

Aula 9

Adicionando condição de parada

► Unidade

Funções: criando uma missão sobre Inteligência Artificial

Questão 1 – Sobre limpar o conteúdo de uma div

Nesta aula, esvaziamos a `<div>` de alternativas antes de adicionarmos novas para que elas não ficassem todas na tela ao mesmo tempo. Sobre esse processo, escolha a alternativa correta:

- a) O ideal seria deletar a `<div>` atual e criar uma nova.
- b) Podemos inserir um texto vazio na propriedade `textContent` da `<div>`.
- c) Não é possível limpar uma `<div>`, pois o conteúdo da página é estático.
- d) A única forma de fazer isso é inserindo um texto em branco na `<div>`.

Alternativa A, incorreta. Embora isso não seja impossível, não é uma necessidade e cria diversos outros problemas.

Alternativa B, correta. Esse é o método mais simples e que resolve todo o processo de limpeza.

Alternativa C, incorreta. O fato de a página ser estática não significa que não possamos limpar o conteúdo.

Alternativa D, incorreta. Não é a única forma, apesar de ser uma forma válida. É importante que você saiba que, na programação, há muitas formas de fazer as coisas.

Questão 2 – Restaurante cheio

Você é responsável por mapear a quantidade de clientes em um restaurante. A capacidade máxima é 200 pessoas. Considerando essas informações, ordene **todos os blocos** para criar uma verificação da quantidade de clientes. Se a capacidade máxima for atingida, o restaurante deverá impedir a entrada de novos clientes.

`return; }`

`if`

`fecharEntrada();`

`(convidados >= 200) {`

Escreva a sequência correta de blocos nas linhas a seguir:

Sequência correta: `if | (convidados >= 200) { | fecharEntrada(); | return;}`

Comentário: conforme vimos em aula, colocamos a condição dentro de um `if`, chamamos as instruções que queremos executar e passamos o comando `return` para encerrar o programa.

Questão 3 – Elevador

Você está criando um algoritmo para simular o funcionamento de um elevador de um prédio que possui 20 andares, porém, apenas o último andar está ativo. Para isso, o algoritmo precisa:

- Verificar se o andar atual é o último;
- Se for o último, o elevador deverá parar;
- Após a parada, as portas do elevador deverão se abrir;
- E, assim, o algoritmo deve ser encerrado.

Agora, ordene **todos os blocos** que representam corretamente o algoritmo descrito.

`abrePortas();`

`if (atual >=`

`return;}`

`andares.length){`

`paraElevador();`

Escreva a sequência correta de blocos nas linhas a seguir:

Sequência correta: `if (atual >= | andares.length){ | paraElevador(); | abrePortas(); | return;}`

Comentário: neste caso, a palavra *length* representa o comprimento do vetor, que possui 20 andares. Assim que o andar chega a 20, o elevador para (chama a função) e as portas se abrem (chama outra função). Por fim, encerramos o algoritmo com *return*.