



## Array e métodos avançados

### JavaScript

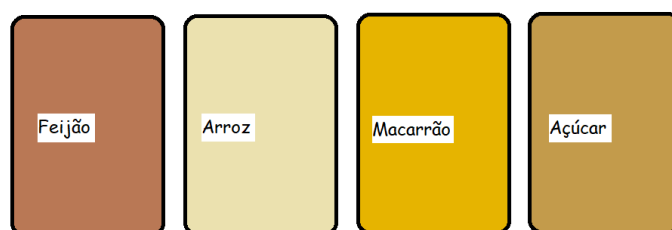
---

#### O que é uma array?

Uma array (ou "vetor", em português) em JavaScript é uma lista de valores. Ela serve para guardar vários dados em uma única variável.

#### *Exemplificando*

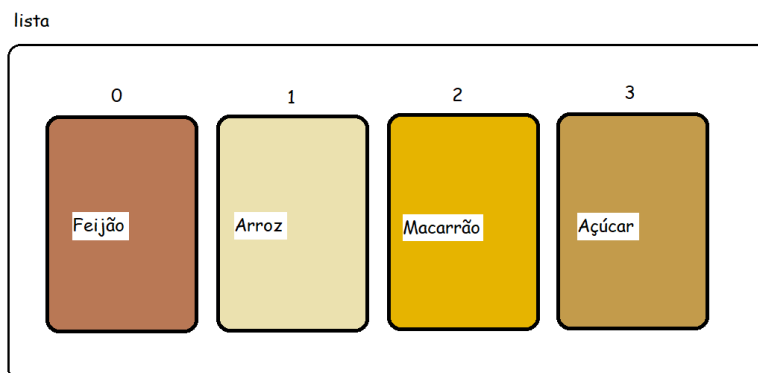
Vamos imaginar que você tem uma **lista de compras** com os seguintes itens: *feijão*, *arroz*, *macarrão* e *açúcar*.



Em **JavaScript**, essa lista pode ser representada assim:

```
const lista = ['feijão', 'arroz', 'macarrão', 'açúcar'];
```

Em uma lista, cada elemento que a compõe é numerada a partir de 0, por exemplo:

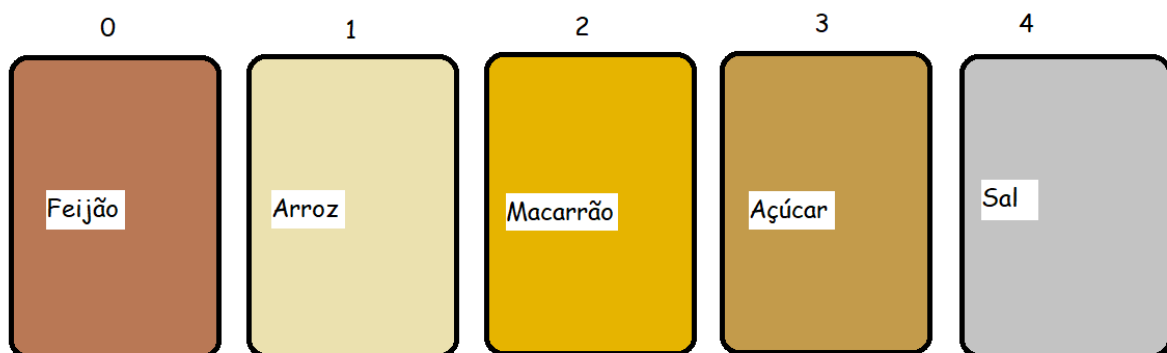


Ou seja, quando você precisa consultar um dos elementos dessa lista, você precisa informar a posição dela dentro da array. Se você precisar selecionar “*feijão*” ele vai estar na posição 0.

```
lista[0] = Feijão
```

Se você criar uma array vazia, `const lista = []`; você pode adicionar os itens em outras linhas `lista[0] = “Feijão”`;

```
const lista = [];  
lista[0] = 'Feijão';  
lista[1] = 'Arroz';  
lista[2] = 'Macarrão';  
lista[3] = 'Açúcar';  
lista[4] = 'Sal';
```



Você também pode criar uma array em um formato de objeto:

```
const lista = new Array('Feijão', 'Arroz', 'Macarrão', 'Açúcar', 'Sal');
```

