

Aula 4

Botão de jogar novamente

► Unidade

Técnicas computacionais: refletindo sobre Inteligência Artificial na escola

Questão 1 – Inicializando contagem

Luana desenvolveu o seguinte código:

```
````javascript
let contagem = 0;
function iniciaContagem() {
 contagem++;
 console.log("Contagem atual: " + contagem);
}
```

Com base na função **`iniciaContagem`**, qual das alternativas a seguir melhor descreve uma função **`reiniciaContagem`**, que reinicia a contagem e mantém as outras funcionalidades do código?

a) 

```
````
function reiniciaContagem() {
    contagem = 0;
    console.log("Contagem reiniciada");
}
```

b)

```
````
function reiniciaContagem() {
 contagem = 0;
 iniciaContagem();
}
```

c) ```

```
function reiniciaContagem() {
 console.log("Contagem atual: " + contagem);
}
```

d) ```

```
function reiniciaContagem() {
 contagem++;
 console.log("Contagem reiniciada: " + contagem);
}
```

**Alternativa A**, correta. Essa alternativa reinicia a variável contagem para 0 e informará a pessoa usuária que a contagem foi reiniciada. Ela não interfere nas operações da função *iniciaContagem*.

**Alternativa B**, incorreta. Essa função reinicia a contagem para 0, mas imediatamente chama *iniciaContagem()*, que incrementa a contagem em 1. Isso contradiz a ideia de uma reinicialização, pois a contagem não permanece em 0 após a execução da função *reiniciaContagem*. A função deveria apenas resetar a contagem para 0 sem aumentá-la.

**Alternativa C**, incorreta. Essa função apenas exibe a contagem atual e não realiza nenhuma ação para reiniciar ou modificar o valor da variável contagem. Para uma função *reiniciaContagem* ser correta, ela precisa efetivamente reiniciar o valor da contagem para 0, o que não é feito.

**Alternativa D**, incorreta. Essa função incrementa a contagem em vez de reiniciá-la. Nessa função, a contagem é aumentada, indo na direção oposta do que é esperado para uma reinicialização.

## Questão 2 – Uma lista de passos

Anderson criou uma lista com o passo a passo necessário para incluir um botão em seu projeto de aceleração. Observe:

No HTML:

- Adicionar um botão com a tag **<button>**;
- Não esquecer de adicionar uma \_\_\_\_\_ descritiva ao botão;

No JavaScript:

- Criar uma \_\_\_\_\_ para o botão;
- Utilizar o método \_\_\_\_\_ para selecionar a classe;
- Não esquecer de que para selecionar uma classe, usamos aspas e ponto antes dela;
- Adicionar um ouvinte ao botão criado usando \_\_\_\_\_;
- Criar uma função que tenha um nome descritivo do que o botão faz.

Qual a ordem correta das palavras faltantes na lista de Anderson?

**query.Selector**

**classe**

**addEventListener**

**variável**

Escreva a sequência correta de blocos nas linhas a seguir:

**Sequência correta:** classe | variável | query.Selector | addEventListener

**Comentário:** primeiro, é preciso definir uma classe no HTML para identificar o botão. Em seguida, no JavaScript, ele deve criar uma variável para representar esse botão. O *querySelector* é usado para selecionar o botão pela classe definida no HTML. Por fim, *addEventListener* é aplicado à variável para adicionar uma função que define o comportamento do botão quando clicado. Essa sequência garante que o botão seja não apenas criado, mas também funcional.

### Questão 3 – Adicionando evento ao botão

O botão que Alex está desenvolvendo gerou um erro. Ele conseguiu identificar que, dentro da função `mostraResultado`, faltou adicionar o evento para a constante `botaoRetornar`. Porém, ele não sabe qual a estrutura e sintaxe correta que deve utilizar para corrigir isso.

Ordene **todos os blocos** abaixo e monte o código que resolve o problema encontrado por Alex.

`.addEventListener`

`const`

`botaoRetornar`

`perguntaNovamente);`

`("click",`

Escreva a sequência correta de blocos nas linhas a seguir:

**Sequência correta:** `const | botaoRetornar | .addEventListener | ("click", | perguntaNovamente);`

**Comentário:** queremos que esse botão apareça na caixa de resultado quando o resultado for exibido. Por isso, vamos adicionar a função `perguntaNovamente` à função `mostraResultado()`. Porém, como é um botão, precisamos fazer esse botão funcionar, por isso vamos adicionar um ouvinte no botão de retornar. Em `botaoRetornar`, adicionamos `addEventListener()`. Dentro do `addEventListener()`, precisamos especificar o que estamos esperando, que é o `click`, e depois chamar a função `perguntaNovamente()`.