

Aula 5

Criando botões de alternativa

► Unidade

Funções: criando uma missão sobre Inteligência Artificial

Questão 1 – Sobre o loop *for*

Nesta aula, utilizamos um loop **for** para percorrer uma lista de alternativas. Sobre o uso desse loop, é correto afirmar que:

- a) Após percorrer os três primeiros elementos da lista, o loop para.
- b) Declaramos uma constante que percorre a lista e para cada elemento da lista executamos as instruções entre chaves.
- c) Declaramos uma constante que percorre a lista e para cada elemento da lista imprimimos seu nome na tela.
- d) Devemos evitar o uso de loops, pois ele deixa o código mais lento. O ideal é chamarmos cada item da lista individualmente, passando um índice.

Alternativa A, incorreta. Para que isso aconteça, é necessário alterar o loop *for* e incluir uma condição de parada que seja igual a 3. No nosso caso, o loop deve percorrer a lista inteira.

Alternativa B, correta. Essa é uma forma simples de interpretar o loop. Outras estruturas de loop, embora façam o mesmo, não deixam essa leitura tão evidente.

Alternativa C, incorreta. Podemos passar qualquer instrução dentro das chaves, não apenas imprimir nomes.

Alternativa D, incorreta. Loops não estão em desuso e são uma das estruturas mais comuns em programação.

Questão 2 – Criando um botão

Luana é síndica de um novo prédio e está criando um formulário para cadastrar os moradores. Esse formulário precisa de um botão que deverá enviar as informações preenchidas dos moradores para o e-mail dela.

Considere que o botão deverá ter a palavra “Envie” escrito e ordene **todos os blocos** abaixo para criá-lo corretamente.

```
botaoEnviar.
```

```
const botaoEnviar =
```

```
textContent = "Envie";
```

```
document.createElement("button");
```

Escreva a sequência correta de blocos nas linhas a seguir:

Sequência correta: `const botaoEnviar = | document.createElement("button"); | botaoEnviar.textContent = "Envie";`

Comentário: para resolver essa questão, é necessário lembrar o uso do método `createElement` para criar um elemento HTML através da linguagem JavaScript, além de utilizar corretamente o método `textContent` para escrever a palavra “envie” no corpo do botão. Dica: antes de utilizar um método de modificação diretamente no botão, é necessário criá-lo no HTML existente.

Questão 3 – Lista de tarefas

Kleber quer criar uma lista de tarefas da semana. Para isso, ele pensou em criar uma função chamada **mostraTarefa**. Para cada tarefa da lista, ele quer que a função:

- Crie um botão que representará esta tarefa;
- Altere o conteúdo de texto do botão para o nome da tarefa.

Agora, ordene **todos os blocos** abaixo para criar este código.

```
const botaoTarefa =
```

```
for(const tarefa of listaTarefas) {
```

```
document.createElement("button");
```

```
function mostraTarefas() {
```

```
botaoTarefa.textContent = tarefa; }}
```

Escreva a sequência correta de blocos nas linhas a seguir:

Sequência correta: `function mostraTarefas() { | for(const tarefa of listaTarefas) { | const botaoTarefa = | document.createElement("button"); | botaoTarefa.textContent = tarefa; }}`

Comentário: O objetivo deste exercício é reforçar o uso do loop para verificar, de forma individual, cada um dos elementos de uma lista e, a partir disso, criar uma instância de um botão para cada elemento da lista, além de alterar o conteúdo de texto utilizando o método `textContent`. Dica: o elemento botão criado deve ser armazenado na constante `botaoTarefa`. Além disso, todas as alterações devem estar inseridas dentro de uma função.