



Array e métodos avançados

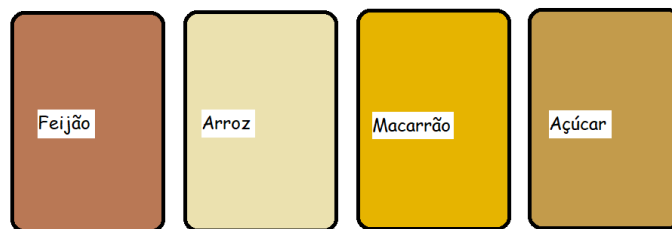
JavaScript

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

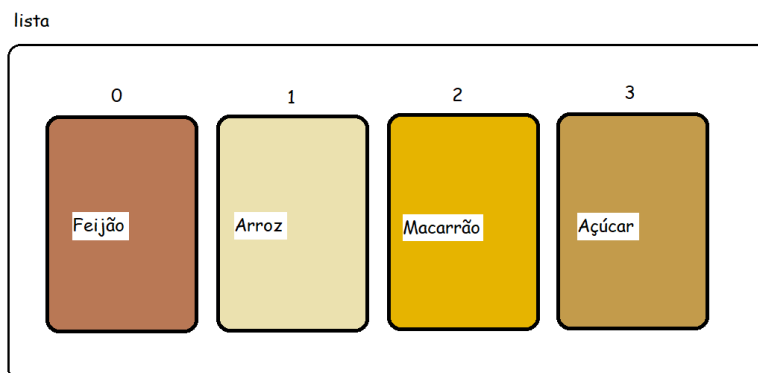
Vamos imaginar que você tem uma **lista de compras** com os seguintes itens: *feijão*, *arroz*, *macarrão* e *açúcar*.



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:

