

1) Uma pessoa está desenvolvendo a tela de boas vindas de uma aplicação feita com JavaScript e criou o seguinte código:

```
let mensagemDeBoasVindas = 'Boas vindas ao App';
alert('mensagemDeBoasVindas');
```

Com base nisso, analise as afirmações abaixo e marque apenas as verdadeiras em relação ao código mostrado acima.

Selecione 2 alternativas corretas:

- a) Ao executar o programa, uma caixa será exibida com a mensagem mensagemDeBoasVindas.
- b) Nenhuma mensagem será exibida ao executar o programa.
- c) Ao executar o programa, uma caixa será exibida com a mensagem Boas vindas ao App.**
- d) O problema nesse código é que a variável mensagem DeBoasVindas não está sendo utilizada corretamente ao ser exibida na função alert.**

2) Em um projeto prático dentro do curso "Iniciando com JavaScript", você está desenvolvendo um jogo no qual os usuários têm que adivinhar o número secreto. Você escreveu um código que mostra uma mensagem de boas-vindas aos jogadores, pedindo que escolham um número entre 1 e 10, e apresenta uma resposta se eles acertarem, conforme a sequência a seguir:

```
alert('Boas-vindas ao jogo do número secreto')
let chute = prompt('Escolha um número entre 1 e 10')

let numeroSecreto = 4

if (chute == numeroSecreto) {
    alert('Acertou')
}
```

Qual variável você precisaria alterar para mudar o número secreto a cada rodada do

jogo? **Selecione 1 alternativa correta:**

- a) alert('Boas vindas ao jogo do número secreto')alert('Boas vindas ao jogo do número secreto')
- b) if (chute == numeroSecreto){

c) let numeroSecreto = 4

d) let chute = prompt('Escolha um número entre 1 e 10')

3) Luis é apaixonado por tecnologia e está dando seus primeiros passos no mundo da programação. Para praticar lógica de programação, ele decidiu desenvolver um sistema de login de usuário único em que é necessário apenas informar a senha para ver o conteúdo do restante do projeto.

No entanto, ao testar o seu projeto, percebeu que não importava a senha passada para o sistema, o acesso sempre era permitido.

Sabendo que você também está estudando lógica de programação, Luis pediu sua ajuda para que vocês consigam encontrar o bug juntos. Para isso, ele disponibilizou o código que escreveu:

```
let senhaDoSistema = "senhaTeste!";

let senha = prompt("Digite a senha do sistema:");

if (senha = senhaDoSistema){
    alert("Acesso ao sistema garantido");
}
```

De acordo com o código disponibilizado, qual é o problema?

Selecione 1 alternativa correta:

a) O código apresenta um bug por conta dos nomes das variáveis, que são muito parecidos e parecem se tratar da mesma coisa.

b) O código está funcionando perfeitamente, e o problema é o navegador de internet. Para o sistema funcionar, basta mudar o navegador.

c) O código apresenta um bug dentro da estrutura condicional proposta, pois o operador de comparação não está sendo utilizado.

d) O código apresenta um bug porque as funções alert e prompt estão invertidas.

4) Você é uma pessoa desenvolvedora de software em uma clínica médica chamada Médica Voll. A empresa deseja que você crie um pequeno jogo de adivinhação para seus pacientes enquanto eles estão na sala de espera. Então, você decide criar um jogo do número secreto usando JavaScript, conforme a seguir:

```
alert('Bem-vindo ao jogo do número secreto')
let chute = prompt('Escolha um número entre 1 e 10')

let numeroSecreto = 4

console.log(chute == numeroSecreto)
if (chute == numeroSecreto) {
    alert('Acertou')
} else {
    alert('O número secreto era ' + numeroSecreto)
}
```

Dada essa estrutura básica de código que você criou, como você modificaria a mensagem do alerta para incluir o número que o usuário escolheu, caso não acertasse o número secreto?

Escolha a alternativa correta:

Obs: Note que o "+" usado serve como uma alternativa ao `> template de strings já apresentado para realizarmos concatenação de strings/caracteres/valores de variáveis no JavaScript.

