Professores

Celso Giusti

Daniel Filho

Marlon Rodrigues

Funções JavaScript

SENAI

JAVASCRIPT

O que é uma Função?

- Bloco de Código que pode ser definido uma vez e executado várias vezes.
- Funções são usadas para encapsular a lógica, reutilizar o Código e organizar o programa.

JAVASCRIPT

Sintaxe Básica

A forma mais comum de definir uma função em JavaScript é usando a palavra chave **function.**

```
function nomeDaFuncao(paramentro1, parametro2) {
    //corpo da função
    return resultado
}
```

JAVASCRIPT

Parâmetros x Argumentos

Parâmetros: variáveis listadas na DEFINIÇÃO da função;

Argumentos: Valores passados para função quando ela é CHAMADA.

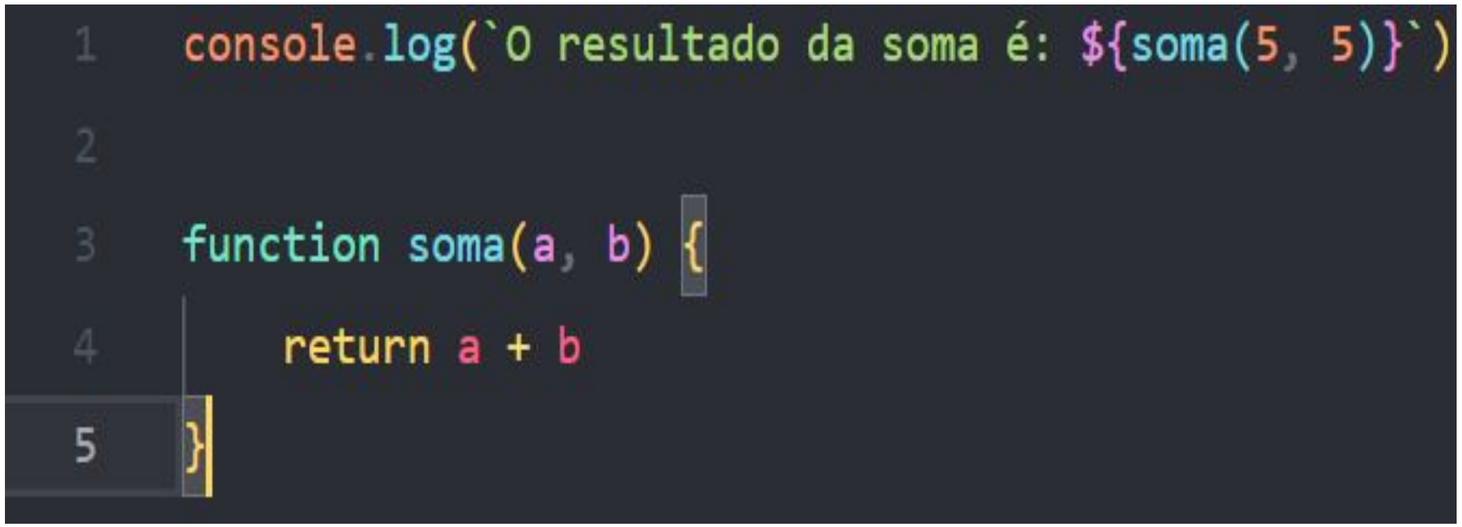
```
function somar(a, b) { // a e b são parâmetros
    return a + b
}

let resultado = somar(5, 3) // 5 e 3 são argumentos
console.log(resultado) // 8
```

AGORA, vamos estudar os tipos de Função.

JAVASCRIPT

Funções Declaradas

Funções definidas usando a palavra-chave **function**. Elas são **hoisted,** ou seja, podem ser chamadas antes da definição do código.



JAVASCRIPT

Funções Expressas

São funções atribuídas a variáveis. Elas não são hoisted, ou seja, só podem ser chamadas após a sua definição.

```
1  const soma = function (a, b) {
2      return a + b
3  }
4  result = soma(4,4)
5  console.log(`O resultado da soma é: ${result}`)
```

JAVASCRIPT

Funções Anônimas

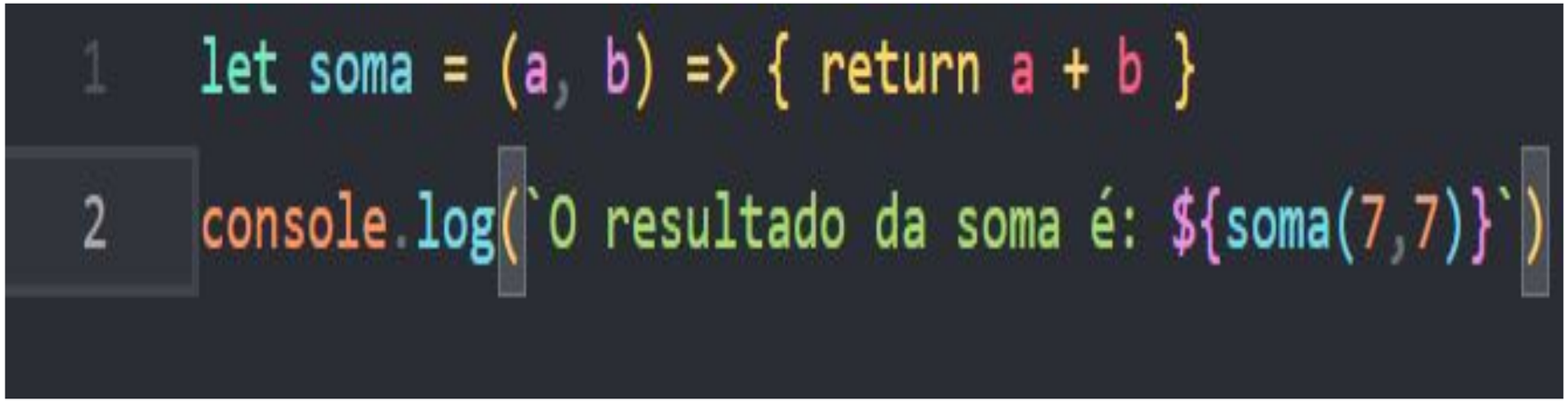
São funções sem nome, geralmente usadas como argumentos para outras funções.

```
1   soma = function(a, b){ // Função sem nome
2       return a + b
3   }
4   console.log(`O resultado da soma é: ${soma}`)
```

JAVASCRIPT

Arrow Functions

Também são funções anônimas, porém, possuem uma sintaxe mais curta.