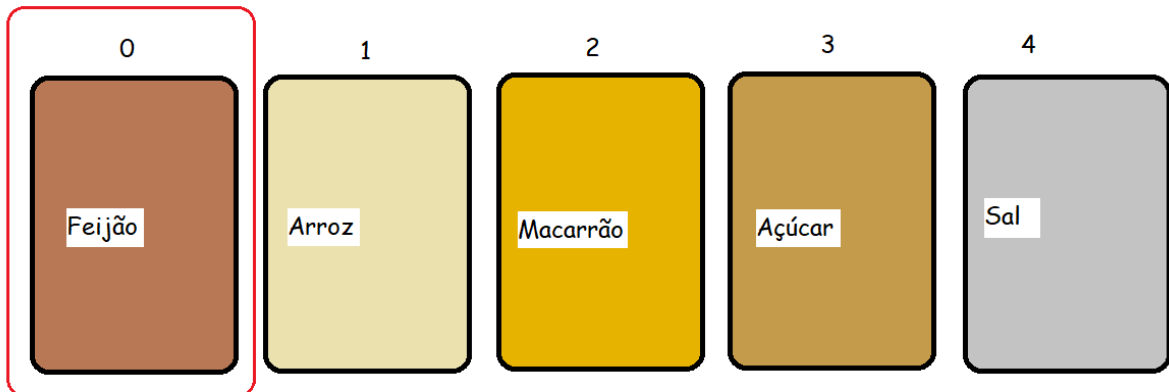
---

## Consultando valores específicos

Vamos imaginar que você quer consultar e incrementar um dos elementos da array em uma variável.

*Exemplificando*

Vamos precisar do **feijão** pra cozinhar uma **feijoada**



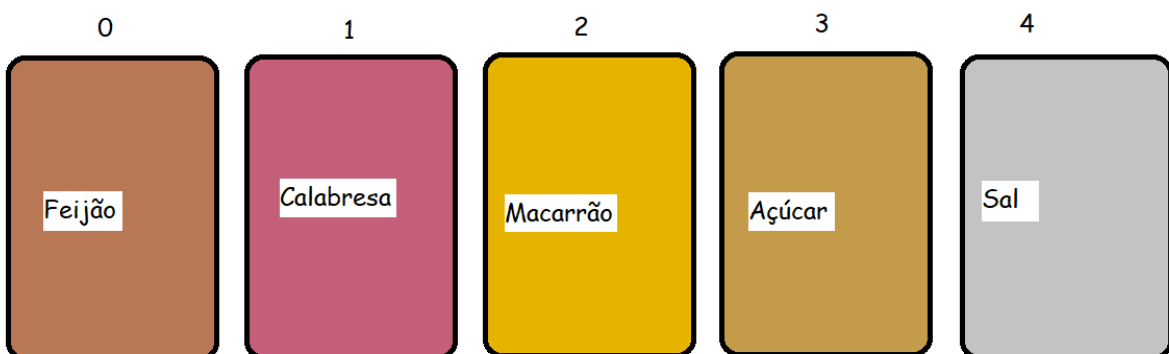
```
let feijoada = lista[0];  
alert(feijoada);
```

`let feijoada = lista[0];` vai colocar o item “*Feijão*” dentro da variável “*feijoada*”.  
`alert(feijoada);` vai nos dar o resultado “*Feijão*”.

Você também pode alocar um novo item à lista ou substituí-lo de outra forma:

```
lista[1] = “Calabresa”;  
alert(lista[1]);
```

`lista[1] = “Calabresa”;` vai substituir o item “*Arroz*” por “*Calabresa*”.  
`alert(lista[1]);` vai nos dar o resultado do item que está na posição 1 da array.



Essa é a nossa lista de compras - até o momento. Como podemos descobrir a quantidade de itens que temos dentro da array?

```
lista.length;  
alert(lista.length);
```

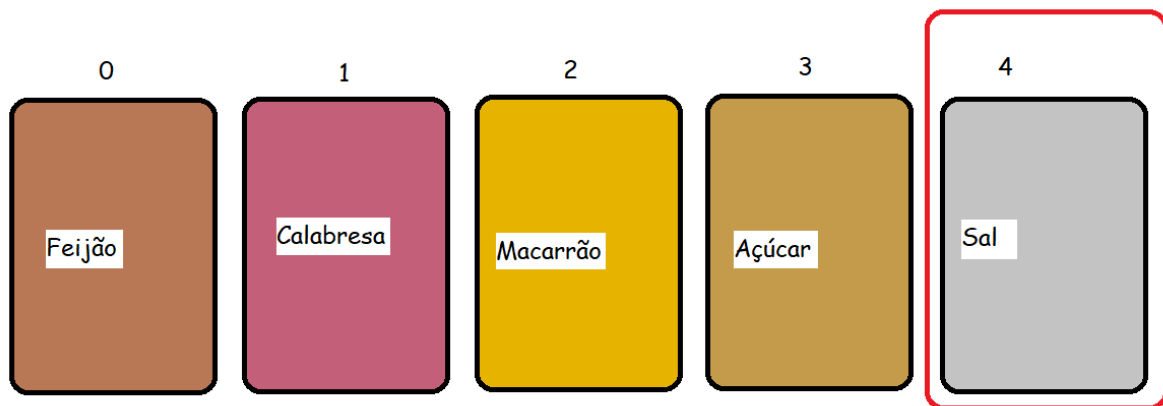
`alert(lista.length);` vai nos dar o resultado da quantidade de itens que temos na array, ou seja = 5.