Selecione 1 alternativa correta:

a) alert('O número secreto era numeroSecreto e você escolheu chute')

b) alert('O número secreto era ' + numeroSecreto + ', mas você escolheu ' +

chute); c) alert('O número secreto era ' + numeroSecreto + chute)

d) alert('O número secreto era ' + numeroSecreto, ' e você escolheu ' + chute)

5) Você é uma pessoa desenvolvedora de software que desenvolve aplicações para o Detran, que é o órgão responsável pela fiscalização e regulamentação do trânsito, habilitação de condutores, registro e licenciamento de veículos no Brasil.

A idade mínima para dirigir no Brasil é 18 anos. Sabendo disso, uma equipe criou o seguinte programa:

```
let idade = prompt("Digite sua idade:");

if (idade >= 18) {
    console.log("Você é maior de idade.");
} else {
    console.log("Você é menor de idade.");
}
```

Com essa base, analise as afirmações abaixo e marque apenas as verdadeiras.

Selecione 2 alternativas corretas:

- a) Caso a idade seja igual a 18, a mensagem do console que será exibida será: Você é menor de idade.
 - b) Caso a idade seja igual a 18, a mensagem do console que será exibida será: Você é maior de idade.**
 - c) Caso a idade seja menor que 18, a mensagem do console que será exibida será: Você é menor de idade.**
 - d) Caso a idade seja maior que 18, a mensagem do console que será exibida será: Você é menor de idade.
- 6) Saber trabalhar com loops de repetição, como o while, é essencial na programação! Os loops permitem automatizar tarefas repetitivas e lidar com grandes quantidades de dados de forma eficiente.

Contudo, é importante usá-los com cuidado para evitar loops infinitos (quando a condição nunca se torna falsa - o que pode travar o programa).

É recomendado sempre ter uma lógica que faça com que a condição eventualmente se torne falsa para que o loop termine corretamente.

Abaixo, temos um código de exemplo:

```
let contador = 1;

while (contador < 4) {
    console.log('Executando a iteração ' + contador);
    contador = contador + 1;
}
```

Ao executar código, podemos afirmar que:

Selecione 2 alternativas corretas:

- a) Um erro será exibido no console indicando que o contador não está definido.
- b) O resultado final será a impressão das mensagens "Executando a iteração 1", "Executando a iteração 2", "Executando a iteração 3" e "Executando a iteração 4" no console.
- c) while (contador < 4){ } começa o loop while. Ele continuará repetindo o bloco de código entre as chaves {} enquanto a condição contador < 4 for verdadeira.**
- d) O resultado final será a impressão das mensagens "Executando a iteração 1", "Executando a iteração 2" e "Executando a iteração 3" no console.**

7) Willian está iniciando no mundo da programação e recentemente descobriu que existe uma maneira de executar um bloco de comandos repetidamente enquanto uma condição pré estabelecida não for satisfeita.

Animado com as possibilidades, ele decidiu treinar a utilização da estrutura while() em um projeto pessoal, de cálculo de médias aritméticas. No entanto, acabou se deparando com um loop infinito, e não conseguiu descobrir o por quê.

Sabendo que você também está estudando lógica de programação, Willian pediu sua ajuda para localizar onde está o erro em seu código:

```
let qtdNumeros = prompt('Digite a quantidade de números para o cálculo da média:');
let soma = 0;
let contador = qtdNumeros;

while(contador > 0){
    let numero = parseInt(prompt('Digite o número:'));
    soma += numero;
}

let media = soma / qtdNumeros;

console.log(media);
```

Qual alteração você poderia sugerir a Willian para que o código tenha o resultado esperado?

Selecione 1 alternativa correta:

a) Quando utilizamos loops com contadores, precisamos garantir que um dia eles cheguem até a condição de parada do loop. No código acima, o contador não tem seu valor alterado e para resolver o loop infinito basta decrementá-lo a cada iteração, adicionando a linha **contador-- dentro do loop.**

b) O código apresenta loop infinito porque a condição de parada não foi definida corretamente. Para o programa funcionar como esperado, é necessário alterar o código de maneira que:

```
while(contador > qtdNumeros){
    //Código omitido
}
```

c) Quando utilizamos loops com contadores, precisamos garantir que um dia eles cheguem até a condição de parada do loop. No código acima, o contador não tem seu valor alterado e para resolver o loop infinito basta **incrementá-lo** a cada iteração, adicionando a linha **contador++ dentro do loop.**

d) O código apresenta loop infinito porque as variáveis não chegam ao valor da condição de parada do laço de repetição. Para resolver isso, basta adicionar a linha **qtdNumeros++ dentro do loop.**

8) A função Math.random em JavaScript gera um número decimal aleatório entre 0 (inclusivo) e 1 (exclusivo) de forma pseudo aleatória. Isso significa que os números gerados podem estar em

qualquer lugar entre 0 (inclusive) e quase 1 (exclusivo), com uma precisão de até 16 casas decimais. A função Math.random usa um valor interno inicial que é geralmente baseado no horário atual do sistema, gerando números pseudo aleatórios.

