
Sessão 11 (05/02/2026) – Módulo 02 – Fundamentos JavaScript – Aula 07 – scope & closures

Questão 1 – Call Stack

O que é a Call Stack no JavaScript?

- a) Um espaço de memória onde objetos e variáveis são armazenados permanentemente.
- b) Uma fila que organiza funções assíncronas aguardando execução.
- c) Uma estrutura de dados que controla a ordem de execução das funções chamadas.
- d) Um mecanismo responsável pela criação do escopo global.
- e) Um tipo especial de heap usado apenas para funções.

Questão 2 – Memory Heap

Sobre o Memory Heap, assinale a alternativa correta:

- a) É responsável por executar funções de forma assíncrona.
- b) Armazena objetos, funções e dados alocados dinamicamente na memória.
- c) Define a ordem de execução das funções no JavaScript.
- d) É limpo automaticamente sempre que uma função termina.
- e) Existe apenas dentro do escopo global.

Questão 3 – Relação entre Call Stack e Memory Heap

Qual afirmação descreve corretamente a relação entre Call Stack e Memory Heap?

- a) Ambos armazenam apenas valores primitivos.
- b) A Call Stack armazena dados, enquanto o Heap executa funções.
- c) A Call Stack controla a execução, enquanto o Heap armazena dados na memória.
- d) O Heap substitui a Call Stack em código assíncrono.
- e) São a mesma estrutura com nomes diferentes.

Questão 4 – Escopo Global

O que caracteriza o escopo global em JavaScript?

- a) Variáveis acessíveis apenas dentro de funções.
- b) Variáveis acessíveis apenas dentro de blocos {}.
- c) Variáveis declaradas dentro de objetos.
- d) Variáveis acessíveis em qualquer parte do código.
- e) Variáveis acessíveis apenas durante a execução da Call Stack.

Questão 5 – Escopo Local (Função)

Sobre o escopo local, é correto afirmar que:

- a) Variáveis locais podem ser acessadas de qualquer função.
- b) Variáveis locais existem apenas dentro da função onde foram declaradas.
- c) Variáveis locais são sempre armazenadas no escopo global.
- d) Variáveis locais não podem acessar variáveis globais.
- e) Variáveis locais só existem enquanto o navegador estiver aberto.

Questão 6 – Escopo Léxico

O que significa dizer que JavaScript possui escopo léxico?

- a) O escopo é definido em tempo de execução.
- b) O escopo depende de quem chama a função.
- c) O escopo é determinado pela posição do código no arquivo.
- d) O escopo muda dinamicamente conforme a Call Stack cresce.
- e) O escopo é definido apenas para funções assíncronas.

Questão 7 – Escopo Léxico (aplicação prática)

Em JavaScript, uma função consegue acessar variáveis externas porque:

- a) O JavaScript copia todas as variáveis para dentro da função.
- b) O escopo léxico permite acesso ao escopo onde a função foi definida.
- c) O this referencia automaticamente o escopo pai.
- d) A Call Stack mantém todas as variáveis vivas.
- e) O Memory Heap elimina variáveis externas após a execução.

Questão 8 – Closure

O que é uma closure?

- a) Uma função que executa apenas uma vez.
- b) Uma função que não possui acesso a variáveis globais.
- c) Uma função que lembra do escopo onde foi criada, mesmo após esse escopo ter sido encerrado.
- d) Uma função criada dentro de um objeto.
- e) Uma função que sempre retorna outra função.

Questão 9 – Closure (condição para existir)

Uma closure ocorre quando:

- a) Uma função é chamada dentro de outra função.
- b) Uma função retorna outra função que utiliza variáveis do escopo externo.
- c) Uma função usa apenas variáveis globais.
- d) Uma função é executada imediatamente após ser criada.
- e) Uma função é armazenada na Call Stack permanentemente.

Questão 10 – Closure e Memória

Por que variáveis de uma closure não são removidas da memória?

- a) Porque estão no escopo global.
- b) Porque a Call Stack impede a remoção.
- c) Porque ainda existem referências ativas a essas variáveis.
- d) Porque o JavaScript nunca libera memória.
- e) Porque closures desativam o garbage collector.