

**INSTITUTO FEDERAL**  
Sul-rio-grandense

Câmpus  
Charqueadas

# Programação Web I

Arrays, Matrizes e Objetos em Javascript

Prof. Pablo Werlang  
[pablowerlang@ifsul.edu.br](mailto:pablowerlang@ifsul.edu.br)

# Arrays, Matrizes e Objetos

## Arrays

- **Array:** Vetores, estruturas capazes de armazenar informações em índices numerados consecutivamente.

```
const meuArray = [];  
meuArray[0] = 10;  
meuArray[1] = 20;  
meuArray[2] = 30;  
meuArray[3] = 40;  
meuArray[4] = 50;  
meuArray[5] = 60;  
meuArray[6] = 70;  
meuArray[7] = 80;  
meuArray[8] = 90;  
meuArray[9] = 100;
```

10	20	30	40	50	60	70	80	90	100
0	1	2	3	4	5	6	7	8	9

```
const meuArray = [10,20,30,40,50,60,70,80,90,100];
```

```
> meuArray  
< ▼ (10) [10, 20, 30, 40, 50, 60, 70, 80, 90, 100] ⓘ  
  0: 10  
  1: 20  
  2: 30  
  3: 40  
  4: 50  
  5: 60  
  6: 70  
  7: 80  
  8: 90  
  9: 100  
  length: 10  
  ► [[Prototype]]: Array(0)
```

# Arrays, Matrizes e Objetos

## Arrays

- **Array:** Vetores, estruturas capazes de armazenar informações em índices numerados consecutivamente.
- São muito úteis quando queremos guardar informações em série usando repetições.

```
const meuArray = [];  
for (let i=0 ; i<5 ; i++){  
    meuArray[i] = 10*(i+1);  
}  
console.log(meuArray);
```

```
▼ (5) [10, 20, 30, 40, 50] ⓘ  
  0: 10  
  1: 20  
  2: 30  
  3: 40  
  4: 50  
  length: 5  
  ► [[Prototype]]: Array(0)
```

# Arrays, Matrizes e Objetos

## Arrays

- **Array:** Vetores, estruturas capazes de armazenar informações em índices numerados consecutivamente.
- São muito úteis quando queremos guardar informações em série usando repetições.
- Podemos guardar quaisquer coisas dentro de um array: *números, strings, objetos, funções, arrays, etc.*

```
const arr = [];  
arr[0] = 'Primeira';  
arr[1] = 'Segunda';  
arr[2] = 'Terceira';  
console.log(arr);
```

```
const notas = [];  
let soma = 0;  
const qtd = parseInt(prompt('Quantas notas deseja informar?'));  
for (let i=0 ; i < qtd ; i++){  
    notas[i] = parseFloat(prompt(`Informe a nota do aluno ${ i+1 }:`));  
    soma = soma + notas[i];  
}  
const media = soma / qtd;  
for (let i=0 ; i < qtd ; i++){  
    const desvio = notas[i] - media;  
    console.log(`Desvio do aluno ${ i+1 }: ${ desvio }`);  
}
```

# Arrays, Matrizes e Objetos

## Arrays

### Iterando sobre arrays

- for
- for ... in
- forEach

Propriedade **length**:  
Quantidade de elementos  
preenchidos no array.

```
console.log(arr.length) // 10
```

```
const arr = [10,20,30,40,50,60,70,80,90,100];

for (let i=0 ; i < arr.length ; i++){
    console.log(i, arr[i]);
}

for (let i in arr){
    console.log(i, arr[i]);
}

arr.forEach(func);
function func(e, i) {
    console.log(i, e); // e == arr[i]
}

arr.forEach( (e,i) => {
    console.log(i, e);
});
```

0	10
1	20
2	30
3	40
4	50
5	60
6	70
7	80
8	90
9	100

# Arrays, Matrizes e Objetos

## Arrays

### Funções úteis para usar com arrays

- Push: Insere elementos no fim do array
- indexOf: Procura um valor e retorna o índice.
- Slice: Extraí uma fatia do array.

```
const arr = [];  
arr.push(5);  
arr.push(1, 2, 3.1415, 4, 'texto');  
console.log(arr);  
// [5,1,2,3.1415,4,"texto"]
```

```
const arr = [10,20,30,40,50,60,70,80,90];  
console.log(arr.indexOf(40)); // 3  
console.log(arr.indexOf(100)); // -1
```

```
const arr = [10,20,30,40,50];  
console.log(arr.slice(2));  
// [30,40,50]
```

```
const arr = [10,20,30,40,50];  
console.log(arr.slice(1,4));  
// [20,30,40]
```

```
const arr = [10,20,30,40,50];  
console.log(arr.slice(-3));  
// [30,40,50]
```