Confira abaixo alguns exemplos:

```
0.5248738910328501  
0.08458620904957355  
0.9347284927568912
```

Sabendo disso, analise as alternativas abaixo e marque apenas a

verdadeira: **Seleciona 1 alternativa correta:**

- a) Para gerar um número entre 1 e 3, podemos usar o código: let numeroAleatorio1a3 = parseInt(Math.random() * 3) + 1;.
 - b) Para gerar um número entre 1 e 3, podemos usar o código: let numeroAleatorio1a3 = parseInt(Math.random() * 4);.
 - c) Para gerar um número entre 1 e 3, podemos usar o código: let numeroAleatorio1a3 = parseInt(Math.random() * 3);.
 - d) nenhuma das alternativas
- 9) Aline está dando seus primeiros passos na programação utilizando o JavaScript e recentemente descobriu uma funcionalidade que achou fascinante: **o operador ternário.**

Ela achou muito interessante a possibilidade que o operador traz para diminuir a quantidade de código escrito em algumas ocasiões e decidiu aplicar na prática, em um de seus projetos pessoais. No entanto, ao tentar fazer essa refatoração, acabou se esquecendo da sintaxe do operador ternário.

Sabendo que você estudou sobre o operador recentemente, ela pediu sua ajuda e mostrou o código que pretende refatorar:

```
let palavraPessoa = "";

if(quantidadePessoas == 1){
    palavraPessoa = "pessoa";
}else{
    palavraPessoa = "pessoas"
}
```

Como você poderia mostrar à Aline um exemplo de como transformar esse trecho de código em outro que tenha o mesmo comportamento, mas utilizando o operador ternário?

Seleccione 1 alternativa correta:

- a) let palavraPessoa = quantidadePessoas ? "pessoa" : "pessoas";
- b) let palavraPessoa = quantidadePessoas == 1 ? "pessoa" ; "pessoas";
- c) let palavraPessoa = if(quantidadePessoas == 1) ? "pessoa" : "pessoas";
- d) let palavraPessoa = quantidadePessoas == 1 ? "pessoa" :
"pessoas";**

10) Suponha que você esteja trabalhando em um sistema de vendas, no qual o usuário informa o total de itens e o valor unitário. Com base em algumas regras, o sistema aplica um desconto de 5% sobre o valor total.

As regras são as seguintes:

- Inclusão de 10 ou mais itens; e
- Valor unitário igual ou superior a R\$ 100,00.

O valor deverá ser exibido num **alert** e depois a função **reiniciar()** deverá ser chamada.

Os seguintes códigos foram escritos para implementar essa funcionalidade e a funcionalidade de reiniciar os valores:

```
function calcular() {  
  
    let quantidade = parseInt(document.getElementById('quantidade').value);  
    let valor = parseFloat(document.getElementById('valor').value);  
  
    let total = quantidade * valor;  
  
    if (quantidade >= 10 || valor >= 100) {  
        total = total - (total / 100 * 5);  
    }  
  
    alert(`Valor total: ${total}`);  
}  
  
function reiniciar() {  
    quantidade.value = '';  
    valor.value = '';  
}
```

Entretanto, os campos não estão sendo limpos após o cálculo do valor total. Escolha TODAS as alternativas que indicam os problemas de lógica do código anterior:

Selecione 2 alternativas corretas:

- a) Na função `reiniciar()`, o código deveria ter sido declarado com `innerHTML` ao invés de `value`.
- b) Faltou declarar um bloco `else`, que é obrigatório, para cada bloco `if`, e colocar para chamar a função `reiniciar()` dentro dele.
- c) Necessário ajustar a função `reiniciar()`, pois as variáveis `quantidade` e `valor` foram declaradas dentro da função `calcular()`.
- d) A função `reiniciar()` não está sendo chamada após a exibição do `alert`.