



# Paradigma Funcional

Prof. Dr. Márcio Teixeira Oliveira



# Ementa

Princípios da programação funcional. Características de linguagens de programação funcional. Lambdas. Streams. Callbacks. Programação reativa. Programação assíncrona

# Lambda

Lambda refere-se geralmente a uma arrow function (função de seta), uma sintaxe introduzida no ECMAScript 6 (ES6) para escrever funções de forma mais concisa. São funções anônimas utilizadas principalmente para operações de ordem superior (como map, filter, e reduce) ou callback

# JavaScript Convencional x Lambda

```
<!DOCTYPE html>
<html>
<body>
<h1>Função JavaScript</h1>

<script>
function hello() {
  return alert("Hello World!");
}
</script>
<form>

  <button onclick="hello()">Mensagem</button>
</form>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
<h1>Função JavaScript</h1>

<script>

const hello = () => {
  alert("Hello World!");
}

</script>
<form>

  <button type="button" onclick="hello()">Mensagem</button>
</form>
</body>
</html>
```

# JavaScript Convencional Parâmetros

```
<!DOCTYPE html>
<html>
<body>
<h1>Soma de Dois Números</h1>
Número 1: <input type="text" id="num1"><br>
Número 2: <input type="text" id="num2"><br>
<button type="button" onclick="somar()">Somar</button>
<script>
function somar() {
let n1 = parseFloat(document.getElementById("num1").value);
let n2 = parseFloat(document.getElementById("num2").value);
    return alert("Soma: "+(n1+n2));
}
</script>
</body>
</html>
```

# JavaScript Lambda

```
<!DOCTYPE html>
<html>
<body>
<h1>Soma de Dois Números</h1>
Número 1: <input type="text" id="num1"><br>
Número 2: <input type="text" id="num2"><br>

<button type="button" onclick="somar()">Somar</button>
```

```
<script>
const somar = () => {
  let numero1 = parseFloat(document.getElementById("num1").value);
  let numero2 = parseFloat(document.getElementById("num2").value);
  alert("A soma é: "+(numero1 + numero2));
}
</script>
</body>
</html>
```

```
switch (x) {
  case "texto":
    text = "Off";
    break;
}
```

# JavaScript Filter

Filter é um método que pode ser utilizado em arrays para satisfazer uma consulta específica. Filter possibilita uma visualização personalizada de um array (mas nunca alterá-lo), desde que sua condição lógica seja verdadeira. Sua sintaxe é:

```
const clientes = [{"nome": "José", "idade": 26},  
                  {"nome": "Carlos", "idade": 55},  
                  {"nome": "Manoel", "idade": 5 },  
                  {"nome": "Paula", "idade": 2 }];
```

```
const maioridade = clientes.filter(({ idade }) => idade > 18 );
```

```
console.log(maioridade)
```

# JavaScript Filter Javascript + HTML

```
<html>
<head>
  <title>Clientes Maiores de Idade</title>
</head>
<body>
<h1>Clientes Maiores de Idade:</h1>
<ul id="lista-maioridade"></ul>
```

```
<script>
const clientes = [
  {"nome": "José", "idade": 26},
  {"nome": "Carlos", "idade": 55},
  {"nome": "Manoel", "idade": 5},
  {"nome": "Paula", "idade": 2}
];
```

Função para realizar  
filtro no array

```
const maioridade = clientes.filter(({ idade }) => idade > 18);
```

```
console.log(maioridade);
```

```
</script>
</body>
</html>
```

Instrução Lógica

Dados a serem  
manipulados



# JavaScript + de 1 Filter

```
<html>
<head>
  <title>Clientes Maiores de Idade</title>
</head>
<body>
  <h1>Clientes Maiores de Idade:</h1>
  <ul id="lista-maioridade"></ul>

  <script>
const clientes = [
  {"nome": "José", "idade": 26},
  {"nome": "Carlos", "idade": 55},
  {"nome": "Manoel", "idade": 5},
  {"nome": "Paula", "idade": 2}
];
```

```
const maioridade = clientes
  .filter(({ idade }) => idade > 18)
  .filter(({ idade }) => idade > 30);

console.log(maioridade);

</script>
</body>
</html>
```

# JavaScript ForEach

```
<html>
<head>
  <title>Clientes Maiores de Idade</title>
</head>
<body>
  <h1>Clientes Maiores de Idade:</h1>
  <ul id="lista-maioridade"></ul>

  <script>
const clientes = [
  {"nome": "José", "idade": 26},
  {"nome": "Carlos", "idade": 55},
  {"nome": "Manoel", "idade": 5},
  {"nome": "Paula", "idade": 2}
];
const maioridade = clientes.filter(({ idade }) => idade > 15);
```

```
console.log(maioridade);
```

```
const listaMaioridade = document.getElementById('lista-
maioridade');
```

```
maioridade.forEach(cliente => {
  const item = document.createElement('li');
  item.textContent = `${cliente.nome} - ${cliente.idade} anos`;
  listaMaioridade.appendChild(item);
});
```

Cría um novo elemento list item

Define o conteúdo de texto para  
apresentar dados: Nome e Idade

Adiciona os dados em listaMaioridade

```
</script>
</body>
</html>
```

# JavaScript AddToArray

```
<html>
<head>
  <title>Clientes</title>
</head>
<body>
  <h2>Adicionar novo cliente</h2>
  Nome: <input type="text" id="nomeCliente" >
  Idade: <input type="number" id="idadeCliente" >
  <button onclick="adicionarCliente()">Adicionar Cliente</button>
  <div id="listaClientes"></div>
<br/>
<script>
  let clientes = [];

  function adicionarCliente() {
    const nome = document.getElementById('nomeCliente').value;
    const idade = document.getElementById('idadeCliente').value;
    clientes.push({nome: nome, idade: idade});
    mostrarClientes();
  }
}
```

```
function mostrarClientes() {
  const lista = document.getElementById('listaClientes');
  lista.innerHTML = '<h3>Clientes: </h3>';

  clientes.forEach(cliente => {
    const item = document.createElement('div');
    item.textContent = `Nome: ${cliente.nome}, Idade:
    ${cliente.idade}`;
    lista.appendChild(item);
  });
}

</script>
</body>
</html>
```

# JavaScript Filter + ArrowFunction

```
<html>
<head>
  <title>Clientes Maiores de Idade</title>
</head>
<body>
<h1>Clientes Maiores de Idade:</h1>
<ul id="lista-maioridade"></ul>

<script>
const clientes = [
  {"nome": "José", "idade": 26},
  {"nome": "Carlos", "idade": 55},
  {"nome": "Manoel", "idade": 5},
  {"nome": "Paula", "idade": 2}
];

const verificarIdade = idade => idade > 15;

const maioridade = clientes.filter(({ idade }) => verificarIdade(idade));
```

```
console.log(maioridade);

const listaMaioridade =
document.getElementById('lista-maioridade');

maioridade.forEach(cliente => {
  const item = document.createElement('li');
  item.textContent = `${cliente.nome} -
${cliente.idade} anos`;
  listaMaioridade.appendChild(item);
});

</script>
</body>
</html>
```