# Arrays, Matrizes e Objetos

## Arrays

### Exercício - Parte 1:

- Gere 100 números aleatórios entre 0 e 1000.
  - Use a função random ao lado.
- Insira os números gerados em um array usando o **push**.
- Remova do array os 5 primeiros e os 5 últimos números.
  - Use **slice** para remover os números.
- Mostre o conteúdo do array no console.

```
// número entre 0 e 10  
const num = Math.random() * 10;
```

```
< (90) [729, 174, 29, 638, 145, 892, 105, 379, 508, 66, 961, 245, 88  
6, 84, 821, 34, 75, 9, 632, 21, 665, 696, 567, 436, 321, 621, 749,  
326, 688, 849, 303, 866, 666, 186, 526, 954, 516, 803, 91, 722, 6  
▶ 4, 563, 276, 126, 947, 222, 466, 712, 116, 354, 788, 214, 571, 69  
5, 175, 904, 296, 821, 104, 501, 895, 3, 574, 93, 394, 432, 805, 6  
76, 999, 426, 605, 917, 941, 541, 34, 176, 468, 138, 264, 277, 57  
5, 504, 765, 567, 661, 845, 562, 916, 86, 275]
```

# Arrays, Matrizes e Objetos

## Arrays

### Funções úteis para usar com arrays

- **Split**: Divide uma string em arrays.
- **Join**: Junta um array transformando em uma string.
- **Sort**: ordena os elementos do array.

```
const text = 'Cada palavra está separada por um espaço';
const words = text.split(' ');
console.log(words);
// ["Cada","palavra","está","separada","por","um","espaço"]
```

```
const nums = [23,534,645,7658,56,324,26,54,78];
const rows = [];
for (let i in nums) {
    rows.push(`<div>${ nums[i] }</div>`);
}
document.body.innerHTML = rows.join('');
```

```
let arr = [6,4,3,7,3,10,8,11,9,2];
arr = arr.sort((a,b) => a-b);
console.log(arr); // [2,3,3,4,6,7,8,9,10,11]
```



# Arrays, Matrizes e Objetos

## Arrays

Funções úteis para usar com arrays

- Includes: Verifica se o array possui o valor.
- Reverse: Inverte ordem do array
- Bônus: Pesquise sobre map e filter.

```
const arr = [6,4,3,7,3,10,8,11,9,2];  
console.log(arr.includes(1)); // false  
console.log(arr.includes(10)); // true
```

```
const arr = [1,2,3,4,5];  
arr.reverse();  
console.log(arr); // [5,4,3,2,1]
```

# Arrays, Matrizes e Objetos

## Arrays

### Funções úteis para usar com arrays

- **Map**: Altera o conteúdo de todos elementos do array.

```
const arr = [1,2,3,4,5,6,7,8,9,10];

const multiplos = arr.map(mult);
function mult(e) {
    return e * 10;
}

const multiplos = arr.map((e) => {
    return e * 10;
});

const multiplos = arr.map(e => e*10);

console.log(multiplos);
// [10,20,30,40,50,60,70,80,90,100]
```

# Arrays, Matrizes e Objetos

## Arrays

### Funções úteis para usar com arrays

- **Filter**: Retorna um novo array com somente alguns elementos do array original.

```
const arr = [1,2,3,4,5,6,7,8,9,10];

const pares = arr.filter(isEven);
function isEven(e) {
  if (e % 2 == 0) {
    return true;
  }
  return false;
}

const pares = arr.filter((e) => {
  return e % 2 == 0;
});

const pares = arr.filter(e => e % 2 == 0);

console.log(pares);
// [2,4,6,8,10]
```

# Arrays, Matrizes e Objetos

## Arrays

### Exercício - Parte 2:

- Use o array do exercício anterior.
- Organize o array em ordem crescente.
- Usando JS, insira o conteúdo do array em uma tabela HTML de 9x10.

