

Laboratório 3

Neste laboratório vamos apresentar a criação de função no JavaScript. Além disso, vamos apresentar o formato JSON (JavaScript Object Notation).

Função:

1 – Crie uma pasta chamada lab03. Nesta pasta, crie um arquivo chamado **funcao** com a extensão **.js (funcao.js)**. Neste arquivo, vamos criar nossa primeira função.

```
function saudacao() {  
    console.log("Função simples que exibe uma mensagem no console");  
}  
  
// Chamando a função  
saudacao();
```

2 – Agora vamos criar uma função que receba dois parâmetros.

```
// Função que recebe dois parâmetros e exibe a soma deles  
function somar(a, b) {  
    console.log("A soma é:", a + b);  
}  
somar(5, 3);
```

3 – Uma função também pode retornar um valor.

```
// Função que recebe dois parâmetros e retorna a soma deles  
function somarRetorno(a, b) {  
    return a + b;  
}  
let resultado = somarRetorno(5, 3);  
console.log("O retorno da soma é:", resultado);
```

4 – Vamos criar uma função anônima no JavaScript. As funções anônimas são muito úteis quando não há necessidade de reutilizar a função ou precisamos passar funções como argumentos.

```
// Função anônima atribuída a uma variável  
let somarAnonimo = function(a, b) {  
    console.log("A soma da função anônima:", a + b);  
};  
somarAnonimo(5, 3);
```

5 – Vamos criar uma *arrow function*. Que são funções anônimas mais curtas e modernas.

```
// Arrow Function que recebe dois parâmetros e retorna a soma deles
let somarArrow = (a, b) => a + b;
resultado = somarArrow(5, 3);
console.log("O retorno da arrow function da soma é:", resultado);
```

6 – Também podemos utilizar uma função como callback em JavaScript.

```
let vetor = [10, 20, 30, 40, 50];

function exibirElemento(elemento, indice) {
  console.log(`Elemento no índice ${indice}: ${elemento}`);
}

function processarCallback(v, callback) {
  console.log("Processando callback dentro da função forEach");
  v.forEach(callback);
}

processarCallback(vetor, exibirElemento);

// Arrow Function como callback
vetor.forEach((elemento, indice) => {
  console.log(`Arrow function exibindo elemento no índice ${indice}: ${elemento}`);
});
```

JSON:

1 – Na pasta lab02, crie um arquivo chamado **json** com a extensão **.js (json.js)**. Neste arquivo iremos trabalhar um pouco com o formato JSON.

Aqui criamos um objeto e o convertemos para string.

```
// Criando um objeto JavaScript
let pessoa = {
  nome: "Adailton Cerqueira",
  idade: 30,
  altura: 1.71,
  cidade: "Salvador",
  hobbies: ["ler", "jogar", "assistir"]
};

console.log("Nome:", pessoa.nome);
console.log("Idade:", pessoa.idade);
console.log("Altura (m):", pessoa.altura);
console.log("Cidade:", pessoa.cidade);
console.log("Hobbies:", pessoa.hobbies.join(", "));

// Convertendo o objeto JavaScript para uma string
let pessoaStr = JSON.stringify(pessoa);
console.log("String JSON:", pessoaStr);
```

2 – Vamos fazer o contrário agora. Iremos pegar uma string no formato JSON e transformaremos em um objeto JavaScript.

```
// String, no formato JSON, com dados de um filme
let filmeStr = '{"titulo": "Inception", "ano": 2010, "diretor": "Christopher Nolan", "generos": ["Ação", "Ficção Científica", "Suspense"], "duracao": 148}';
console.log("String JSON:", filmeStr);

// Convertendo a string para um objeto JavaScript
let filme = JSON.parse(filmeStr);

// Acessando e exibindo as propriedades do objeto JavaScript
console.log("Título:", filme.titulo);
console.log("Ano:", filme.ano);
console.log("Diretor:", filme.diretor);
console.log("Gêneros:", filme.generos.join(", "));
console.log("Duração (min):", filme.duracao);
```