JAVASCRIPT

Arrow Imediatas IIFE

São funções que são executadas imediatamente após a sua definição.

JAVASCRIPT

Funções de Callbacks

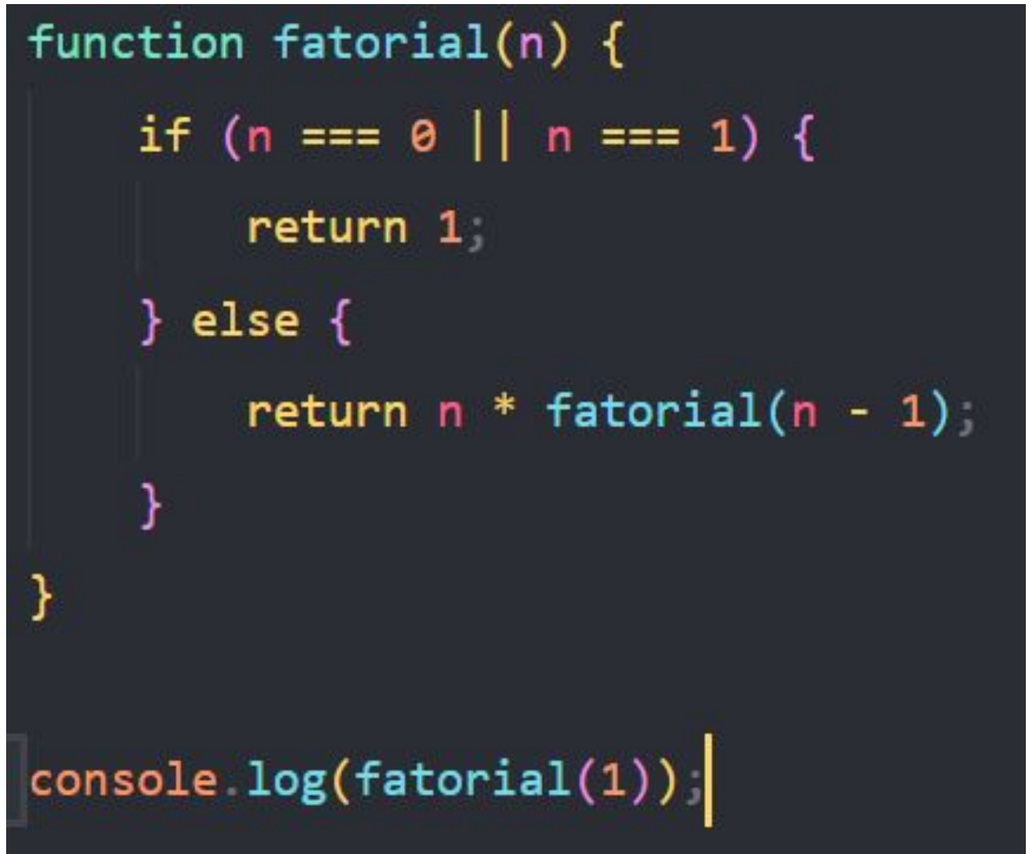
Uma função de call-back é uma função passada como ARGUMENTO para outra função, que é então invocada dentro da função externa.

```
1  function executarCallback(callback){ // função passada como parâmetro
2      callback()
3  }
4  executarCallback(() => {console.log('Callback Executado')})
5  // Arrow Function sendo passada via argumento
```

JAVASCRIPT

Funções Recursivas

São funções que chamam a si mesma.



```
function fatorial(n) {
    if (n === 0 || n === 1) {
        return 1;
    } else {
        return n * fatorial(n - 1);
    }
}

console.log(fatorial(1));
```

JAVASCRIPT

Funções Assíncronas

Uma função assíncrona em JavaScript é uma função que pode realizar operações que demoram um certo tempo para serem concluídas (como requisições de rede, leitura de arquivos, consultas a banco de dados) sem bloquear a execução do código.

Para isso, usamos a palavra-chave async para definir a função como assíncrona e await para esperar a resolução de uma operação assíncrona dentro dela.

JAVASCRIPT

Funções Assíncronas

```
async function buscarDados() {
    console.log("Iniciando a busca...");

    const resposta = await fetch("https://viacep.com.br/ws/18275243/json/");
    const dados = await resposta.json();

    console.log("Dados recebidos:", dados);
}

buscarDados();
console.log("Essa mensagem aparece antes da resposta da API!");
```

JAVASCRIPT

Funções Assíncronas

- async define que a função buscarDados() será assíncrona.
- await fetch(...) pausa a execução da função até que a resposta da API seja recebida.
- await resposta.json() converte a resposta para JSON.
- A linha "Essa mensagem aparece antes da resposta da API!" é exibida antes da resposta porque a chamada da API é assíncrona e não bloqueia o restante do código.

Escola SENAI "Italo Bologna"

Av. Goiás, 139 – Itu/SP

Telefone

(11) 2396-1999

Instagram

@senai.itu

Facebook

/senai.itu

Site

https://sp.senai.br/unidade/itu/