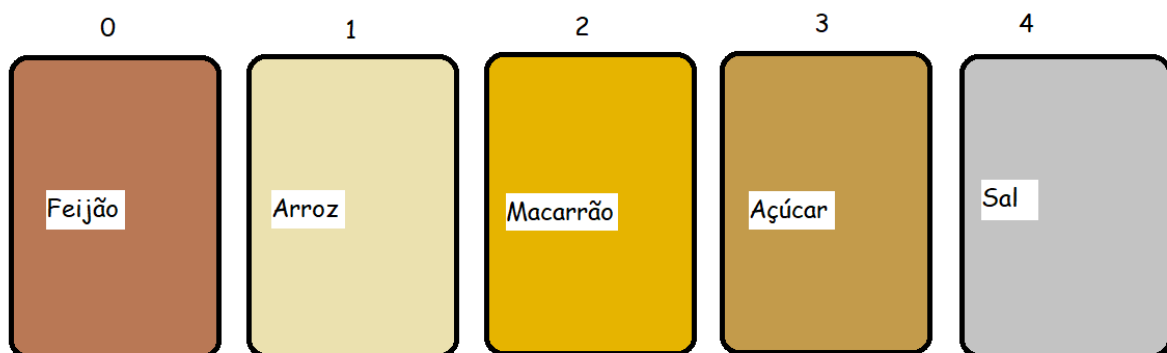


Ou seja, quando você precisa consultar um dos elementos dessa lista, você precisa informar a posição dela dentro da array. Se você precisar selecionar “*feijão*” ele vai estar na posição 0.

```
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:

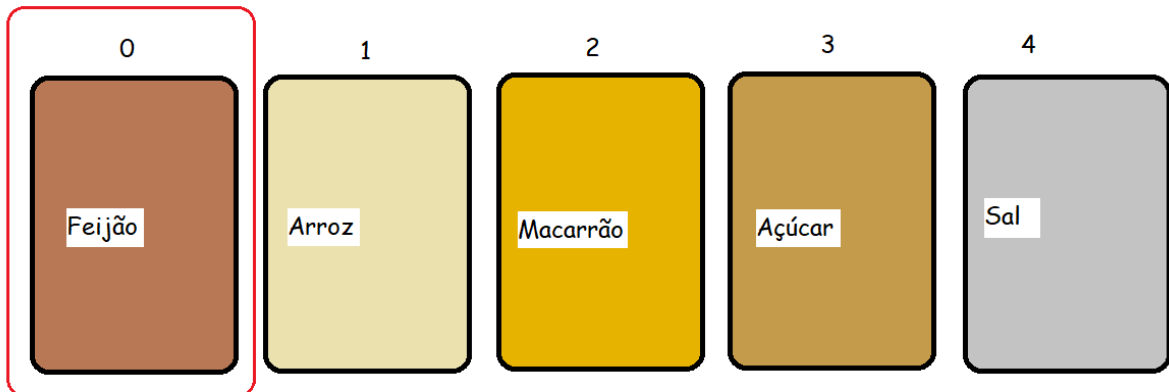
```
const lista = new Array('Feijão', 'Arroz', 'Macarrão', 'Açúcar', 'Sal');
```

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**



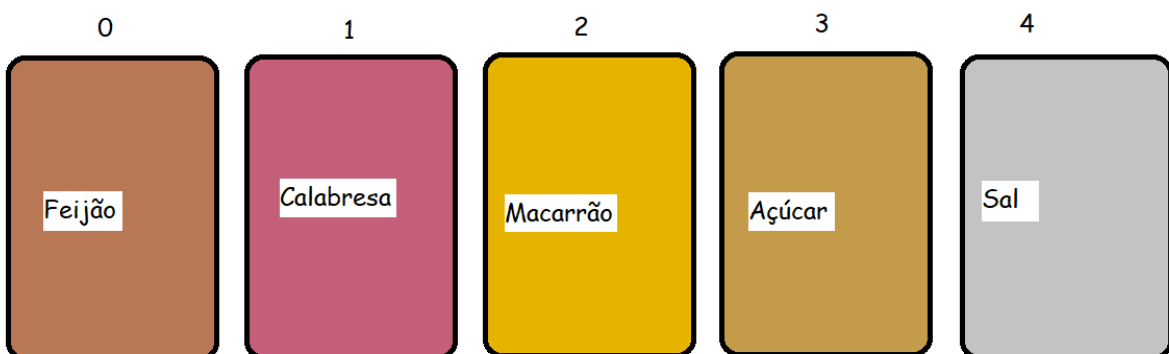
```
let feijoada = lista[0];  
alert(feijoada);
```

`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?

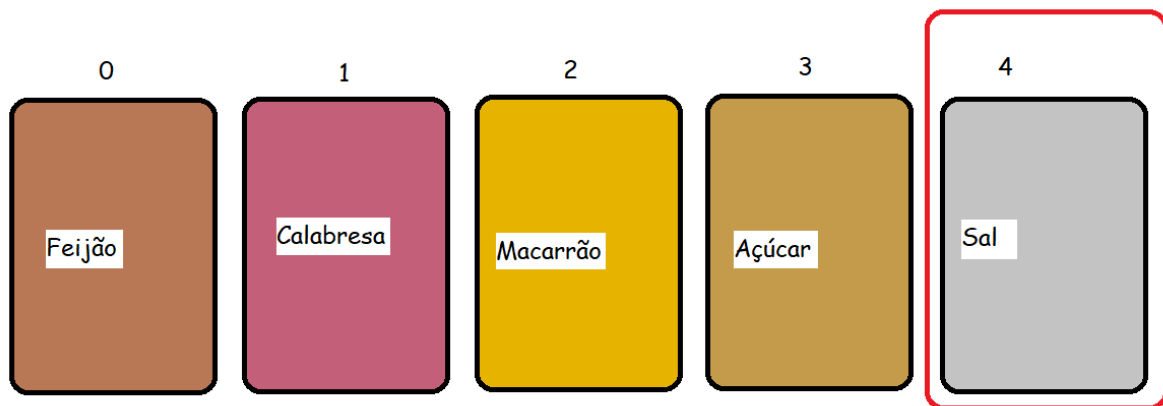
```
lista.length;  
alert(lista.length);
```

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

Mais métodos

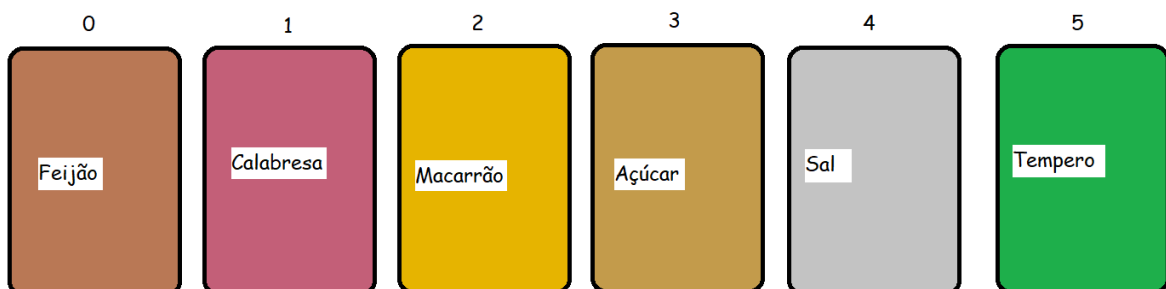
Encontrando o último item de um array.

```
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.

```
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; }
```