8	16	35	37	39	40	55	60	102	120
143	144	146	155	161	183	212	217	231	231
235	240	248	262	273	284	290	293	300	307
358	359	385	389	393	396	413	425	434	440
462	466	468	481	491	494	501	503	513	521
530	538	538	543	548	566	569	579	588	628
653	659	680	701	719	731	750	765	799	821
821	832	852	860	876	879	907	915	929	933
933	934	940	945	969	972	976	990	993	997

# Arrays, Matrizes e Objetos

## Objetos

Objetos no Javascript são parecidos com arrays, mas podem ser indexados por uma string:

```
const pessoa = {  
  nome: 'Fulano',  
  sobrenome: 'Silva',  
  idade: 19,  
};  
pessoa.salario = 2345.67;  
  
console.log(pessoa.nome);  
console.log(pessoa['sobrenome']);  
  
const field = prompt('Informe qual campo deseja saber:');  
if (pessoa[field]){  
  alert(`${ field }: ${ pessoa[field] }`);  
}  
else {  
  alert('Este campo não existe');  
}
```

# Arrays, Matrizes e Objetos

## Objetos

Funções úteis para usar com objetos

- **Object.keys**: Retorna um array com os índices (chaves) do objeto.
- **Object.values**: Retorna um array com os valores do objeto.
- **Object.entries**: Retorna um array onde cada element é um array com o par [ chave, valor ] do objeto.

**ATENÇÃO:** Objetos não possuem a propriedade **length**.

```
const obj = { foo: 'bar', num: 100 };  
  
console.log(Object.keys(obj)); // [ 'foo', 'num' ]  
console.log(Object.values(obj)); // [ 'bar', 100 ]  
console.log(Object.entries(obj)); // [ ['foo', 'bar'], ['num', 100] ]
```

# Arrays, Matrizes e Objetos

## Objetos

Podemos percorrer um objeto usando `for...in`

```
const pessoa = {  
  nome: 'Fulano',  
  sobrenome: 'Silva',  
  idade: 19,  
  salario: 2345.67,  
};  
  
let str = '';  
for (let key in pessoa) {  
  const value = pessoa[key];  
  str += `

<span>Chave: ${key}</span>  
    <span>Valor: ${value}</span>  
  </div>`;  
}  
document.body.innerHTML = str;


```

```
div {  
  margin-bottom: 3px;  
  border: 1px solid black;  
  display: flex;  
}  
  
span {  
  width: 200px;  
  padding: 5px 10px;  
}
```

Chave: nome	Valor: Fulano
Chave: sobrenome	Valor: Silva
Chave: idade	Valor: 19
Chave: salario	Valor: 2345.67

# Arrays, Matrizes e Objetos

## Exercício:

- Crie uma página com 4 caixas de texto, como no exemplo ao lado.
- Ao clicar no botão, guarde todas informações do livro dentro de um objeto, e mostre a informação “Livro cadastrado com sucesso”.
- Mostre as informações do livro abaixo.

**CADASTRAR**

**Livro cadastrado com sucesso.**

O livro **O Senhor dos Anéis: A Sociedade do Anel** foi publicado pela editora **George Allen & Unwin** em **1954** e escrito por **J.R.R. Tolkien**.

**Cadastro de Livros**

**Título**  
O Senhor dos Anéis: A Sociedade do Anel

**Autor**  
J.R.R. Tolkien

**Ano**  
1954

**Editora**  
George Allen & Unwin

**CADASTRAR**



# Arrays, Matrizes e Objetos

## Matrizes

Matrizes é como chamamos arrays multidimensionais.

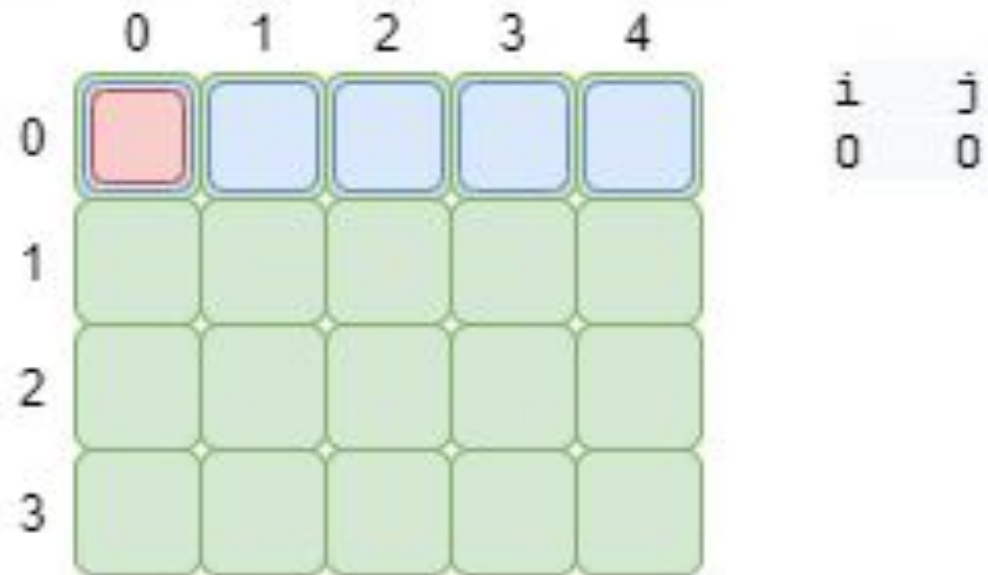
```
const mat = [  
  [ 1, 2, 3, 4, 5],  
  [ 6, 7, 8, 9, 10],  
  [11, 12, 13, 14, 15],  
  [16, 17, 18, 19, 20],  
];
```

