, 30, 40, 50]; // Você pode adicionar mais números conforme necessário

// Inicialize a variável para armazenar o total

var total = 0;

// Loop para somar todos os números no array

```
for (var i = 0; i < numeros.length; i++) {  
    total = total + numeros[i];  
}
```

// Calcule a média e exiba o resultado

console.log('A média é: ' + (total / numeros.length));

2) O objetivo agora é criar uma função simples em JavaScript que converta a temperatura de Celsius para Fahrenheit.

Este desafio serve como uma introdução básica às operações matemáticas e à sintaxe do JavaScript. A função deve aceitar um número representando a temperatura em Celsius e retornar a temperatura convertida em Fahrenheit. Para realizar esta conversão, você precisa aplicar a fórmula: $(\text{Fahrenheit} = \text{Celsius} \times \frac{9}{5} + 32)$. Este problema é ideal para entender como declarar funções, realizar operações matemáticas básicas e retornar valores em JavaScript. Além disso, é uma boa oportunidade para praticar a declaração de variáveis e a compreensão de como a linguagem lida com tipos de dados numéricos.

É importante para essa tarefa escrever um código claro e preciso, demonstrando compreensão dos fundamentos da linguagem. A precisão na utilização de operadores matemáticos e na estrutura da função são essenciais para a correta implementação desta simples, mas importante tarefa de programação.

Qual das seguintes funções em JavaScript converte corretamente a temperatura de Celsius para Fahrenheit?

a) ``function celsiusParaFahrenheit(celsius) { var fahrenheit = celsius * (9 / 5); return fahrenheit + 32; }``

b) ``function celsiusParaFahrenheit(celsius) { return celsius * 9 / 5 + 32; }``

c) ``function celsiusParaFahrenheit(celsius) { var fahrenheit = (celsius * 9) / 5; return fahrenheit + 32; }``

d) ``function celsiusParaFahrenheit(celsius) { return (celsius * 9) / 5 + 32; }``

<https://jseditor.io/>

`// Definição da função que converte Celsius para Fahrenheit`

```
function celsiusParaFahrenheit(celsius) {  
    return celsius * 9 / 5 + 32;  
}
```

`// Exemplo de uso da função`

```
var temperaturaCelsius = 25; // Exemplo de temperatura em Celsius  
var temperaturaFahrenheit = celsiusParaFahrenheit(temperaturaCelsius);
```

`// Exibir o resultado no console`

```
console.log('Temperatura em Fahrenheit: ' + temperaturaFahrenheit);
```

3) Neste cenário, você tem a tarefa de criar uma função em JavaScript que gerencie um pequeno inventário de produtos. A função deve ser capaz de adicionar um novo produto ao inventário, que é representado por um objeto. Cada produto tem um ``id``, ``nome`` e ``quantidade``. Sua função deve aceitar três parâmetros - `id`, `nome` e `quantidade` - e adicionar um novo objeto de produto ao inventário se o `id` não existir. Se o `id` já existir, a função deve atualizar a quantidade do produto existente. Este desafio testa sua habilidade de trabalhar com objetos em JavaScript, incluindo a declaração de variáveis, a atribuição de valores e a manipulação de propriedades de objetos.

A chave para resolver este problema está na compreensão de como criar e manipular objetos em JavaScript, além de utilizar estruturas condicionais para verificar a existência de um `id` no inventário. A correta utilização de variáveis e a manipulação de objetos são fundamentais para

implementar esta função de forma eficiente e eficaz.

Dado um objeto `inventario` que armazena produtos, qual das seguintes funções em JavaScript implementa corretamente a lógica descrita acima?

a) ``function adicionarAoInventario(inventario, id, nome, quantidade) { if(inventario[id]) { inventario[id].quantidade += quantidade; } else {inventario[id] = { nome: nome, quantidade: quantidade }; } }`

b) ``function adicionarAoInventario(inventario, id, nome, quantidade) { if(!inventario[id]) { inventario[id] = { nome: nome, quantidade: quantidade };}else { inventario[id].quantidade += quantidade; } }`

c) ``function adicionarAoInventario(inventario, id, nome, quantidade) {inventario[id] = inventario[id] || {}; inventario[id].nome = nome;inventario[id].quantidade = (inventario[id].quantidade || 0) + quantidade; }`

d) ``function adicionarAoInventario(inventario, id, nome, quantidade) {inventario[id] = inventario[id] || { nome: nome, quantidade: 0 };inventario[id].quantidade += quantidade; }`