---

## Mais métodos

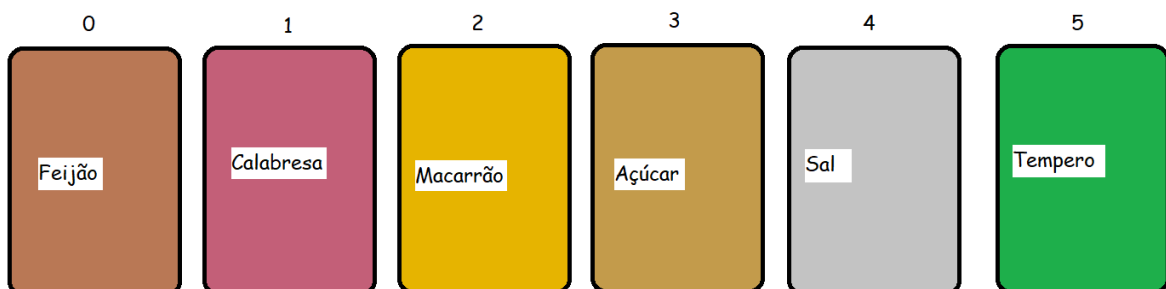
Encontrando o último item de um array.

```
alert(lista[lista.length-1]);
```



Adicionando mais um item.

```
lista.push("Tempero");
```



Confirmando se uma array é realmente uma array com retorno True ou False.

```
Array.isArray(lista);
```

JOIN - Substitui o separador “,” dos elementos de uma array.

```
document.getElementById("teste").innerHTML=lista.join(*);
```

POP - Removendo o último item da array.

```
lista.pop();
```

PUSH - Adicionando um novo valor à array.

```
lista.push();
```

SHIFT - Remove o primeiro item e todos os índices são alterados.

```
lista.shift();
```

UNSHIFT - Adiciona um valor no início da array.

```
lista.unshift();
```

DELETE - Deleta um valor mas deixa a posição vazia, sem alterar o índice.

```
delete lista[0];
```

SPLICE - Insere um ou mais itens na posição em que é inserida no método.

```
lista.splice(1,0, "item1", "item 2")
```

CONCAT - Concatenando duas arrays em uma só.

```
const listaConcatenada = lista.concat(lista2);
```

**SLICE** - Remove uma parte de uma array.

No exemplo abaixo, “2” é responsável por remover a array e tudo que vem antes desse índice. Em “3” ele remove o índice 3 da array e tudo que vem após ele.

```
const teste1 = lista.slice(2);  
const teste2 = lista.slice(2,3);
```

**SORT** - Deixa a array em ordem alfabética.

```
lista.sort();
```

**REVERSE** - Inverte a array que está em ordem alfabética.

```
lista.reverse();
```

**SORT em números** - Deixando a array em ordem numérica.

```
lista.sort(function(a,b){return a-b});
```

**REVERSE em números** - Invertendo a array em ordem numérica.

```
lista.sort(function(a,b){return b-a});
```

Utilizando função para encontrar maior e menor número de uma array

```
function MaiorNumero(array) {  
    return Math.max.apply(null, array);  
}  
  
function MenorNumero(array) {  
    return Math.min.apply(null, array);  
}
```

Filtrando números através de função

```
const maior20 = numeros.filter(Filtragem);  
  
function Filtragem(array) { return value > 20; }
```

forEach tem como objetivo aplicar alguma função para cada item da lista informada

```
const numeros = [2, 10, 8];

let resultado = "";

numeros.forEach((numero) => {
  resultado += (numero * 2) + "<br>";
});

document.getElementById("teste").innerHTML = resultado;
```

MAP - Mapear o array e aplica a função direcionada à ele

```
const numeros = [1, 2, 3];

const dobrados = numeros.map((numero) => numero * 2);

document.getElementById("teste").innerHTML = dobrados;
```

REDUCE - Mapeia o array e aplica também uma função direcionada à ele, o exemplo abaixo reduz a array somando todos os números que fazem parte de “numeros”

```
const numeros = [1, 2, 3, 4];

const soma = numeros.reduce((acumulador, valorAtual) => acumulador + valorAtual, 0);

document.getElementById("teste").innerHTML = soma;
```

**SOME** - Tem o objetivo de mapear a array e nos retornar um resultado false ou true. No exemplo abaixo, ele nos informa se na array numeros há número par.

```
const numeros = [4, 2, 6];

const temImpar = numeros.some((numero) => numero % 2 === 1);

document.getElementById("teste").innerHTML = temImpar;
```

**FIND** - Tem o objetivo de encontrar e retornar apenas o primeiro elemento da array que cumpre o requisito da função que é estabelecida nele. No exemplo abaixo, ele pede o primeiro número da lista que é maior que 10.

```
const numeros = [10, 5, 8, 20, 21];

const primeiroMaiorQue10 = numeros.find((numero) => numero > 10);

document.getElementById("teste").innerHTML = primeiroMaiorQue10;
```

**EVERY** - Tem o objetivo de retornar verdadeiro apenas se todos os elementos da lista estiverem de acordo com a função que é estabelecida. No exemplo abaixo, ele apenas retorna true se todos os elementos da lista numeros forem par.

```
const numeros = [1, 4, 6];

const todosSaoPares = numeros.every((numero) => numero % 2 === 0);

document.getElementById("teste").innerHTML = todosSaoPares;
```