Nada mais é que um array, onde cada elemento é um array também.

```
element == row[j] == mat[i][j]
```

Como percorrer?

```
for (let i in mat){  
  const row = mat[i];  
  for (let j in row){  
    const element = row[j];  
    // code  
  }  
}
```



# Arrays, Matrizes e Objetos

## Matrizes

```
const pessoas = [
  {
    nome: 'João',
    idade: 16,
    email: 'joao.silva@gmail.com',
  },
  {
    nome: 'Marcelo',
    idade: 38,
    email: 'marcelozucca@gmail.com',
  },
  {
    nome: 'Maria',
    idade: 19,
    email: 'mama2001@gmail.com',
  },
  {
    nome: 'Rafaela',
    idade: 12,
    email: 'rafafanti@gmail.com',
  },
  {
    nome: 'Guilherme',
    idade: 24,
    email: 'guilhermecardoso@gmail.com',
  },
];
```

É muito comum usar arrays de objetos.

```
<input>
<button>BUSCA</button>
<div id="message"></div>
```

```
const message = document.querySelector('#message');
document.querySelector('button').addEventListener('click', () => {
  const input = document.querySelector('input');
  message.innerHTML = `<span>Pessoa não encontrada</span>`;
  pessoas.forEach(p => {
    if (input.value == p.nome){
      message.innerHTML = `
        <span>Nome: ${p.nome}</span>
        <span>Idade: ${p.idade}</span>
        <span>E-mail: ${p.email}</span>
      `;
    }
  });
});
```

# Arrays, Matrizes e Objetos

## Matrizes

```
const pessoas = [  
  {  
    nome: 'João',  
    idade: 16,  
    email: 'joao.silva@gmail.com',  
  },  
  {  
    nome: 'Marcelo',  
    idade: 38,  
    email: 'marcelozucca@gmail.com',  
  },  
  {  
    nome: 'Maria',  
    idade: 19,  
    email: 'mama2001@gmail.com',  
  },  
  {  
    nome: 'Rafaela',  
    idade: 12,  
    email: 'rafafanti@gmail.com',  
  },  
  {  
    nome: 'Guilherme',  
    idade: 24,  
    email: 'guilhermecardoso@gmail.com',  
  },  
];
```

```
<input id="name">  
<input id="age">  
<input id="email">  
<button>INSERIR</button>
```

```
const name = document.querySelector('#name');  
const age = document.querySelector('#age');  
const email = document.querySelector('#email');  
document.querySelector('button').addEventListener('click', () => {  
  const pessoa = {  
    nome: name.value,  
    idade: age.value,  
    email: email.value,  
  }  
  pessoas.push(pessoa);  
});
```

# Arrays, Matrizes e Objetos

## Exercício:

- Crie uma página com duas caixas de texto e dois botões, para o cadastro de carros e suas placas.
- Ao clicar em **INSERIR**, o par de informações das caixas de texto devem ser inseridas em um array de objetos com os campos **modelo** e **placa**.
- Ao clicar no botão **REMOVER**, o primeiro elemento da lista deve ser removido. Use a função [shift](#) ou `slice`.
- A lista mostrada no documento deve ser atualizada a cada inserção e remoção.

Modelo do carro

Placa do carro

**INSERIR** **REMOVER**

Carro	Placa
Fiesta	QWE3R56
Polo	ABC1D23
Cronos	FSD8Y27
Cruze	KKP2X63

```
const carros = [  
  {  
    modelo: "NOME",  
    placa: "PLACA",  
  },  
  ...  
];
```