<https://jseditor.io/>

```
function adicionarAoInventario(inventario, id, nome, quantidade) {  
  if (inventario[id]) {  
    inventario[id].quantidade += quantidade;  
  } else {  
    inventario[id] = { nome: nome, quantidade: quantidade };  
  }  
}
```

```
let inventario = {};
```

```
// Adiciona e atualiza produtos  
adicionarAoInventario(inventario, '001', 'Caneta', 10);  
adicionarAoInventario(inventario, '001', 'Caneta', 5);  
adicionarAoInventario(inventario, '002', 'Lápis', 20);
```

```
// Exibe o inventário formatado  
console.log('Inventário (JSON):');  
console.log(JSON.stringify(inventario, null, 2));
```

4) Você foi incumbido de configurar um ambiente de desenvolvimento básico para um novo projeto em JavaScript. Isso inclui a instalação de um editor de texto adequado (como Visual Studio Code, Atom ou Sublime Text), a configuração de um interpretador JavaScript (como Node.js), e a criação de um arquivo JavaScript simples para testar o ambiente. Seu primeiro teste consiste em escrever um script que imprima "Olá, Mundo!" no console. Este desafio é uma introdução fundamental ao desenvolvimento de software, pois permite que você entenda como configurar e utilizar ferramentas essenciais para a programação em JavaScript.

A tarefa requer conhecimento básico sobre instalação de software e configuração de um ambiente de desenvolvimento, além da habilidade de escrever e executar um script simples em JavaScript. A

precisão na execução desses passos iniciais é fundamental para garantir um ambiente de desenvolvimento funcional e pronto para projetos mais complexos.

Qual das seguintes linhas de código em JavaScript imprimirá corretamente "Olá, Mundo!" no console, assumindo que o ambiente de desenvolvimento foi configurado corretamente?

- a) ``window.print("Olá, Mundo!");``
- b) ``console.log("Olá, Mundo!");``
- c) ``document.write("Olá, Mundo!");``
- d) ``alert("Olá, Mundo!");``

PREMISSA: NODE.JS

helloWorld.js

```
-----  
// helloWorld.js  
console.log("Olá, Mundo!");  
-----
```

node helloWorld.js

5) Neste desafio, você deve escrever um script JavaScript que manipule informações de produtos em uma loja. Cada produto possui um ``nome``, ``preço`` e ``quantidade`` em estoque. Sua tarefa é criar variáveis para cada um desses atributos, atribuir valores a elas e, em seguida, calcular o valor total do estoque de um produto (preço multiplicado pela quantidade). Este exercício prático testa sua capacidade de declarar variáveis em JavaScript, atribuir valores a elas e realizar operações matemáticas básicas. Além disso, você deve garantir que as variáveis sejam nomeadas de maneira clara e representativa, facilitando a compreensão e a manutenção do código.

Este desafio é fundamental para entender como as variáveis funcionam em JavaScript, incluindo a escolha dos tipos de dados corretos (por exemplo, strings para nomes e números para preços e quantidades) e a aplicação de operações matemáticas para calcular valores. A precisão na manipulação dessas variáveis é fundamental para obter o resultado correto.

Considere que você está criando variáveis para um produto em JavaScript. Qual das seguintes sequências de código declara corretamente as variáveis e calcula o valor total do estoque do produto?

- a) ``var nomeProduto = "Caneta"; var precoUnitario = 1.50; var qtdEstoque = 100; var totalValorEstoque = precoUnitario + qtdEstoque;``
- b) ``var produto = "Caneta"; var valor = 1.50; var estoque = 100; var totalEstoque = valor + estoque;``
- c) ``var nome = "Caneta"; var preco = 1.50; var quantidade = 100; var valorTotalEstoque = preco * quantidade;``
- d) ``var nome = "Caneta"; var precoUnitario = "1.50"; var quantidadeEmEstoque = "100"; var valorTotalEstoque = parseFloat(precoUnitario) * parseInt(quantidadeEmEstoque);``

<https://jseditor.io/>

```
// Declaração das variáveis  
var nome = "Caneta";           // Nome do produto  
var preco = 1.50;              // Preço unitário do produto  
var quantidade = 100;          // Quantidade em estoque
```

```
// Cálculo do valor total do estoque
var valorTotalEstoque = preco * quantidade;

// Impressão do valor total do estoque formatado com duas casas decimais
console.log(quantidade + " " + nome + " de " +preco);
console.log("Valor Total do Estoque: R$ " + valorTotalEstoque.toFixed(2));
```